NeuRSS: Enhancing AUV Localization and Bathymetric Mapping With Neural Rendering for Sidescan SLAM

Yiping Xie, Jun Zhang, Nils Bore, and John Folkesson, *Senior Member, IEEE*

***Abstract*—Implicit neural representations and neural rendering have gained increasing attention for bathymetry estimation from sidescan sonar (SSS). These methods incorporate multiple observations of the same place from SSS data to constrain the elevation estimate, converging to a globallynt bathymetric model. However, the quality and precision of the bathymetric estimate are limited by the positioning accuracy of the autonomous underwater vehicle (AUV) equipped with the sonar. The global positioning estimate of the AUV relying on dead reckoning (DR) has an unbounded error due to the absence of a geo-reference system like GPS underwater. To address this challenge, we propose in this article a modern and scalable framework, *NeuRSS*, for SSS SLAM based on DR and loop closures (LCs) over large timescales, with an elevation prior provided by the bathymetric estimate using neural rendering from SSS. This framework is an iterative procedure that improves localization and bathymetric mapping. Initially, the bathymetry estimated from SSS using the DR estimate, though crude, can provide an important elevation prior in the *nonlinear least-squares* (NLSs) optimization that estimates the relative pose between two LC vertices in a pose graph. Subsequently, the global pose estimate from the SLAM component improves the positioning estimate of the vehicle, thus improving the bathymetry estimation. We validate our localization and mapping approach on two large surveys collected with a surface vessel and an AUV, respectively. We evaluate their localization results against the ground truth and compare the bathymetry estimation against data collected with multibeam echo sounders (MBESs).**

***Index Terms*—Deep learning, marine robots, simultaneous localization and mapping, underwater navigation.**

Received 9 May 2024; revised 21 August 2024; accepted 29 October 2024. This work was supported in part by the Wallenberg AI, Autonomous Systems and Software Program (WASP) through by the Knut and Alice Wallenberg Foundation, in part by Stiftelsen för Strategisk Forskning (SSF) through the Swedish Maritime Robotics Centre (SMaRC) under Grant IRC15-0046, in part by the Alice Wallenberg Foundation through Mobile Underwater System Tools (MUST) Project that provided the Hugin AUV, in part by the National Academic Infrastructure for Supercomputing in Sweden (NAISS), and in part by the Swedish National Infrastructure for Computing (SNIC) at Berzelius through the Swedish Research Council under Grant 2022-06725 and Grant 2018-05973. *(Corresponding author: Yiping Xie.)*

Guest Editor: H. Singh.

Yiping Xie and John Folkesson are with the Robotics, Perception and Learning Division, KTH Royal Institute of Technology, SE-100 44 Stockholm, Sweden (e-mail: yipingx@kth.se; johnf@kth.se).

Jun Zhang is with the Institute of Computer Graphics and Vision, Graz University of Technology, 8010 Graz, Austria (e-mail: jun.zhang@tugraz.at).

Nils Bore is with Ocean Infinity, SE-426 71 Västra Frölunda, Sweden (e-mail: nils.bore@oceaninfinity.com).

Digital Object Identifier 10.1109/JOE.2024.3501317

I. Introduction

SMALL autonomous underwater vehicles (sAUVs) equipped with sidescan sonar (SSS) are often used for hydrogeological surveys and seabed mapping. Nonetheless, in the absence of GPS and an a priori map of the surveyed area, the dead reckoning (DR) estimate of their global position can drift significantly over time. Existing underwater positioning systems analogous to GPS, such as long baseline (LBL) and ultrashort baseline (USBL), require external infrastructure for the deployment of beacons/transponders. As a small-form and low-cost sensor, SSS provides a promising and cost-effective solution for sAUV navigation and mapping due to its ability of generating high-resolution images with wide swath.

Traditionally, bathymetric maps are usually constructed with multibeam echo sounders (MBESs), which can be cost-prohibitive and too large for low-cost sAUVs. For SSS data, the range and the azimuth angle of the returns are known, but the information of the elevation angle is lost due to projection, which is essential for bathymetry reconstruction. However, since the changes of the returned intensities indicate changes of the incidence angle, it is possible to extract information of the slope of the seafloor from SSS data. Although reconstruction of the seafloor from a single line of SSS imagery is mathematically ill-posed, this problem can be constrained adequately in areas that have been observed from multiple viewpoints. One can notice the similarities between bathymetry reconstruction from SSS and 3-D reconstruction from camera images, in the sense that for optical camera images, there is also 1-D information lost during the projection, that is, the range instead of the elevation angle. As a result, many approaches from computer vision and computer graphics for 3-D reconstruction using camera images can be adapted for the same task using sonars. Examples of these are, in the early days, shape-from-shading (SfS) techniques [1], [2], [3], [4], and more recently, data-driven methods using convolutional neural networks (CNNs) [5], [6] and inverse rendering based on implicit neural representations [7], [8], [9], [10].

As for underwater navigation, simultaneous localization and mapping (SLAM) techniques can reduce the drift in the AUV's DR estimate using onboard sensor measurements. Most of the state-of-the-art graph-based underwater SLAM solutions for unstructured environment target AUVs equipped with MBES [11]. But the limited coverage of MBES makes loop closure (LC) detection sparse, especially due to the scarcity of distinguishable

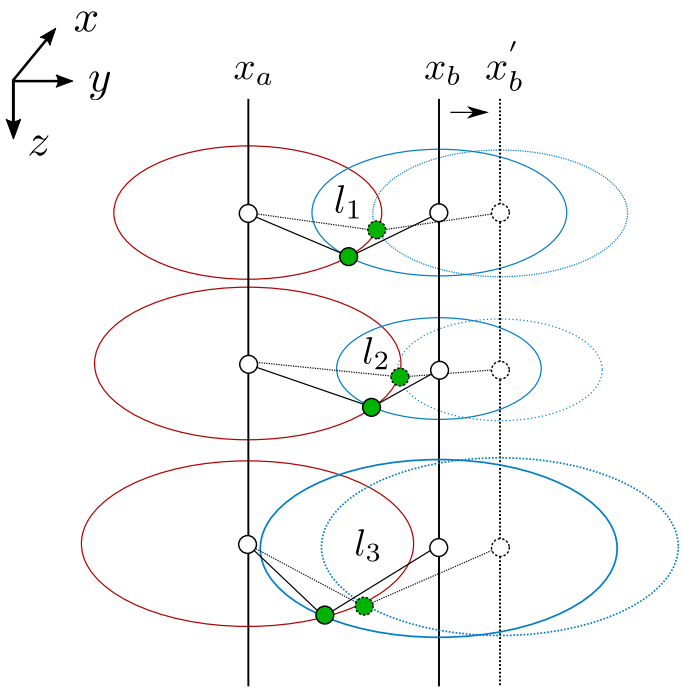


Fig. 1. Illustration of landmark elevation degeneracy with pure y-translation motion, in the sensor forward-lateral-down frame. Here, we show two submaps from two parallel survey lines with three landmarks (green dots). In the NLS optimization, we usually fix pose $\boldsymbol{x}_a$, namely, fixing the solid red circles (indicating the range measurements). Without any priors on landmarks, $\boldsymbol{x}_b$ and all the landmarks can move together in y-z plane, in this case, $\boldsymbol{x}_b$ positive translation along the y-axis to $\boldsymbol{x}_b'$, landmarks moving up along the z-axis (from solid strokes to dashed strokes), where all SSS range and bearing measurements are still fulfilled.

features on the seabed. SSS, on the other hand, has wider coverage, which potentially allows more LC detections between overlapping submaps constructed from adjacent survey lines, and can potentially resolve smaller features. However, due to the elevation ambiguity inherent to the SSS sensor, graph-based SSS SLAM faces elevation degeneracy (see Fig. 1), making the overall NLS optimization rank-deficient. Such degeneracy, similar to forward-looking sonar (FLS) [12], [13], [14], [15], [16], results in large errors in landmark on and unrobustness in the relative pose estimation between two LC vertices of a pose graph when a LC is detected. The degeneracy case is especially common in standard lawn-mower pattern[1] surveys, since most matches can only be made between parallel lines where the SSS sees the feature from the same side but different distances.

