Tightly Coupled, Graph-Based DVL/IMU Fusion and Decoupled Mapping for SLAM-Centric Maritime Infrastructure Inspection

Alexander Thoms, *Graduate Student Member, IEEE*, Gabriel Earle, *Graduate Student Member, IEEE*, Nicholas Charron, and Sriram Narasimhan, *Member, IEEE*

***Abstract*—In this article, we address the problem of simultaneous localization and mapping (SLAM)-centric maritime infrastructure inspection [using unmanned surface vehicles (USVs)] via novel approaches in tightly-coupled, graph-based DVL/IMU fusion and decoupled mapping. As our first contribution, we formalize the preintegration of linear velocity measurements, obtained by a Doppler velocity log (DVL), in combination with angular velocity measurements, obtained by an inertial measurement unit (IMU), as binary factors encoding relative position. To evaluate state estimation improvements imparted by DVL/IMU fusion, we implement our proposed factor within a state-of-the-art, graph-based lidar-visual-inertial (LVI) SLAM system as our second contribution. Accuracy and robustness improvements are demonstrated in simulation by comparing maximum *a posteriori* pose estimates with and without DVL/IMU fusion against ground truth poses. As our third contribution, we propose a map generation framework for downstream inspection applications decoupled from SLAM. In our framework, volumetric data (captured by sonar, lidar, etc.) is transformed into a common world coordinate frame using extrinsic calibrations and SLAM pose estimates as input. Our framework operates over the complete set of raw volumetric data, whereas SLAM systems (both online and offline) typically operate over a subset of down-sampled volumetric data. To address the processing of additional volumetric data, we present innovations in refined pose correction and staged filtering for user-controlled denoising. We experimentally evaluate our map generation framework against the LVI SLAM system adopted for this study using real-world data and demonstrate improvements to map quality metrics important to inspection.**

***Index Terms*—Doppler velocity log, inspection, mapping, sensor fusion, simultaneous localization and mapping (SLAM), unmanned surface vehicles (USVs).**

Manuscript received 28 April 2022; revised 14 December 2022; accepted 3 April 2023. Date of publication 13 June 2023; date of current version 14 July 2023. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada through their Research Tools and Instruments Grant, and in part by the Graduate Fellowship program by the University of California, Los Angeles. *(Alexander Thoms, Gabriel Earle, and Nicholas Charron are co-first authors.) (Corresponding author: Sriram Narasimhan.)*

Associate Editor: B. Englot.

Alexander Thoms and Sriram Narasimhan are with the Sensing and Robotics for Infrastructure Laboratory, University of California Los Angeles, Los Angeles, CA 90095 USA (e-mail: adthoms@ucla.edu; snarasim@ucla.edu).

Gabriel Earle and Nicholas Charron are with the Structural Dynamics Identification and Controls Laboratory, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: gearle@uwaterloo.ca; nicholas.charron@uwaterloo.ca).

As an open source contribution, we release our mapping software with associated tools: https://github.com/nickcharron/3d_map_builder.

Digital Object Identifier 10.1109/JOE.2023.3265742

I. Introduction

A LARGE inventory of maritime infrastructure (such as water-spanning bridges, ports, and dams) exists at the land-water interface, presenting an environment where vulnerabilities and deterioration mechanisms occur above and below the surface. Below-surface inspection has traditionally centered on the use of human divers to manually probe and photograph the condition of submerged components as a means of qualitative assessment [1]. In many cases, visibility, safety, and cost factors preclude the use of divers or limit their ability to perform inspection-related tasks. Above-surface inspections encounter similar challenges, with access and scale being limiting factors. As a result, infrastructure management agencies [2] have identified the need for scalable technologies that enable the quantitative data collection of these assets in adverse conditions, including global navigation satellite system (GNSS) deprivation.

To address the above need, we develop an unmanned surface vehicle (USV) capable of producing high-quality volumetric maps (both above and below the surface of water) for maritime infrastructure inspection. For state estimation, a graph-based simultaneous localization and mapping (SLAM) approach is taken, enabling the USV to generate a volumetric map of its environment while simultaneously localizing itself relative to the generated map. Here, we define graph-based SLAM as SLAM formulations that can be modeled using factor graphs and posed as nonlinear least squares optimization problems [3]. For our study, we adopt a graph-based lidar-visual-inertial (LVI) SLAM system [4], which tightly couples the visual-inertial (VI) system from [5] with the lidar-inertial (LI) system from [6]. In SLAM, tight-coupling refers to the processing and fusion of measurements directly within a single estimator while loose-coupling refers to the fusion of state estimates (from separate estimators) within a common estimator. Graph-based LVI SLAM systems [4], [7], [8] have generally shown to improve accuracy and robustness over their VI and LI SLAM counterparts, hence its adoption. Note that it is common to distinguish SLAM systems from odometry systems with integrated loop-closing in the literature, though we do not make this distinction and use the terms interchangeably in this article.

Works extending graph-based LVI SLAM toward additional sensor fusion are limited, with [9] and [10] demonstrating

accuracy and robustness improvements over their base LVI system via tightly coupled fusion methods. Motivated by these observed improvements, our first contribution is the formulation of a tightly coupled Doppler velocity log (DVL)/inertial measurement unit (IMU) fusion method (herein referred to as DVL fusion) for graph-based SLAM. To accomplish this, we adapt the preintegrated velocity factor by Wisth et al. [9] to account for the scale factor error inherent to DVL measurements (herein, we refer to the adapted preintegrated velocity factor as a DVL factor). In doing so, we formalize the preintegration of linear velocity measurements, obtained by a DVL, in combination with angular velocity measurements, obtained by an IMU, to give a compound measurement on the USV's relative change in position. Our tightly coupled, graph-based fusion strategy departs from existing DVL/IMU fusion methods [11], [12], [13], [14], which are formulated for filter-based state estimators and do not allow for refined estimates of error propagation and bias terms if loosely coupled with a graph-based SLAM approach.

As our second contribution, we implement and evaluate our formalisms within the adopted LVI SLAM system. By tightly coupling our DVL factor to the existing graph, we extend the system state for an additional velocity bias term to account for any drift in DVL measurements, with the observation of this bias term made possible by our fusion strategy. The extended system is then compared against the base LVI SLAM system in simulation, where accuracy and robustness improvements (imparted by DVL fusion) are observed.

As our third contribution, we present a novel mapping approach, decoupled from SLAM, for downstream inspection applications. This contribution is motivated by SLAM systems (both online and offline) that downsample and discard volumetric data, which reduces map density and (consequently) degrades environmental structure. This loss of information directly conflicts with the prioritization of fine-scale resolution for inspection applications, justifying a separate map generation process. In our decoupled approach, the entire set of volumetric data (from any generic range sensor) is transformed into the world coordinate frame using extrinsic sensor calibrations and SLAM pose estimates as input. For keyframe-based SLAM systems, our method corrects high-rate, locally consistent odometry with low-rate, globally consistent maximum *a posteriori* (MAP) SLAM poses for refined pose estimation. To control for noise, filters can be applied to volumetric data subsets of a user-specified size to tune for filter performance. To evaluate our decoupled mapping framework, horizontally and vertically oriented lidars, two ruggedized RGB cameras, an IMU, and a multibeam profiling sonar are time-synchronized and integrated with our USV platform and deployed for data collection near a water-spanning bridge. The map generated by our framework is then qualitatively and quantitatively evaluated against the map produced by the adopted LVI SLAM system under the categories of density, noise, and retention of environmental structure. Quantitative results show our approach excels in all categories.

The rest of this article is organized as follows. Section II discusses related work; Section III presents our problem formulation in the context of the adopted LVI SLAM system from which we propose our extension; Section IV presents the mathematical framework of our proposed factor along with its implementation; Section V evaluates the LVI SLAM system with and without DVL fusion in simulation; Section VI provides details on our proposed decoupled mapping framework along with results obtained from real-world experimentation. Finally, Section VII concludes this article.

II. Related Works

Research has demonstrated that robotic mapping systems, including those based on SLAM [15], [16], [17], [18] and structure from motion (SfM) [19], [20], [21], are effective in automating routine visual inspection of infrastructure; however, few studies have focused on maritime infrastructure inspection (above and below surface) using USVs. In [22], the authors outfit an USV with three 3-D SICK lidars, a Blueview MB2250 imaging sonar, and an inertial navigation system (INS) with integrated GNSS. Using a graph-based lidar SLAM approach, the authors reconstruct a water-spanning bridge by parenting sonar submaps to their respective lidar submaps after lidar submap global alignment. During submap generation, the USV's state is propagated forward (as an initial estimate for scan registration) via motion modeling, where the authors fuse IMU rates, DVL velocities, and GNSS positions within a Kalman filter framework. The authors further integrate an adaptive error model into their framework accounting for GNSS satellite count, GNSS reported dilution of precision, IMU reported velocity errors, and conditions on lidar returns. In [23], the authors forego the use of an INS with GNSS and instead rely on an iterative closest point (ICP) scan matching algorithm using a Velodyne HDL-64E. The authors note the work as preliminary and did not use a SLAM approach given the challenges of using surface reconstruction techniques on moving marine structures, limiting the quality of mapping and reconstruction. Similar to [22], the authors reconstructed the below surface map by transforming sonar data (also collected by a BlueView MB2250 imaging sonar) into the vehicle frame along the trajectory obtained by their ICP scan matching approach. In [24], the authors followed a similar sensor configuration to [22], substituting the imaging sonar with an Imagenex Model 837B Delta T multibeam profiling sonar. The authors additionally equip their USV with an omnidirectional camera, though visual information is neither used in their SLAM nor for 3-D reconstruction. By extracting planar features within lidar data, the authors incorporate these landmarks (via a data association step) within an extended-Kalman filter (EKF) SLAM approach. Filtered vehicle states from the EKF SLAM are used in an integrated 3-D reconstruction process, combining data from a horizontally oriented 3-D lidar, a vertically oriented 2-D lidar, and the multibeam profiling sonar into volumetric maps and (optionally) meshes. The 3-D reconstruction process remains coupled to the EKF SLAM approach, requiring downsampling of volumetric data to remain computationally tractable. Moreover, the 3-D reconstruction process aggregates volumetric data from all range sensors to a common time point, ignoring temporal offsets between sensors and motion compensation. Distinct from these works, our contributions advance

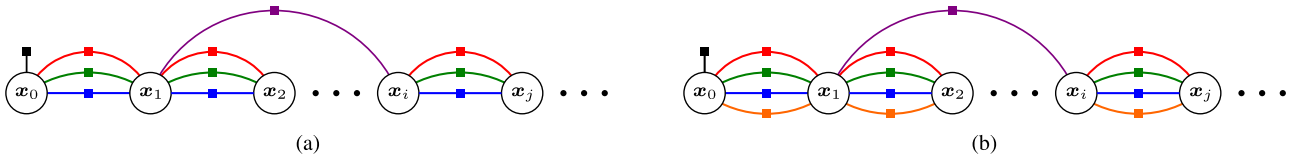


Fig. 1. (a) LVI-SAM factor graph. Source: Adapted from [4]. (b) LVI-SAM factor graph with our proposed DVL factors. The factors are prior (black), loop closure (purple), lidar odometry (red), visual-inertial odometry (green), IMU preintegration (blue), DVL preintegration (orange). Indices i, j represent consecutive keyframes.

the state of SLAM-centric maritime infrastructure inspection toward tightly-coupled, graph-based DVL fusion and decoupled mapping, the former improving state estimation accuracy and robustness and the latter improving inspection-related metrics for mapping.

III. Problem Formulation

For this study, we formalize and implement our proposed DVL factor within LVI-SAM [4]. LVI-SAM combines the lidar-inertial system (LIS) from LIO-SAM [6] with the visual-inertial system (VIS) from VINS-MONO [5] into an LVI odometry system with integrated loop-closing. The two subsystems can function independently when failure is detected in one of them, or jointly when enough features are detected [4]. We adopt LVI-SAM as it is open-source software, allowing our work to focus on the required extensions of graph-based SLAM for DVL fusion; moreover, LVI-SAM yields state-of-the-art results over previous LI and VI systems at the time of its release. Notation used to describe LVI-SAM and supporting machinery follows that presented in [8] and [9].

