Received: 24 August 2023 | Revised: 11 March 2024 | Accepted: 22 May 2024

DOI: 10.1002/rob.22375

RESEARCH ARTICLE

WILEY

Pose-graph underwater simultaneous localization and mapping for autonomous monitoring and 3D reconstruction by means of optical and acoustic sensors

Alessandro Bucci[1,2] | Alessandro Ridolfi[1,2] | Benedetto Allotta[1,2]

[1]Department of Industrial Engineering, University of Florence, Florence, Italy

[2]Interuniversity Center of Integrated Systems for the Marine Environment (ISME), Genova, Italy

Correspondence
Alessandro Bucci, Department of Industrial Engineering, University of Florence, via di Santa Marta 3, 50139 Florence, Italy.
Email: alessandro.bucci@unifi.it

Abstract

Modern mobile robots require precise and robust localization and navigation systems to achieve mission tasks correctly. In particular, in the underwater environment, where Global Navigation Satellite Systems cannot be exploited, the development of localization and navigation strategies becomes more challenging. Maximum A Posteriori (MAP) strategies have been analyzed and tested to increase navigation accuracy and take into account the entire history of the system state. In particular, a sensor fusion algorithm relying on a MAP technique for Simultaneous Localization and Mapping (SLAM) has been developed to fuse information coming from a monocular camera and a Doppler Velocity Log (DVL) and to consider the landmark points in the navigation framework. The proposed approach can guarantee to simultaneously locate the vehicle and map the surrounding environment with the information extracted from the images acquired by a bottom-looking optical camera. Optical sensors can provide constraints between the vehicle poses and the landmarks belonging to the observed scene. The DVL measurements have been employed to solve the unknown scale factor and to guarantee the correct vehicle localization even in the absence of visual features. Furthermore, to evaluate the mapping capabilities of the SLAM algorithm, the obtained point cloud is elaborated with a Poisson reconstruction method to obtain a smooth seabed surface. After validating the proposed solution through realistic simulations, an experimental campaign at sea was conducted in Stromboli Island (Messina), Italy, where both the navigation and the mapping performance have been evaluated.

KEYWORDS

3D robotic mapping, field robotics, SLAM, underwater navigation

1 | INTRODUCTION

From geology to exploration and surveillance of archeological sites and from the oil and gas industry to reconnaissance for military purposes, exploring and understanding seas and oceans is a matter of primary importance. Considering their human hostile nature, since the 1960s, seas and oceans have been explored with the aid of robots. The first Unmanned Underwater Vehicles (UUVs) were teleoperated ones and are referred in the technical literature as Remotely Operated Vehicles (ROVs). A cable, usually called an umbilical cable, acts as a constant connection providing power and communications, and specialized operators are thus able to control the vehicle using the feedback forwarded by the onboard sensors. In the last decades Autonomous Underwater Vehicles (AUVs), which are completely autonomous, have gained interest with respect to ROVs. Indeed, such vehicles do not require human intervention (except for deployment and recovery), are usually equipped with electric batteries, and possess dedicated systems used to control their motion. Since the demanded tasks of underwater vehicles have become more and more challenging (Ferri et al., 2017; Prats et al., 2012), researchers and scientists are following the tide of change and are pushing the boundaries of AUVs capabilities by integrating cutting-edge technologies. Indeed, autonomous inspection (Cashmore et al., 2014), intervention (Youakim et al., 2020) strategies for underwater installations, exploration planning solutions (Vidal et al., 2020), and autonomous coverage approaches (Paull et al., 2012) have become essential tools to execute demanding and hazardous subsea operations.

One of the most significant and complex tasks in autonomous underwater exploration is to retrieve the vehicle's pose within the surrounding environment, making use of precise and reliable navigation and localization systems, which are necessary regardless of the kind of mission or task the underwater vehicle is required to perform. In addition to this, perceptual devices (such as optical cameras and acoustic devices) able to sense the surrounding environment have been earning attention throughout the last decades to acquire data for monitoring and inspection purposes. The use of optical and acoustic equipment to aid navigation has emerged as a relevant alternative or support to traditional navigation sensors.

Several algorithms have been developed throughout the years to increase the navigation and localization capabilities of the AUVs relying on Bayesian estimators, such as Kalman filtering and Maximum A Posteriori (MAP) estimators. Both Extended Kalman Filter (EKF) (Dissanayake et al., 2001) and least-squares optimization (Dellaert & Kaess, 2006) have been used extensively in Simultaneous Localization and Mapping (SLAM) research in the past (Zhang et al., 2018). Earlier SLAM research has used EKF algorithms where the state vector contained the latest robot pose and the positions of the observed features. However, it has been shown that EKF–SLAM could result in inconsistent estimates (Castellanos et al., 2004; Julier & Uhlmann, 2001), as the estimated covariance from the algorithm can violate the theoretically achievable lower bounds (Dissanayake et al., 2001; Huang & Dissanayake, 2007). On the contrary, optimization-based SLAM uses a state vector containing all the robot poses and all the features observed. Considering that relinearization is performed during each iteration step, there is no inconsistency issue in optimization-based SLAM, and thus, the quality of the estimate is higher than that of EKF–SLAM.

Consequently, to overcome the limitations introduced by the Kalman filter strategies, which condense all the history into the last estimation, a sensor fusion MAP algorithm has been developed for underwater navigation in the context of this work. Due to the complexity of retrieving navigation information in the underwater environment, a sensor fusion approach has been used. The performance and robustness of the visual SLAM algorithm heavily rely on the quality of the images and salient features. Consequently, the visual SLAM system has been fused with other sensing algorithms, such as the Doppler Velocity Log (DVL). Very few works still exist on underwater SLAM fusing data from a monocular camera and a DVL (Kim & Eustice, 2013; Ozog & Eustice, 2013). Despite that, fusing an optical and an acoustic sensor in a MAP-based framework can take advantage of both sensors, which have an excellent complement to each other. This developed solution can be employed to locate the vehicle and map the seabed at the same time in a unified framework. Thus, an underwater visual acoustic SLAM strategy which integrates DVL with a visual SLAM system has been developed to perform accurate navigation and mapping tasks at the same time. Particular attention has been focused on the design of scale factor ambiguity resolution and extrinsic calibration optimization procedure and on implementing a reset procedure to reduce the computational burden. Furthermore, the proposed strategy has been tested with both simulated and experimental data to evaluate the navigation performance and has been compared with an Unscented Kalman Filter (UKF)-based algorithm, whose performance has been accurately discussed in authors' previous works (Bucci et al., 2021, 2023).

The paper is organized as follows: state-of-the-art SLAM strategies are detailed in Section 2, whereas Section 3 is dedicated to introduce the MAP estimation approach. Section 4 outlines the development of the factor graph framework, whereas some improvements and peculiarities of the proposed SLAM strategy are reported in Section 5. While navigation results obtained from a simulated environment and from an experimental campaign are reported, respectively, in Sections 6 and 7, an analysis of the mapping capabilities is depicted in Section 8. Finally, Section 9 draws conclusions.

2 | RELATED WORKS

Many estimation problems in robotics have an underlying optimization problem (Dellaert, 2021). In most of these optimization problems, the objective to be maximized or minimized is composed of many different factors (e.g., a Global Navigation Satellite System [GNSS] measurement is applied to the pose of the vehicle at a particular time and can be referred to as an unary factor, an Inertial

Measurement Unit [IMU] measurement can be related to two vehicle states at adjacent times and can represent an odometry factor). The use of factorial graphs in the design of algorithms for robotic applications has three main advantages. First, since many optimization problems in robotics have the property of locality, factorial graphs can model a wide variety of problems in all robotics domains, such as tracking, navigation, and mapping. Second, by clearly exposing the structure of the problem, reflection on factorial graphs offers many opportunities to improve the performance of the algorithms employed to solve the problem. Indeed, many classical algorithms, which can be viewed as the application of the elimination algorithm to a particular type of factorial graph, are only optimal for a small class of problems. In many applications, knowledge of the specific structure of the problem domain can improve the execution time of inference by orders of magnitude. Similarly, well-known algorithmic ideas from linear algebra can be generalized to factorial graphs, leading, for example, to incremental inference algorithms. Third, apart from performance considerations, factorial graphs are useful when designing and thinking about how to model a problem, providing a common language to express ideas to collaborators and users of a particular algorithm. After working with factor graphs for a while, one begins to identify factor types as a particularly useful design unit. A factor type specifies how many variables a factor is related to and the semantics associated with the function to be calculated.

MAP estimation has recently become the standard approach for modern SLAM strategies (Cadena et al., 2016). Indeed, while fixed-lag smoothers and filtering solutions restrict the inference within a window of the latest states or to the latest state, respectively, MAP strategies estimate the entire history of the system by solving a nonlinear optimization problem. Both fixed-lag smoothers and filters marginalize older states, collapsing the corresponding information (usually) in a Gaussian prior. This approach can lead to reduced robustness against outlier data (Forster et al., 2016). Since MAP strategies can quickly lead to an unsuitable approach for real-time applications, the development of incremental smoothing techniques has arisen as the state-of-the-art approach. Such techniques can reuse previously calculated quantities when new measurements or variables are added (Kaess et al., 2008, 2012). In particular, in Kaess et al. (2012), a Bayes tree data structure is employed to perform incremental optimization on the factor graph. Also, the adopted solution possesses the ability to identify and update only a small subset of variables by accurately selecting the ones affected by the new measurement. A complete review can be found in Grisetti et al. (2020) and the references therein.

Considering the underwater domain, two works have been taken as inspiration for the development of the factor graph employed in the proposed SLAM strategy. Westman and Kaess (2019) propose an algorithm to generate pose-to-pose constraints for pairs of SONAR images and to fuse these resulting pose constraints with the vehicle odometry in a pose-graph optimization framework. In Franchi et al. (2021), Ultra-Short BaseLine measurements are exploited as observations within the onboard navigation filter, where the localization task is solved as a MAP estimation problem. Both these solutions rely on Incremental Smoothing and Mapping 2 (iSAM2) (Kaess et al., 2008, 2012), which is the last evolution of the iSAM solution developed in Georgia Tech Smoothing and Mapping (GTSAM) (Dellaert, 2012). Furthermore, other graph-based SLAM strategies have been proposed to fuse the data obtained by the navigation sensors and the perception sensors, both acoustic and optical. In Fallon et al. (2013) this approach is used in an AUV for mine counter measurement and localization. While the graph is initialized by pose node from a Global Positioning System, a nonlinear least-square optimization is performed with the DVL and IMU-based Dead Reckoning (DR) estimations and the SONAR images. In Huang and Kaess (2015), an acoustic structure from a motion algorithm for recovering three-dimensional (3D) scene structure from multiple 2D SONAR images while at the same time localizing the SONAR is presented.

Turning to visual SLAM, ORB-SLAM (Mur-Artal et al., 2015) is one of the most complete and simple algorithms, and the whole system is calculated around Speeded Up Robust Features (ORB) feature points, with features such as rotational scale invariance and fast detection. ORB-SLAM2 (Mur-Artal & Tardós, 2017) is upgraded from ORB-SLAM, supporting monocular, binocular, and RGB-D modes, and has good adaptability. Finally, the latest ORB-SLAM3 (Campos et al., 2021) algorithm fuses optical images with inertial sensors. The excellent characteristics of the ORB-SLAM2 algorithm, which can achieve centimeter-level precision on the ground, represent an incentive for its application in underwater environments. Consequently, the visual part of the developed SLAM algorithm takes inspiration from the ORB-SLAM2 framework. Referring to the vision-based SLAM algorithm for underwater navigation and mapping, Hong and Kim (2020) address a visual mapping method for precise camera trajectory estimation and 3D reconstruction of underwater ship hull surfaces using a monocular camera as the primary sensor. Du et al. (2017) propose an underwater visual SLAM system using a stereo camera, which has been tested in a circular pool.

Finally, an acoustic-visual-inertial SLAM strategy has been proposed by Rahman et al. (2018a, 2018b). Data coming from a mechanical scanning SONAR, a stereo camera, and proprioceptive inertial sensors are fused in a tightly coupled nonlinear optimization to estimate the vehicle trajectory and reconstruct the surrounding environment. There are few works where the DVL measurements are fused with other perception sensors in SLAM strategies. In Ozog and Eustice (2013), a SLAM method, which uses a very sparse point cloud derived from a DVL to add constraints to a piecewise-planar framework, is proposed. While the focus of the paper is the evaluation of visual saliency to select which imagery has to be considered informative in terms of a SLAM algorithm, a strategy where a monocular camera is employed together with an odometry algorithm based on DVL, IMU and pressure Depth Sensor (DS) is presented in Kim and Eustice (2013). Fiducial markers are also integrated into a visual SLAM framework with DVL, IMU, and DS in Westman and Kaess (2018).

The main contribution of this work with respect to the state of the art is the development of a sensor fusion algorithm relying on a MAP technique to fuse information coming from a monocular camera and a DVL and to consider the landmark points in the navigation framework. In particular, considering the changing visibility conditions, which affect the underwater environment, fusing visual and acoustic measurements increases the robustness of the algorithm. The fusion of an optical and an acoustic sensor in a MAP-based framework can benefit from both sensors, which complement each other. The improvements related to the scale factor resolution and to the reset procedures have been developed to face the main issues of the SLAM algorithms, such as the computational load and the size of the factor graph.