This degeneracy is a well-known issue in triangulation, as a result, most SSS SLAM approaches address such elevation degeneracy by assuming the seafloor is locally flat [17], [18], [19], and as for FLS SLAM approaches, they use the planar assumption [20]. This assumption only works when the structure of seafloor is relatively benign, for example, slowly sloping. The assumption will break when dealing with complex seabed such as rocky areas, mountains, and ridges. However, in these scenarios, an estimate of the seafloor would be of great help to constraining the elevation angle of the returns, serving as an elevation prior in the NLS optimization, yielding more robust relative pose estimation. The easiest way to estimate the rough bathymetry is to linearly interpolate altimeter readings so that a crude bathymetric model can be obtained, but the errors are still quite large if the seabed is complex. On the other hand, incorporating information from SSS data would significantly improve the bathymetric model.

Bathymetry estimation from SSS requires accurate pose estimation, while SSS SLAM suffers from elevation degeneracy without a precise bathymetric estimate. Nevertheless, one can address such a chicken-and-egg problem by iteratively improving the estimates of the bathymetry and the pose estimation. Following this principle, we present a framework,[2] *NeuRSS*, that leverages the advances of neural rendering to estimate bathymetry from SSS [7], [8], which provides an elevation before addressing the degeneracy in SSS SLAM problems and showcases that one can significantly improve the AUV's DR estimate and subsequently produce high-quality bathymetry using SSS data from a standard survey.

The contribution of this work is to extend our previous work [7], [8], [21] and combine them into a new framework. For the sonar scattering model in [8], we extend it to be able to model shadows. For the back-end of the SSS SLAM framework in [21], we extend it to a submap-based optimization with a prior map to better address the elevation degeneracy. We evaluate the proposed NeuRSS framework numerically using two field data sets containing both SSS and MBES collected simultaneously.

II. Related Work

A. SSS SLAM

Early works used stochastic maps to estimate the AUV's position from SSS imagery with an extended Kalman filter (EKF) [22], [23], [24], however, EKF-based SLAM approaches suffer from scalability as the size of the state vector grows, especially in a large underwater environment, e.g., an open sea. Later, Fallon et al. [17] proposed a graph-based SLAM framework that utilizes incremental smoothing and mapping (iSAM) to fuse acoustic ranging and SSS measurements for on-board applications in real time. Similarly, Bernicola et al. [18] demonstrated using iSAM to correct trajectories with SSS images. Issartel et al. [19], aiming to address the false SSS data associations issue, proposed to use switchable observation constraints in the pose graph. However, they [17], [18], [19] all use the flat seafloor assumption for the sonar measurement model, which prevents the NLS optimization from landmark elevation degeneracy (pure y-translation). But the error introduced by this assumption increases as the terrain gets more complex, harming the performance of localization. In [21] and [25], the 3-D landmark positions together with relative pose transformation between the reference pose and current pose are estimated in the NLS optimization with an elevation prior on the landmark, provided by interpolating between the altitudes of the two poses. However, the underlying assumption is that the terrain is relatively benign.

[1] A lawn-mower pattern that has the vehicle perform the survey as a series of long parallel lines.

[2] The source code link will be provided here upon acceptance.

B. FLS SLAM

Another line of work that is highly relevant is FLS SLAM [12], [13], [14], [15], [16], where the degeneracy cases are in general more complex than SSS SLAM. Different robot motions on this degeneracy are discussed in [12], where the majority of the cases causing such does not exist in SSS SLAM with a standard lawn-mower pattern, except the pure y-translation. Westman et al. [14], [15] proposed to examine the eigenvalues of the Jacobian matrix of landmarks to address these degeneracy cases, whereas Wang et al. [16] modeled the terrain as Gaussian processes (GPs) and incorporated the terrain factors into the factor graph.

C. SSS Bathymetry Reconstruction

Early works to estimate bathymetry from SSS rely on traditional SfS techniques and Lambertian models. Notably, Coiras et al. [4] used a Lambertian model for the sonar ensonification process and obtained an approximation of the surface gradients from sonar intensities by inverting the image formation. They showed that convergence can be improved by gradually increasing the resolution of the predicted bathymetry. Recently, data-driven approaches [5], [6] using CNNs have been proposed to learn the missing elevation directly from SSS images in a supervised-learning fashion. However, the "ground truth" bathymetry is needed for creating a training set, which is not always practical underwater. Neural rendering [7], [8] methods that leverage the continuity and differentiablity of implicit neural representations have been recently proposed to fit many sidescan lines into a self-consistent bathymetry with a global optimization. Specifically, a multilayer perceptron (MLP) with sine activation functions, known as SIREN [26], was used to represent the bathymetry where the gradients of the bathymetry were constrained by SSS intensities through a Lambertian model. Extended from [7], a nadir model was proposed in [8] to model the nadir region in SSS waterfall images so that the optimization can converge without any external bathymetric data, e.g., altimeter readings. However, acoustic shadows cannot be explained by the Lambertian model in [8]. Furthermore, all the aforementioned works assume access to high-accuracy navigation estimates.

III. Neural Rendering for Bathymetry Estimation

A prerequisite of our neural rendering pipeline is the assumption that the vehicle's trajectory is already corrected. In this section, we present the neural rendering pipeline with an extended Lambertian model based on implicit neural representations.

A. Implicit Neural Representation

In this approach, the bathymetry is represented using an implicit neural representation, specifically a function $\Phi_\theta : \mathbb{R}^2 \to \mathbb{R}$, which maps 2-D spatial coordinates, i.e., Euclidean easting and northing x, y, to the corresponding height of the seafloor $\tilde{h}$. This function is parameterized by a fully connected neural network with parameters θ, specifically, a variant of an MLP that employs sinusoidal activation functions, known as SIREN [26].

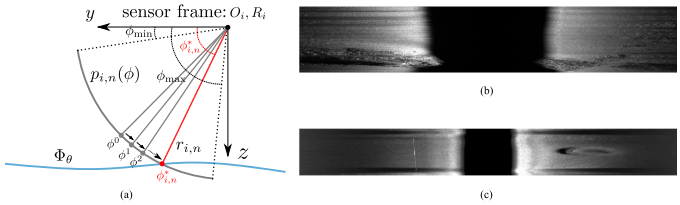


Fig. 2. (a) Illustration of the gradient descent approach to find the intersection between the elevation arc and the seafloor, parameterized by SIREN. (b) Example of an SSS image in Data set 1. (c) Example of an SSS image in Data set 2, showing the sinkhole on the seabed.

Given a SSS survey in a data set of the form $D = \{I_i, \boldsymbol{x}_i, h_i\}_{i=1}^N$, containing N pings of SSS intensities $I_i \in \mathcal{I}$, estimates of the 6-D AUV poses $\boldsymbol{x}_i$ and altimeter readings h_i. Combining the positioning estimates of the AUV and h_i, we have sparse bathymetric measurements on the seafloor surface $\{p_i^{xyz}\}_{i=1}^N$ along the AUV trajectory in the world coordinates Easting, Northing, Up (ENU), which can be directly used to constrain the unknown mapping Φ_θ

$$\mathcal{L}_H = \frac{1}{|\{p_i^{xyz}\}|} \sum_i \|\Delta^{\Phi_\theta}(p_i^{xyz})\| \tag{1}$$

where

$$\Delta^{\Phi_\theta}(p) = \Phi_\theta(p_x, p_y) - p_z \tag{2}$$

is the signed vertical distance to the seafloor, $|\cdot|$ denotes the size of the set $\{p_i^{xyz}\}$ and $\|\cdot\|$ denotes the L_2 norm.

Besides the constraint $\mathcal{L}_H$ from the altimeter readings, the measured returned intensity of every pixel in the SSS images $I_{i,n}$ at given ping i and given bin n can be modeled using neural rendering given a sidescan scattering model. The difference between the rendered SSS intensity $\tilde{I}_{i,n}$ and the measured intensity $I_{i,n}$ can form the intensity loss to further constrain the surface normal of the bathymetry Φ_θ

$$\mathcal{L}_I = \frac{1}{|\{I_{i,n}\}|} \sum_{I_{i,n}} \|\tilde{I}_{i,n} - I_{i,n}\|. \tag{3}$$

The following section introduces the sidescan scattering model and the process of neural rendering.

B. Sidescan Scattering Model

SSS emits a fan-shaped beam to the side of the AUV with a narrow beam along the travel direction and a wide beam in the azimuth direction, and then records the returned echos at fixed intervals of time. The recorded backscatter intensities are arranged in a vector, often referred to as a *ping*, which can be stacked "row-by-row" as the vehicle moves along to form a "waterfall" image [see Fig. 2(b) and (c)]. Each item in the vector, often referred as a *bin*, stores the amplitude and two-way travel time of the returns. The travel time is used to calculate the distance of returns from the sonar array, combined with the sound-speed profile (SVP).