A. Existing Framework

Let the complete history of observed states $\mathcal{X}_k$ for all observable points up to time t_k be generalized as

$$\mathcal{X}_k \triangleq \{\boldsymbol{x}_i\}_{i \in \mathsf{K}_k} \tag{1}$$

where $\boldsymbol{x}_i$ represents the robot's state at time t_i, with i belonging to the set of keyframes K_k. In LVI-SAM, $\boldsymbol{x}_i$ is formalized as

$$\boldsymbol{x}_i \triangleq [\mathbf{R}_i, \mathbf{p}_i, \mathbf{v}_i, \mathbf{b}_i^{\omega}, \mathbf{b}_i^{a}] \tag{2}$$

where $\mathbf{R}_i \in \mathrm{SO}(3)$ is the rotation matrix, $\mathbf{p}_i \in \mathbb{R}^3$ is the position vector, $\mathbf{v}_i \in \mathbb{R}^3$ is the velocity vector, $\mathbf{b}_i^{\omega} \in \mathbb{R}^3$ and $\mathbf{b}_i^{a} \in \mathbb{R}^3$ are the bias vectors for IMU angular velocity and linear acceleration, respectively.

LVI-SAM solves the MAP of $\mathcal{X}_k$ by jointly optimizing visual-inertial odometry (from the VIS), lidar odometry, IMU preintegration, and loop closure constraints in a factor graph using iSAM2 [25]. Visual-inertial odometry constraints are obtained via joint minimization of visual reprojection and IMU residuals over a sliding window of visual features. From the LIS, lidar odometry constraints are derived from scan-matching, where LOAM [26] features are extracted from the current lidar keyframe and matched against a global lidar feature map. IMU preintegration constraints follow directly from [27], where high-frequency IMU measurements are preintegrated between consecutive keyframes. Lastly, loop-closure constraints are added when a candidate keyframe K_p (first detected by the VIS) matches with a nonconsecutive query keyframe K_q. The relative transformation between these keyframes is further validated and refined by the LIS. Fig. 1(a) illustrates LVI-SAM's factor graph, while (3) formalizes the MAP of $\mathcal{X}_k$ as a nonlinear least squares optimization problem

$$\mathcal{X}_k^{\star} = \arg\min_{\mathcal{X}_k} \|\mathbf{r}_0\|^2_{\boldsymbol{\Sigma}_0} + \sum_{(i,j)\in\mathsf{K}_k} \left(\sum_{\mathcal{Z}_k \in [\mathcal{L}_{ij}, \mathcal{C}_{ij}, \mathcal{I}_{ij}]} \|\mathbf{r}_{\mathcal{Z}_k}\|^2_{\boldsymbol{\Sigma}_{\mathcal{Z}_k}} + \sum_{\nu \in [\omega, a]} \left\| \mathbf{r}_{\mathbf{b}_{ij}^{\nu}} \right\|^2_{\boldsymbol{\Sigma}_{\mathbf{b}_{ij}^{\nu}}} \right) + \sum_{(p,q)\in\mathsf{K}_k} \|\mathbf{r}_{\mathcal{L}_{pq}}\|^2_{\boldsymbol{\Sigma}_{\mathcal{L}_{pq}}} \tag{3}$$

where $\mathbf{r}_0$ is the residual error on the robot's initial state; $\|\cdot\|^2_{\boldsymbol{\Sigma}}$ is the Mahalanobis distance with covariance $\boldsymbol{\Sigma}$; (i, j) are indices for consecutive keyframes; $\mathbf{r}_{\mathcal{Z}_k}$ are the residual errors associated with lidar odometry $\mathcal{L}_{ij}$, visual-inertial odometry $\mathcal{C}_{ij}$, and IMU preintegration $\mathcal{I}_{ij}$, which form the complete set of measurements $\mathcal{Z}_k$ for consecutive keyframes; $\mathbf{r}_{\mathbf{b}_{ij}^{\nu}}$ are the residuals associated with bias terms arising from IMU preintegration; (p, q) are indices for nonconsecutive keyframes as determined by loop-closure detection. We refer the reader to the original article [4] for further details on LVI-SAM's system architecture and existing factors.

B. Framework Extension

To extend LVI SLAM toward DVL fusion, we add a linear velocity bias term to $\boldsymbol{x}_i$. We further add our proposed DVL factors as binary factors linking consecutive keyframes as illustrated in Fig. 1(b). We extend the set of residuals to include those arising from preintegration $\mathbf{r}_{\mathcal{D}_{ij}}$ and linear velocity bias $\mathbf{r}_{\mathbf{b}_{ij}^{v}}$ such that: $\mathcal{Z}_k \in [\mathcal{L}_{ij}, \mathcal{C}_{ij}, \mathcal{I}_{ij}, \mathcal{D}_{ij}]$ and $\nu \in [\omega, a, v]$. Our extension generalizes to any graph-based SLAM formulation that follows (1), which may include additional observed states (ex. landmarks) as in [8] and [9].

IV. Proposed DVL Factor

We derive our DVL factor from preintegrated linear velocity measurements obtained from a DVL in combination with angular velocity measurements obtained from an IMU. A discussion of our proposed factor's mathematical formalism and implementation follows.

A. Mathematical Formalism

As per [27], angular velocity and linear acceleration measurements captured by an IMU within its frame $\mathtt{B}$ are formulated,

respectively, as

$$ {}_{\mathtt{B}}\tilde{\boldsymbol{\omega}}_{\mathtt{WB}}(t) = {}_{\mathtt{B}}\boldsymbol{\omega}_{\mathtt{WB}}(t) + \mathbf{b}^{\omega}(t) + \boldsymbol{\eta}^{\omega}(t) \tag{4} $$

$$ {}_{\mathtt{B}}\tilde{\mathbf{a}}(t) = \mathbf{R}^{\top}_{\mathtt{WB}}(t)({}_{\mathtt{W}}\mathbf{a}(t) - {}_{\mathtt{W}}\mathbf{g}) + \mathbf{b}^{a}(t) + \boldsymbol{\eta}^{a}(t) \tag{5} $$

where $\boldsymbol{\eta}^{\omega}(t)$ and $\boldsymbol{\eta}^{a}(t)$ are angular velocity and linear acceleration Gaussian white noise terms, respectively, ${}_{\mathtt{W}}\mathbf{g}$ is the constant gravity vector in $\mathtt{W}$, and bias terms $\{\mathbf{b}^{\nu}(t) \mid \nu \in [\omega, a]\}$ are slowly time-varying quantities that are modeled using Brownian motion [27]

$$ \dot{\mathbf{b}}^{\nu}(t) = \boldsymbol{\eta}^{b\nu}(t) \tag{6} $$

where the integration of (6) over the time interval $[t_i, t_j]$ between two consecutive key frames i and j gives

$$ \mathbf{b}^{\nu}_{j} = \mathbf{b}^{\nu}_{i} + \boldsymbol{\eta}^{b\nu d} \tag{7} $$

where $\mathbf{b}^{\nu}_{i} := \mathbf{b}^{\nu}(t_i)$ and $\boldsymbol{\eta}^{b\nu d}$ is discrete Gaussian white noise with covariance $\boldsymbol{\Sigma}_{\mathbf{b}^{\nu}_{ij}} \doteq \Delta t_{ij}\mathrm{Cov}(\boldsymbol{\eta}^{b\nu d})$. According to [12], linear velocity measurements captured by a DVL within its frame $\mathtt{B}_{\mathtt{o}}$ are defined as

$$ {}_{\mathtt{B}_{\mathtt{o}}}\tilde{\mathbf{v}}(t) = s_d\left({}_{\mathtt{B}_{\mathtt{o}}}\mathbf{v}(t)\right) + \mathbf{b}^{v}(t) + \boldsymbol{\eta}^{v}(t) \tag{8} $$

where s_d is a scaling factor accounting for environmental factors such as the variation of sound speed in water and aquatic terrain conditions, $\mathbf{b}^{v}(t)$ is linear velocity bias and follows the same Brownian motion model presented in (6), and $\boldsymbol{\eta}^{v}(t)$ is the linear velocity Gaussian white noise term.

As IMU and DVL measurements are practically captured in discrete time, (4) and (8) are rewritten using subscript notation while also dropping the frame decorator for brevity

$$ \tilde{\boldsymbol{\omega}}_k = \boldsymbol{\omega}_k + \mathbf{b}^{\omega}_k + \boldsymbol{\eta}^{\omega}_k \tag{9} $$

$$ \tilde{\mathbf{v}}_k = s_d\left(\mathbf{v}_k\right) + \mathbf{b}^{v}_k + \boldsymbol{\eta}^{v}_k. \tag{10} $$

Given [9] preintegrates linear velocity measurements (obtained via kinematic sensing) for their factor, we apply the author's preintegrated relative change in position measurement and noise in (11) and (12), respectively, with slight modification to account for s_d

$$ \Delta\tilde{\mathbf{p}}_{ij} \triangleq \sum_{k=i}^{j-1}\left[\Delta\tilde{\mathbf{R}}_{ik}\left(\frac{\tilde{\mathbf{v}}_k - \mathbf{b}^{v}_i}{s_d}\right)\Delta t\right] \tag{11} $$

$$ \begin{aligned} \delta\mathbf{p}_{ij} \triangleq \sum_{k=i}^{j-1}\Bigg[&\Delta\tilde{\mathbf{R}}_{ik}\frac{\boldsymbol{\eta}^{v}_k}{s_d}\Delta t \\ &- \Delta\tilde{\mathbf{R}}_{ik}\left(\frac{\tilde{\mathbf{v}}_k - \mathbf{b}^{v}_i}{s_d}\right)^{\wedge}\delta\boldsymbol{\phi}_{ik}\Delta t\Bigg] \end{aligned} \tag{12} $$

where k is an intermediary measurement between keyframes i and j, and $(\cdot)^{\wedge}$ maps vectors to skew symmetric matrices belonging to $\mathrm{SO}(3)$. Supplementary equations for the incremental preintegrated rotation measurement and associated noise are defined according to (13) and (14), respectively, and directly follow from [27]

$$ \Delta\tilde{\mathbf{R}}_{ik} \triangleq \prod_{k=i}^{k-1}\mathrm{Exp}\left(\left(\tilde{\boldsymbol{\omega}}_k - \mathbf{b}^{\omega}_i\right)\Delta t\right) \tag{13} $$

$$ \delta\boldsymbol{\phi}_{ik} \triangleq \sum_{k=i}^{k-1}\Delta\tilde{\mathbf{R}}^{\top}_{k+1,k}\mathbf{J}^{k}_{r}\boldsymbol{\eta}^{\omega}_{k}\Delta t \tag{14} $$

where $\mathbf{J}^{k}_{r}$ is the right Jacobian of $\mathrm{SO}(3)$ such that $\mathbf{J}^{k}_{r} \doteq \mathbf{J}^{k}_{r}((\tilde{\boldsymbol{\omega}}_k - \mathbf{b}^{\omega}_i)\Delta t)$. The residual error on relative position and its associated covariance are given by (15) and (16), respectively. Using (15), our proposed DVL factor residual is formalized as $\mathbf{r}_{\mathcal{D}_{ij}} \triangleq [\mathbf{r}^{\top}_{\Delta\tilde{\mathbf{p}}_{ij}}]$, where we exclude rotation residuals from our factor to avoid double counting the IMU signal. Note that $\mathbf{r}_{\mathcal{D}_{ij}}$ is evaluated over $\mathbf{b}^{\omega}$ and $\mathbf{b}^{v}$ and may be updated using the first order approximation $\Delta\tilde{\mathbf{p}}_{ij}(\mathbf{b}^{\omega}_i, \mathbf{b}^{v}_i)$, which is derived in Appendix A when small bias updates $\mathbf{b} \leftarrow \bar{\mathbf{b}} + \delta\mathbf{b}$ are made during optimization. Matrices $\mathbf{A}$ and $\mathbf{B}$ in (16) are derived in Appendix B

$$ \mathbf{r}_{\Delta\tilde{\mathbf{p}}_{ij}} = \mathbf{R}^{\top}_{i}\left(\mathbf{p}_j - \mathbf{p}_i\right) - \Delta\tilde{\mathbf{p}}_{ij}\left(\mathbf{b}^{\omega}_i, \mathbf{b}^{v}_i\right) \tag{15} $$

$$ \boldsymbol{\Sigma}^{\mathcal{D}}_{i,k+1} = \mathbf{A}\boldsymbol{\Sigma}^{\mathcal{D}}_{i,k}\mathbf{A}^{\top} + \mathbf{B}\boldsymbol{\Sigma}^{\mathcal{D}}_{\boldsymbol{\eta}}\mathbf{B}^{\top}. \tag{16} $$