3 | MAP ESTIMATION

A navigation and mapping problem is a problem where the unknown state variables $X = \{x_1, x_2, \ldots, x_M\}$ constituted of poses and landmarks have to be determined given the measurements $Z = \{z_1, z_2, \ldots, z_N\}$. The MAP estimator maximizes the posterior density $p(X|Z)$ of the states X given the measurements Z:

$$X^{MAP} = \underset{X}{\operatorname{argmax}}\, p(X)l(Z|X) = p(X)\prod_{i=1}^{N} l(z_i|X), \tag{1}$$

where $l(z_i|X)$ is the likelihood distribution and an additive Gaussian noise is assumed in all measurement models, as reported in Equation (2).

$$p(z_i|X) = \mathcal{N}(h_i(X), \Sigma_i) \propto \exp\left(-\frac{1}{2}\|h_i(X) - z_i\|^2_{\Sigma_i}\right), \tag{2}$$

where $h_i(X)$ is the measurement function, which maps the state estimate X into a predicted value $\hat{z}_i$ of the measurement z_i and Σ_i is the covariance matrix, which summarizes the uncertainty of the measurement model. By applying the monotonic logarithmic function and the Gaussian model previously introduced, the optimization problem can be simplified into a nonlinear least-square problem:

$$X^{MAP} = \underset{X}{\operatorname{argmin}} \sum_{i=1}^{N} \|h_i(X) - z_i\|^2_{\Sigma_i}, \tag{3}$$

where

$$\|h_i(X) - z_i\|^2_{\Sigma_i} = (h_i(X) - z_i)^\top \Sigma_i^{-1} (h_i(X) - z_i) \tag{4}$$

is the Mahalanobis distance.

The nonlinear problem can be solved through standard methods, such as the Gauss–Newton or the Levenberg–Marquardt algorithms, which iteratively converge to the solution by solving the linear approximation of the nonlinear system. More information can be found in Grisetti et al. (2020) and Dellaert and Kaess (2017).

4 | FACTOR GRAPH FRAMEWORK DEVELOPMENT

The mathematical modeling of the factors used to represent the measurement constraints to solve the autonomous navigation and mapping problem is presented. Inspired by Westman and Kaess (2018, 2020), the factors described below have been employed, where it is necessary to consider that the information included in some factors can be derived from measurements not coming from a single sensor. The state of the system at instant i is defined as a complete pose belonging to $SE(3)$, which can be expressed mathematically as

$$T_{x_i} = \begin{bmatrix} R_i & \mathbf{t}_i \\ \mathbf{0}^{1\times 3} & 1 \end{bmatrix}, \tag{5}$$

where $R_i \in SO(3)$ is the rotation matrix and $\mathbf{t}_i \in \mathbb{R}^3$ represents the translation vector. Defining the set of poses at time k with $\mathcal{X}_k$, such that $\mathcal{X}_k = \{T_{x_i}\}_{i=0,1,\ldots,k}$, it is possible to define the optimization problem and, in particular, Equation (4) on the smooth manifold $SE(3)$. Considering a transformation from the state x_i to the state x_j constrained with an odometry measurement $z_{i,j}$ with covariance $\Sigma_{i,j}$, Equation (4) becomes

$$\|f_{ij}(x_i, x_j) \ominus z_{i,j}\|^2_{\Sigma_{i,j}} = \left\|\log\left(T^{-1}_{z_{i,j}} T^{-1}_{x_i} T_{x_j}\right)\right\|^2_{\Sigma_{i,j}}. \tag{6}$$

The symbol $\ominus$ encodes the logarithmic map from the manifold to an element of the $SE(3)$ Lie algebra, where $f_{ij}(\cdot)$ represents the measurement function applied to the poses T_{x_i} and T_{x_j}. For ease of explanation, T_{x_i} can be represented with the vector $[X_{x_i} \quad Y_{x_i} \quad Z_{x_i} \quad \phi_{x_i} \quad \theta_{x_i} \quad \psi_{x_i}] \in \mathbb{R}^6$, and the measurement function becomes

$$f_{ij}(x_i, x_j) = [X_{x_{i,j}} \quad Y_{x_{i,j}} \quad Z_{x_{i,j}} \quad \phi_{x_{i,j}} \quad \theta_{x_{i,j}} \quad \psi_{x_{i,j}}]^\top. \tag{7}$$

In contrast, for a measurement z_i that indicates local information on the state x_i with covariance Σ_i, Equation (4) is

$$\|f_i(x_i) \ominus z_i\|^2_{\Sigma_i} = \left\|\log\left(T^{-1}_{z_i} T_{x_i}\right)\right\|^2_{\Sigma_i}, \tag{8}$$

where the measurement function $f_i(\cdot)$ applied to the pose T_{x_i} can be defined as

$$f_i(x_i) = [X_{x_i} \quad Y_{x_i} \quad Z_{x_i} \quad \phi_{x_i} \quad \theta_{x_i} \quad \psi_{x_i}]^\top. \tag{9}$$

The information from the available onboard sensors has been encoded as measurement factors to constrain the optimization, whose solution represents the MAP estimate. Inspired by Westman and Kaess (2019), the following factors have been included:

- a relative 4D pose-to-pose constraint on x, y, and z translation and yaw rotation, thanks to the measurements coming from the DVL and the yaw estimated by the attitude estimator (Allotta et al., 2015, 2016);

- a unary 2D constraint on pitch and roll rotations, obtained from the attitude estimation filter;
- a unary 1D constraint on z translation thanks to the DS measurements;
- a unary constraint on x and y translation exploiting GNSS observations;
- a relative 6D pose-to-pose constraint on x, y, and z translation and roll, pitch, and yaw rotation, thanks to the relative pose estimated through the monocular camera and properly scaled with the algorithm described in Section 5.1;
- a camera-based landmark constraint on the vehicle pose and the landmark position for each feature seen with the monocular camera over multiple images.

The implemented approach adds a new state only when at least one observation from GNSS, DVL, DS, or, when the visibility is acceptable, the camera is available. The link between adjacent nodes is maintained by collapsing the relative motion *XYZ–Y* in a single compound constraint, where simple DR is performed between the two consecutive nodes with the last acquired DVL measurements. The pose T_{x_i} can be represented with a vector $[X_{x_i} \quad Y_{x_i} \quad Z_{x_i} \quad \phi_{x_i} \quad \theta_{x_i} \quad \psi_{x_i}] \in \mathbb{R}^6$ that encodes the state at the generic instant. Mathematically, at time k, the optimization problem can be written as

$$\begin{aligned}\mathcal{X}_k^* = \underset{X}{\operatorname{argmax}} & \sum_{i=1}^{k-1} \left(\| m_{XYZ-Y}(x_{i-1}, x_i) \ominus o_{i-1,i} \|^2_{\Sigma_{o_{i-1,i}}} + \| m_{RP}(x_i) \ominus r_i \|^2_{\Sigma_{r_i}} \right) \\ & + \sum_{i \in \mathcal{Z}} \| m_Z(x_i) - z_i \|^2_{\Sigma_{z_i}} \\ & + \sum_{i \in \mathcal{G}} \| m_{XY}(x_i) - \boldsymbol{g}_i \|^2_{\Sigma_{g_i}} \\ & + \sum_{i,j \in C} \| m_{XYZ-RPY}(x_i, x_j) \ominus p_{i,j} \|^2_{\Sigma_{p_{i,j}}} \\ & + \sum_{j \in \mathcal{LM}, i \in C} \rho \left(\| \boldsymbol{p}_{ij} - \pi_i(T_{x_i} \boldsymbol{P}_j) \|^2_{\Sigma_{lm_j}} \right) \\ & + \| T_{x_0} \ominus T_{x_{prior}} \|^2_{\Sigma_{lm_j}},\end{aligned} \tag{10}$$

$\{m_{XYZ-Y}(\cdot), o_{i-1,i}, \Sigma_{o_{i-1,i}}\}$, $\{m_{RP}(\cdot), r_i, \Sigma_{r_i}\}$, $\{m_Z(\cdot), z_i, \Sigma_{z_i}\}$, $\{m_{XY}(\cdot), \boldsymbol{g}_i, \Sigma_{g_i}\}$, $\{m_{XYZ-RPY}(\cdot), p_{i,j}, \Sigma_{o_{i,j}}\}$ are the measurement functions, the measured values and covariances associated with the previously introduced factors. In particular, $o_{i-1,i}$, r_i, and $p_{i,j}$ represent, on $SE(3)$, the observation for the *XYZ–Y* part, the RP part and the camera-based *XYZ–RPY* part, respectively, $z_i \in \mathbb{R}$ is the depth measurement, $\boldsymbol{g}_i \in \mathbb{R}^2$ is the GNSS measurement. The measurement functions are:

$$\begin{aligned} & m_{XYZ-Y}(x_{i-1}, x_i) = [X_{x_{i-1,i}} \quad Y_{x_{i-1,i}} \quad Z_{x_{i-1,i}} \quad \psi_{x_{i-1,i}}]^\top, \\ & m_{RP}(x_i) = [\phi_{x_i} \quad \theta_{x_i}]^\top, \\ & m_Z(x_i) = [Z_{x_i}], \\ & m_{XY}(x_i) = [X_{x_i} \quad Y_{x_i}]^\top, \\ & m_{XYZ-RPY}(x_i, x_j) = [X_{x_{i,j}} \quad Y_{x_{i,j}} \quad Z_{x_{i,j}} \quad \phi_{x_{i,j}} \quad \theta_{x_{i,j}} \quad \psi_{x_{i,j}}]^\top. \end{aligned} \tag{11}$$

Thanks to the features extracted from optical images and matched through multiple keyframes, it is possible to optimize map point locations $\boldsymbol{P}_j \in \mathbb{R}^3$ and keyframe poses $T_{x_i} \in SE(3)$ minimizing the reprojection error with respect to the matched keypoints $\boldsymbol{p}_{ij} \in \mathbb{R}^2$. The error term for the observation of a map point j in a keyframe i is

$$\boldsymbol{e}_{ij} = \boldsymbol{p}_{ij} - \pi_i(T_{x_i} \boldsymbol{P}_j), \tag{12}$$

where $\pi_i(\cdot)$ is the projection function:

$$\pi_i(T_{x_i} \boldsymbol{P}_j) = \begin{bmatrix} f_x \frac{x_{ij}}{z_{ij}} + c_x \\ f_y \frac{y_{ij}}{z_{ij}} + c_y \end{bmatrix}, \tag{13}$$

where (f_x, f_y) and (c_x, c_y) are, respectively, the focal length and the principal point of the camera and $[x_{ij} \quad y_{ij} \quad z_{ij}]^\top$ are the coordinates of the point. The cost function to be minimized can be defined as

$$f_{LM}(x_i) = \rho \left(\| \boldsymbol{p}_{ij} - \pi_i(T_{x_i} \boldsymbol{P}_j) \|^2_{\Sigma_{lm_j}} \right), \tag{14}$$

where $\rho(\cdot)$ is the Huber robust cost function and Σ_{lm_j} is the covariance matrix associated with the scale at which the keypoint i was detected. While $\mathcal{Z}, \mathcal{G}$, and C are the set of pose nodes for which DS, GNSS and camera measurements, respectively, occur, $\mathcal{LM}$ is the set of landmark nodes. $T_{x_{prior}}$ is the prior constraint on the first pose, which is necessary to anchor the state evolution to a global coordinate frame (Figure 1).

In terms of implementation, the GTSAM library has been used as the back-end for the localization solution. Further information can be found in Kaess et al. (2008, 2012) and Dellaert (2012). iSAM2, which is the latest evolution of the iSAM solution developed in GTSAM, allows only the typical small subset of variables affected by a new measurement, that is, the measurement function and associated covariances, to be identified and updated, thus limiting the computational load of the estimation, offering a trade-off between accuracy and efficiency. Several issues affect the vision in underwater environments, which can negatively influence the employment of visual SLAM algorithms. Specifically, while scattering reduces light intensity causing a loss of contrast and haze in underwater images, light absorption leads to a decrease in the color quality of underwater images. Light attenuation in water introduces degradation in underwater images, such as poor colors, decreased contrast, haziness, and blurring, making them hardly usable for the filter. Thus it is necessary to guarantee that the visual part of the navigation framework, which is dependent on uncontrollable external conditions, can be correctly inserted or removed from the factor graph. Only when the visual SLAM algorithm is correctly initialized and the current scale factor is computed, it is possible to insert keyframe poses and map point locations in the factor graph. Edges computed thanks to other onboard sensors, which do not suffer from visibility limitations, are inserted in the whole factor graph during the entire mission. The developed system, through the map points obtained from the vision system and the DVL beams, can build a map of the surrounding environment independently from the visibility. Indeed, when the reduced visibility impedes the usage of the visual-based features as map points, DVL-based beams can be employed to build an approximated map of the sea bottom. The quality and resolution of the produced map depend on the availability of the visual landmarks. Still, thanks to the DVL measurements, the reconstruction can be performed for the whole mission. Considering that, when an AUV

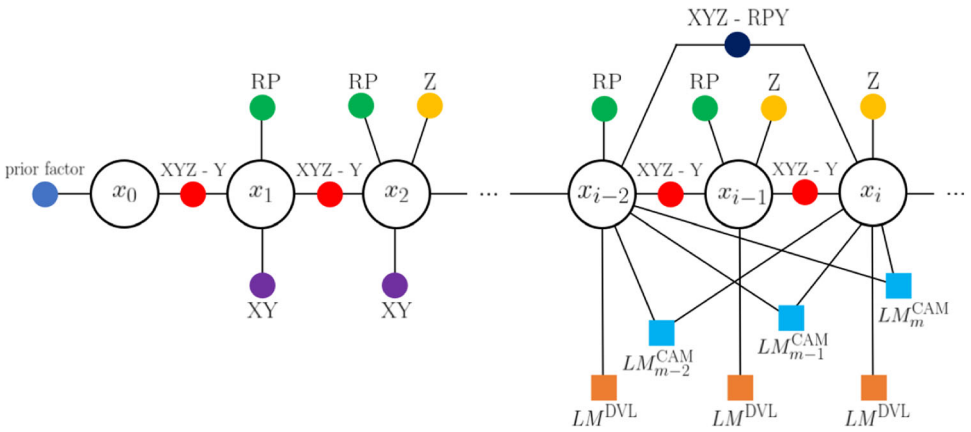

FIGURE 1 Example of the factor graph at the iteration i constrained with vision-based landmarks and all the onboard sensors. [Color figure can be viewed at wileyonlinelibrary.com]

accomplishes an underwater mission, the sea bottom texture can change very fast, and its depth can increase rapidly, the possibility to guarantee reconstruction of the surrounding environment, even approximated, represents a helpful advantage. Obviously, it is necessary to highlight that the DVL beams cannot be employed as landmark nodes in the factor graph. Still, they can only be added to the map utilizing the sensor geometrical model. It is important to remark that underwater SLAM fusing camera, and DVL sensors can increase the localization accuracy and robustness thanks to the excellent complement between these two sensors: DVL provides reliable motion estimates for underwater visual SLAM, extending SLAM's robustness and operation even without visual features, and vision, when applicable, helps the estimation process by introducing visual landmarks which increase the constraints on the vehicle position.