Similar to [7] and [8], we use the Lambertian model for the scattering process. For $I_{i,n}$, denoting the measured returned

intensity from the ensonified point on the seafloor $p_{i,n}$, its returned intensity can be approximated by

$$\tilde{I}_{i,n} = K\Phi(p_{i,n})R(p_{i,n})\left\|\cos(\alpha)\right\|^2 \tag{4}$$

where K is the normalizing constant, Φ is the beam pattern of the sonar, R is the reflectivity of the seafloor and α is the incidence angle. The incidence angle can be calculated given sonar's pose and the bathymetry model Φ_θ as follows.

Assuming isovelocity SVP, the ensonified volume $p_{i,n}$ is at a fixed distance (slant range) $r^s_{i,n}$ away from the sonar, parameterized by the elevation angle $\phi^g_{i,n}$ along an arc referred to as an *isotemporal curve* [7]

$$p_{i,n} = t_i + r^s_{i,n}R_i\left[0, -\cos(\phi^g_{i,n}), \sin(\phi^g_{i,n})\right]^T \tag{5}$$

where t_i is the translation and R_i is the rotation matrix for ping i. Given the estimated bathymetry Φ_θ, we can determine the elevation angle $\phi^g_{i,n}$ by using gradient descent (GD) algorithm to find where the arc intersects with the current estimated seafloor (see Fig. 2), assuming the arc only has one intersection with the seafloor within sonar's vertical sensor opening $[\phi_{\text{min}}, \phi_{\text{max}}]$

$$\phi^{k+1} = \phi^k - \frac{\lambda}{r_{i,n}}\frac{d}{d\phi}(\Delta^{\Phi_\theta}(p_{i,n}(\phi^k)))^2 \tag{6}$$

where $\Delta^{\Phi_\theta}(p)$ is the signed vertical distance to the seafloor, and λ is the step size in the GD. Once we find the optimal $\phi^g_{i,n}$, we can define the surface normal at $p_{i,n}$ with respect to Φ_θ, given the gradient components ∇_x, ∇_y

$$N^{\Phi_\theta}(p) = \left[-\nabla_x\Phi_\theta(p_x, p_y), -\nabla_y\Phi_\theta(p_x, p_y), 1\right]^T. \tag{7}$$

The ray from sonar to the isotemporal curve can be simply defined as

$$r(\phi^g_{i,n}) = r^s_{i,n}R_i\left[0, -\cos(\phi^g_{i,n}), \sin(\phi^g_{i,n})\right]^T. \tag{8}$$

Given $N^{\Phi_\theta}(p)$ and $r(\phi^g_{i,n})$, we can compute the Lambertian scattering contribution, $M^{\Phi_\theta}_{i,n}$, using $\cos^2$ of the incidence angle

$$M^{\Phi_\theta}_{i,n} = \left\|\cos(\alpha)\right\|^2 = \left(r(\phi^g_{i,n})^T\hat{N}^{\Phi_\theta}(p_{i,n})\right)^2. \tag{9}$$

Besides the Lambertian contribution in (4), we also model and estimate the gain, beam pattern and reflectivity jointly with the bathymetry, similarly as [7] and [8]. The beam pattern $\Phi(\phi)$ is modeled as a radial basis function (RBF) with kernels evenly spread across ϕ at fixed positions. Reflectivity $R(p)$ of the whole surveyed area is also modeled as a 2-D RBF with kernels spread spatially. The gain parameter A_i is estimated for each sidescan line. Note that the assumption of a single intersection between the arc and the seafloor may not hold in complex seafloor geometries, such as a shipwreck. In these cases, sampling-based techniques from [9] and [10] can be used to resolve the SSS elevation ambiguity instead of GD.

C. Nadir and Shadows

Neither nadir area in the SSS data nor the shadows can be explained by the Lambertian scattering model, thus, we propose to extend the traditional Lambertian model to handle both components in the SSS data, inspired by recent advances of volumetric rendering using neural implicit surfaces [27], [28].

Nadir area is when the sound pulse travels through the water column before hitting the seafloor, where the corresponding arc has no intersections with the seafloor. Since we perform a fixed-number of steps gradient descent to calculate where the arc intersects with the seafloor, for the nadir area, the signed vertical distance $\Delta^{\Phi_\theta}(p(\phi^*))$ for the optimal grazing angle ϕ^* would be far from zero. We propose to weight the intensity for nadir area with its volume density, computed as [8]: $\sigma(p(\phi^*)) = \exp(-(\Delta^{\Phi_\theta}(p(\phi^*))^2)/\sigma_s)$, so that the volume density at nadir area is close to zero but one other wise. σ_s is a spread parameter that can be manually tuned to control the smoothness of the volume density function.

It is also well known that the shadows in the sidescan data cannot be explained by the Lambertian model. Similarly, for the shadows in the data, the gradient descent procedure would find an intersection between the arc and the seafloor, but the found intersection is occluded. Inspired by [9] and [10], we propose to sample a few points along the ray backwards, starting from the intersection and compute the accumulated transmittance at the intersection $T(p(\phi^*)) = \exp\left(-\int_0^{r^s}\rho(u)du\right)$, which is used to indicate if $p(\phi^*)$ is occluded. Here, the particle density ρ is computed as in [9] and [10]. We use $T(p(\phi^*))$ to weight the predicted SSS intensity, since if $T(p(\phi^*))$ is close to zero, it indicates that the intersection point is occluded, which corresponds to shadows in the sidescan images. The extended Lambertian model for the complete intensity rendering is given by

$$\tilde{I}_{i,n} = A_i\Phi(\phi^*_{i,n})R(\phi^*_{i,n})M^{\Phi_\theta}_{i,n}(\phi^*_{i,n})\sigma^{\Phi_\theta}(\phi^*_{i,n})T^{\Phi_\theta}(\phi^*_{i,n}). \tag{10}$$

D. Neural Rendering Algorithm

Algorithm 1 outlines the SIREN training process using neural rendering. The inputs are the data set D collected during the survey and the learned parameters are $\{\Phi_\theta, \Phi, R, A_i\}$ containing the bathymetry estimation Φ_θ parameterized by SIREN, the beam pattern Φ parameterized by 1-D Gaussian kernels, the reflectivity of the seafloor R parameterized by 2-D Gaussian kernels and gains per sidescan lines A_i as scalar variables. The initialization of Φ_θ can be random as in [7] and [8] or one could apply bilinear interpolation given $\{\boldsymbol{x}_i, h_i\}$ to obtain an initial estimate of Φ_θ, which in theory would fasten the convergence of the SIREN training afterwards. Lines $2-15$ outline the SIREN training for a fixed number of steps (K) with a decaying learning rate. Specifically, lines $3-6$ calculate the height loss within a batch of random samples with batch size M_H given the current bathymetry estimate Φ^k_θ. Lines $8-13$ calculate the intensity loss within a batch of data (batch size M_I), where line 9 applies a fixed number of steps GD to find the optimal elevation angle ϕ for each sample given the current bathymetry estimate Φ^k_θ. Line 10 computes the beam pattern and reflectivity at ϕ given the current estimate of Φ^k, R^k and line 11 calculates the corresponding Lambertian component M, the volume density σ and the accumulated transmittance T of the seafloor intersection. Lines $12-13$ construct the intensity

Algorithm 1: Φ_θ Algorithm.

Data: $D = \{I_{i,n}, \hat{\boldsymbol{x}}_i, h_i\}$
Result: $\Phi_\theta, \Phi, R, A_i$
1 initialization
2 **for** $k = 0$ *to* $K-1$ **do**
3 random sample M_H samples from $\{\hat{\boldsymbol{x}}_i, h_i\}$
4 **for** $m_H = 0$ *to* M_H **do**
5 $p^{xyz}_{m_H} \leftarrow \hat{\boldsymbol{x}}_{m_H}, h_{m_H}$
6 $\mathcal{L}_H \leftarrow p^{xyz}_{m_H}, \Phi^k_\theta$
7 random sample M_I samples from $\{\hat{\boldsymbol{x}}_i, I_{i,n}\}$
8 **for** $m_I = 0$ *to* M_I **do**
9 $\phi_{m_I} \leftarrow \mathrm{GD}(\hat{\boldsymbol{x}}_{m_I}, \Phi^k_\theta, r_{m_I})$
10 $\Phi^k_{m_I}, R^k_{m_I} \leftarrow \hat{\boldsymbol{x}}_{m_I}, \phi_{m_I}, r_{m_I}$
11 $M_{m_I}, \sigma_{m_I}, T_{m_I} \leftarrow \hat{\boldsymbol{x}}_{m_I}, \phi_{m_I}, r_{m_I}, \Phi^k_\theta$
12 $\hat{I}_{m_I} \leftarrow A^k_{m_I}, \Phi^k_{m_I}, R^k_{m_I}, M_{m_I}, \sigma_{m_I}, T_{m_I}$
13 $\mathcal{L}_I \leftarrow I_{m_I}, \hat{I}_{m_I}$
14 $\Phi^{k+1}_\theta, \Phi^{k+1}, R^{k+1}, A^{k+1} \leftarrow \mathrm{OptimizerStep}(\mathcal{L}_H, \mathcal{L}_I, \Phi^k_\theta, \Phi^k, R^k, A^k)$
15 update the learning rate
16 **return** $\Phi^K_\theta, \Phi^K, R^K, A^K$

loss within the batch and line 14 updates $\{\Phi_\theta, \Phi, R, A_i\}$ at the same time using gradient-based optimization.