B. Implementation

To preintegrate discrete angular (9) and linear (10) velocity measurements for our proposed DVL factor, time-ordered discrete DVL and IMU measurements, formalized in (17) and (18), respectively, are temporally reconciled to form time-ordered pseudosensor measurements according to (19)

$$ Q_{\mathcal{D}} = \{\mathcal{D}_l \triangleq \{{}_{\mathtt{B}}\tilde{\mathbf{v}}, t\}_l\}_{l\in[n]} \tag{17} $$

$$ Q_{\mathcal{I}} = \{\mathcal{I}_k \triangleq \{{}_{\mathtt{B}}\tilde{\boldsymbol{\omega}}, {}_{\mathtt{B}}\tilde{\mathbf{a}}, t\}_k\}_{k\in[m]} \tag{18} $$

$$ Q_{\mathcal{P}} = \{\mathcal{P}_k \triangleq \{{}_{\mathtt{B}}\tilde{\boldsymbol{\omega}}, {}_{\mathtt{B}}\tilde{\mathbf{v}}_{\text{interp}}, t\}_k\}_{k\in[m]} \tag{19} $$

where ${}_{\mathtt{B}}\tilde{\mathbf{v}}$ is obtained by transforming ${}_{\mathtt{B}_{\mathtt{o}}}\tilde{\mathbf{v}}$ into $\mathtt{B}$ using the extrinsic calibration $\mathbf{T}_{\mathtt{B}_{\mathtt{o}}\mathtt{B}}$ between the DVL and IMU; $\mathcal{D}_l$, $\mathcal{I}_k$, and $\mathcal{P}_k$ are time-stamped, discrete DVL, IMU, and pseudosensor measurements, respectively; $[n] \triangleq [1, \ldots, n]$ and $[m] \triangleq [1, \ldots, m]$ are sets of indexing integers. $\mathcal{D}_l$ and $\mathcal{I}_k$ are temporally reconciled by linearly interpolating between $\mathcal{D}_{l-1}$ and $\mathcal{D}_l$ at the time of IMU measurement t_k such that $t_{l-1} \leq t_k \leq t_l$. This interpolation step gives $\{{}_{\mathtt{B}}\tilde{\mathbf{v}}_{\text{interp}}\}_k$, which is then paired with $\{{}_{\mathtt{B}}\tilde{\boldsymbol{\omega}}, t\}_k$ for preintegration over all $\mathcal{P}_k$ between consecutive keyframes (which we effectively call DVL preintegration). In doing so, we maintain DVL preintegration at the IMU rate, thus, leveraging the granularity of angular velocity measurements. Algorithm 1 details the temporal reconciliation of DVL and IMU measurements in full. Note that algorithmically determining which DVL measurements bound t_k to obtain $\{{}_{\mathtt{B}}\tilde{\mathbf{v}}_{\text{interp}}\}_k$ requires the time difference between the first and last elements of $Q_{\mathcal{I}}$ to be equal to or greater than the time difference between consecutive DVL measurements.

To impart robustness against lidar and vision degeneracy, the extended LVI-SLAM system fully relies on IMU and DVL

Algorithm 1: Temporal Reconciliation of DVL and IMU Measurements for Our Proposed DVL Factor.

Input: $Q_{\mathcal{D}}, Q_{\mathcal{I}}, Q_{\mathcal{P}} = \varnothing$
Output: $Q_{\mathcal{P}}$
// Initialize first DVL and IMU measurements
$\mathcal{D}_{l-1} \leftarrow Q_{\mathcal{D}}$.Front()
$Q_{\mathcal{D}}$.PopFront()
$t_{k-1} \leftarrow Q_{\mathcal{I}}$.Front().GetTime()
while $t_{k-1} \leq \mathcal{D}_{l-1}$.GetTime() **do**
  $t_{k-1} \leftarrow Q_{\mathcal{I}}$.Front().GetTime()
  $Q_{\mathcal{I}}$.PopFront()
end while
// Perform temporal reconciliation
while $Q_{\mathcal{D}}$ and $Q_{\mathcal{I}}$ not empty **do**
  $\{{}_{\mathrm{B}}\tilde{\boldsymbol{\omega}}, {}_{\mathrm{B}}\tilde{\mathbf{a}}, t\}_k \leftarrow \mathcal{I}_k \leftarrow Q_{\mathcal{I}}$.Front()
  $t_{k-1} \leftarrow t_k$
  $Q_{\mathcal{I}}$.PopFront()
  $\mathcal{D}_l \leftarrow Q_{\mathcal{D}}$.Front()
  if $t_k \geq \mathcal{D}_l$.GetTime() **then**
    $\mathcal{D}_{l-1} \leftarrow \mathcal{D}_l$
    while $t_k \geq \mathcal{D}_l$.GetTime() **do**
      $Q_{\mathcal{D}}$.PopFront()
      $\mathcal{D}_l \leftarrow Q_{\mathcal{D}}$.Front()
    end while
  end if
  $\{{}_{\mathrm{B}}\tilde{\mathbf{v}}_{\text{interp}}\}_k \leftarrow \text{LinearlyInterpolate}(t_k, \mathcal{D}_l, \mathcal{D}_{l-1})$
  $\mathcal{P}_k \leftarrow \{\{{}_{\mathrm{B}}\tilde{\boldsymbol{\omega}}, t\}_k, \{{}_{\mathrm{B}}\tilde{\mathbf{v}}_{\text{interp}}\}_k\}$
  $Q_{\mathcal{P}}$.PushBack($\mathcal{P}_k$) // preintegrate in separate process
end while

factors when degeneracy is detected. To compare robustness improvements fairly, we also modify the base LVI-SLAM system to rely only on IMU factors during degeneracy. When both the LIS and VIS detect failure, bias terms are set to zero (in both the base and extended system) as they are no longer observable, and thus, cannot be optimized for. In testing the extended LVI-SLAM system, we observe preintegrated IMU odometry incurs higher drift than preintegrated DVL odometry and thus, set IMU factor residuals on relative position to zero.

V. DVL Fusion Evaluation

In lieu of real-world USV data sets containing lidar, camera, IMU, and DVL sensors [28], [29], [30], DVL fusion (within the adopted LVI SLAM system) is evaluated in simulation, assuming the following.

1) The extrinsic calibration between the DVL and IMU is equal to its ground truth quantity.
2) The scale factor s_d from (8) is a known constant.
3) The DVL operates with no signal dropout.

In doing so, our experiments serve as a proof-of-concept for graph-based DVL fusion, with Section VII outlining future work in which these assumptions may be addressed.

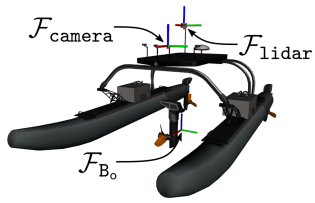


Fig. 2. Lidar, camera, IMU, and DVL sensors integrated with an USV platform in the VRX simulator. DVL and IMU frames coincide with $\mathcal{F}_{\text{B}_\text{o}}$, while lidar and camera frames correspond to $\mathcal{F}_{\text{lidar}}$ and $\mathcal{F}_{\text{camera}}$, respectively. Note that a propulsion system at $\mathcal{F}_{\text{B}_\text{o}}$ is included to simulate current.

A. Simulation

The Virtual RobotX (VRX) simulator [31] is chosen for this study. Built atop the Gazebo simulator [32], VRX has been developed for USV simulations with enhancements over Gazebo's existing physics, including the following:

1) wave and water surface representation for approximating USV motion, sensor feedback, and visual rendering;
2) wind and buoyancy simulation.

Within VRX, lidar, camera, IMU, and DVL sensors are integrated with the USV platform illustrated in Fig. 2 . Ground truth poses of the USV throughout data collection, as well as ground truth extrinsic calibrations between all sensors, are provided by the simulator.

B. Sensors

Sensors are modeled after commercial devices that are economically viable for deployable USVs. The lidar is modeled after the *Velodyne VLP16 lidar*, which is a mechanical 16-beam lidar with 360° of horizontal coverage and 30° of vertical coverage. The range and publishing frequency parameters are set to default manufacturer specifications, with the number of points output per revolution set to 1875. Downsampling points from 60 000 points (achievable on real hardware) to 1875 points per revolution is necessary to ensure real-time simulation performance while maintaining scans dense enough for LVI-SAM's scan registration algorithm. Furthermore, the time for each lidar point is output to utilize LVI-SAM's lidar motion compensation feature for refined lidar odometry. Lastly, range measurement noise is assumed white-noise Gaussian (standard deviation: 0.01 m) to suit manufacturer specification.

Camera sensors are modeled after *Flir BlackflyS* cameras, which are equipped with *Fujinon* wide field-of-view (FOV) lenses that cover 185° in the horizontal direction and 140° in the vertical direction. These RBG cameras have a resolution of 3.2 megapixels and have been configured to capture images at 20 frames per second. Cameras are intrinsically calibrated in OpenCV [33] using the Kannala Brandt [34] camera model, which has been approximated using the first three radial distortion parameters k_1, k_2, k_3 in simulation. Gaussian white noise (standard deviation: 0.007 pixels) is assumed for each color channel, with intrinsic camera calibrations included as input to LVI-SAM to undistort images. For this study, a single forward-facing camera is used to pass visual information to LVI-SAM, though we acknowledge existing work does allow

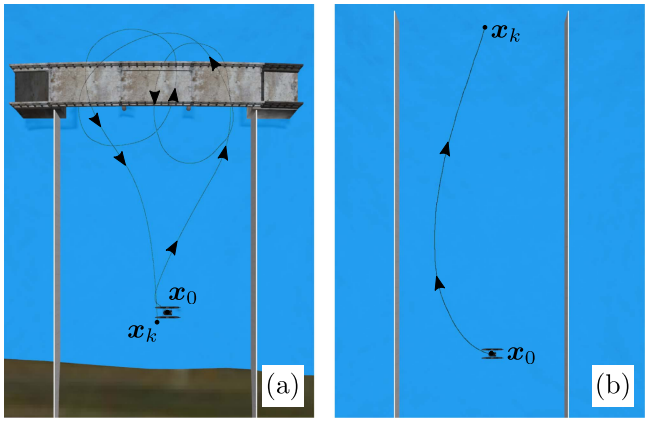


Fig. 3. (a) Environment 1: Contains built infrastructure offering rich texture and structure for visual key-point detection and scan registration. (b) Environment 2: Solid opaque white walls bound the USV to model the corridor effect. Along the entire length of wall, no visual or structural distinctions can be made, eliciting degeneracy in lidar and vision. Ground truth trajectories (solid black lines) are superimposed approximately to scale.

visual information from multiple, time-synchronized cameras to be used in SLAM for improved visual keypoint detection and tracking [35].

The IMU is modeled after the *Xsens MTi-30 AHRS*, which captures angular velocity and linear acceleration about all three axes at a configured rate of 200 Hz. Intrinsic calibration parameters are obtained from angular velocity and linear acceleration Allan standard deviation plots [36] to model white noise and random walk.

Lastly, the DVL is modeled after the *Water Linked DVL A50*, which reports a linear velocity measurement accuracy of $\pm 0.1\%$. As such, we corrupt ground truth linear velocity measurements $\tilde{\mathbf{v}}_k$ with zero-mean Gaussian white noise (standard deviation: 1e-3$\tilde{\mathbf{v}}_k$). The Waterlink DVL A50 operates adaptively from 4–24 Hz according to altitude, though our DVL has been configured to operate with a constant publishing frequency of 20 Hz assuming a constant altitude. According to the manufacturer, the white noise term is dominant for the DVL A50; therefore, we assume linear velocity bias evolves with a random walk standard deviation of 1e-4$\tilde{\mathbf{v}}_k$ in lieu of available data. Table I summarizes sensor parameters and intrinsic calibrations used in simulation.