Turning to the specific strategies for DVL and camera-based factor graph constraint computation, the following approaches have been employed. While the primary application field of a DVL is vehicle navigation through a DR strategy that computes the AUV position by integrating the measured linear velocity, the DVL has four acoustic beams, each pointing in a different direction, which can be employed to acquire the 3D location of 4 points of the sea bottom during each speed measurement. The points located thanks to the DVL beams cannot be employed as additional constraints in the navigation pose graph because they do not link any node of the graph. Still, they can easily be used to increase the number of points in the estimated map of the sea bottom. Indeed, by knowing the vehicle's actual position from the navigation algorithm, the location of the four beams can be converted from the DVL frame to the North, East, and Down (NED) reference system. The visual SLAM algorithm employed in the developed navigation framework is a feature-based monocular SLAM system that operates to estimate the camera trajectory and an environment map. The basic idea of the SLAM system introduced in the navigation filter takes inspiration from the algorithms proposed by Mur-Artal et al. (2015) and Mur-Artal and Tardós (2017). Furthermore, following the results reported by Zacchini et al. (2019) and Bucci et al. (2022), where accurate comparisons between several feature detectors are explained, the ORB feature detector has been chosen as the preferable solution instead of Scale Invariant Feature Transform, Speeded Up Robust Features, and Accelerated-KAZE. Considering that a monocular camera is employed, a scale factor ambiguity to be solved features the visual-based estimate.

5 | FACTOR GRAPH FRAMEWORK IMPROVEMENTS

5.1 | Scale factor ambiguity resolution

This procedure, which is executed every time the visual SLAM algorithm is correctly initialized, has two main purposes, the scale factor ambiguity resolution and accurate compensation of the fixed rototranslation between the camera and the body frames. This transformation is represented as a similarity transformation composed of a scale factor s, a translation vector $\boldsymbol{t}_{c,b} = \begin{bmatrix} t_{c,b}^x & t_{c,b}^y & t_{c,b}^z \end{bmatrix}^\top$ and a rotation matrix $R_c^b = R_z(\psi_c^b)R_y(\theta_c^b)R_x(\phi_c^b)$. It is based on comparing the trajectories estimated through the DVL, the other inertial sensors, and the camera. It is necessary to notice that until the scale factor has not been estimated, the measurements obtained thanks to the visual SLAM algorithm are not inserted in the whole factor graph. Considering this algorithm's two purposes and that, usually, underwater vehicles for survey missions execute planar trajectories at constant depth, the problem has been solved with a two-step algorithm. In particular, while the first part of the algorithm determines a closed-form solution for the x- and y-directions, yaw rotation, and scale factor, the second part optimizes the whole scaled rototranslation with an iterative algorithm. This framework has been adopted due to the limitations introduced by the particular motion executed by the AUV. Indeed, on the one hand, the optimal closed-form solution estimated with 3D points that almost lie on a plane

cannot correctly estimate the roll and pitch angles of the rigid transformation between the two considered reference frames. On the other hand, the iterative algorithm locally converges and requires an initial guess in the neighborhood of the exact solution, which can be measured directly on the vehicle or evaluated through the closed-form solution.

The two steps of the algorithm are described in detail. First, the closed-form solution is found by computing the trajectory alignment transformation with a translational component on the *xy*-plane of the trajectory estimated with the DVL and the camera and with a rotational component computed with respect to the perpendicular axis to this plane. Given the DVL-based positions $\{\boldsymbol{p}_i^{DVL}\}_{i=1}^{N}$ and the camera-based positions $\{\boldsymbol{p}_i^{CAM}\}_{i=1}^{N}$, it is necessary to determine the optimal similarity transformation $S^* = \left\{s^*, R_c^{b*}, \boldsymbol{t}_{c,b}^*\right\} = \left\{s^*, \psi_c^{b*}, t_{c,b}^{x*}, t_{c,b}^{y*}\right\}$ that satisfies the minimization problem reported in Equation (15).

$$S^* = \underset{s,R_c^b,\boldsymbol{t}_{c,b}}{\operatorname{argmin}} \sum_{i=1}^{N} \left\| \boldsymbol{p}_i^{DVL} - sR_c^b \boldsymbol{p}_i^{CAM} - \boldsymbol{t}_{c,b} \right\|^2, \tag{15}$$

where it is necessary to suppose that

$$R_c^b = R_z\left(\psi_c^b\right), \tag{16}$$

$$\boldsymbol{t}_{c,b} = \begin{bmatrix} t_{c,b}^x & t_{c,b}^y & 0 \end{bmatrix}^\top. \tag{17}$$

The solution to this least-squares problem can be found using the method explained in Umeyama (1991).

The second step works with Ceres Solver, an open-source library that provides a rich set of tools to construct and solve an optimization problem. Ceres solves robustified bounds constrained nonlinear least-squares problems of the form:

$$\min_{\mathbf{x}} \frac{1}{2} \sum_i \rho_i \left(\left\| f_i\left(x_{i_1}, \ldots, x_{i_k}\right) \right\|^2 \right). \tag{18}$$
$$l_j \le x_j \le u_j$$

The expression $\rho_i\left(\left\| f_i\left(x_{i_1}, \ldots, x_{i_k}\right)\right\|^2\right)$ represents the residual block, where $\rho_i(\cdot)$ is the loss function used to reduce the influence of outliers on the solution and $f_i(\cdot)$ is the cost function that depends on the parameters block $\{x_{i_1}, \ldots, x_{i_k}\}$. l_j and u_j are the lower and upper bounds on the parameter block x_j.

Defining the state $\boldsymbol{x} = \begin{bmatrix} s & \phi_c^b & \theta_c^b & \psi_c^b & t_{c,b}^x & t_{c,b}^y & t_{c,b}^z \end{bmatrix}^\top$, the loss function is assumed to be the identity function, the cost function is the same as in the first step of the algorithm

$$f(\boldsymbol{x}) = \boldsymbol{p}_i^{DVL} - sR_c^b \boldsymbol{p}_i^{CAM} - \boldsymbol{t}_{c,b}, \tag{19}$$

where, unlike the previous case, it is supposed that

$$R_c^b = R_z\left(\psi_c^b\right) R_y\left(\theta_c^b\right) R_z\left(\phi_c^b\right), \tag{20}$$

$$\boldsymbol{t}_{c,b} = \begin{bmatrix} t_{c,b}^x & t_{c,b}^y & t_{c,b}^z \end{bmatrix}^\top. \tag{21}$$

The initial guess and the upper and lower bounds are computed thanks to the values estimated in the closed-form solution. Considering that this is a small problem with few parameters and relatively dense Jacobians, dense QR factorization is the method of choice (Björck, 1996). It is necessary to notice that the procedure is applied when a fixed number of positions estimated through the strategy based on the camera and through the one based on the DVL have been acquired. Then, the two optimizations are applied to the acquired data. To evaluate the goodness of the estimated outcomes of the scale factor ambiguity algorithm, a metric has been employed. In particular, first, the estimated rototranslation and scale factor value are applied to the trajectory estimated through the camera. Second, the obtained trajectory is compared with the DVL-based one to evaluate the distance between each corresponding point. If the sum of the distances is lower than a threshold value, the estimated rototranslation and scale factor are considered acceptable (Figure 2).

5.2 | Reset procedures

Although iSAM2 reduces the variables to be optimized to a small subset, it is necessary to apply a reset procedure to maintain a limited factor graph and avoid increasing nodes and edges. In particular, considering that the presence of visual landmark nodes constrains several pose nodes, the computational burden tends to increase at every iteration step, and the factor graph is more arduous to be managed. Two-factor graph reset procedures have been developed to avoid the increase of the graph size, where the first is dedicated to compacting the factor graph without reducing the visual landmark nodes, and the second operates on the whole factor graph reducing all the information to the ones contained in the last node. While the first reset strategy will be called keyframe reset, the second one will be referred to as global reset. One of the two reset strategies is applied when the number of pose nodes of the factor graph reaches a value equal to N. The status of the factor graph is checked to decide which one of the two strategies is applied. In particular, the keyframe reset procedure is recalled only if the visual SLAM algorithm is active and for a maximum number of consecutive times equal to p. The last condition is set to maintain control of the increase of the execution time of each filter iteration. When the visual SLAM part of the navigation algorithm is not working due to the external visibility conditions or when the factor graph is reset for the $(p + 1)$th time, the global reset algorithm is employed. It is necessary to notice that the keyframe reset procedure does not delete all the information contained in the previous pose graph. Still, only the ones related to the IMU, DVL, and DS measurements are removed. Indeed, this information is compressed in a new framework, which contains all the properties to be transferred from the previous to the following factor graph. On the contrary, the global factor reset reduces all the information to be transferred to the new factor graph to the ones in the last node of the previous factor graph.

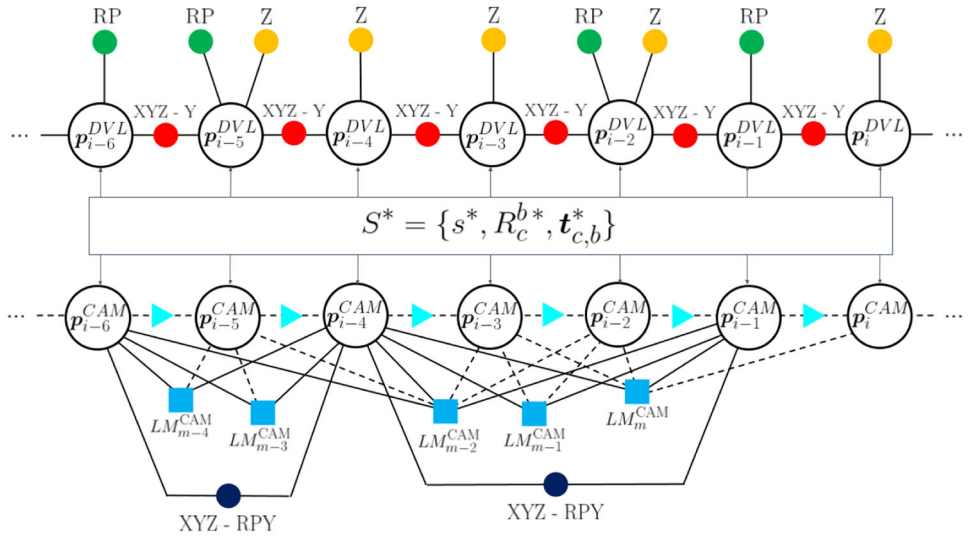

FIGURE 2 Comparison of the two-factor graphs (e.g., the DVL-based graph on the top and the camera-based graph on the bottom of the image) employed for the scale factor ambiguity resolution. The dashed lines in the bottom graph are the edges which are not reported in the whole graph. For ease of reading, a one-to-one association between the two graphs is considered. CAM, camera; DVL, Doppler Velocity Log. [Color figure can be viewed at wileyonlinelibrary.com]

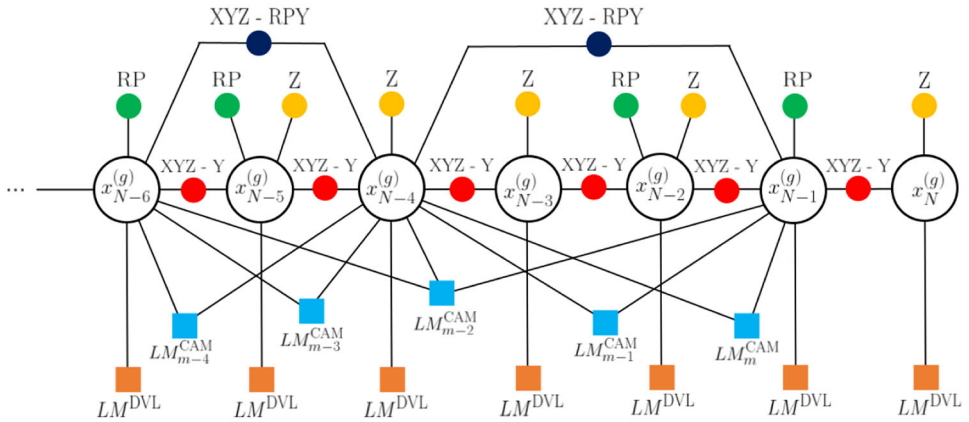

FIGURE 3 Last nodes of the factor graph g constrained with vision-based landmarks and all the onboard sensors. CAM, camera; DVL, Doppler Velocity Log. [Color figure can be viewed at wileyonlinelibrary.com]

Both the reset strategies are now analyzed in detail to outline which information is passed from the previous to the actual graph and how these measurements are compressed in the new framework. Considering the keyframe reset procedure and referring to Figures 3 and 4, the following actions are performed to obtain the graph $g + 1$ from the graph g.

- The $i + 1$ keyframe pose nodes are transferred from the previous to the actual factor graph. The first keyframe node, as the one associated with the state $x_k^{(g)}$, is constrained with a prior factor with the last estimated value. All the subsequent $i - 1$ keyframe nodes are determined by an *XYZ–RPY* factor obtained from each last estimated value and the associated covariance.
- All the $m + 1$ visual landmark points are transferred from the previous to the actual factor graph. They are employed to maintain constraints between all the keyframe pose nodes. Each landmark node is reported in the current graph with its last estimate and covariance and all the vision-based edges.
- The last pose node associated with the state $x_N^{(g)}$, even if it is not a keyframe node, is transferred to the actual graph to be employed as starting point to insert the acquired measurements as constraints. This node is constrained to the last keyframe node with an *XYZ–RPY* odometry factor computed from the last pose estimated values of the two nodes. The relative rototranslation transformation is thus computed and applied as a constraint.