IV. SSS SLAM With Elevation Prior

The full 6-D AUV state is defined as $[x, y, z, \phi, \theta, \psi]$ in three Cartesian and three rotation dimensions, where absolute measurements of the depth z, roll ϕ, and pitch θ, are measured using a pressure sensor and inertial sensors, respectively. The heading ψ is measured using a compass and integrated with DVL measurements to propagate estimates of x and y, which will drift over time. The motion of the AUV $\boldsymbol{x}_i$ can be modeled as a Gaussian distribution $\boldsymbol{x}_i = \mathcal{N}(f(\boldsymbol{x}_{i-1}, \boldsymbol{u}_i), \Sigma_i)$, where $\boldsymbol{u}_i$ is the vehicle's control input, $f(\cdot)$ is its motion model and Σ_i is the covariance of the additive Gaussian noise that parameterizes the uncertainty of the DR.

A. SSS Measurement Model

A landmark on the seafloor $\boldsymbol{l}$ gives a range measurement, namely, the slant range, r_s, and a bearing measurement from the constraint that $\boldsymbol{l}$ lays within the horizontal sensor opening in the y–z plane. These two measurements paired together can be written as

$$\boldsymbol{z}_m = \begin{pmatrix} r_m \\ 0 \end{pmatrix} = \hat{\boldsymbol{z}}_m + \boldsymbol{\eta} = \begin{pmatrix} \sqrt{\boldsymbol{\pi}(\boldsymbol{l}_m) \cdot \boldsymbol{\pi}(\boldsymbol{l}_m)} \\ (1,0,0) \cdot \boldsymbol{\pi}(\boldsymbol{l}_m) \end{pmatrix} + \boldsymbol{\eta} \quad (11)$$

where $\boldsymbol{l}_m \in \mathbb{R}^3$ is the 3-D landmark in the world frame, r_m is the slant range of $\boldsymbol{l}_m$ and $\boldsymbol{\eta}$ is the measurement noise. $\boldsymbol{\pi}(\cdot)$ is a function that transforms a 3-D landmark from world frame to the sensor frame (denoted by s)

$$\boldsymbol{\pi}(\boldsymbol{l}) = {}^s\bar{\boldsymbol{l}} = {}^p\boldsymbol{T}_s^{-1} \cdot \boldsymbol{T}_p^{-1} \cdot \bar{\boldsymbol{l}}. \quad (12)$$

Here $\boldsymbol{T}_p \in \mathrm{SE}(3)$ denotes the AUV body pose at current ping (denoted by p) that contains the mth keypoint, and ${}^p\boldsymbol{T}_s$ is the transformation from AUV body frame to its sensor frame. $\bar{\boldsymbol{l}} \in \mathbb{E}^3$ denotes the homogeneous representation of $\boldsymbol{l}$.

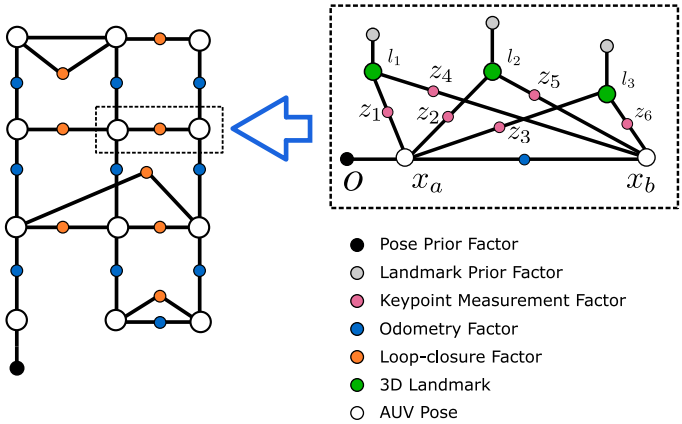


Fig. 3. Factor graph formulation of the proposed framework. Left: Pose graph of the global optimization. Right: Factor graph for submap-based relative pose estimation with the elevation prior.

B. Submap-Based Relative Pose Estimation With a Prior Map

For a submap-based approach, we reasonably assume that the DR error within submaps is small enough to be neglected. Then for (12), we need one more transformation from the center pose of the submap to the pose at the ping corresponding to the landmark, ${}^p\boldsymbol{T}_c$

$$\boldsymbol{\pi}(\boldsymbol{l}) = {}^s\bar{\boldsymbol{l}} = {}^c\boldsymbol{T}_s^{-1} \cdot {}^p\boldsymbol{T}_c^{-1} \cdot \boldsymbol{T}_p^{-1} \cdot \bar{\boldsymbol{l}} \quad (13)$$

where ${}^c\boldsymbol{T}_s$ now is the sensor offset, which is usually considered as known. Now we can describe the two-view submap-based sonar optimization as the following. Given a submap, S_a, consisting of N_a sidescan pings which has an overlapping area (containing M_s landmarks) with another submap, S_b with N_b pings, we denote the poses of the center ping for S_a and S_b as $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$, respectively. We formulate this optimization with an a priori map as a factor graph [see Fig. 3(right)], and estimate poses of the center of the two submaps as well as M_s 3-D point landmarks, $X = [\boldsymbol{x}_a, \boldsymbol{x}_b, \boldsymbol{l}_1, \boldsymbol{l}_2, \ldots, \boldsymbol{l}_{M_s}]$, by solving a *nonlinear least-squares* (NLS) problem under Gaussian noise with a *maximum a posteriori* (MAP) estimator

$$\begin{aligned} X^* = \underset{X}{\operatorname{argmin}} & \sum_{m=1}^{M_s} \|\hat{\boldsymbol{z}}_m - h(\boldsymbol{x}_a, \boldsymbol{l}_m)\|^2_{\Sigma_{am}} \\ & + \sum_{m=M_s+1}^{2M_s} \|\hat{\boldsymbol{z}}_m - h(\boldsymbol{x}_b, \boldsymbol{l}_m)\|^2_{\Sigma_{bm}} \\ & + \|\hat{\boldsymbol{x}}_b - \boldsymbol{x}_b\|^2_{\Sigma_b} + \phi_x(\boldsymbol{x}_a) + \sum_{m=1}^{M_s} \phi_l(\boldsymbol{l}_m). \end{aligned} \quad (14)$$

Here, we use the notation $\|x\|^2_\Sigma := x^T\Sigma^{-1}x$ to denote Mahalanobis distance. $h(\cdot)$ is the measurement model in (11), $\hat{\boldsymbol{x}}_b$ is calculated from the DR data and $\phi_x(\boldsymbol{x}_a)$ is the prior on $\boldsymbol{x}_a$ whose uncertainty approaches zero, meaning we treat $\boldsymbol{x}_a$ as fixed and only adjust $\boldsymbol{x}_b$. Note that $\phi_l(\boldsymbol{l}_m)$ is the prior on the landmarks obtained from the a priori map, which is to address the degeneracy illustrated in Fig. 1. As we discussed before, both $\boldsymbol{x}_b$ and $\boldsymbol{l}_m$ are unknown and they can be simultaneously adjusted to satisfy the

constraints without any priors on the landmarks, which means there are no unique solutions. To tackle this degeneracy, we propose to incorporate the information from sidescan data by employing the neural-rendering-based techniques outlined in Section III to construct an initial estimate of the bathymetry as a prior, providing an elevation prior for the landmarks, which is critical to solving the optimization robustly. The landmarks are marginalized out and the relative pose estimation from the MAP estimator is going to be added to the pose graph as LC constraints for localization.

C. Incremental Smoothing and Mapping

We use a pose graph formulation for AUV localization, where the only variables are poses, so that we could exploit the sparsity introduced by this formulation and solve the optimization through iSAM [29].