TABLE I
SENSOR PARAMETERS AND INTRINSIC CALIBRATIONS USED IN SIMULATION TO APPROXIMATE REAL HARDWARE

Parameter Description	Value
Lidar: Velodyne VLP16 lidar	
Number of lasers [-]	16
Coverage (horizontal, vertical) [°]	(360, 30)
Points per revolution [-]	1875
Publishing frequency [Hz]	10
Range (min, max, σ) [m]:	(0.1, 130, 0.01)
Camera: Flir BlackflyS with Fujinon wide field-of-view lense	
Image:	
Resolution [mega pixels]	3.2
Dimensions $(w \times h)$ [pixels]	(2048×1536)
Gaussian white noise (σ) [pixels]	0.007
Coverage (horizontal, vertical) [°]	(185,140)
Publishing frequency [Hz]	20
Intrinsic calibrations:	
Focal length (f_x, f_y) [pixels]	$(763.159, 762.408)$
Camera centre (c_x, c_y) [pixels]	$(1017.926, 791.398)$
Kannala Brandt distortion parameters $(k_1, k_2, k_3, k_4, k_5)$ [pixels]	$(1.000, 3.866\text{e-}3, 2.494\text{e-}2, -2.006\text{e-}2, 4.700\text{e-}3)$
IMU: Xsens MTi-30 AHRS	
Degrees of freedom [-]	6
Publishing frequency [Hz]	200
Intrinsic calibrations:	
White noise:	
Gyroscope $\sigma_{\boldsymbol{\omega}} \left[\frac{rad}{s}\frac{1}{\sqrt{Hz}}\right]$	5.7e-4
Accelerometer $\sigma_{\mathbf{a}} \left[\frac{m}{s^2}\frac{1}{\sqrt{Hz}}\right]$	9.4e-4
Random walk:	
Gyroscope $\sigma_{\mathbf{b}^\omega} \left[\frac{rad}{s^2}\frac{1}{\sqrt{Hz}}\right]$	3.7e-6
Accelerometer $\sigma_{\mathbf{b}^a} \left[\frac{m}{s^3}\frac{1}{\sqrt{Hz}}\right]$	2.4e-6
DVL: Water Linked DVL A50	
Publishing frequency [Hz]	20
Operating altitude (min, max) [m]	(0.05, 50)
Linear velocity[†] (max, σ) [m/s]:	(3.75, 1e-3$\tilde{\mathbf{v}}_k$)
Intrinsic calibrations:	
Random walk:	
Linear velocity[*] $\sigma_{\mathbf{b}^v} \left[\frac{m}{s}\frac{1}{\sqrt{Hz}}\right]$	1e-4$\tilde{\mathbf{v}}_k$

[*] $\tilde{\mathbf{v}}_k$ is the ground truth discrete linear velocity of the DVL

C. Environment

Two simulation environments are developed for DVL fusion evaluation. The first environment, illustrated in Fig. 3(a), contains infrastructure typically found near the land-water interface, with partially submerged cinder block walls and a concrete bridge offering complex texture and structure favorable for lidar and vision. The second environment, illustrated in Fig. 3(b), is deprived of texture and structure, with opaque white walls bounding the USV to elicit degeneracy in lidar and vision.

From an initial state $\boldsymbol{x}_0$, the USV is manually teleoperated with an input velocity of 1.20 m/s to a terminal state $\boldsymbol{x}_k$. In Environment 1, the USV navigates toward the concrete bridge, then around both bridge piers to elicit loop closure detection before returning to its starting location for a total traveled distance of approximately 265 m. In Environment 2, the USV navigates down the corridor formed by both walls, traveling approximately 38 m in a winding path.

D. Results

Accuracy improvements are assessed by comparing $\mathcal{X}_k^\star$ with and without DVL fusion against the ground truth trajectory provided by the simulator. Absolute trajectory error (ATE) is computed using the methods of Zhang et al. [37], which serves as a metric for global pose accuracy. Accuracy improvements with and without DVL fusion are summarized in Table II.

TABLE II
ACCURACY IMPROVEMENTS TO LVI-SAM WITH AND WITHOUT DVL FUSION OBSERVED IN SIMULATION

Environment	DVL fusion	Absolute trajectory error (ATE)* $\mu(\sigma)$	
		Translation [m]	Rotation [°]
1	with	**0.114 (0.042)**	3.495 (1.191)
	without	0.194 (0.075)	**2.641 (0.844)**
2	with	**1.878 (0.003)**	0.023 (0.018)
	without	2.863 (0.977)	0.023 (0.018)

* Bold values indicate best performance.

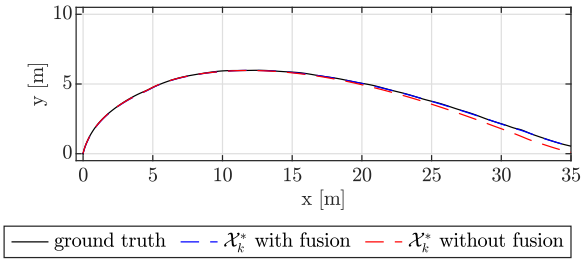


Fig. 4. Maximum *a posteriori* trajectory of the USV generated by LVI-SAM with and without DVL fusion in Environment 2. Without DVL fusion, the USV dead reckons off of preintegrated IMU measurements during lidar and vision degeneracy, which occurs over the full duration of data collection. With DVL fusion, the USV dead reckons off of preintegrated pseudosensor measurements, which combine linear velocity measurements from the DVL and angular velocity measurements from the IMU. During lidar and vision degeneracy, biases are no longer observable and are set to zero.

Results show accuracy improvements made with DVL fusion, where translation ATE is reduced by 0.080 m from 0.194 to 0.114 m in Environment 1. Rotation ATE marginally increases by 0.854°, which is attributed to the joint minimization of rotation and translation residuals in $\mathcal{X}_k^\star$, considering $\mathbf{r}_{\mathcal{D}_{ij}}$ only constrains relative changes in position. With lidar and vision fully degenerating in Environment 2, DVL fusion reduces translation ATE over the base LVI system, where only IMU factors are added to the graph. Robustness imparted by DVL fusion is illustrated in Fig. 4, where the USV is able to dead reckon off of preintegrated DVL odometry. In contrast, preintegrated IMU odometry diverges from ground truth near the end of data collection, incurring a higher translation ATE than DVL fusion.

VI. DECOUPLED MAPPING

With demonstrated improvements to state estimation with DVL fusion, we aim to further improve SLAM-centric maritime infrastructure inspection via our decoupled mapping framework. In preface, we mathematically define volumetric scans and volumetric maps according to (20) and (21), respectively

$$ {}_{\mathtt{S_i}}\mathcal{V}_{K_j} \triangleq \{\boldsymbol{v}_k \triangleq \{x, y, z, \boldsymbol{w}\}_k\}_{k\in K_j} \tag{20} $$

$$ {}_{\mathtt{W}}\mathcal{V} \triangleq \cup_{S_i\in S} \cup_{K_j\in K} {}_{\mathtt{W}}\mathcal{V}_{\mathtt{WS_i},K_j} \tag{21} $$

where $\boldsymbol{v}_k$ is volumetric data point k consisting of $\{x, y, z\} \in \mathbb{R}^3$ position relative to sensor frame $\mathtt{S_i}$ and $\boldsymbol{w}$, which is an arbitrary set of scalar values (ex. intensity, time); ${}_{\mathtt{W}}\mathcal{V}_{\mathtt{WS_i},K_j}$ is the expression of ${}_{\mathtt{S_i}}\mathcal{V}_{K_j}$ in the world coordinate frame $\mathtt{W}$, and S and K are sets of range sensors (i.e., lidar, sonar) and scans, respectively, used to construct ${}_{\mathtt{W}}\mathcal{V}$.

Next, we distinguish volumetric maps generated by SLAM (herein referred to as SLAM maps) and volumetric maps generated for downstream inspection applications (herein referred to as inspection maps). For SLAM systems operating above the surface, volumetric maps maintained by lidar-based odometry methods may be accessed upon SLAM session completion for downstream tasks. To improve scan-registration performance and to remain computationally tractable, lidar-based odometry methods (including feature-based [6], [26], [38], [39] and direct [40], [41] methods) generally down-sample and discard volumetric data. Feature-based methods, which extract and use only the most salient points before scan registration [42], retain a small subset of the total volumetric data collected with environmental structure lost in the process. Direct lidar odometry methods, which attempt scan-registration on dense clouds, suffer from heavy down-sampling, with even state-of-the-art direct lidar odometry [42] voxelizing lidar scans with 0.25 m resolution. Consequently, SLAM map density is reduced and environmental structure is lost. Inspection maps, in contrast, prioritize high point density and low map noise to improve semantic labeling [43], surface reconstruction [44], and point cloud segmentation and classification [45]. Achieving inspection map performance, thus, directly conflicts with SLAM performance requirements, justifying a separate process for inspection map generation.

TABLE III
FILTERS AVAILABLE WITHIN MAPBUILDER

Filter	Description
Cropbox	Removes points inside or outside of a user-defined rectangular volume
Voxel Grid	Downsamples volumetric measurements according to a prescribed resolution.
ROR	Radius outlier removal. Removes outliers based on point density
DROR	Dynamic radius outlier removal. Removes outliers based on point density and distance from scan origin

A. Proposed Framework

We propose a novel point-cloud generation framework (herein referred to as *MapBuilder*) that decouples the map generation process from SLAM. Our framework allows any number or type of volumetric data to be combined into a single map, with noise removal and scan cropping techniques available at various stages along the map-building pipeline. Moreover, our framework can be used with any SLAM algorithm, provided the algorithm provides access to estimated poses. Fig. 5 illustrates the MapBuilder pipeline as a block diagram, with details on MapBuilder's input, processing, and filtering following.

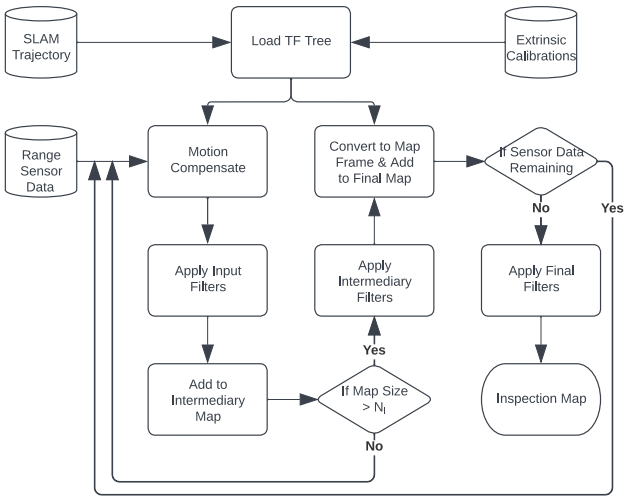


Fig. 5. Block diagram of the *MapBuilder* software. MapBuilder is designed for any number or type of volumetric data to be combined into a single map. N_I is the number of scans in each intermediary map.

TABLE IV
FILTERS APPLIED TO THE INSPECTION MAP OF WELLAND CANAL BRIDGE 13

Stage	Filter	Sensor S_i	
		$\texttt{lidar}_\texttt{h}$ and $\texttt{lidar}_\texttt{v}$	sonar
	CropBox		
	$\{x, y, z\}_{\min}$	{-60, -60, -60}	{-30, -30, -30}
	$\{x, y, z\}_{\max}$	{60, 60, 60}	{30, 30, 30}
	remove points	outside	outside
${}_{\texttt{S}_\texttt{i}}\mathcal{V}_{K_j}$	**CropBox**		
	$\{x, y, z\}_{\min}$	{-2, -2, -2}	{-2, -2, -2}
	$\{x, y, z\}_{\max}$	{2, 2, 2}	{2, 2, 2}
	remove points	inside	inside
	DROR*		
	$\{\beta, \alpha, k_{\min}, SR_{\min}\}$	{3.0, 0.2, 4, 0.01}	{3.0, 0.5, 4, 0.01}

* For parameter descriptions, see [48].

TABLE V
QUANTITATIVE COMPARISON BETWEEN SLAM MAP AND INSPECTION MAP FOR WELLAND CANAL BRIDGE 13

Metrics	SLAM Map	Inspection Map
	μ for local neighborhood radius $r = 0.25$m*	
Surface Density	337.124	**31986.9**
Roughness	3.637e-2	**2.576e-2**
Planarity	4.686e-1	**6.546e-1**

* Bold values indicate best performance.