All the DVL-based landmarks are reported in the global NED reference frame using the poses estimated with the graph g, and they are employed to build the point cloud for the seabed reconstruction. Even though the whole graph has been reset, the

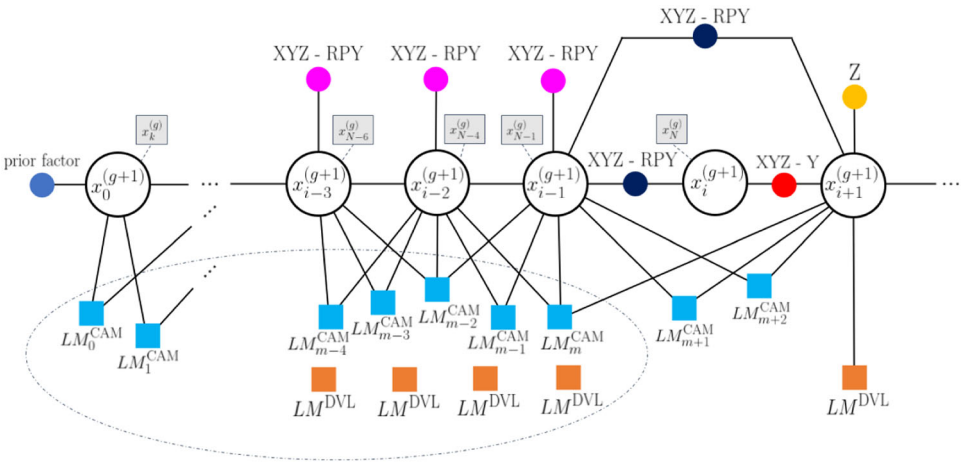

FIGURE 4 First nodes of the factor graph $g + 1$ after the employment of the keyframe reset procedure. The values in the gray boxes represent the corresponding states taken from the previous factor graph g and transferred to the actual graph $g + 1$. CAM, camera; DVL, Doppler Velocity Log. [Color figure can be viewed at wileyonlinelibrary.com]

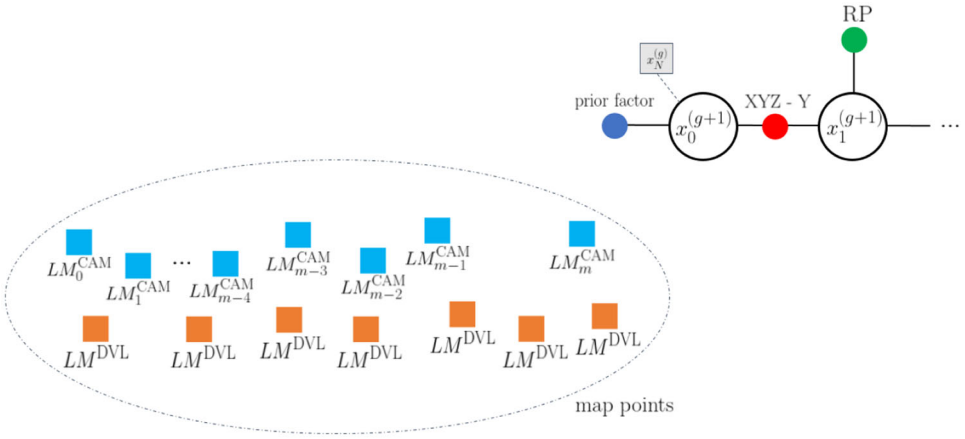

FIGURE 5 First nodes of the factor graph $g + 1$ after the employment of the global reset procedure. The values in the gray boxes represent the corresponding states taken from the previous factor graph g and transferred to the actual graph $g + 1$. CAM, camera; DVL, Doppler Velocity Log. [Color figure can be viewed at wileyonlinelibrary.com]

visual SLAM part, if the visibility is acceptable, continues to compute poses and visual landmarks, which are inserted in the new graph and connected to the keyframe nodes passed from the previous graph. Furthermore, until a new keyframe is not computed, the new nodes are inserted thanks to the DVL-based DR, the DS measurements, and the attitude estimator filter outputs.

Considering the global reset procedure and referring to Figures 3 and 5, the following actions are performed to obtain the graph $g + 1$ from the graph g.

- Only the last pose node associated with the state $x_N^{(g)}$ is transferred to the actual graph to be employed as a starting point to insert the acquired measurements as constraints. It is constrained with a prior factor with the last estimated value.
- The visual landmarks and the keyframe poses are not transferred from the previous to the actual graph. All positions of the estimated DVL-based and visual landmarks are saved as estimated in the last optimization of the previous graph, and they are employed to build the point cloud for the seabed reconstruction.

Even if the visibility is acceptable, the visual SLAM algorithm is reinitialized, the scale factor is again computed, and no information is transferred from the vision-based part of the previous graph. Despite the loss of some helpful information, the global reset procedure is necessary to limit the algorithm's computation burden. When the global reset procedure is applied, the whole graph is condensed in its last estimation. The estimated map is saved, and the estimated positions of the landmarks are no longer optimized. In this context, some information may be lost, but it is necessary to guarantee the algorithm to work on board the robot and work with many nodes. Indeed, the navigation and mapping performance might be affected by the lost information. Still, the gain in terms of computational

burden is higher than the loss in terms of performance (the estimated map is preserved, and the estimated positions are condensed in the last estimation as in a Kalman filter framework).

6 | NAVIGATION RESULTS IN SIMULATED ENVIRONMENT

To validate the developed DVL and camera-based SLAM algorithm, realistic simulations were performed by means of the UUV Simulator. In particular, while navigation performance has been evaluated thanks to a Monte Carlo simulation, mapping capabilities have been analyzed with a lawnmower survey at a constant depth over a simulated seabed generated with a known mathematical function $z = f(x, y)$. The obtained results have been employed to evaluate the goodness of the whole algorithm and some of its main features, such as the reset procedure and the scale factor computation algorithm. To focus attention on the navigation and mapping capabilities of the filter, the DVL and the camera have been modeled thanks to the simulator features. The realistic simulations were based on the dynamic model of FeelHippo AUV implemented in the UUV Simulator and on modeling all the onboard sensors. In particular, the DVL beams have been modeled by applying a noise in the measured value, which determines a noise in the measured velocity. The camera has been modeled with a noise in the pixel position of the acquired image, which influences both the vehicle and landmark position estimation. The data employed for the sensor modeling have been chosen by considering the datasheet of the devices mounted on FeelHippo AUV, the vehicle used in the experimental campaign.

During the Monte Carlo simulations, the position filter was fed with the data coming from the simulated sensors, as the GNSS, when the vehicle was higher than a fixed depth, DS, DVL and camera. To increase adherence to the real data set, the DVL speed measurements have been published with a 5 Hz rate, and the camera acquired images with a frequency of 10 Hz. The proposed strategies have been tested on a vehicle whose dynamic behavior has been simulated using the model implemented in the UUV Simulator, which has traveled a rectangular path at a fixed depth of 2 m. A Monte Carlo

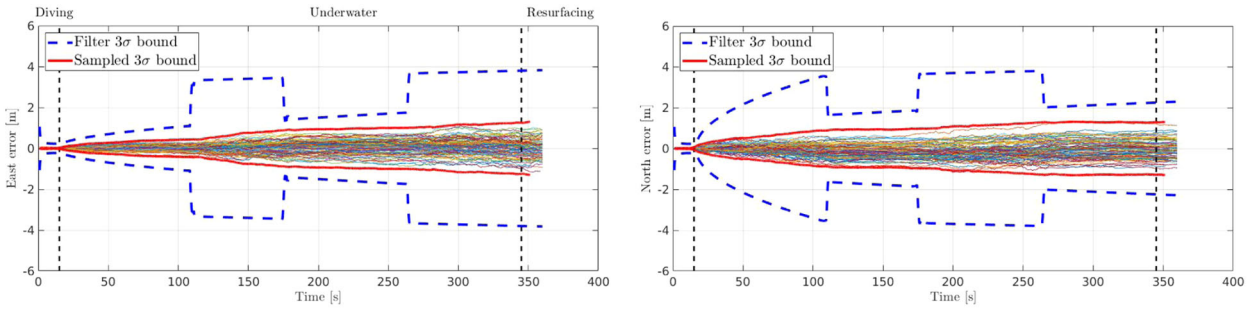

FIGURE 6 East and North position estimation errors versus their 3σ bounds obtained from 100 simulation analysis with the SLAM algorithm. The σ values are computed as the square root of the corresponding diagonal element of the estimated covariance matrix. SLAM, Simultaneous Localization and Mapping. [Color figure can be viewed at wileyonlinelibrary.com]

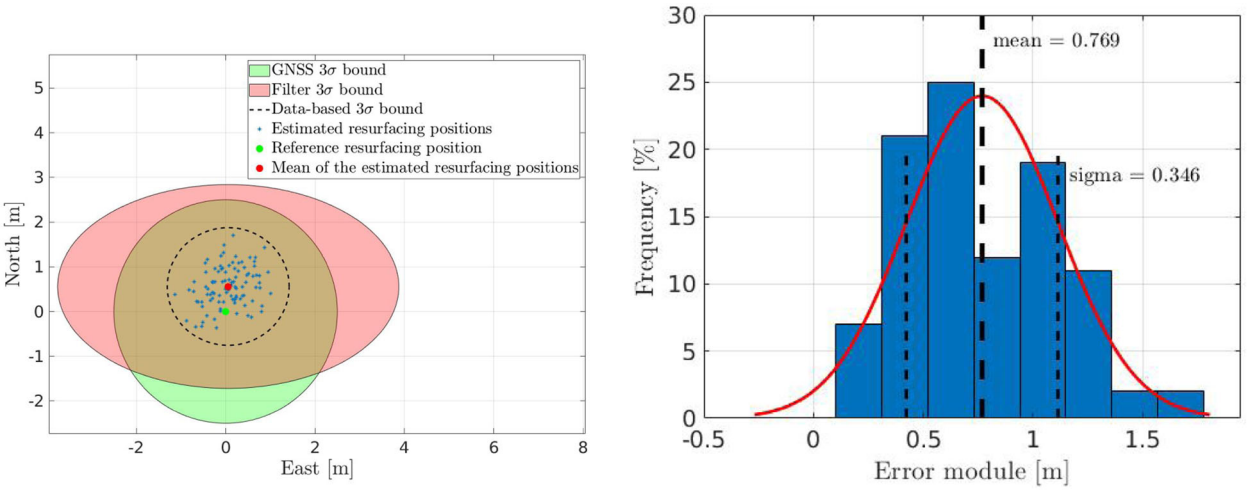

FIGURE 7 On the left, the estimated resurfacing positions versus the theoretical GNSS fix position obtained from 100 simulation analyses for the SLAM algorithm. On the right, histograms containing the estimated resurfacing position errors were obtained from 100 simulation analyses for the SLAM algorithm. GNSS, Global Navigation Satellite System; SLAM, Simultaneous Localization and Mapping. [Color figure can be viewed at wileyonlinelibrary.com]

simulation with 100 iterations has been performed. The position errors and the estimated 3σ bounds along the East and North directions are reported in Figure 6. The covariance trend follows the trajectory described by the vehicle. Still, the SLAM algorithm, due to the presence of visual landmarks that constrain the vehicle position, provides an elliptic 3σ bound with a major axis perpendicular to the direction followed by the vehicle. Despite its particular shape, the 3σ bound continuously diverges when the vehicle is under the sea surface, and no position measurements are available, correctly representing the behavior of the AUV.

Furthermore, as in Section 5, the estimated resurfacing position has been compared with the theoretical first GNSS fix and its 3σ bound. The resurfacing positions estimated in all the Monte Carlo simulations fall inside the 3σ bound, guaranteeing reasonable estimations. Furthermore, it is possible to compare the 3σ bound estimation obtained from the filter and the 3σ bound estimation obtained from the simulated data, evaluating the latter by computing the best normal distribution approximating the estimated resurfacing positions with respect to the theoretical ones (Figure 7).