The joint probability distribution of the AUV poses $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N]$, the LC constraints $Z = [\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_M]$ between two poses, and the DR constraints between successive poses $U = [\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_N]$ are given by

$$P(X, U, Z) = P(\boldsymbol{x}_0) \prod_{i=1}^{N} P(\boldsymbol{x}_i | \boldsymbol{x}_{i-1}, \boldsymbol{u}_i) \prod_{j=1}^{M} P(\boldsymbol{x}_{b_j} | \boldsymbol{x}_{a_j}, \boldsymbol{z}_j). \tag{15}$$

A MAP estimate of the AUV poses attempts to find the most likely $\boldsymbol{x}$ by solving the optimization incrementally so that we can avoid the large drift over a long time.

V. Full System Overview

Finally, we give an overall view of the whole framework, combining the method for SSS SLAM with the elevation prior introduced in Section IV and the bathymetry reconstruction using neural rendering, from Section III.

We assume data association $D_a = \{(\alpha_m, \beta_m, \gamma_m)\}_{m=1}^{M}$, is known, where the measurement of landmark $\boldsymbol{l}_{\gamma_m}$ is obtained from sensor state $\boldsymbol{x}_{\alpha_m}$ and $\boldsymbol{x}_{\beta_m}$. Algorithm 2 outlines the NeuRSS framework given SSS, altimeter readings, DR $\{\hat{\boldsymbol{x}}_i^0\}$ and data association D_a, which can be run iteratively to improve navigation estimates, represented by a pose graph G and the bathymetry estimate, represented by a SIREN Φ_θ. Line 3 trains a SIREN given the current navigation estimate using Algorithm 1, where the resultant estimated bathymetry Φ_θ^j is used to compute the landmarks 3-D positions $\mathcal{L}^j$ given D_a in line 4. Lines $5-25$ describe the iSAM algorithm to estimate the vehicle's poses at all pings' timestamps ($i = 1$ to N) using Φ_θ^j as the elevation prior. Specifically, the pose graph G is constructed at every ping using DR constraints between the two consecutive pings (lines 7 and 11). At the same time, if the submap S_b with size N_b ending at the current ping i has more than $thres1$ landmarks $\mathcal{L}_\gamma^j$, which can also be observed from a previous submap S_a with size N_a, a LC is triggered (lines $12-24$). Lines $16-22$ describe that the relative pose from the center of S_a to the center of S_b is estimated using RANSAC, where for each iteration r, M_s landmarks are randomly sampled from $\mathcal{L}_\gamma^j$ (line 17), i.e., $\mathcal{L}_s^j$. Given the initial estimate of $\hat{\boldsymbol{x}}_a^j, \hat{\boldsymbol{x}}_b^j$ before solving the NLS optimization, we can compute the triangulation error e^r_{before} on $\mathcal{L}_s^{j\complement}$ (lines $18-19$). Line 20 solves the NLS optimization using (14) with the elevation prior ϕ_l using the Levenberg-Marquardt (LM) algorithm to get the optimal estimate $\boldsymbol{x}_a^{*,r}, \boldsymbol{x}_b^{*,r}$, which are then used to compute the triangulation error e^r_{after}. After a fixed iterations of RANSAC, if the estimated pose $\boldsymbol{x}_a^{*,r^*}, \boldsymbol{x}_b^{*,r^*}$ that gives the smallest $e^r_{\text{after}}/e^r_{\text{before}}$, is smaller than a threshold $0 < thres2 < 1$, this relative pose estimate is added to G, line 24. G is updated for every ping (line 25) until the end of the survey to obtain the pose estimates for the full survey, $\{\hat{\boldsymbol{x}}_i^{j+1}\}$, line 26. Lines $2-26$ can be run several iterations until both of the navigation estimate $\{\hat{\boldsymbol{x}}_i^J\}$ and bathymetry estimate Φ_θ^J are converged.

Algorithm 2: NeuRSS Framework.

Data: $D = \{I_{i,n}, \hat{\boldsymbol{x}}_i^0, h_i\}, D_a = \{(\alpha_m, \beta_m, \gamma_m)\}$
Result: $\Phi_\theta, \Phi, R, A_i$

1 **for** *iteration* $j = 0$ *to* J **do**
2 $\quad G = \emptyset$
3 $\quad \Phi_\theta{}^j \leftarrow \{I_{i,n}, \hat{\boldsymbol{x}}_i^j, h_i\}$ using Algorithm 1
4 $\quad \mathcal{L}^j \triangleq \{\boldsymbol{l}_{\gamma_m}^j\} \leftarrow \{\hat{\boldsymbol{x}}_i^j\}, \Phi_\theta{}^j, D_a$
5 $\quad$ **for** *loop all pings* $i = 1$ *to* N **do**
6 $\quad\quad$ **if** $i < N_b$ **then**
7 $\quad\quad\quad G \leftarrow$ AddDRedge$(\hat{\boldsymbol{x}}_{i-1}^j, \hat{\boldsymbol{x}}_i^j)$
8 $\quad\quad\quad$ Continue
9 $\quad\quad \mathcal{B} = \{i - N_b, \ldots, i\} \cap \{\beta_m\}$
10 $\quad\quad$ **if** $|\mathcal{B}| < thres1$ **then**
11 $\quad\quad\quad G \leftarrow$ AddDRedge$(\hat{\boldsymbol{x}}_{i-1}^j, \hat{\boldsymbol{x}}_i^j)$
12 $\quad\quad$ **else**
13 $\quad\quad\quad \mathcal{A} = \{\alpha_m | \beta_m \in \mathcal{B}\}$
14 $\quad\quad\quad \mathcal{L}_\gamma^j = \{\boldsymbol{l}_{\gamma_m}^j \in \mathcal{L}^j | \beta_m \in \mathcal{B}\}$
15 $\quad\quad\quad a, b \leftarrow$ center $\mathcal{A}, \mathcal{B}$
16 $\quad\quad\quad$ **for** *RANSAC loop* $r = 0$ *to* R **do**
17 $\quad\quad\quad\quad \mathcal{L}_s^j = \{\boldsymbol{l}_{\gamma_m}^j | \gamma_m \in_R \mathcal{L}_\gamma^j\}$ where $|\mathcal{L}_s^j| = M_s$
18 $\quad\quad\quad\quad \mathcal{L}_s^{j\complement} = \{\boldsymbol{l}_{\gamma_m}^j \in \mathcal{L}_\gamma^j : \boldsymbol{l}_{\gamma_m}^j \notin \mathcal{L}_s^j\}$
19 $\quad\quad\quad\quad e^r_{\text{before}} \leftarrow \text{tri}_{\text{err}}(\mathcal{L}_s^{j\complement}, \hat{\boldsymbol{x}}_a^j, \hat{\boldsymbol{x}}_b^j)$
20 $\quad\quad\quad\quad$ Solve $\boldsymbol{x}_a^{*,r}, \boldsymbol{x}_b^{*,r}$ using Eq. 14
21 $\quad\quad\quad\quad e^r_{\text{after}} \leftarrow \text{tri}_{\text{err}}(\mathcal{L}_s^{j\complement}, \boldsymbol{x}_a^{*,r}, \boldsymbol{x}_b^{*,r})$
22 $\quad\quad\quad r^* = \text{argmin}\{e^r_{\text{after}}/e^r_{\text{before}}\}$
23 $\quad\quad\quad$ **if** $e^{r^*}_{after}/e^{r^*}_{before} < thres2$ **then**
24 $\quad\quad\quad\quad G \leftarrow$ AddLCedge$(\boldsymbol{x}_a^{*,r^*}, \boldsymbol{x}_b^{*,r^*})$
25 $\quad\quad$ update G
26 $\quad \{\hat{\boldsymbol{x}}_i^{j+1}\} \leftarrow \{\hat{\boldsymbol{x}}_i^j, D_a, \Phi_\theta{}^j\}$
27 **return** $\Phi_\theta^J, \Phi^J, R^J, A^J, \{\hat{\boldsymbol{x}}_i^J\}$

VI. Experiments

A. Data Sets and Vehicles

Two data sets have been collected with two vehicles (see Fig. 4) for testing the proposed approach. A nearshore surface vessel *MMT Ping* equipped with a real-time kinematic (RTK) GPS, a hull-mounted EdgeTech 4200 sidescan and a Reson SeaBat 7125 MBES has collected Data set 1, where the surveyed area contains a large mountain and a ridge. Data set 2 was collected by a Kongsberg Hugin 3000 AUV equipped with a Honeywell HG9900 inertial navigation system (INS), aided with a Nortek 500 kHz Doppler velocity log (DVL), an EdgeTech 2205 sidescan and a Kongsberg EM2040 MBES, where the surveyed terrain is relatively benign with a sinkhole on the seabed. Data set 1 was collected on the surface so that we could

(a) (b)

Fig. 4. MMT survey vessel ping (a) and Hugin AUV (b).