1) Input: MapBuilder receives input from three separate modules: *extrinsic calibrations*, *SLAM trajectory*, and *range sensor data*. Extrinsic calibrations include the set of rigid body transformations $\mathbf{T}_{\texttt{S}_\texttt{i}\texttt{S}_\texttt{j}}$ relating all range sensors to a common frame $\texttt{B}$. SLAM trajectory includes the full set of time-stamped MAP poses $\bar{\mathbf{T}}^{\text{MAP}}_{\texttt{WB}} = \{\mathbf{T}^{\text{MAP}}_{\texttt{WB},i}, t_i\}_{i \in \texttt{K}_k}$, with the option to correct and add high-rate, locally consistent poses to $\bar{\mathbf{T}}^{\text{MAP}}_{\texttt{WB}}$ from available odometry subsystems according to Algorithm 2. This feature is particularly useful for keyframe-based SLAM approaches, where nonkeyframe poses are excluded when solving $\mathcal{X}^{\star}_k$, thus, reducing trajectory granularity. Lastly, range sensor data includes: $\{{}_{\texttt{S}_\texttt{i}}\mathcal{V}_{K_j} \mid \forall S_i \in S, \forall K_j \in K\}$.

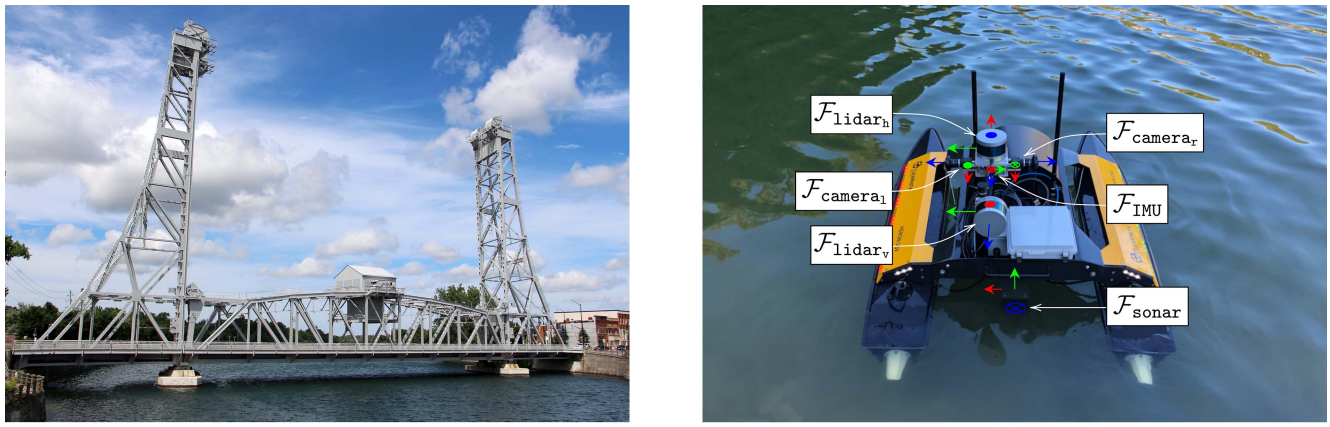
Algorithm 2: High-Rate, Locally Consistent Odometry Corrected by Low-Rate, Globally Consistent MAP Poses.

Input: $\bar{\mathbf{T}}^{\text{MAP}}_{\texttt{WB}}$, $\bar{\mathbf{T}}^{\text{odom}}_{\texttt{W'B}}$
Output: $\bar{\mathbf{T}}^{\text{refined}}_{\texttt{WB}}$

// Initialize TF Tree object with MAP pose
$\texttt{tfMAP}$.InitializeObject($\bar{\mathbf{T}}^{\text{MAP}}_{\texttt{WB}}$)
// Calculate corrections
$\texttt{tfCorrect}$.InitializeObject()
$j \leftarrow 0$
for all i in $\bar{\mathbf{T}}^{\text{odom}}_{\texttt{W'B}}$ **do**
  $t_i \leftarrow \bar{\mathbf{T}}^{\text{odom}}_{\texttt{W'B}}$.GetTimeAtIndex($i$)
  $t_j \leftarrow \bar{\mathbf{T}}^{\text{MAP}}_{\texttt{WB}}$.GetTimeAtIndex($j$)
  // Get correction at or near time of MAP pose
  if $t_i \geq t_j$ **then**
    $\mathbf{T}^{\text{interp}}_{\texttt{WB}} \leftarrow \texttt{tfMAP}$.GetTransformAtTime($t_i$)
    $\mathbf{T}^{\text{odom}}_{\texttt{W'B},i} \leftarrow \bar{\mathbf{T}}^{\text{odom}}_{\texttt{W'B}}$.GetTransformAtIndex($i$)
    $\mathbf{T}_{\texttt{WW'}} = \mathbf{T}^{\text{interp}}_{\texttt{WB}} \times \mathbf{T}^{\text{odom}}_{\texttt{W'B},i}$.GetInverse()
    $\texttt{tfCorrect}$.AddStampedTransform($\{\mathbf{T}_{\texttt{WW'}}, t_i\}$)
    $j \leftarrow j + 1$
  end if
end for
// Correct odometry
$\bar{\mathbf{T}}^{\text{refined}}_{\texttt{WB}} \leftarrow \varnothing$
for all i in $\bar{\mathbf{T}}^{\text{odom}}_{\texttt{W'B}}$ **do**
  $t_i \leftarrow \bar{\mathbf{T}}^{\text{odom}}_{\texttt{W'B}}$.GetTimeAtIndex($i$)
  $\mathbf{T}^{\text{interp}}_{\texttt{WW'}} \leftarrow \texttt{tfCorrect}$.GetTransformAtTime($t_i$)
  $\mathbf{T}^{\text{refined}}_{\texttt{WB}} \leftarrow \mathbf{T}^{\text{interp}}_{\texttt{WW'}} \times \bar{\mathbf{T}}^{\text{odom}}_{\texttt{W'B}}$.GetTransformAtIndex($i$)
  $\bar{\mathbf{T}}^{\text{refined}}_{\texttt{WB}}$.AddStampedTransform($\{\mathbf{T}^{\text{refined}}_{\texttt{WB}}, t_i\}$)
end for
return $\bar{\mathbf{T}}^{\text{refined}}_{\texttt{WB}}$

2) Processing: First, extrinsic calibrations and SLAM trajectory are combined into a transformation tree (TF Tree) data structure [46]. Conceptually, a TF Tree originates from a single parent node, representing an arbitrarily defined frame common to all sensors (i.e., frame $\texttt{W}$), with edges representing time-stamped coordinate transformations between frames. Once constructed, the TF tree may be queried at the time of data capture for sensor $\texttt{S}_\texttt{i}$, retrieving $\mathbf{T}_{\texttt{WS}_\texttt{i}}$. If this time does not coincide with any time-stamped transformations stored in the TF tree, spherical linear interpolation [47] is used, allowing our framework to handle scans that are asynchronous to SLAM trajectory. For mapping maritime infrastructure, sonar and supplementary lidar sensors may collect volumetric data at a rate variable to the lidar used in the SLAM approach, thus, requiring interpolation to finely resolve sensor pose.

Before transforming ${}_{\texttt{S}_\texttt{i}}\mathcal{V}_{K_j}$ into ${}_{\texttt{W}}\mathcal{V}_{\texttt{WS}_\texttt{i},K_j}$, sensor data are optionally motion compensated if each $\boldsymbol{w}_k \in {}_{\texttt{S}_\texttt{i}}\mathcal{V}_{K_j}$ contains a time value. Motion compensation again uses the TF tree to



(a) (b)

Fig. 6. (a) Perspective photograph of *Welland Canal Bridge 13*, comprised of two partially submerged concrete piers and a steel superstructure, is used to evaluate our proposed decoupled mapping framework. Source: [50] (b) USV platform outfit with: Two Flir BlackflyS RGB cameras with Fujinon wide field-of-view (FOV) lenses, corresponding to $\mathcal{F}_{\texttt{camera}_\texttt{l}}$ and $\mathcal{F}_{\texttt{camera}_\texttt{r}}$; two Velodyne VLP16 lidars oriented horizontally and vertically, corresponding to $\mathcal{F}_{\texttt{lidar}_\texttt{h}}$ and $\mathcal{F}_{\texttt{lidar}_\texttt{v}}$ respectively; an Xsens MTi-30 AHRS inertial measurement unit corresponding to $\mathcal{F}_{\texttt{IMU}}$; an Imagenex DT100 multibeam profiling sonar corresponding to $\mathcal{F}_{\texttt{sonar}}$.

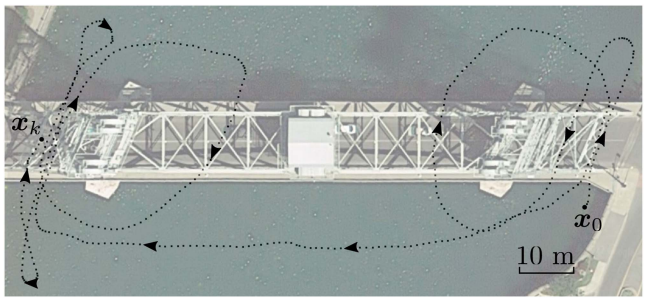


Fig. 7. Maximum *a posteriori* trajectory of the USV generated by LVI-SAM, superimposed on a satellite image of *Welland Canal Bridge 13* approximately to scale. Source: Adapted from [55].

query $\mathbf{T}_{\texttt{WS}_\texttt{i}}$ for each ${}_{\texttt{S}_\texttt{i}}\boldsymbol{v}_k$, aggregating them back into ${}_{\texttt{S}_\texttt{i}}\mathcal{V}_{K_j}$. Note this optional processing is only effective when the SLAM approach provides preintegrated IMU (or DVL) odometry that can be fused with corrected odometry using Algorithm 2.

3) Filtering: Volumetric data is optionally cropped and filtered using the methods outlined in Table III as a means of noise removal.

Filters may be applied at three stages of the map building process to suit performance needs. First, filters may be applied directly to each volumetric scan, with the *dynamic radius outlier removal* (DROR) [48] filter specifically designed to remove outliers from a single lidar scan. At this stage, the cropbox filter may also be applied to remove distant, noisy measurements, and near measurements of the robot itself. In the second stage of filtering, ${}_{\texttt{W}}\mathcal{V}_{\texttt{WS}_\texttt{i},K_j}$ are aggregated into intermediary maps

$$ {}_{\texttt{W}}\mathcal{V}_I = \cup_{K_j \in I} {}_{\texttt{W}}\mathcal{V}_{\texttt{WS}_\texttt{i},K_j} \tag{22}$$

where I is a set of consecutive scans of user-defined cardinality N_I. When applying filters to intermediary maps, the performance of the *radius outlier removal* (ROR) [49] filter improves over its application to a single scan, given the denser set of points to discriminate inliers versus outliers. In the last stage, filters may be applied to ${}_{\texttt{W}}\mathcal{V}_{\texttt{WS}_\texttt{i},K_j}$ for each S_i, where the voxel grid filter may be used to reduce final surface noise.

B. Evaluation

To evaluate MapBuilder, real-world experiments are conducted near *Welland Canal Bridge 13* [see Fig. 6(a)], which is a water-spanning bridge situated in tranquil water. Details on the USV platform deployed for data collection [see Fig. 6(b)] and extrinsic calibrations performed follow.

1) USV Platform: Lidar, camera, and IMU sensors described in Section V-B are integrated with the USV platform. From studies conducted in simulation, the lidar remains horizontally oriented, while the IMU has been mounted near the lidar to minimize contaminate centripetal forces, given LVI-SAM assumes lidar and inertial frame origins coincide. Cameras are oriented in side-facing directions to maximize the number of tracked features during lengthwise passes of infrastructure. For volumetric mapping of maritime infrastructure, supplementary sensing is provided by a vertically oriented Velodyne VLP16 lidar and an Imagenex DT100 multibeam profiling sonar. All sensors interface with a microcontroller, which provides analog signals to start, trigger, and time-synchronize sensors to a common clock.