Analyzing the results obtained from the lawnmower survey at a constant depth of 5 m and comparing the estimated trajectory with the reference path provided by the simulator, it is possible to notice that the divergence over time of the navigation error is reduced (see Figure 8). Indeed, even if a global loop closure on the visual keyframes is not

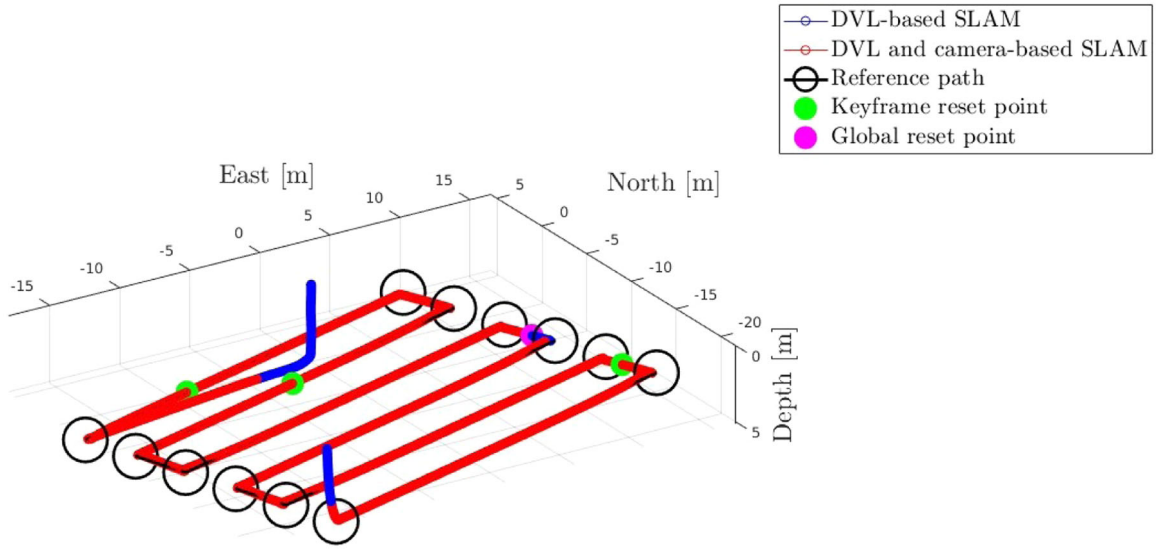

FIGURE 8 Three-dimensional plot of the estimated trajectory in the NED reference system, where the reset points and the areas where the vision part of the algorithm is not used are highlighted. The camera-based part of the SLAM algorithm is not employed due to the lack of features and/or when the scale factor ambiguity has not been solved yet. DVL, Doppler Velocity Log; NED, North, East, and Down; SLAM, Simultaneous Localization and Mapping. [Color figure can be viewed at wileyonlinelibrary.com]

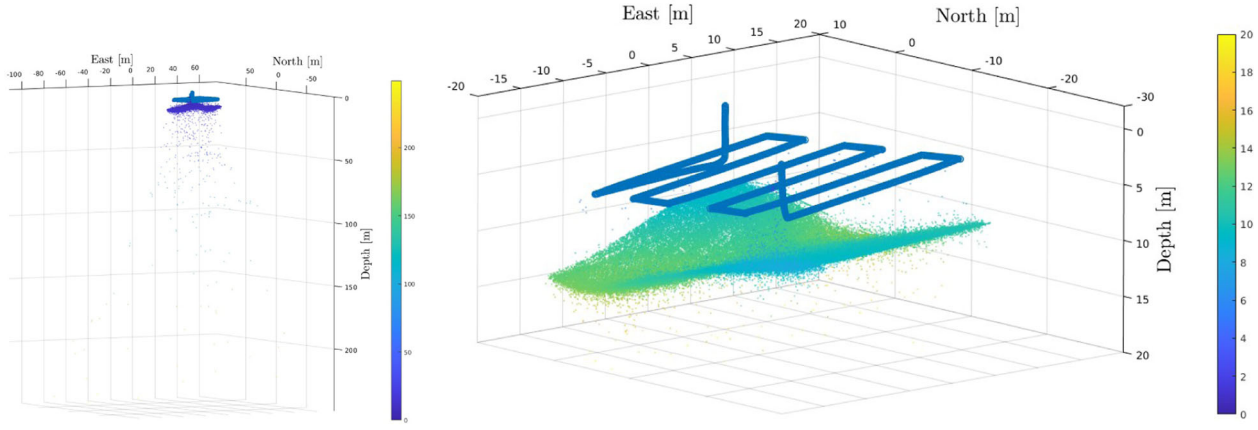

FIGURE 9 Representation of the point cloud and the traveled trajectory estimated through the SLAM algorithm. While on the left image the entire point cloud is reported and, due to the presence of outliers, the depth scale is too extended, on the right image a zoom on the region of interest is performed. The values reported in the color map are in meters. SLAM, Simultaneous Localization and Mapping. [Color figure can be viewed at wileyonlinelibrary.com]

performed, the presence of highly accurate DVL measurements can maintain a low estimation error drift. Furthermore, Figure 8 shows the estimated trajectory on the NED frame, where it is possible to notice the points where the system has been reset. Considering that the simulated seabed has been textured with a feature-rich image, it is necessary to see that the visual part of the SLAM algorithm continues to work for the whole trajectory. Thus, both reset strategies have been employed to limit the computational burden. Figure 9 reports the estimated trajectory and the generated point cloud. It is possible to evaluate the algorithm mapping capabilities by comparing the estimated point cloud and the function employed to simulate the seabed. Considering that several outliers are kept in the point cloud during the SLAM algorithm, which negatively influences the seabed reconstruction, the estimated landmarks are elaborated to eliminate the wrong points and to downsample the cloud. The outliers come from the camera-based feature extraction and matching. They are caused by the complex environment where the AUV is performing its mission. Consequently, the seabed reconstruction capabilities of the developed algorithm are analyzed in Section 8, where the employed post-processing strategies are described.

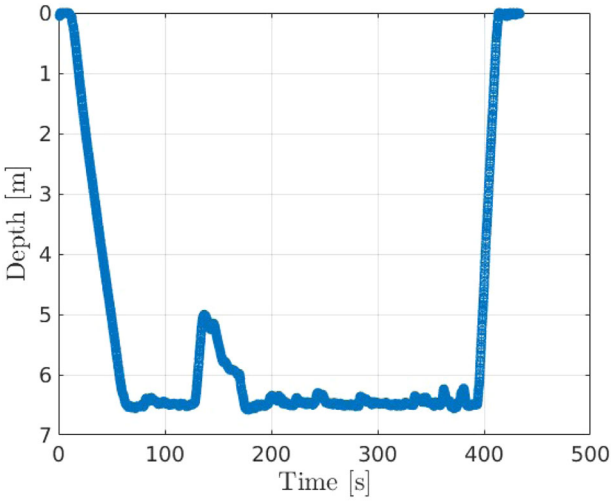

FIGURE 10 FeelHippo AUV depth profile during the mission in Stromboli Island, Messina (Italy). AUV, Autonomous Underwater Vehicle. [Color figure can be viewed at wileyonlinelibrary.com]

7 | EXPERIMENTAL RESULTS

The presented navigation and mapping strategy has been tested and validated by employing experimental data recorded in Stromboli Island, Messina (Italy), in September 2022, during an autonomous underwater mission performed in the framework of the project PATHFinder. During its autonomous navigation, performed at a constant depth (see Figure 10), along a preprogrammed path, the payload sensors were switched on, and the vehicle acquired both acoustic and optical data. GNSS readings obtained from the satellites of the Galileo system were collected before FeelHippo AUV (Allotta et al., 2017) dove and after it resurfaced. They have been employed as ground truth to compute the resurfacing error and to globally

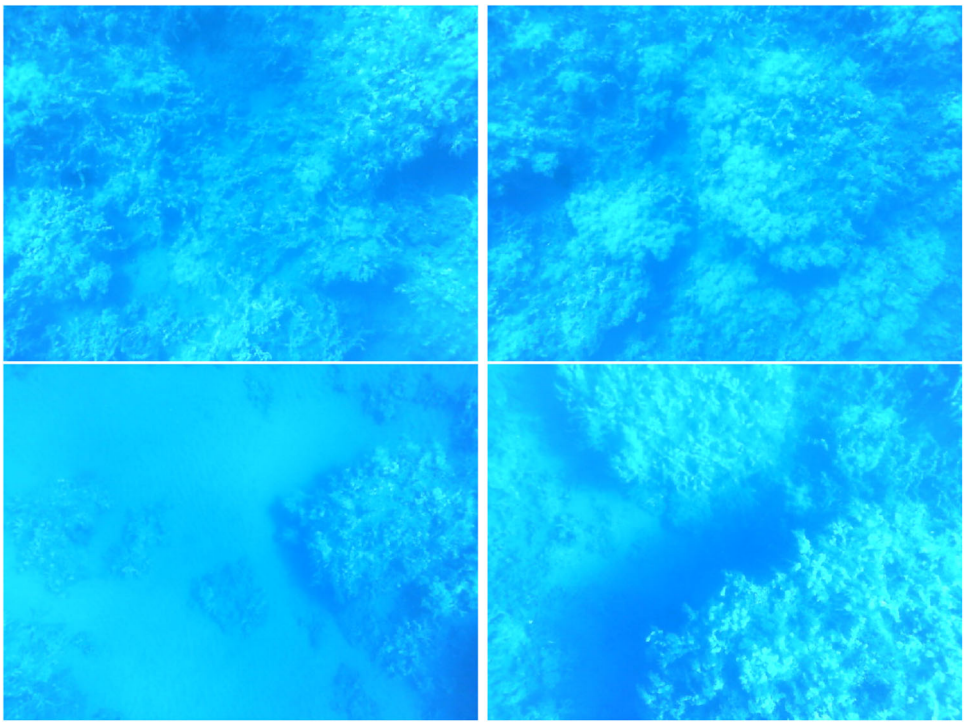

FIGURE 11 Examples of images acquired by FeelHippo AUV during the autonomous mission performed in Stromboli Island, Messina (Italy). AUV, Autonomous Underwater Vehicle. [Color figure can be viewed at wileyonlinelibrary.com]

FIGURE 12 FeelHippo AUV before an on-field underwater mission. AUV, Autonomous Underwater Vehicle. [Color figure can be viewed at wileyonlinelibrary.com]

TABLE 1 FeelHippo AUV main properties.

Weight (kg)	35
Dimensions (mm)	600 × 640 × 500
Maximum depth (m)	30
Maximum longitudinal speed (m/s)	1
Battery life (h)	3

Abbreviation: AUV, Autonomous Underwater Vehicle.

TABLE 2 Navigation performance for the mission accomplished in Stromboli Island, Messina (Italy): resurfacing error.

Navigation strategy	Error (m)
UKF algorithm	1.943
SLAM algorithm	0.899

Note: The results obtained from the SLAM algorithm have been computed before the outlier rejection, which has been employed only for the seabed 3D reconstruction.

Abbreviations: 3D, three-dimensional; SLAM, Simultaneous Localization and Mapping; UKF, Unscented Kalman Filter.

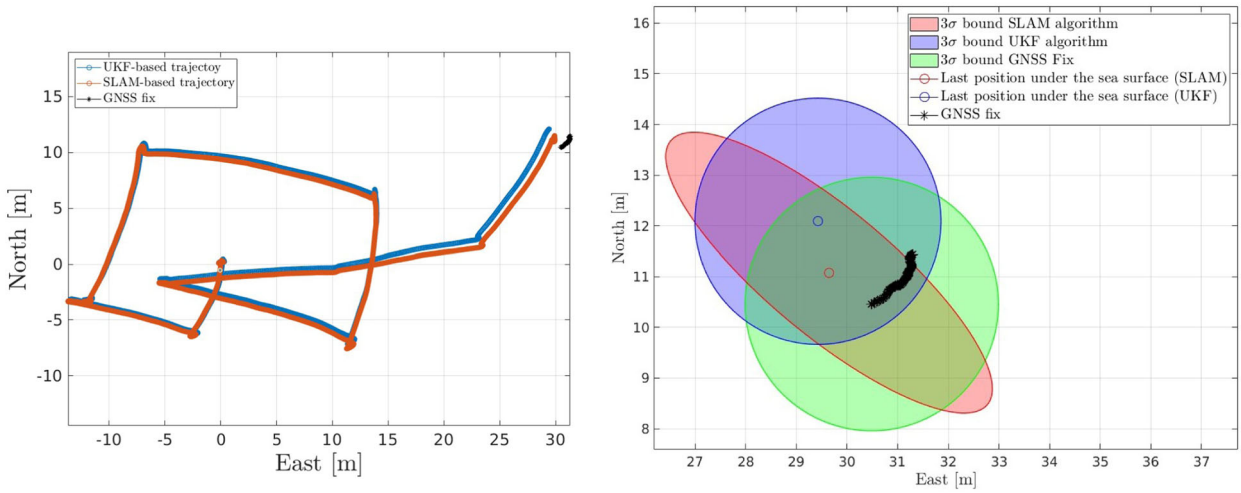

FIGURE 13 On the left, comparison between the trajectories estimated with the SLAM algorithm and the UKF algorithm during the mission accomplished in Stromboli Island, Messina (Italy). A ground truth, when the vehicle was under the sea surface, is not available, but the first GNSS fix when the vehicle resurfaces can be employed as a reference to evaluate the resurfacing error. On the right, the 3σ bound of the last positions under the sea surface is estimated with the SLAM and UKF algorithms and the first GNSS fix measurement with its accuracy 3σ bound. The trajectory estimated through the SLAM algorithm has been computed before the outlier rejection, which has been employed only for the seabed 3D reconstruction. 3D, three-dimensional; GNSS, Global Navigation Satellite System; SLAM, Simultaneous Localization and Mapping; UKF, Unscented Kalman Filter. [Color figure can be viewed at wileyonlinelibrary.com]

reference the trajectory and the map. Samples of images acquired by the camera mounted on FeelHippo AUV are reported in Figure 11.

FeelHippo AUV (see Figure 12) is a compact vehicle capable of performing missions in shallow waters. The main features of FeelHippo AUV are reported in Table 1. Furthermore, the sensors available on board are listed as follows:

- U-blox 7P precision GNSS;
- Orientus Advanced Navigation IMU;
- KVH DSP 1760 single-axis high precision Fiber Optic Gyroscope;
- Nortek DVL1000 DVL, measuring linear velocity and acting as DS;
- Teledyne BlueView M900 2D Forward Looking SONAR (FLS);
- two Microsoft Lifecam Cinema forward- and bottom-looking cameras.

The developed SLAM strategy has been compared with the Standard UKF algorithm chosen from the navigation strategies proposed by Bucci et al. (2023). The position resurfacing error values and covariances have been evaluated on the North–East plane. Figure 13 reports the estimated trajectories and an analysis of the resurfacing errors with their 3σ bound. From Table 2, analyzing the results from the GNSS resurfacing error, it is easily noticeable that both the proposed strategies are acceptable in terms of navigation estimation quality. As can be seen in both simulation and experimental results, employing GTSAM (or any other factor graph-based solutor) to solve navigation problems with odometry factors determines an elliptical shape of the covariance, where the uncertainty is more significant along the direction perpendicular to the motion direction. Despite the initial circular shape of the covariance when the vehicle is on the sea surface, it becomes elliptical due to the available measurements and the employed strategy.