TABLE I
DATA SETS DETAILS

Survey	1	2
Vehicle	MMT Ping	Hugin
Avg altitude (m)	∼17	∼19
Duration	3 hr	12 min
Trajectory (km)	7.3	0.7
Survey area	∼350 m × 300 m	∼300 m × 250 m
Bathymetry depth range (m)	9-25	78-92
MBES Bathymetry resolution (m)	0.5	1
Number of sidescan pings	58 217	3350
Sidescan range (m)	∼50	∼170
Sidescan frequency (kHz)	850	410

have GPS for accurate positioning. The MBES data are used as the ground truth bathymetry during evaluation. Data set 2 was collected as a part of a long mission (24 h) without any external navigation aid, where the DR of Hugin will have inevitable drift. However, since Data set 2 is only 12 min and given the high quality of Hugin's INS system, we reasonably assume the relative trajectory error is very little within Data set 2. Therefore, the DR data and MBES collected are used as the ground truth for evaluation. The main characteristics of the two data sets are summarized in Table I.

B. SIREN Training Details

In this work, we leave the architecture of SIREN as it is in [6], a 5-layer MLP with hidden layer size 128. As for the sidescan, we downsample all the raw data to 512 bins from 5734 bins (Data set 1) and 20816 bins (Data set 2), where all 512 bins are used in the loss calculation. All positional data is normalized to $[-1,1]$. For SIREN's initialization, instead of initializing the weights randomly as in [6], we first train SIREN for ten epochs using the heightmap linearly interpolated from the sparse altimeter readings, so that the training later could converge faster. As for the training hyperparameters, we train 800 epochs using Adam optimizer with a learning rate 2×10^{-4} that linearly decays by a factor of 0.995 every 2 epochs. For each mini-batch, we randomly selected 200 SSS pings and 800 altimeter readings. For the GD, we optimize 30 steps with $\lambda = 2.0$. To compute the transmittance T, we sample 30 points along the ray backwards 2 m.

C. Evaluation of the NeuRSS Framework

In this section, we seek to assess and validate the amenability of the proposed NeuRSS framework on large industrial-scale surveys. For this, we have designed two sets of experiments. For both experiment setups, we corrupt the ground truth trajectory by adding Gaussian noise to the yaw of the vehicle, $5e-3\,rd/s$, to simulate the vehicle's navigation estimates (DR) that inherent cumulative drift. As for the front-end, we generate the "perfect data association" using ground truth trajectory and bathymetry. We first construct SSS submaps every 200 pings along the trajectory, and for two submaps that have sufficient overlaps (usually from sections of two parallel adjacent track lines), we use SIFT features to extract feature points from one SSS submap, the reference frame. Then we associate the feature points in the reference waterfall image to their 3-D landmarks coordinates using ground truth bathymetry (sidescan draping [30]) and then projected them back to the other SSS submap, the current frame.

TABLE II
SLAM ATE (M)

	$\{\hat{\boldsymbol{x}}_i^0\}$ DR	$\{\hat{\boldsymbol{x}}_i^1\}$ No Prior	$\{\hat{\boldsymbol{x}}_i^1\}$ Linear	$\{\hat{\boldsymbol{x}}_i^1\}$ SIREN
Dataset 1	9.751	8.563	6.232	**2.074**
Dataset 2	7.346	5.349	2.551	**2.060**

In Experiment 1, we have gauged the effects of the landmark elevation prior in the back-end NLS optimization and the SLAM performance on both data sets. We run the NLS optimization on the corrupted trajectory with three setups: no landmark elevation prior, an elevation prior provided using linear interpolation between the altimeter readings of reference frame and current frame, and the elevation prior given by the estimated SIREN bathymetry from SSS using our neural rendering approach. The results of the optimization have been compared based on two error metrics, relative translation error (RTE) translation and absolute trajectory error (ATE), against the ground truth.

In Experiment 2, we have demonstrated how the proposed approach can be used iteratively to improve navigation estimates and subsequently for bathymetric mapping with SSS. On both data sets, we have applied $J = 2$ iterations of Algorithm 2 starting from the corrupted trajectory ($\{\hat{\boldsymbol{x}}_i^0\}$, DR). We compare the estimated trajectories and bathymetric maps against their ground truth, respectively.

VII. RESULTS

A. Exp 1: Elevation Prior on NLS Optimization

As mentioned in Algorithm 2, line 24, a threshold $thres2$ is used to determine if the relative pose estimation from NLS optimization should be added to the pose graph G as LC constraints. This parameter controls whether a LC constraint is considered to have a robust relative pose estimate after optimization. Fig. 5 shows the RTE with different $thres2$ for all submap pairs from Data set 1 after NLS optimization using no elevation prior, linear interpolated elevation prior or the elevation prior from the estimated SIREN bathymetry from SSS. We can observe that the relative pose estimation is not robust at all when no elevation prior is provided, indicating the elevation degeneracy case. Note that we obtain much more robust performance on relative pose estimation when we use Algorithm 1 to estimate bathymetry and treat it as a map known a priori, providing the elevation prior.

TABLE III
RESULTS WITH SIREN

	$\{\hat{\boldsymbol{x}}_i^0\}$ (DR) ATE	$\{\hat{\boldsymbol{x}}_i^1\}$ ATE	$\{\hat{\boldsymbol{x}}_i^2\}$ ATE	Φ_θ^0 Err.	Φ_θ^0 Abs. Err.	Φ_θ^1 Err.	Φ_θ^1 Abs. Err.	Φ_θ^2 Err.	Φ_θ^2 Abs. Err.
Dataset 1	9.751	2.074	0.822	-0.037±0.259	0.124	-0.044±0.161	0.082	-0.025±0.155	0.069
Dataset 2	7.346	2.060	1.669	0.152±0.536	0.373	0.139±0.498	0.287	0.137±0.492	0.284

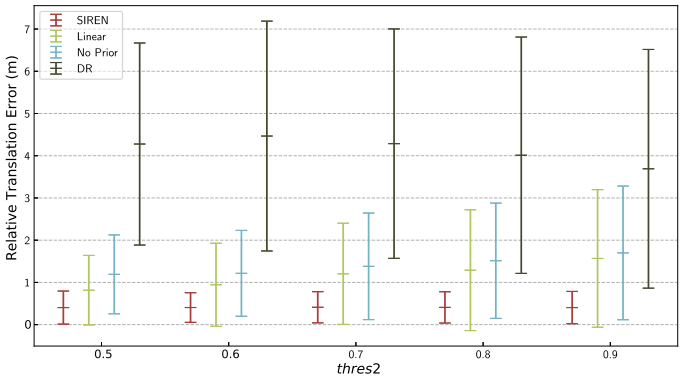


Fig. 5. Mean and standard deviation of the RTE at $thres2 = 0.5, 0.6, 0.7, 0.8, 0.9$. Black bars are the DR errors before optimization. Red bars and green bars are errors after optimization using SIREN estimates and linear interpolation of altimeter readings as the elevation prior, respectively. Blue bars are the errors without using any elevation priors.

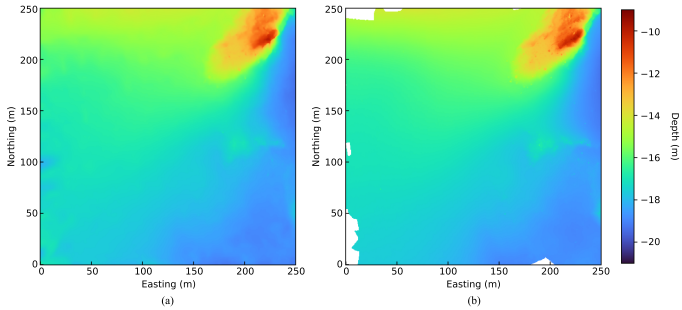


Fig. 6. Final estimated bathymetry from data set 1. (a) Bathymetry from SSS. (b) Bathymetry from MBES.