2) Extrinsic Calibration: Following the intrinsic calibration of cameras and the IMU described in Section V-B, sensors are extrinsically calibrated such that:

1) $\mathbf{T}_{\texttt{lidar}_\texttt{h},\texttt{camera}_\texttt{l}}$ and $\mathbf{T}_{\texttt{lidar}_\texttt{h},\texttt{camera}_\texttt{r}}$ are obtained using the method of Zhou et al. [51];
2) $\mathbf{T}_{\texttt{IMU},\texttt{camera}_\texttt{l}}$ is obtained using the Kalibr toolbox [52];
3) $\mathbf{T}_{\texttt{lidar}_\texttt{h},\texttt{lidar}_\texttt{v}}$ and $\mathbf{T}_{\texttt{lidar}_\texttt{h},\texttt{sonar}}$ are estimated using a manual calibration tool developed within MapBuilder.

Note that the *patch test* [53] and *multibeam-IMU boresight automatic calibration* [54] methods, both of which are

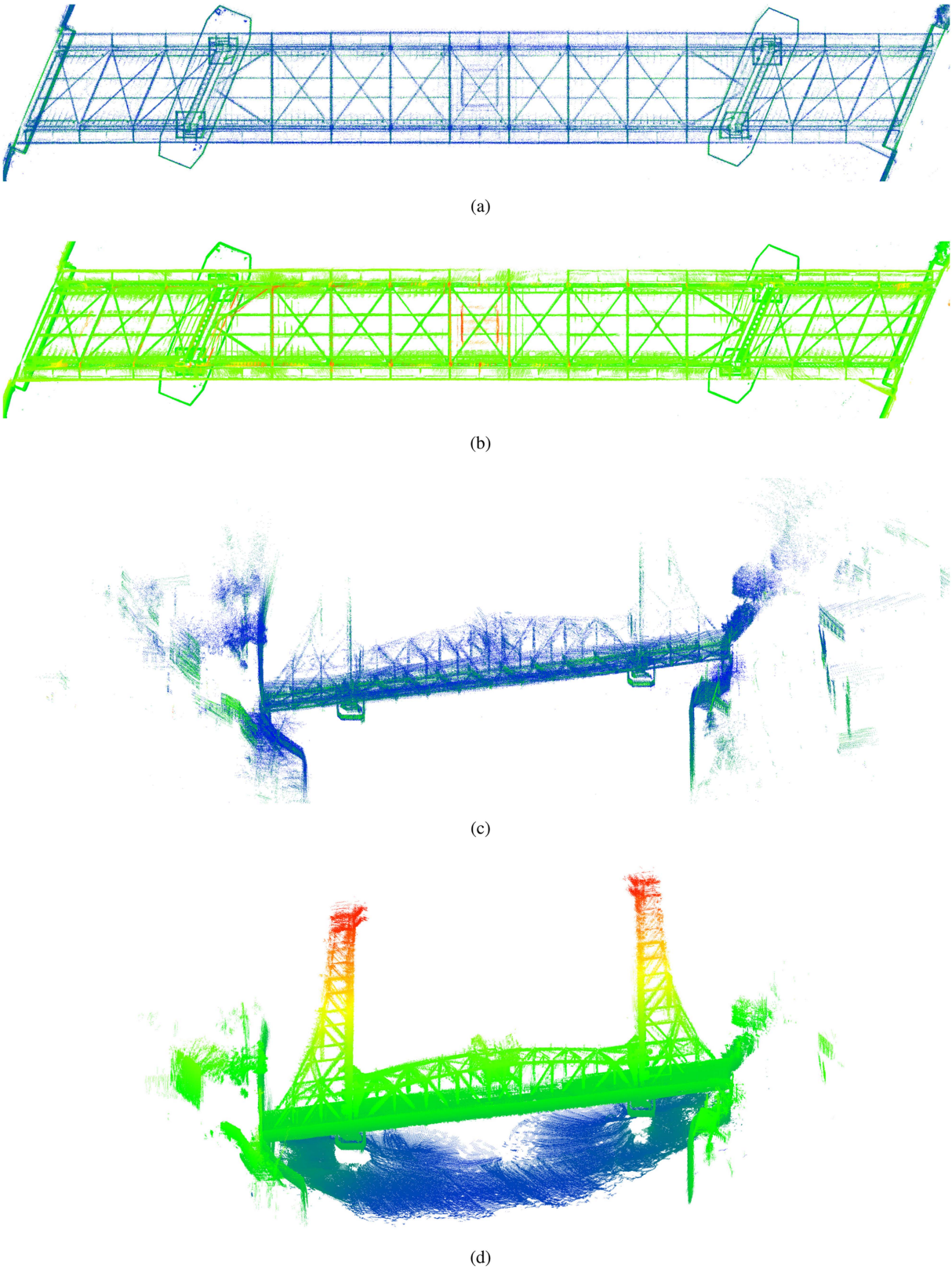


Fig. 8. SLAM map produced by LVI-SAM viewed (a) top–down and (c) perspective versus inspection map generated by our proposed decoupled mapping framework (b) top–down and (d) perspective. Volumetric data from the vertically oriented lidar and multibeam profiling sonar is excluded from (b) to contrast density enhancement and noise removal provided by our framework when compared to (a).

Algorithm 3: Manual Extrinsic Calibration Between Reference Sensor $\mathtt{S_r}$ and Candidate Sensor $\mathtt{S_c}$.

Input: $\bar{\mathbf{T}}_{\mathtt{WB}}^{\text{refined}}$, $\mathbf{T}_{\mathtt{BS_r}}$, $\{{}_{\mathtt{S_i}}\mathcal{V}_{K_j} \mid S_i = \{S_r, S_c\}, \forall K_j \in K\}$

// Initialize
$\mathbf{T}_{\mathtt{S_rS'_c}}^{\text{prev}}$.SetToIdentity(); $k_{\text{prev}} \leftarrow \varnothing$; $N_{\text{prev}} \leftarrow \varnothing$
${}_{\mathtt{W}}\mathcal{V}_{\mathtt{WS_r}} \leftarrow \varnothing$; ${}_{\mathtt{W}}\mathcal{V}_{\mathtt{WS_c}} \leftarrow \varnothing$
$\mathtt{tfRefined}$.InitializeObject($\bar{\mathbf{T}}_{\mathtt{WB}}^{\text{refined}}$)
// Listen for user input
while true **do**
  if keypress **then**
    // Get user input
    $\{k, N, \mathtt{x}, \mathtt{y}, \mathtt{z}, \mathtt{roll}, \mathtt{pitch}, \mathtt{yaw}\} \leftarrow$ GetInput()
    $\mathbf{T}_{\mathtt{S_rS'_c}}$.SetTransform($\mathtt{x}, \mathtt{y}, \mathtt{z}, \mathtt{roll}, \mathtt{pitch}, \mathtt{yaw}$)
    $\mathtt{isWindowReset} \leftarrow (k_{\text{prev}} \neq k$ or $N_{\text{prev}} \neq N)$
    // Transform candidate sensor data into world frame
    if $\mathbf{T}_{\mathtt{S_rS'_c}}^{\text{prev}} \neq \mathbf{T}_{\mathtt{S_rS'_c}}$ or $\mathtt{isWindowReset}$ **then**
      ${}_{\mathtt{W}}\mathcal{V}_{\mathtt{WS_c}} \leftarrow \varnothing$
      for j from k to $k+N$ **do**
        $t_j \leftarrow {}_{\mathtt{S_c}}\mathcal{V}_{K_j}$.GetScanTime()
        $\mathbf{T}_{\mathtt{WB}} \leftarrow$ $\mathtt{tfRefined}$.GetTransformAtTime(t_j)
        $\mathbf{T}_{\mathtt{WS'_c}} \leftarrow \mathbf{T}_{\mathtt{WB}} \times \mathbf{T}_{\mathtt{BS_r}} \times \mathbf{T}_{\mathtt{S_rS'_c}}$
        ${}_{\mathtt{W}}\mathcal{V}_{\mathtt{WS_c}} \leftarrow {}_{\mathtt{W}}\mathcal{V}_{\mathtt{WS_c}} \cup$ TransformScan($\mathbf{T}_{\mathtt{WS'_c}}, {}_{\mathtt{S_c}}\mathcal{V}_{K_j}$)
      end for
    end if
    // Transform reference sensor data into world frame
    if $\mathtt{isWindowReset}$ **then**
      ${}_{\mathtt{W}}\mathcal{V}_{\mathtt{WS_r}} \leftarrow \varnothing$
      for j from k to $k+N$ **do**
        $t_j \leftarrow {}_{\mathtt{S_r}}\mathcal{V}_{K_j}$.GetScanTime()
        $\mathbf{T}_{\mathtt{WB}} \leftarrow \mathtt{tfRefined}$.GetTransformAtTime ($t_j$)
        $\mathbf{T}_{\mathtt{WS_r}} \leftarrow \mathbf{T}_{\mathtt{WB}} \times \mathbf{T}_{\mathtt{BS_r}}$
        ${}_{\mathtt{W}}\mathcal{V}_{\mathtt{WS_r}} \leftarrow {}_{\mathtt{W}}\mathcal{V}_{\mathtt{WS_r}} \cup$ TransformScan($\mathbf{T}_{\mathtt{WS_r}}, {}_{\mathtt{S_r}}\mathcal{V}_{K_j}$)
      end for
    end if
    // Publish map for visualization
    Publish(${}_{\mathtt{W}}\mathcal{V}_{\mathtt{WS_r}}$).SetCloudColor(red)
    Publish(${}_{\mathtt{W}}\mathcal{V}_{\mathtt{WS_c}}$).SetCloudColor(blue)
    // Reset
    $\mathbf{T}_{\mathtt{S_rS'_c}}^{\text{prev}} \leftarrow \mathbf{T}_{\mathtt{S_rS'_c}}$; $k_{\text{prev}} \leftarrow k$; $N_{\text{prev}} \leftarrow N$
  end if
end while

optimization-based methods for sonar-to-IMU extrinsic calibration, could not be used at the time of this work due to strict data collection requirements; moreover, these methods assume relatively flat or gradually sloping bathymetry and thus do not generalize to cases where the multibeam profiling sonar is angled toward submerged infrastructure. For lidar-to-lidar extrinsic calibration, no target or targetless methods (to our knowledge) exist for the case of minimally overlapping FOVs, hence the development and application of the manual calibration tool summarized in Algorithm 3.

C. Results

The USV platform described in Section VI-B1 is deployed for data collection near the right pier of Welland Canal Bridge 13 as illustrated in Fig. 7. From an initial state $\boldsymbol{x}_0$, the USV proceeds to circumnavigate both bridge piers before reaching a terminal state $\boldsymbol{x}_k$. The MAP trajectory of the USV generated by LVI-SAM is illustrated approximately to scale.

In generating the inspection map, lidar-inertial odometry (provided by LVI-SAM's LIS) is corrected using Algorithm 2 against MAP poses, which additionally factor visual-inertial odometry and loop-closure constraints into their solution. For refined interkeyframe pose estimates for the vertically oriented lidar, multibeam profiling sonar, and scans from the horizontally oriented lidar discarded by LVI-SAM, Algorithm 2 is again applied to correct high-rate preintegrated IMU odometry with the corrected lidar-inertial odometry. By inputting these corrected poses to MapBuilder as $\bar{\mathbf{T}}_{\mathtt{WB}}^{\text{refined}}$, range sensor data are then motion-compensated. Table IV summarizes the filters applied to the inspection map, with qualitative and quantitative comparisons between SLAM and inspection maps following. Note that we limit our comparison to LVI-SAM to highlight the benefits of inspection map generation over the direct use of its SLAM map.

1) SLAM Versus Inspection Map Qualitative Comparison: Fig. 8 qualitatively illustrates the difference between SLAM and inspection maps generated for Welland Canal Bridge 13. When comparing maps from a top–down perspective in Fig. 8(a) and (b), the inspection map is only constructed using the horizontally oriented lidar to illustrate density and structure improvements offered by our decoupled approach while controlling for noise. In Fig. 8(c) and (d), the inspection map is constructed from both lidars and the multibeam profiling sonar, better characterizing the bridge's superstructure and submerged structure.