To evaluate the agreement between estimation errors and estimated uncertainty, the 3σ bounds during the resurfacing phase are presented. This is summarized in Figure 13, where the 3σ bounds for the filters and the GNSS are presented. In all the analyzed cases, the position provided by the filter (with its confidence bounds) appears to guarantee a reasonable prediction of the vehicle's true position when it resurfaces. The employed GNSS has an expected accuracy on the order of meters and the 2D error can be represented as a 2D Gaussian distribution whose components are independently distributed.

Focusing attention on the SLAM algorithm and its mapping capabilities, Figure 14 reports the SLAM-based estimated trajectory and the generated point cloud. It is possible to evaluate the algorithm mapping capabilities by comparing the estimated point cloud with a bathymetry of the region around the island. As for the test in simulated environments, several outliers are kept in the point cloud during the SLAM algorithm operation, which negatively influences the seabed reconstruction. Consequently, the seabed reconstruction capabilities of the developed algorithm and the comparison with the ground truth bathymetry are reported in Section 8, where the employed post-processing strategies are described.

The scale factor computation procedure has been applied to estimate the scale factor between the DVL-based trajectory and the visual part of the algorithm before fusing them in the whole factor graph. In particular, approximate values of the relative position and orientation between the DVL and the camera have been provided as input to the algorithm, but their values have been kept as variables in the optimization process. The scale factor between the DVL-based trajectory and the visual SLAM has been solved with the developed algorithm, and the results have been reported in Table 3. It is necessary to highlight that the proposed strategy can compensate for

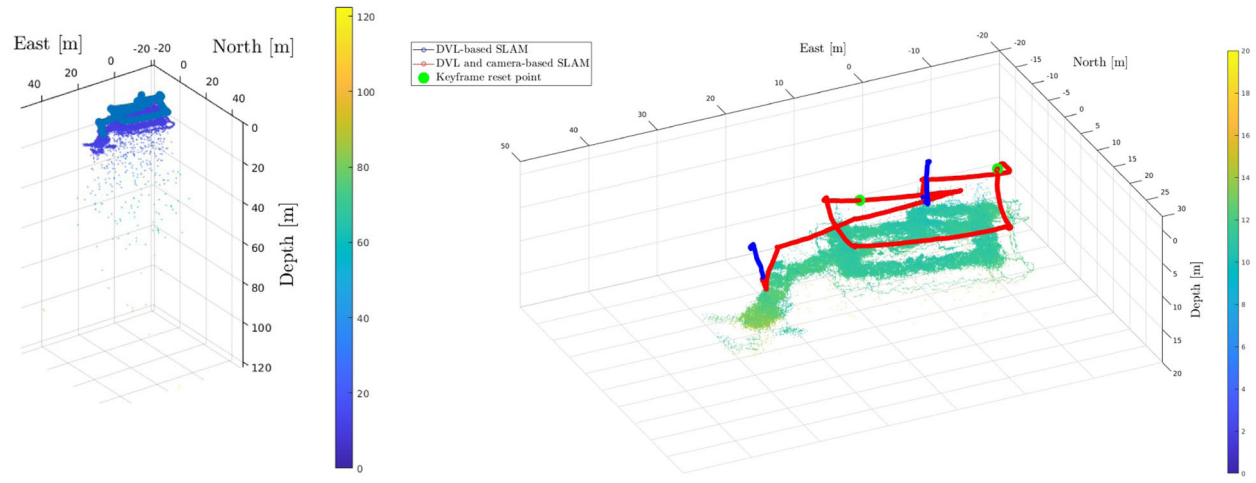

FIGURE 14 Representation of the point cloud and the traveled trajectory estimated through the SLAM algorithm during the mission in Stromboli Island, Messina (Italy). While on the left image the entire point cloud is reported and, due to the presence of outliers, the depth scale is too extended, on the right image a zoom on the region of interest is performed. Furthermore, on the right image the reset points and the areas where the vision part of the algorithm is not used are highlighted. The values reported in the color map are in meters. DVL, Doppler Velocity Log; SLAM, Simultaneous Localization and Mapping. [Color figure can be viewed at wileyonlinelibrary.com]

TABLE 3 Estimated scale factor and rototranslation transform between Doppler Velocity Log and camera reference systems.

Parameter	Initial guess	Value after step 1	Value after step 2
s	0.0	5.448	5.529
ϕ_c^b (deg)	0.0	0.0	−0.005
θ_c^b (deg)	90.0	90.0	89.477
ψ_c^b (deg)	0.0	10.119	8.43
$t_{c,b}^x$ (m)	0.24	0.24	0.233
$t_{c,b}^y$ (m)	0.07	0.06	0.076
$t_{c,b}^z$ (m)	0.05	0.05	0.049

the alignment error between the camera and the DVL frames. Indeed, due to uncontrollable external conditions (e.g., loosening of the screws during the vehicle preparation, collisions during the diving procedure), the camera rotated around its z-axis during the autonomous mission of an unknown quantity which has been estimated and compensated by the algorithm. The resurfacing error value is equal to 0.899 m, indicating a high navigation accuracy of the proposed strategy with respect to the GNSS fixes obtained when the vehicle resurfaced.

Finally, regarding the computational burden, the execution time of the filter has been subject of the analysis. The sum of the requested time to perform the measurement insertion in the factor graph and the optimization process has been considered. For what

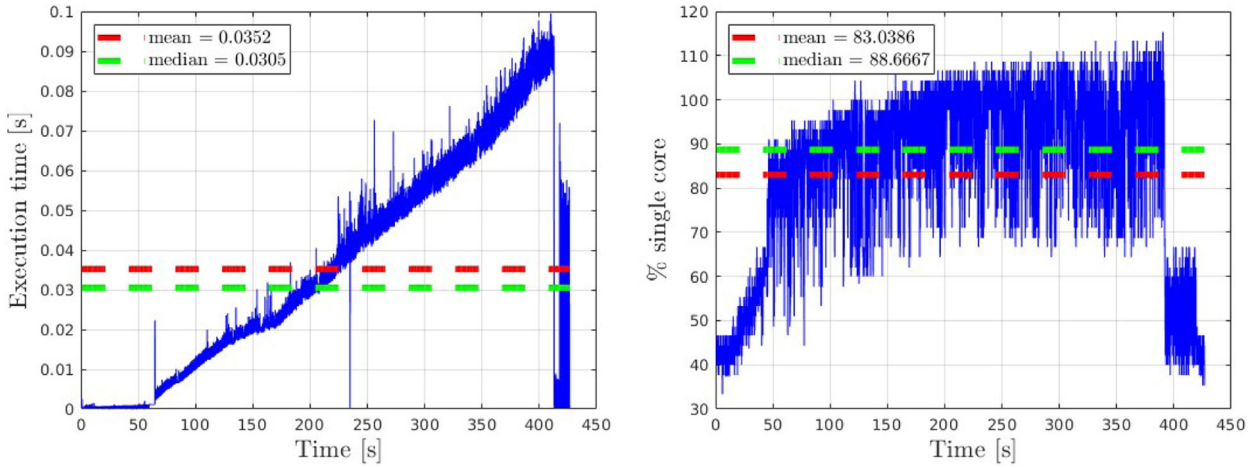

FIGURE 15 On the left, execution time of the SLAM filter, calculated at each iteration as the sum of the requested time for measurement insertion in the factor graph and for the optimization process. On the right, Central Processing Unit burden analysis. In red and green are, respectively, reported the mean and the median. SLAM, Simultaneous Localization and Mapping. [Color figure can be viewed at wileyonlinelibrary.com]

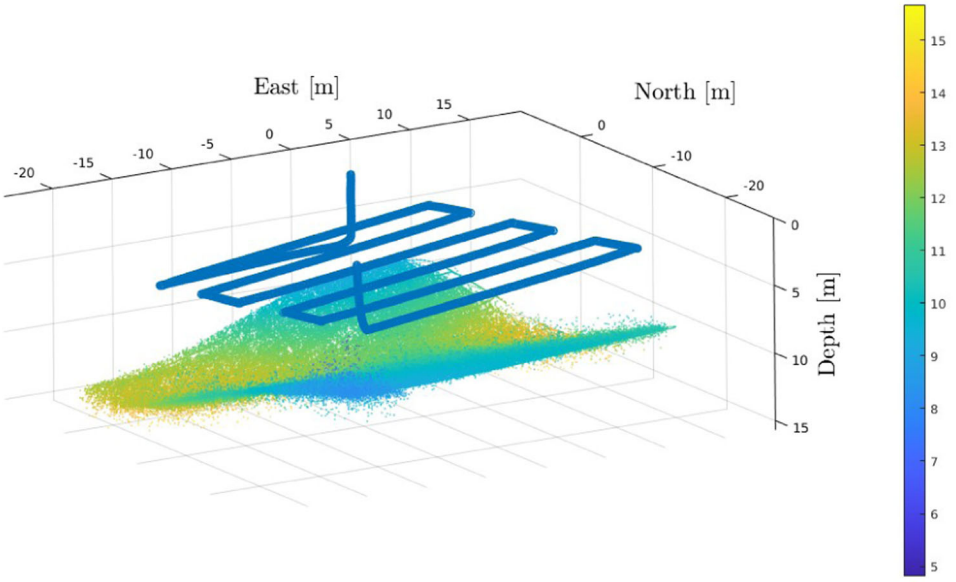

FIGURE 16 Filtered point cloud and estimated trajectory traveled by the simulated vehicle. The values reported in the color map are in meters. [Color figure can be viewed at wileyonlinelibrary.com]

concern the Central Processing Unit analysis, the output of the command *top* has been recorded to store the data. The results can be found in Figure 15. It is necessary to notice that the instants where the visual part of the algorithm is initialized and stopped can be easily highlighted thanks to its influence on the execution time of each iteration. Indeed, despite the SLAM algorithm optimizes only the last nodes thanks to the properties of the iSAM2 library, handling a continuously growing point cloud increases the required computational cost. When the vehicle resurfaces and the visual part of the algorithm is excluded due to visibility limitations, the necessary computational burden drastically decreases. Indeed, the point cloud is saved, and only the position nodes are updated when new measurements are acquired.

8 | MAPPING PERFORMANCE ANALYSIS

Mapping the surrounding environment is a common task in underwater exploration, and it is fundamental to enhance the vehicle capabilities to find objects of potential interest. The point clouds obtained from the SLAM algorithm have been processed with an automatic tool to obtain a 3D reconstruction of the sea bottom. The

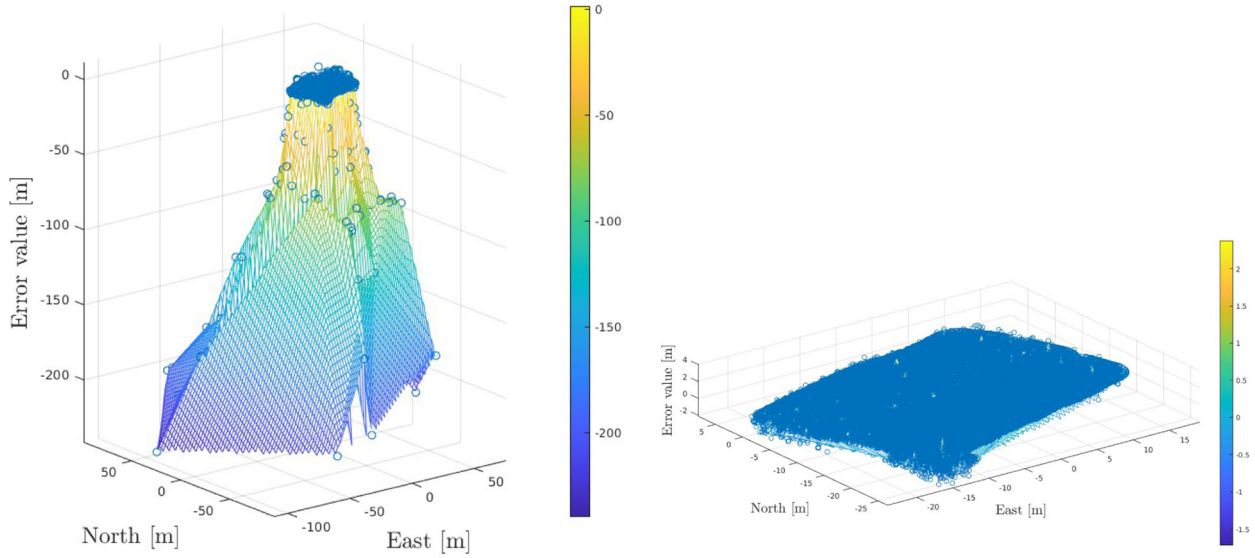

FIGURE 17 Representation of the error point clouds computed by comparing the reference sea bed function and the estimated point cloud and generation of the estimated error maps before (left image) and after (right image) the filtering procedure. The values reported in the color map are in meters. [Color figure can be viewed at wileyonlinelibrary.com]

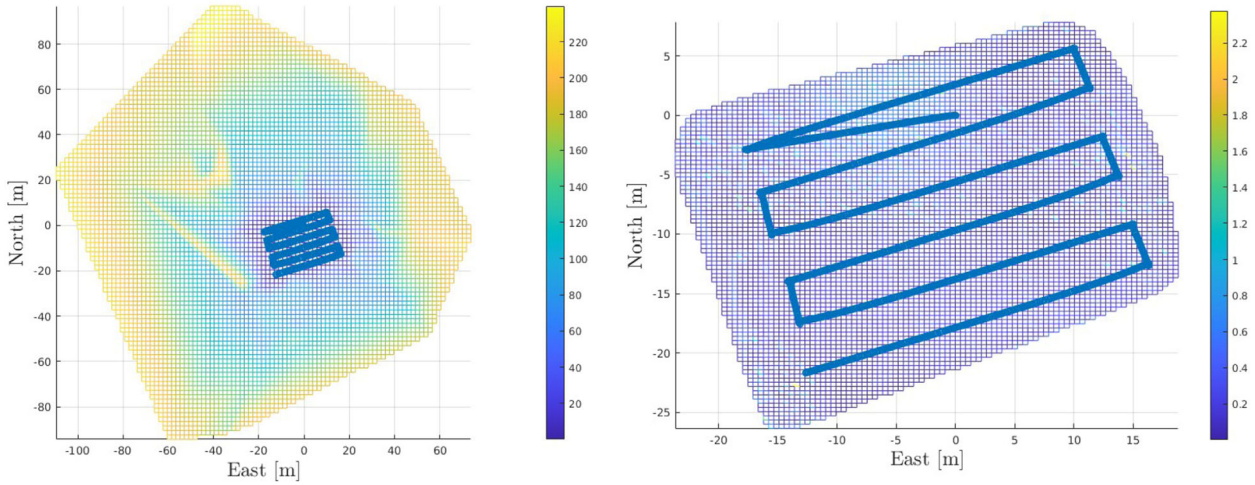

FIGURE 18 Comparison between the estimated error maps before (left image) and after (right image) the filtering procedure with respect to the traveled trajectory by the simulated vehicle. The values reported in the color map are in meters. The reported maps are, respectively, the projection on the North–East plane of the data shown in Figure 17. [Color figure can be viewed at wileyonlinelibrary.com]

developed reconstruction strategy takes as input the estimated point cloud and the geographical coordinate of a reference point and automatically generates a 3D reconstruction and a georeferenced depth map, thanks to the employment of the functions implemented in the open-source libraries Point Cloud Library (Rusu & Cousins, 2011) and Open3d (Zhou et al., 2018).