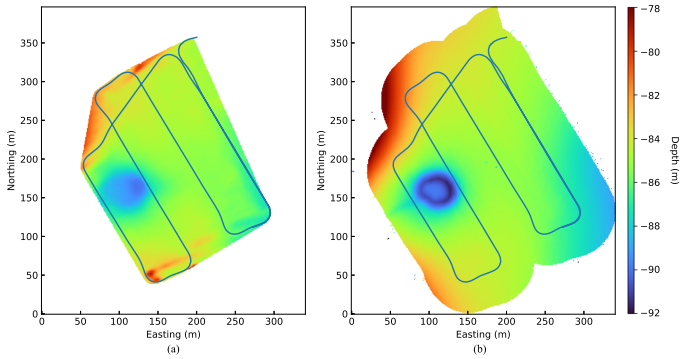


Fig. 7. Final estimated bathymetry from data set 2 (AUV trajectory in blue). Note that we use the SSS trajectory to create a boarder of the reconstructed area because the quality outside of the boarder degrades fast due to under-constraints. (a) Bathymetry from SSS. (b) Bathymetry from MBES.

This indicates that the converged bathymetry estimated from SSS using neural rendering, though having errors due to the inaccurate positioning, still provides a more valuable approximation compared to simply interpolating all the altimeter readings, due to the rich information extracted from SSS imagery.

Table II shows the ATE when we run the full system (for one iteration) using both data sets, where we can also observe that using the proposed method as the elevation prior gives the smallest trajectory error (in bold) in both cases. Note that the terrain in Data set 1 is much more complex than Data set 2, which is the main reason that the performance using linear interpolation to provide the elevation prior is much worse than that using our estimated SIREN bathymetry on Data set 1 (6.232 m versus 2.074 m), compared to Data set 2 (2.551 m versus 2.060 m).

B. Exp 2: Iterative Refinement of Navigation and Mapping

Table III shows the SLAM ATE (m) and the error (m) on reconstructed bathymetry Φ_θ^j at each iteration, $j = 0, 1, 2$. Inspecting the ATE, on both data sets, we can observe that another iteration ($j = 2$) could further improve the trajectory estimates, but again, the improvement on Data set 2 is less than that on Data set 1, due to the terrain in Data set 2 being relatively benign and the features on the seabed mainly being clustered around the sinkhole. We can also notice that the errors on the estimated bathymetry in Table III also decrease each iteration because of the better estimates on the positioning of the SSS, resulting in a final 0.069 m MAE on Data set 1 and 0.284 m MAE on Data set 2. Fig. 8 shows the estimated trajectory compared against the ground truth (red) for Data set 1, the entire mission in (a) and zoom-in section in (b), where it illustrates how our proposed approach can iteratively improve the navigation estimates from DR ($\{\hat{\boldsymbol{x}}_i^0\}$, blue) with ATE 9.751 m to $\{\hat{\boldsymbol{x}}_i^1\}$ (orange) with ATE 2.074 m and $\{\hat{\boldsymbol{x}}_i^2\}$ (green) with ATE 0.822 m.

Fig. 6 shows heightmaps of the final estimated bathymetry from SSS, Φ_θ^2, and the ground truth bathymetry constructed from MBES, for Data set 1. We can see that the topographic details of the mountain and the ridge are fairly well reconstructed. We zoom in and show the ridge in 3-D as a mesh in Fig. 9, where one can notice the effect of sonar pose estimates on the quality of the reconstructed topology. We can observe that, Φ_θ^2, in Fig. 9(b) manages to reconstruct more details on the abyssal hills on top of the ridge compared to Φ_θ^1 in (a). However, compared to MBES in (c), we can see the limits of the proposed approach, even though the MAE over the whole surveyed area is less than 10 cm.

For Data set 2, the final estimated bathymetry is shown in Fig. 7(a) with AUV transit in blue, together with the ground truth in (b). Note that the sinkhole can be clearly seen in the estimated bathymetry, but the shape and dimension have noticeable errors, compared to the bathymetric map from MBES. This is largely because reconstructing bathymetry from SSS is an ill-posed optimization problem, especially in this case, where we

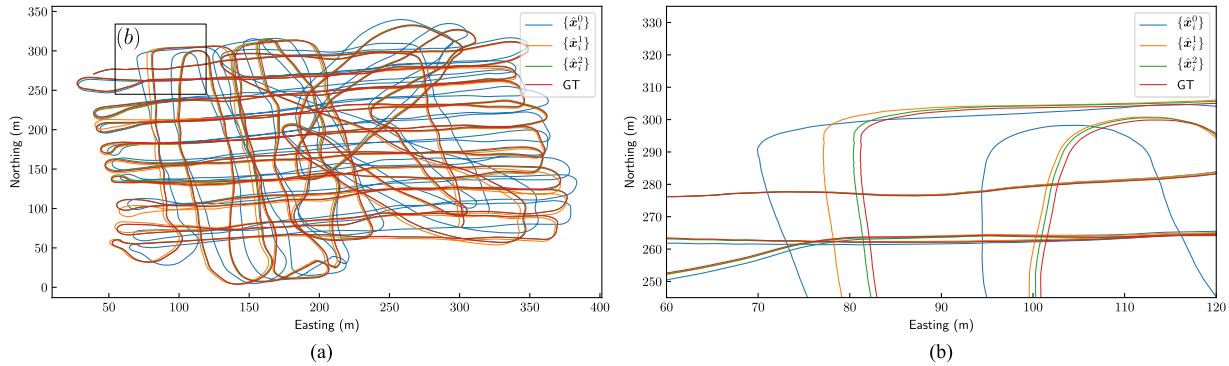


Fig. 8. Optimized trajectory estimates of the vehicle (orange and green), DR estimates (blue), as well as the ground truth for mission 1.

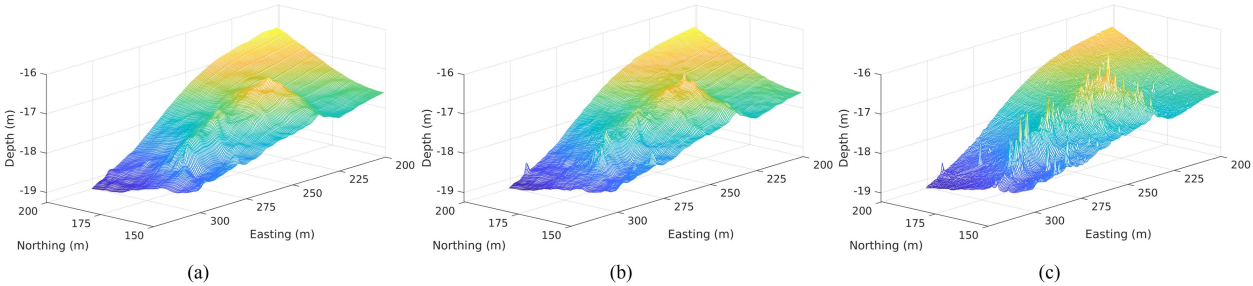


Fig. 9. Zoomed in sections of bathymetry in Data set 1. (a) Φ_θ^1 from SSS. (b) Φ_θ^2 from SSS. (c) Bathymetry from MBES.

lack sufficient repeated observations from different viewpoints (unlike the case in Data set 1, see Fig. 8). For similar reason, we can also see that the quality of the reconstructed bathymetric map deters at the perimeter. However, one can reasonably speculate that the reconstruction quality could improve given more survey lines, for example, perpendicular to the ones in Fig. 7.

VIII. Conclusion

We have presented NeuRSS, a neural rendering-based framework for reconstructing bathymetry from SSS data and DR estimates in a self-supervising manner. The proposed framework has been tested on two field data sets collected with different robots. We demonstrated that the bathymetric estimates from SSS using neural rendering play an important role in addressing the elevation degeneracy in the NLS optimization for estimating the relative poses between two submaps. Compared with interpolating between altimeter readings, the elevation prior provided by incorporating SSS with neural rendering results in much more robust optimization, especially when the terrain is complex, as in Data set 1. We also show that the proposed approach can be run iteratively to improve navigation and bathymetric estimates for high-quality bathymetric mapping using SSS data from standard surveys.

The current major limitation of this work is that we did not address the data association problem in the front-end of the SSS SLAM pipeline. Automatic data association in SSS imagery is still an active and open research question, largely due to the unique challenges that come from the special sensor modality of SSS. In [21] and [25], canonical transformation under flat seafloor assumption has been applied to SSS images to reduce geometric and radiometric distortions before feature-based matching [21] and dense matching [25]. One possible future work is to incorporate the bathymetric estimates from neural rendering into SSS canonical transformation and apply similar matching approaches for automatic data association.

Another possible future work is to use MBES and SSS data for AUV SLAM and superresolution bathymetric mapping in a neural rendering framework, leveraging the strengths of both sensors, namely MBES's ability to directly measure the 3-D seafloor geometry and SSS's wide swath range and high-resolution imagery.