2) SLAM Versus Inspection Map Quantitative Comparison: To quantify density, noise, and retention of environmental structure in SLAM and inspection maps, surface density, roughness, and planarity metrics are according to (23), (24), and (25), respectively. For this study, an empirical local neighborhood radius of $r = 0.25$ m is adopted

$$\text{surface density} = N/\left(\pi r^2\right) \tag{23}$$

$$\text{roughness} = d\left({}_{\mathtt{S_{lidar_h}}}\boldsymbol{v}_k, w\right) \tag{24}$$

$$\text{planarity} = (\lambda_2 - \lambda_3)/\lambda_1 \tag{25}$$

where N is the number of neighbors within r of ${}_{\mathtt{S_{lidar_h}}}\boldsymbol{v}_k$, w is the best fitting plane computed on the neighbors of ${}_{\mathtt{S_{lidar_h}}}\boldsymbol{v}_k$ within r, $d({}_{\mathtt{S_{lidar_h}}}\boldsymbol{v}_k, w)$ is the shortest distance from ${}_{\mathtt{S_{lidar_h}}}\boldsymbol{v}_k$ to w, and $\{\lambda_1, \lambda_2, \lambda_3\}$ are the eigenvalues computed from a singular value decomposition of the covariance tensor [56]

$$\Sigma_k = \frac{1}{N} \sum_{n \in \mathcal{V}_k^N} (\boldsymbol{v}_n - \bar{\boldsymbol{v}})(\boldsymbol{v}_n - \bar{\boldsymbol{v}})^\top \tag{26}$$

where $\mathcal{V}_k^N$ is the set of N neighbors within r of $\boldsymbol{v}_n$ and $\bar{\boldsymbol{v}} = \text{med}_{n\in\mathcal{V}_k^N}(\boldsymbol{v}_n)$ is the medoid. Note that planarity serves as a proxy for reconstructing geometrically accurate models that are Manhattan in nature [20], where higher planarity values indicate a higher probability of a volumetric data point being associated with structural elements as they are mainly comprised of planar surfaces. Table V summarizes mean metric values for both SLAM and inspection maps, where inspection map performance exceeds SLAM map performance across all metrics.

VII. Conclusion

In this article, three novel contributions are proposed to address SLAM-centric maritime infrastructure inspection using USVs. First, we formulate a tightly coupled, graph-based DVL/IMU fusion strategy that generalizes to any graph-based SLAM formulation. Second, we implement our fusion strategy within a state-of-the-art, graph-based lidar-visual-inertial SLAM system and demonstrate accuracy and robustness improvements in simulation. We show benefits to translation absolute trajectory error in environments typical of maritime infrastructure while also showing improvements to robustness in environments where lidar and vision degenerate. Lastly, we show our decoupled mapping approach for downstream inspection applications improves map quality over volumetric maps generated via SLAM. To best utilize precise SLAM trajectories with high-rate range sensor data, our decoupled mapping approach corrects high-rate SLAM poses (from lidar-based odometry, IMU preintegration, etc.) to generate a dense map tuned specifically for inspection. Map metrics important to downstream inspection tasks are calculated and show that our framework outperforms the map produced by SLAM across all categories. In future work, a graph-based SLAM system addressing DVL scale factor estimation, online extrinsic DVL-IMU calibration, and DVL signal drop-out will be developed. This system will be evaluated using real-world experiments across a variety of maritime environments.

Appendix A

Derivations related to our DVL factor's residual term and covariance (as presented in Section IV-A) are provided in this Appendix. Derivations follow similarly to [9], [27], and [57].

A. First-Order Approximation of Relative Position Under Small Bias Updates

Given a bias update $\mathbf{b} \leftarrow \bar{\mathbf{b}} + \delta\mathbf{b}$ where $\delta\mathbf{b}$ is relatively small, the first order approximation of $\Delta\tilde{\mathbf{p}}_{ij}(\mathbf{b}_i^\omega, \mathbf{b}_i^v)$ in (15) is derived by taking the first-order Taylor series expansion

$$\Delta\tilde{\mathbf{p}}_{ij}\left(\mathbf{b}_i^\omega, \mathbf{b}_i^v\right) \simeq \Delta\tilde{\mathbf{p}}_{ij}\left(\bar{\mathbf{b}}_i^\omega, \bar{\mathbf{b}}_i^v\right) + \frac{\partial\Delta\bar{\mathbf{p}}_{ij}}{\partial\mathbf{b}^\omega}\delta\mathbf{b}_i^\omega + \frac{\partial\Delta\bar{\mathbf{p}}_{ij}}{\partial\mathbf{b}^v}\delta\mathbf{b}_i^v. \tag{27}$$

Expressions for Taylor series expansion terms are found by following the same derivation procedure for $\Delta\tilde{\mathbf{p}}_{ij}(\hat{\mathbf{b}}_i)$ in [57, Sec. 1]:

$$\begin{aligned}
\Delta\tilde{\mathbf{p}}_{ij}(\hat{\mathbf{b}}_i) &= \sum_{k=i}^{j-1}\left[\Delta\tilde{\mathbf{R}}_{ik}(\hat{\mathbf{b}}_i)\left(\frac{\tilde{\mathbf{v}}_k - \bar{\mathbf{b}}_i^v - \delta\mathbf{b}_i^v}{s_d}\right)\Delta t\right] \\
&= \sum_{k=i}^{j-1}\left[\Delta\tilde{\mathbf{R}}_{ik}(\bar{\mathbf{b}}_i)\text{Exp}\left(\frac{\partial\Delta\bar{\mathbf{R}}_{ik}}{\partial\mathbf{b}^\omega}\delta\mathbf{b}_i^\omega\right)\right. \\
&\qquad \left.\times\left(\frac{\tilde{\mathbf{v}}_k - \bar{\mathbf{b}}_i^v - \delta\mathbf{b}_i^v}{s_d}\right)\Delta t\right] \\
&\simeq \sum_{k=i}^{j-1}\left[\Delta\tilde{\mathbf{R}}_{ik}(\bar{\mathbf{b}}_i)\left(\mathbf{I} + \left(\frac{\partial\Delta\bar{\mathbf{R}}_{ik}}{\partial\mathbf{b}^\omega}\delta\mathbf{b}_i^\omega\right)^\wedge\right)\right. \\
&\qquad \left.\times\left(\frac{\tilde{\mathbf{v}}_k - \bar{\mathbf{b}}_i^v - \delta\mathbf{b}_i^v}{s_d}\right)\Delta t\right] \\
&= \Delta\tilde{\mathbf{p}}_{ij}(\bar{\mathbf{b}}_i) \\
&\quad + \sum_{k=i}^{j-1}\left[-\Delta\tilde{\mathbf{R}}_{ik}(\bar{\mathbf{b}}_i)\frac{\delta\mathbf{b}_i^v}{s_d}\Delta t\right] \\
&\quad + \sum_{k=i}^{j-1}\left[\Delta\tilde{\mathbf{R}}_{ik}(\bar{\mathbf{b}}_i)\left(\frac{\partial\Delta\bar{\mathbf{R}}_{ik}}{\partial\mathbf{b}^\omega}\delta\mathbf{b}_i^\omega\right)^\wedge\right. \\
&\qquad \left.\times\left(\frac{\tilde{\mathbf{v}}_k - \bar{\mathbf{b}}_i^v}{s_d}\right)\Delta t\right] \\
&= \Delta\tilde{\mathbf{p}}_{ij}(\bar{\mathbf{b}}_i) \\
&\quad + \sum_{k=i}^{j-1}\left[-\Delta\tilde{\mathbf{R}}_{ik}(\bar{\mathbf{b}}_i)\frac{\Delta t}{s_d}\delta\mathbf{b}_i^v\right] \\
&\quad + \sum_{k=i}^{j-1}\left[-\Delta\tilde{\mathbf{R}}_{ik}(\bar{\mathbf{b}}_i)\left(\frac{\tilde{\mathbf{v}}_k - \bar{\mathbf{b}}_i^v}{s_d}\right)^\wedge\right. \\
&\qquad \left.\times\frac{\partial\Delta\bar{\mathbf{R}}_{ik}}{\partial\mathbf{b}^\omega}\Delta t\delta\mathbf{b}_i^\omega\right]
\end{aligned} \tag{28}$$

giving

$$\begin{aligned}
\frac{\partial\Delta\bar{\mathbf{p}}_{ij}}{\partial\mathbf{b}^\omega} &= -\sum_{k=i}^{j-1}\Delta\tilde{\mathbf{R}}_{ik}(\bar{\mathbf{b}}_i)\left(\frac{\tilde{\mathbf{v}}_k - \bar{\mathbf{b}}_i^v}{s_d}\right)^\wedge\frac{\partial\Delta\bar{\mathbf{R}}_{ik}}{\partial\mathbf{b}^\omega}\Delta t \\
\frac{\partial\Delta\bar{\mathbf{p}}_{ij}}{\partial\mathbf{b}^v} &= -\sum_{k=i}^{j-1}\Delta\tilde{\mathbf{R}}_{ik}(\bar{\mathbf{b}}_i)\frac{\Delta t}{s_d}.
\end{aligned} \tag{29}$$

Appendix B
Iterative Noise Propagation

To derive expressions for $\mathbf{A}$ and $\mathbf{B}$ in (16), we look toward (12) and consider the interval from i to $k+1$

$$\begin{aligned}
\delta\mathbf{p}_{i,k+1} &\simeq \delta\mathbf{p}_{ik} + \Delta\tilde{\mathbf{R}}_{ik}\boldsymbol{\eta}_k^v\Delta t \\
&\quad - \tilde{\mathbf{R}}_{ik}\left(\frac{\tilde{\mathbf{v}}_k - \mathbf{b}_i^v}{s_d}\right)^\wedge\delta\boldsymbol{\phi}_{ik}\Delta t.
\end{aligned} \tag{30}$$

In combination with the rotation noise model from [27], (30) may be represented in matrix form as

$$\begin{bmatrix}\delta\boldsymbol{\phi}_{i,k+1}\\ \delta\mathbf{p}_{i,k+1}\end{bmatrix} = \begin{bmatrix}\Delta\tilde{\mathbf{R}}_{k,k+1}^{\top} & \boldsymbol{O}_{3\times3}\\ -\tilde{\mathbf{R}}_{ik}(\frac{1}{s_d}(\tilde{\mathbf{v}}_k-\mathbf{b}_i^v))^{\wedge}\Delta t & \boldsymbol{I}_{3\times3}\end{bmatrix}\begin{bmatrix}\delta\boldsymbol{\phi}_{ik}\\ \delta\mathbf{p}_{ik}\end{bmatrix} + \begin{bmatrix}\mathbf{J}_r^k\Delta t & \boldsymbol{O}_{3\times3}\\ \boldsymbol{O}_{3\times3} & \Delta\tilde{\mathbf{R}}_{ik}\Delta t\end{bmatrix}\begin{bmatrix}\boldsymbol{\eta}_k^{\omega}\\ \boldsymbol{\eta}_k^{v}\end{bmatrix} \tag{31}$$

which more compactly can be written as

$$\begin{bmatrix}\delta\boldsymbol{\phi}_{i,k+1}\\ \delta\mathbf{p}_{i,k+1}\end{bmatrix} = \mathbf{A}\begin{bmatrix}\delta\boldsymbol{\phi}_{ik}\\ \delta\mathbf{p}_{ik}\end{bmatrix} + \mathbf{B}\begin{bmatrix}\boldsymbol{\eta}_k^{\omega}\\ \boldsymbol{\eta}_k^{v}\end{bmatrix} \tag{32}$$

matching $\boldsymbol{\eta}_{i,k+1}^{\Delta} = \mathbf{A}\boldsymbol{\eta}_{ik}^{\Delta} + \mathbf{b}\boldsymbol{\eta}_k$ presented in [57], allowing (32) to incrementally calculate $\boldsymbol{\Sigma}_{\mathcal{D}_{i,k+1}}$ from an initial condition of $\boldsymbol{\Sigma}_{\mathcal{D}_{ii}} = \boldsymbol{O}_{6\times6}$.

ACKNOWLEDGMENT

The authors would like to thank Dirk Friesen for the engineering and assembly of electrical and mechanical hardware integrated with the USV platform, as well as their assistance with data collection.