Analyzing the point cloud obtained from the navigation algorithm applied in both simulated and real environments, it is necessary to notice that some points can be classified as outliers. Therefore, the need arises to eliminate them because they are excessively bulky and negatively influence the mesh realization. For each point, a fixed number of neighbors is defined to estimate the mean of the average distance, and a point is considered an outlier if the average distance to its neighbors is above a specified threshold (Rusu et al., 2008). The outlier-eliminating process, therefore, leads to a significant decrease in points, making the representations more uniform. Subsequently, the point cloud is processed with a smoothing method to filter out the noise of the measurements on the processed points. In particular, median filtering of the 3D point cloud data is performed.

The 3D mesh generation algorithm is applied to the filtered point cloud. The Poisson algorithm (Kazhdan et al., 2006) is applied, and its parameters have been chosen to trade off between the required computational time and the resolution of the obtained reconstruction. It is necessary to note that the depth value and the limit density of the points at which the reconstruction is cut have been chosen to make a compromise between reconstruction speed and estimation quality.

TABLE 4 Mean with the associated covariance and median values of the depth error in the presence and in the absence of the filtering procedure.

Parameter	Before filtering	After filtering
Mean (m)	0.2767	0.2002
Covariance (m)	7.6212	0.0386
Median (m)	0.1469	0.1465

First, considering that in the simulated environment created with UUV Simulator the seabed can be shaped with a mathematical function $z = f(x, y)$, it is possible to evaluate the performance of the filtering algorithm. The simulated seabed has been textured with an image rich in features to facilitate the correct behavior of the visual part of the SLAM algorithm. Figure 16 reports the 3D filtered point cloud with the estimated trajectory. This point cloud can be compared with the one directly obtained from the SLAM algorithm and reported in Figure 9. Two error maps have been created with the point clouds, as before and after the filtering procedure, to analyze the improvements in seabed reconstruction. The outlier points are correctly removed, and the point cloud size is reduced to increase its easiness of management by the reconstruction algorithm (see Figures 17 and 18). As can be retrieved from Table 4, the mean error is reduced by approximately 35% thanks to the outlier removal process, which also strongly decreases the associated covariance. Finally, the 3D point cloud has been processed with the reconstruction algorithm to build a smoothed mesh, which correctly follows the shape of the simulated sea bottom (see Figure 19).

Turning to the processing of the results obtained during the mission performed in Stromboli Island, Messina (Italy), a 3D reconstruction and a geolocalized map of the sea bottom are reported. The reconstruction comprises around 240*k* points obtained as output from the SLAM algorithm. First, the outlier points have been removed (see Figure 20), and then, the 3D point cloud has been processed with the Poisson reconstruction algorithm to build the 3D mesh, which is shown in Figure 21. The good matching between the reference bathymetry, whose data have been provided to the authors by the National Research Council and Presidenza del Consiglio dei Ministri-Dipartimento della Protezione Civile through a specific agreement (see Figure 22), and the

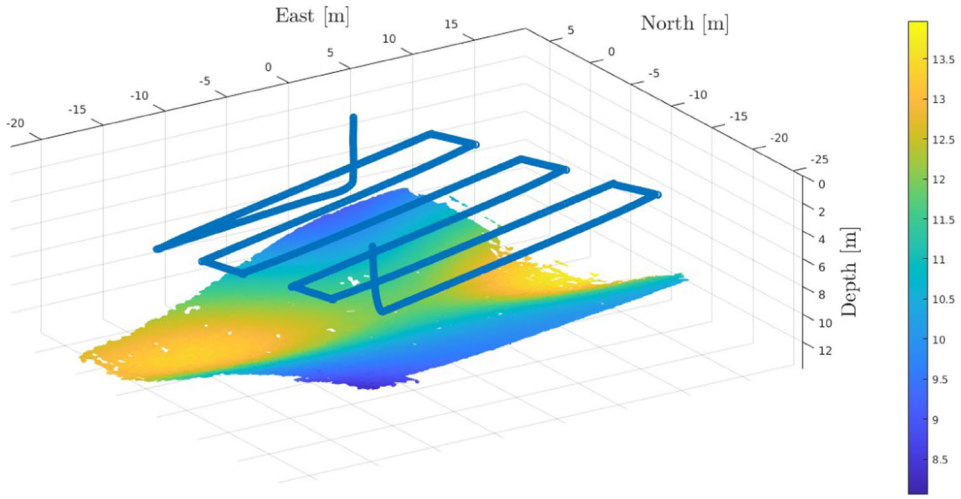

FIGURE 19 Resulting sea bottom three-dimensional mesh reconstruction and estimated trajectory traveled by the simulated vehicle. The values reported in the color map are in meters. [Color figure can be viewed at wileyonlinelibrary.com]

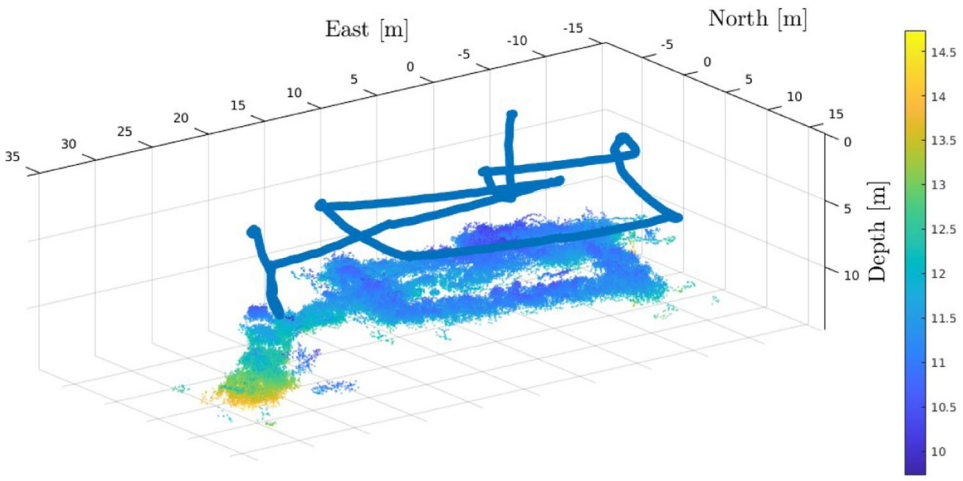

FIGURE 20 Filtered point cloud and estimated trajectory traveled by the vehicle during the autonomous mission accomplished in Stromboli Island, Messina (Italy). The values reported in the color map are in meters. [Color figure can be viewed at wileyonlinelibrary.com]

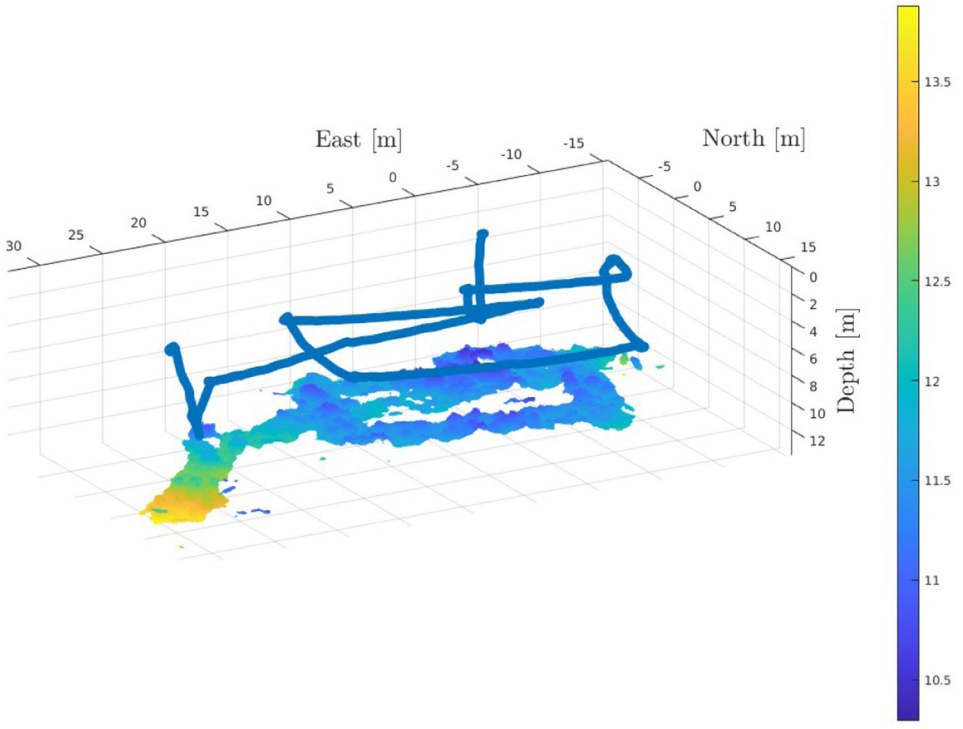

FIGURE 21 Resulting sea bottom three-dimensional mesh reconstruction and estimated trajectory traveled by the vehicle during the autonomous mission accomplished in Stromboli Island, Messina (Italy). The values reported in the color map are in meters. [Color figure can be viewed at wileyonlinelibrary.com]

estimated 3D reconstruction can also be observed to prove the reconstruction's goodness. The provided bathymetry has a horizontal resolution of 5 m. Thus only an approximate comparison can be performed, but it can be sufficient to have a simple evaluation of the generated point cloud. All the points of the cloud that lies in each square generated from the ground truth bathymetry are employed to compute the mean point and perform the comparison (Figure 23 and Table 5). To evaluate the goodness of the 3D reconstruction is possible to distinguish between the squares where the trajectory is superimposed and the ones where the vehicle does not go. As seen from both Figure 23 and Table 5, the reconstruction error, evaluated as the mean error, is strongly different in those regions. Indeed, it is about 0.210 m in the squares where the vehicle goes and about 0.3885 m in the other areas.

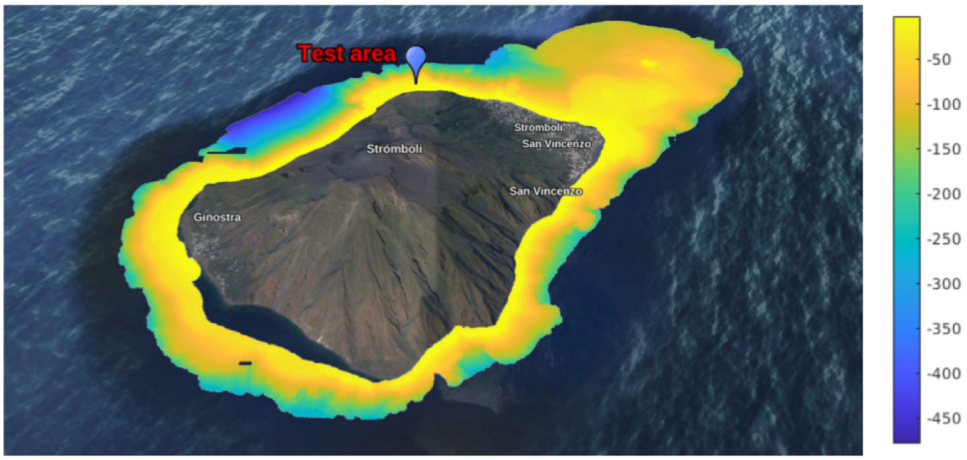

FIGURE 22 Reference bathymetry of the sea bottom around Stromboli Island, Messina (Italy). The test area, where FeelHippo AUV performed its autonomous mission, is highlighted. The values reported in the color map are in meters. [Color figure can be viewed at wileyonlinelibrary.com]

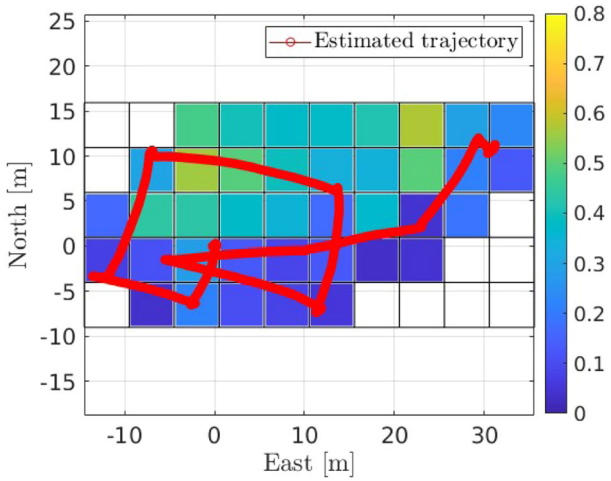

FIGURE 23 Estimated error bathymetry map with respect to the trajectory traveled by the vehicle during the autonomous mission accomplished in Stromboli Island, Messina (Italy). The values reported in the color map are in meters. [Color figure can be viewed at wileyonlinelibrary.com]

9 | CONCLUSION AND FUTURE DEVELOPMENTS

Considering that Kalman filtering condenses the vehicle's history in the last estimate and covariance, a MAP strategy based on factor graphs has been developed to overcome these limitations and include visual landmarks in the estimation process. Visual features are sometimes difficult to be found in underwater environments due to visibility and texture issues. Consequently, the strategy fuses DVL measurements with a visual SLAM system to simultaneously perform accurate navigation and mapping tasks. DVL beam data can be employed for speed measurement and to obtain an approximated knowledge of the sea bottom. Both simulated and experimental data have been employed to evaluate the capabilities of the developed strategy. The experimental data have been acquired during trials at Stromboli Island, Messina (Italy).