Another limitation is that our method is mostly suited to offline optimization instead of real-time applications. The main reason is that to train a SIREN to converge to a self-consistent bathymetric map with high quality and fidelity, we need the multiple repeated observations from SSS from different viewpoints. Nevertheless, if one can explore the idea of combining MBES and SSS at the same time, it is possible to run dense SLAM on an embedded platform in real time.

References

[1] R. Li and S. Pai, "Improvement of bathymetric data bases by shape from shading technique using side-scan sonar images," in *Proc. IEEE OCEANS Conf.*, Honolulu, HI, USA, 1991, vol. 1, pp. 320–324.

[2] D. Langer and M. Hebert, "Building qualitative elevation maps from side scan sonar data for autonomous underwater navigation," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1991, pp. 2478–2483, vol. 3.

[3] A. E. Johnson and M. Hebert, "Seafloor map generation for autonomous underwater vehicle navigation," *Auton. Robots*, vol. 3, no. 2, pp. 145–168, 1996.

[4] E. Coiras, Y. Petillot, and D. M. Lane, "Multiresolution 3-D reconstruction from side-scan sonar images," *IEEE Trans. Image Process.*, vol. 16, no. 2, pp. 382–390, Feb. 2007.

[5] Y. Xie, N. Bore, and J. Folkesson, "Bathymetric reconstruction from sidescan sonar with deep neural networks," *IEEE J. Ocean. Eng.*, vol. 48, no. 2, pp. 372–383, Apr. 2023.

[6] Y. Xie, N. Bore, and J. Folkesson, "Neural network normal estimation and bathymetry reconstruction from sidescan sonar," *IEEE J. Ocean. Eng.*, vol. 48, no. 1, pp. 218–232, Jan. 2023.

[7] N. Bore and J. Folkesson, "Neural shape-from-shading for survey-scale self-consistent bathymetry from sidescan," *IEEE J. Ocean. Eng.*, vol. 48, no. 2, pp. 416–430, Apr. 2023.

[8] Y. Xie, N. Bore, and J. Folkesson, "Sidescan only neural bathymetry from large-scale survey," *Sensors*, vol. 22, no. 14, 2022, Art. no. 5092.

[9] M. Qadri, M. Kaess, and I. Gkioulekas, "Neural implicit surface reconstruction using imaging sonar," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2023, pp. 1040–1047.

[10] Y. Xie, G. Troni, N. Bore, and J. Folkesson, "Bathymetric surveying with imaging sonar using neural volume rendering," *IEEE Robot. Autom. Lett.*, vol. 9, no. 9, pp. 8146–8153, Sep. 2024.

[11] I. Torroba, C. I. Sprague, N. Bore, and J. Folkesson, "PointNetKL: Deep inference for GICP covariance estimation in bathymetric SLAM," *IEEE Robot. Autom. Lett.*, vol. 5, no. 3, pp. 4078–4085, Jul. 2020.

[12] T. A. Huang and M. Kaess, "Towards acoustic structure from motion for imaging sonar," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 758–765.

[13] T. A. Huang and M. Kaess, "Incremental data association for acoustic structure from motion," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 1334–1341.

[14] E. Westman, A. Hinduja, and M. Kaess, "Feature-based SLAM for imaging sonar with under-constrained landmarks," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 3629–3636.

[15] E. Westman and M. Kaess, "Degeneracy-aware imaging sonar simultaneous localization and mapping," *IEEE J. Ocean. Eng.*, vol. 45, no. 4, pp. 1280–1294, Oct. 2020.

[16] J. Wang, T. Shan, and B. Englot, "Underwater terrain reconstruction from forward-looking sonar imagery," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2019, pp. 3471–3477.

[17] M. F. Fallon, M. Kaess, H. Johannsson, and J. J. Leonard, "Efficient AUV navigation fusing acoustic ranging and side-scan sonar," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 2398–2405.

[18] L. Bernicola, D. Gueriot, and J.-M. L. Caillec, "A hybrid registration approach combining SLAM and elastic matching for automatic side-scan sonar mosaic," in *Proc. IEEE OCEANS Conf.*, 2014, pp. 1–5.

[19] M. Issartel, D. Guériot, N. Aouf, and J.-M. L. Caillec, "Robust SLAM for side scan sonar image mosaicking," in *Proc. IEEE OCEANS Conf.*, 2017, pp. 1–10.

[20] Y. Xu, R. Zheng, S. Zhang, and M. Liu, "Robust inertial-aided underwater localization based on imaging sonar keyframes," *IEEE Trans. Instrum. Meas.*, vol. 71, 2022, Art. no. 7501812.

[21] 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, pp. 674–683, 2023.

[22] I. T. Ruiz, Y. Petillot, and D. M. Lane, "Improved AUV navigation using side-scan sonar," in *Proc. MTS/IEEE OCEANS Conf.*, 2003, vol. 3, pp. 1261–1268.

[23] I. T. Ruiz, S. d. Raucourt, Y. Petillot, and D. M. Lane, "Concurrent mapping and localization using sidescan sonar," *IEEE J. Ocean. Eng.*, vol. 29, no. 2, pp. 442–456, Apr. 2004.

[24] S. Reed, I. T. Ruiz, C. Capus, and Y. Petillot, "The fusion of large scale classified side-scan sonar image mosaics," *IEEE Trans. Image Process.*, vol. 15, no. 7, pp. 2049–2060, Jul. 2006.

[25] J. Zhang, Y. Xie, L. Ling, and J. Folkesson, "A dense subframe-based SLAM framework with side-scan sonar," 2023, *arXiv:2312.13802*.

[26] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, and G. Wetzstein, "Implicit neural representations with periodic activation functions," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2020, vol. 33 pp. 7462–7473.

[27] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing scenes as neural radiance fields for view synthesis," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 405–421.

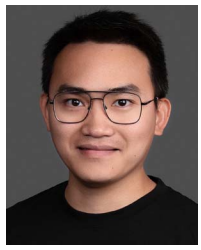
[28] P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang, "NeuS: Learning neural implicit surfaces by volume rendering for multi-view reconstruction," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2021, vol. 34, pp. 27171–27183.

[29] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "iSAM2: Incremental smoothing and mapping using the Bayes tree," *Int. J. Robot. Res.*, vol. 31, no. 2, pp. 216–235, 2012.

[30] N. Bore and J. Folkesson, "Modeling and simulation of sidescan using conditional generative adversarial network," *IEEE J. Ocean. Eng.*, vol. 46, no. 1, pp. 195–205, Jan. 2021.

Yiping Xie received the M.Sc. degree in computer science and the Ph.D. degree in robotics from Royal Institute of Technology (KTH), Stockholm, Sweden, in 2019, and 2024, respectively.

He is currently a Researcher with the Swedish Maritime Robotics Centre (SMaRC) and Robotics Perception and Learning (RPL) Lab, KTH. His research interests include perception for underwater robots, bathymetric mapping, and localization with sonars.



Jun Zhang received the B.Eng. degree in electrical engineering and automation and the M.Sc.Eng. degree in vehicle operation engineering from Northwestern Polytechnical University (NPU), Xi'an, China, in 2012 and 2015, respectively, and the Ph.D. degree in engineering from the Australian National University (ANU), Canberra, ACT, Australia, in 2021.

He is a research staff with the Institute of Computer Graphics and Vision, Graz University of Technology, Graz, Austria. His research interests include aerial/underwater robot localization and 3-D reconstruction, in particular, simultaneous localization and mapping (SLAM) using acoustic and/or visual sensors.

Nils Bore received the M.Sc. degree in mathematical engineering from the Faculty of Engineering, Lund University, Lund, Sweden, in 2012, and the Ph.D. degree in computer vision and robotics from the Robotics Perception and Learning Lab, Royal Institute of Technology (KTH), Stockholm, Sweden, in 2018.

He is currently an AI Researcher with Ocean Infinity. In his work at OI, he and his team are developing solutions for multi-beam alignment and bathymetric SLAM. His research interests include robotic sensing and mapping, with recent scientific output focusing on applications of specialized neural networks to underwater sonar data.

John Folkesson (Senior Member, IEEE) received the B.A. degree in physics from Queens College, City University of New York, New York, NY, USA, in 1983, and the M.Sc. degree in computer science, and the Ph.D. degree in robotics from the Royal Institute of Technology (KTH), Stockholm, Sweden, in 2001 and 2006, respectively.

He is currently an Associate Professor of robotics with the Robotics, Perception and Learning Lab, Center for Autonomous Systems, KTH. His research interests include navigation, mapping, perception, and situation awareness for autonomous robots.