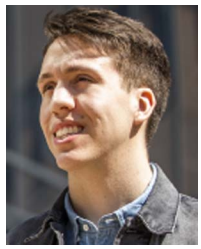
REFERENCES

[1] T. M. Browne et al., "Underwater bridge inspection," USA Federal Highway Admin., Office of Bridge Technol., Washington, DC, USA, Tech. Rep. FHWA-NHI-10-027, 2010.
[2] J. Shen et al., "Underwater inspection of bridge substructures using imaging technology," Federal Highway Admin., Washington, DC, USA, Tech. Rep. FHWA-HIF-18-049, 2018.
[3] F. Dellaert et al., "Factor graphs for robot perception," *Found. Trends Robot.*, vol. 6, no. 1/2, pp. 1–139, 2017.
[4] T. Shan, B. Englot, C. Ratti, and D. Rus, "LVI-SAM: Tightly-coupled lidar-visual-inertial odometry via smoothing and mapping," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2021, pp. 5692–5698.
[5] T. Qin, P. Li, and S. Shen, "VINS-Mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, Aug. 2018.
[6] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "LIO-SAM: Tightly-coupled lidar inertial odometry via smoothing and mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 5135–5142.
[7] S. Zhao, H. Zhang, P. Wang, L. Nogueira, and S. Scherer, "Super odometry: IMU-centric lidar-visual-inertial estimator for challenging environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 8729–8736.
[8] D. Wisth, M. Camurri, S. Das, and M. Fallon, "Unified multi-modal landmark tracking for tightly coupled lidar-visual-inertial odometry," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 1004–1011, Apr. 2021.
[9] D. Wisth, M. Camurri, and M. Fallon, "Vilens: Visual, inertial, lidar, and leg odometry for all-terrain legged robots," *IEEE Trans. Robot.*, vol. 39, no. 1, pp. 309–326, Feb. 2023.
[10] T.-M. Nguyen, M. Cao, S. Yuan, Y. Lyu, T. H. Nguyen, and L. Xie, "VIRAL-fusion: A visual-inertial-ranging-lidar sensor fusion approach," *IEEE Trans. Robot.*, vol. 38, no. 2, pp. 958–977, Apr. 2022.
[11] A. Tal, I. Klein, and R. Katz, "Inertial navigation system/doppler velocity log (INS/DVL) fusion with partial DVL measurements," *Sensors*, vol. 17, no. 2, 2017, Art. no. 415.
[12] D. Wang, X. Xu, Y. Yao, T. Zhang, and Y. Zhu, "A novel SINS/DVL tightly integrated navigation method for complex environment," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 7, pp. 5183–5196, Jul. 2020.
[13] L. Luo, Y. Zhang, T. Fang, and N. Li, "A new robust Kalman filter for SINS/DVL integrated navigation system," *IEEE Access*, vol. 7, pp. 51386–51395, 2019.
[14] G. Fukuda, D. Hatta, X. Guo, and N. Kubo, "Performance evaluation of IMU and DVL integration in marine navigation," *Sensors*, vol. 21, no. 4, 2021, Art. no. 1056.
[15] J. Han, J. Park, T. Kim, and J. Kim, "Precision navigation and mapping under bridges with an unmanned surface vehicle," *Auton. Robots*, vol. 38, no. 4, pp. 349–362, 2015.
[16] N. Charron, E. McLaughlin, S. Phillips, K. Goorts, S. Narasimhan, and S. L. Waslander, "Automated bridge inspection using mobile ground robotics," *J. Struct. Eng.*, vol. 145, no. 11, 2019, Art. no. 04019137.
[17] S. Phillips and S. Narasimhan, "Automating data collection for robotic bridge inspections," *J. Bridge Eng.*, vol. 24, no. 8, 2019, Art. no. 04019075.
[18] A. Palomer, P. Ridao, and D. Ribas, "Inspection of an underwater structure using point-cloud slam with an AUV and a laser scanner," *J. Field Robot.*, vol. 36, no. 8, pp. 1333–1344, 2019.
[19] D. Lattanzi and G. R. Miller, "3D scene reconstruction for robotic bridge inspection," *J. Infrastructure Syst.*, vol. 21, no. 2, 2015, Art. no. 04014041.
[20] A. Khaloo, D. Lattanzi, A. Jachimowicz, and C. Devaney, "Utilizing UAV and 3D computer vision for visual inspection of a large gravity dam," *Front. Built Environ.*, vol. 4, 2018, Art. no. 31.
[21] S. Zhao, F. Kang, J. Li, and C. Ma, "Structural health monitoring and inspection of dams based on UAV photogrammetry with image 3D reconstruction," *Autom. Construction*, vol. 130, 2021, Art. no. 103832.
[22] J. C. Leedekerken, M. F. Fallon, and J. J. Leonard, "Mapping complex marine environments with autonomous surface craft," in *Experimental Robotics*. Berlin, Germany: Springer, 2014, pp. 525–539.
[23] G. Papadopoulos, H. Kurniawati, A. S. B. M. Shariff, L. J. Wong, and N. M. Patrikalakis, "Experiments on surface reconstruction for partially submerged marine structures," *J. Field Robot.*, vol. 31, no. 2, pp. 225–244, 2014.
[24] J. Han and J. Kim, "Three-dimensional reconstruction of a marine floating structure with an unmanned surface vessel," *IEEE J. Ocean. Eng.*, vol. 44, no. 4, pp. 984–996, Oct. 2019.
[25] 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.
[26] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time," in *Proc. Robot.: Sci. Syst.*, 2014, vol. 2, pp. 1–9.
[27] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual–inertial odometry," *IEEE Trans. Robot.*, vol. 33, no. 1, pp. 1–21, Feb. 2017.
[28] S. Griffith, G. Chahine, and C. Pradalier, "Symphony lake dataset," *Int. J. Robot. Res.*, vol. 36, no. 11, pp. 1151–1158, 2017.
[29] Y. Cheng, M. Jiang, J. Zhu, and Y. Liu, "Are we ready for unmanned surface vehicles in inland waterways? The usvinland multisensor dataset and benchmark," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 3964–3970, Apr. 2021.
[30] A. Shurin et al., "The autonomous platforms inertial dataset," *IEEE Access*, vol. 10, pp. 10191–10201, 2022.
[31] B. Bingham et al., "Toward maritime robotic simulation in gazebo," in *Proc. IEEE/MTS OCEANS Conf.*, 2019, pp. 1–10.
[32] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *Proc.IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2004, vol. 3. pp. 2149–2154.
[33] G. Bradski, "The openCV library," *Dr Dobb's J.: Softw. Tools Professional Programmer*, vol. 25, no. 11, pp. 120–123, 2000.
[34] J. Kannala and S. Brandt, "A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 8, pp. 1335–1340, Aug. 2006.
[35] L. Zhang, D. Wisth, M. Camurri, and M. Fallon, "Balancing the budget: Feature selection and tracking for multi-camera visual-inertial odometry," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 1182–1189, Apr. 2022.
[36] *IEEE Standard Specification Format Guide and Test Procedure for Single-Axis Interferometric Fiber Optic Gyros*, IEEE Standard 952-1997, 1997, pp. 1–84.
[37] Z. Zhang and D. Scaramuzza, "A tutorial on quantitative trajectory evaluation for visual(-inertial) odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2018, pp. 7244–7251.
[38] T. Shan and B. Englot, "LeGO-LOAM: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 4758–4765.
[39] H. Ye, Y. Chen, and M. Liu, "Tightly coupled 3D lidar inertial odometry and mapping," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2019, pp. 3144–3150.
[40] E. Nelson, "Berkeley localization and mapping!." Accessed: Apr. 3, 2023. [Online]. Available: https://github.com/erik-nelson/blam
[41] M. Palieri et al., "LOCUS: A multi-sensor lidar-centric solution for high-precision odometry and 3D mapping in real-time," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 421–428, Apr. 2021.

[42] K. Chen, B. Lopez, A.-a. Agha-mohammadi, and A. Mehta, "Direct lidar odometry: Fast localization with dense point clouds," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 2000–2007, Apr. 2022.
[43] E. McLaughlin, N. Charron, and S. Narasimhan, "Automated defect quantification in concrete bridges using robotics and deep learning," *J. Comput. Civil Eng.*, vol. 34, no. 5, 2020, Art. no. 04020029.
[44] M. Berger et al., "A survey of surface reconstruction from point clouds," in *Proc. Comput. Graph. Forum*, 2017, vol. 36, pp. 301–329.
[45] E. Grilli, F. Menna, and F. Remondino, "A review of point clouds segmentation and classification algorithms," *Int. Arch. Photogrammetry, Remote Sens. Spatial Inf. Sci.*, vol. 42, 2017, Art. no. 339.
[46] T. Foote, "tf: The transform library," in *Proc. IEEE Conf. Technol. Practical Robot Appl.*, 2013, pp. 1–6.
[47] K. Shoemake, "Animating rotation with quaternion curves," in *Proc. 12th Annu. Conf. Comput. Graph. Interact. Techn.*, 1985, pp. 245–254.
[48] N. Charron, S. Phillips, and S. L. Waslander, "De-noising of lidar point clouds corrupted by snowfall," in *Proc. 15th Conf. Comput. Robot Vis.*, 2018, pp. 254–261.
[49] R. B. Rusu and S. Cousins, "3D is here: Point cloud library (PCL)," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 1–4. [Online]. Available: http://pointclouds.org/
[50] C. Hanchey, "Welland canal bridge 13 - main street bridge (Welland, Ontario)." Accessed: Apr. 3, 2023. [Online]. Available: https://flic.kr/p/XgW2FY
[51] L. Zhou, Z. Li, and M. Kaess, "Automatic extrinsic calibration of a camera and a 3D lidar using line and plane correspondences," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 5562–5569.
[52] P. Furgale, J. Rehder, and R. Siegwart, "Unified temporal and spatial calibration for multi-sensor systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 1280–1286.
[53] D. Gueriot, J. Chedru, S. Daniel, and E. Maillard, "The patch test: A comprehensive calibration tool for multibeam echosounders," in *Proc. IEEE/MTS OCEANS Conf. Exhib. Conf.*, 2000, vol. 3, pp. 1655–1661.
[54] N. Seube and R. Keyetieu, "Multibeam echo sounders-IMU automatic boresight calibration on natural surfaces," *Mar. Geodesy*, vol. 40, no. 2/3, pp. 172–186, 2017.
[55] Google. Accessed: Apr. 3, 2023. [Online]. Available: https://earth.google.com/web/,42.99172334,-79.25141246,175.12019361a,122.0026763d,35y,-5.63877461h,0t,0r
[56] T. Hackel, J. D. Wegner, and K. Schindler, "Contour detection in unstructured 3D point clouds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1610–1618.
[57] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "Supplementary material to: IMU preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation," Georgia Inst. Technol., Atlanta, GA, USA, Tech. Rep. GT-IRIM-CP&R-2015-001, 2015.

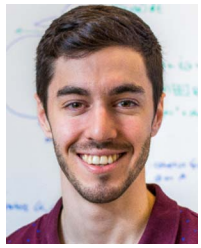
Alexander Thoms (Graduate Student Member, IEEE) received the BA.Sc. degree in civil engineering from the University of Waterloo, Waterloo, Canada in 2018. He is currently working toward the Ph.D. degree in mechanical engineering with the University of California, Los Angeles, USA.

His research interests include sensor fusion and state estimation for robotic inspection, particularly in marine and postdisaster scenarios.



Gabriel Earle (Graduate Student Member, IEEE) received the BA.Sc. degree in civil engineering from George Mason University, Fairfax, VA, USA, in 2020. He is currently working toward the Ph.D. degree in civil engineering with the University of Waterloo, Waterloo, Canada.

His research interests include developing interactive virtual reality simulations of infrastructure lifecycle planning.



Nicholas Charron received the BA.Sc. degree in civil engineering in 2016 from the University of Waterloo, Waterloo, Canada, where he is currently working toward the Ph.D. degree in civil and mechatronics engineering.

His research interests include simultaneous localization and mapping-based robotic systems for automating visual inspections of infrastructure.

Sriram Narasimhan (Member, IEEE) received the B.Eng. degree from Osmania University, Hyderabad, India, in 1993, the M.Eng. degree from the Asian Institute of Technology, Klong Luang, Thailand, in 1996, and the Ph.D. degree in structural engineering from Rice University, Houston, TX, USA, in 2005.

He is currently a Professor jointly appointed with the Departments of Civil and Environmental Engineering and Mechanical and Aerospace Engineering, University of California, Los Angeles, CA, USA. His research interests include sensing and robotics, emphasizing infrastructure condition assessment and inspections.