During the experimental campaign, FeelHippo AUV was the only vehicle involved; nevertheless, since the proposed solution is not tailored to a particular vehicle, its outcomes can be deemed as general, and future developments will include the testing of the navigation strategy on other vehicles. No global reset has been employed due to the length of the mission in the experimental campaign. Necessarily, future works will include more extended missions to evaluate the performance of the global reset procedure not only in the simulation environment and on real data. Furthermore, progress on the developed algorithms still needs to be made. Integrating the developed estimation strategy within the attitude estimation filter could represent an important subject to be investigated. Indeed, developing a unique filter that works on both the attitude and position estimation in a coupled way could increase navigation precision. Concerning the strategies for graph edge computation, including constraints obtained from acoustic FLS images, which have been employed for speed estimation in Bucci et al. (2023), the pose-graph framework would push forward the performance of the navigation filter. Despite the intrinsic characteristic (low resolution, influence of the viewpoint) of FLS images poses relevant issues to face, the employment of an additional acoustic sensor can be useful to apply the developed strategy in scenarios with reduced visibility. Introducing a new sensor will necessarily increase the number of landmarks in the factor graph, but, on the contrary, it will be helpful to face out against possible changes in the external conditions. Employing heterogeneous sensors will lead to a strategy that can be adapted to several external conditions and provide an accurate robot pose estimation in almost every scenario. Finding the correct trade-off between the computational burden and the employment of all the available sensor information will be an exciting point to be investigated. Finally, from a mapping-based point

NaN	NaN	0.476	0.403	0.375	0.380	0.416	0.575	0.292	0.233
NaN	0.303	0.560	0.493	0.396	0.336	0.341	0.494	0.213	0.133
0.164	0.440	0.432	0.377	0.348	0.171	0.367	0.044	0.188	NaN
0.076	0.114	0.285	0.128	0.087	0.130	0.036	0.045	NaN	NaN
NaN	0.040	0.217	0.104	0.078	0.055	NaN	NaN	NaN	NaN

Note: The values are in meters and, while the green background is related to the areas where the trajectory is superimposed to the map, the blue background corresponds to the regions where the vehicle does not go.

TABLE 5 Values of the estimated error bathymetry map on the North–East plane (each value is related to the corresponding data in the map in Figure 23 thanks to its position in the table).

of view, a multivehicle solution for autonomously fusing the underwater environment reconstructions could represent a coherent continuation of the research activity carried out so far. The proposed SLAM strategy could operate onboard of each vehicle and, by employing relative or absolute position measurements, the estimated maps could be fused in a unique more detailed reconstruction.

ACKNOWLEDGMENTS

The activities leading to the presented results have been carried out in the context of the PATHfinder European project, part of the ESA's NAVISP Program, and with the support of the National Research Council and Presidenza del Consiglio dei Ministri-Dipartimento della Protezione Civile.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Alessandro Bucci http://orcid.org/0000-0001-6787-3613

REFERENCES

Allotta, B., Conti, R., Costanzi, R., Fanelli, F., Gelli, J., Meli, E. et al. (2017) A low cost autonomous underwater vehicle for patrolling and monitoring. *Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment*, 231(3), 740–749.

Allotta, B., Costanzi, R., Fanelli, F., Monni, N. & Ridolfi, A. (2015) Single axis FOG aided attitude estimation algorithm for mobile robots. *Journal of Mechatronics*, 30, 158–173.

Allotta, B., Costanzi, R., Fanelli, F., Monni, N. & Ridolfi, A. (2016) Underwater Vehicles attitude estimation in presence of magnetic disturbances. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm, Sweden, pp. 2612–2617.

Björck, A. (1996) *Numerical methods for least squares problems*. Philadelphia, US: Society for Industrial and Applied Mathematics.

Bucci, A., Franchi, M., Ridolfi, A., Secciani, N. & Allotta, B. (2023) Evaluation of UKF-based fusion strategies for autonomous underwater vehicles multisensor navigation. *IEEE Journal of Oceanic Engineering*, 48(1), 1–26.

Bucci, A., Ridolfi, A., Franchi, M. & Allotta, B. (2021) Covariance and gain-based federated unscented Kalman filter for acoustic-visual-inertial underwater navigation. In: *OCEANS 2021: San Diego-Porto*, San Diego, CA, USA, pp. 1–7.

Bucci, A., Zacchini, L., Franchi, M., Ridolfi, A. & Allotta, B. (2022) Comparison of feature detection and outlier removal strategies in a mono visual odometry algorithm for underwater navigation. *Applied Ocean Research*, 118, 102961.

Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J. et al. (2016) Past, present, and future of simultaneous localization and mapping: toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6), 1309–1332.

Campos, C., Elvira, R., Rodríguez, J.J.G., Montiel, J.M.M. & Tardós, J.D. (2021) ORB-SLAM3: an accurate open-source library for visual, visual-inertial, and multimap SLAM. *IEEE Transactions on Robotics*, 37(6), 1874–1890.

Cashmore, M., Fox, M., Larkworthy, T., Long, D. & Magazzeni, D. (2014) AUV mission control via temporal planning. In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, China, pp. 6535–6541.

Castellanos, J.A., Neira, J. & Tardós, J.D. (2004) Limits to the consistency of EKF-based SLAM. *IFAC Proceedings Volumes*, 37(8), 716–721.

Dellaert, F. (2012) *Factor graphs and GTSAM: a hands-on introduction*. Georgia Institute of Technology. Technical Report.

Dellaert, F. (2021) Factor graphs: exploiting structure in robotics. *Annual Review of Control, Robotics, and Autonomous Systems*, 4(1), 141–166.

Dellaert, F. & Kaess, M. (2006) Square root SAM: simultaneous localization and mapping via square root information smoothing. *The International Journal of Robotics Research*, 25(12), 1181–1203.

Dellaert, F. & Kaess, M. (2017) Factor graphs for robot perception. *Foundations and Trends in Robotics*, 6(1–2), 1–139.

Dissanayake, M., Newman, P., Clark, S., Durrant-Whyte, H. & Csorba, M. (2001) A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Robotics and Automation*, 17(3), 229–241.

Du, P., Han, J., Wang, J., Wang, G., Jing, D., Wang, X. et al. (2017) View-based underwater SLAM using a stereo camera. In: *OCEANS 2017—Aberdeen*, Aberdeen, UK, 2017, pp. 1–6.

Fallon, M.F., Folkesson, J., McClelland, H. & Leonard, J.J. (2013) Relocating underwater features autonomously using sonar-based SLAM. *IEEE Journal of Oceanic Engineering*, 38(3), 500–513.

Ferri, G., Ferreira, F. & Djapic, V. (2017) Multi-domain robotics competitions: the CMRE experience from SAUC-E to the European robotics league emergency robots. In: *OCEANS 2017—Aberdeen*, Aberdeen, UK, pp. 1–7.

Forster, C., Carlone, L., Dellaert, F. & Scaramuzza, D. (2016) On-manifold preintegration for real-time visual-inertial odometry. *IEEE Transactions on Robotics*, 33(1), 1–21.

Franchi, M., Bucci, A., Zacchini, L., Ridolfi, A., Bresciani, M., Peralta, G. et al. (2021) Maximum a posteriori estimation for AUV localization with USBL measurements. *IFAC-PapersOnLine*, 54(16), 307–313.

Grisetti, G., Guadagnino, T., Aloise, I., Colosi, M., Della Corte, B. & Schlegel, D. (2020) Least squares optimization: from theory to practice. *Robotics*, 9(3), 51.

Hong, S. & Kim, J. (2020) Three-dimensional visual mapping of underwater ship hull surface using piecewise-planar SLAM. *International Journal of Control, Automation and Systems*, 18, 564–574.

Huang, S. & Dissanayake, G. (2007) Convergence and consistency analysis for extended Kalman filter based SLAM. *IEEE Transactions on Robotics*, 23(5), 1036–1049.

Huang, T.A. & Kaess, M. (2015) Towards acoustic structure from motion for imaging sonar. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Hamburg, Germany. IEEE, pp. 758–765.

Julier, S. & Uhlmann, J. (2001) A counter example to the theory of simultaneous localization and map building. In: *Proceedings of the 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*, Seoul, Korea (South), vol. 4, pp. 4238–4243.

Kaess, M., Johannsson, H., Roberts, R., Ila, V., Leonard, J.J. & Dellaert, F. (2012) iSAM2: incremental smoothing and mapping using the Bayes tree. *The International Journal of Robotics Research*, 31(2), 216–235.

Kaess, M., Ranganathan, A. & Dellaert, F. (2008) iSAM: incremental smoothing and mapping. *IEEE Transactions on Robotics*, 24(6), 1365–1378.

Kazhdan, M., Bolitho, M. & Hoppe, H. (2006) Poisson surface reconstruction. In: *Proceedings of the Fourth Eurographics Symposium on Geometry Processing (SGP '06)*. Eurographics Association, Goslar, Germany, pp. 61–70.

Kim, A. & Eustice, R.M. (2013) Real-time visual SLAM for autonomous underwater hull inspection using visual saliency. *IEEE Transactions on Robotics*, 29(3), 719–733.

Mur-Artal, R., Montiel, J.M.M. & Tardós, J.D. (2015) ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31(5), 1147–1163.

Mur-Artal, R. & Tardós, J.D. (2017) ORB-SLAM2: an open-source SLAM system for monocular, stereo, and RGB-D cameras. *IEEE Transactions on Robotics*, 33(5), 1255–1262.

Ozog, P. & Eustice, R.M. (2013) Real-time SLAM with piecewise-planar surface models and sparse 3D point clouds. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Tokyo, Japan, pp. 1042–1049.

Paull, L., Saeedi, S., Seto, M. & Li, H. (2012) Sensor driven online coverage planning for autonomous underwater vehicles. In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vilamoura-Algarve, Portugal, pp. 2875–2880.

Prats, M., Ribas, D., Palomeras, N., García, J.C., Nannen, V., Wirth, S. et al. (2012) Reconfigurable AUV for intervention missions: a case study on underwater object recovery. *Intelligent Service Robotics*, 5, 19–31.

Rahman, S., Li, A.Q. & Rekleitis, I. (2018a) Sonar visual inertial SLAM of underwater structures. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, QLD, Australia, pp. 5190–5196.

Rahman, S., Li, A.Q. & Rekleitis, I.M. (2018b) SVIn2: sonar visual-inertial SLAM with loop closure for underwater navigation. ArXiv, abs/1810.03200.

Rusu, R.B. & Cousins, S. (2011) 3D is here: Point Cloud Library (PCL). In: *2011 IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, pp. 1–4.

Rusu, R.B., Marton, Z.C., Blodow, N., Dolha, M. & Beetz, M. (2008) Towards 3D point cloud based object maps for household environments. *Robotics and Autonomous Systems*, 56(11), 927–941.

Umeyama, S. (1991) Least-squares estimation of transformation parameters between two point patterns. *IEEE Computer Architecture Letters*, 13(04), 376–380.

Vidal, E., Palomeras, N., Istenič, K., Gracias, N. & Carreras, M. (2020) Multisensor online 3D view planning for autonomous underwater exploration. *Journal of Field Robotics*, 37(6), 1123–1147.

Westman, E. & Kaess, M. (2018) *Underwater AprilTag SLAM and calibration for high precision robot localization*. Pittsburgh, PA: Carnegie Mellon University. Technical Report Number: CMU-RI-TR-18-43.

Westman, E. & Kaess, M. (2019) Degeneracy-aware imaging sonar simultaneous localization and mapping. *IEEE Journal of Oceanic Engineering*, 45(4), 1280–1294.

Westman, E. & Kaess, M. (2020) Degeneracy-aware imaging sonar simultaneous localization and mapping. *IEEE Journal of Oceanic Engineering*, 45(4), 1280–1294.

Youakim, D., Cieslak, P., Dornbush, A., Palomer, A., Ridao, P. & Likhachev, M. (2020) Multirepresentation, multiheuristic A* search-based motion planning for a free-floating underwater vehicle-manipulator system in unknown environment. *Journal of Field Robotics*, 37(6), 925–950.

Zacchini, L., Bucci, A., Franchi, M., Costanzi, R. & Ridolfi, A. (2019) Mono visual odometry for autonomous underwater vehicles navigation. In: *OCEANS 2019 - Marseille*, Marseille, France, pp. 1–5.

Zhang, Y., Zhang, T. & Huang, S. (2018) Comparison of EKF based SLAM and optimization based SLAM algorithms. In: *2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, Wuhan, China, pp. 1308–1313.

Zhou, Q.Y., Park, J. & Koltun, V. (2018) Open3D: a modern library for 3D data processing. arXiv:1801.09847.

How to cite this article: Bucci, A., Ridolfi, A. & Allotta, B. (2024) Pose-graph underwater simultaneous localization and mapping for autonomous monitoring and 3D reconstruction by means of optical and acoustic sensors. *Journal of Field Robotics*, 1–21. https://doi.org/10.1002/rob.22375