# Pose-graph underwater Simultaneous Localization And Mapping for autonomous monitoring by means of optical and acoustic sensors

Alessandro Bucci[1], Alessandro Ridolfi[1], and Benedetto Allotta[1]

[1]Universita degli Studi di Firenze Dipartimento di Ingegneria Industriale

August 26, 2023

**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 (GNSSs) 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, thanks to the onboard sensors, 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 absence of visual features. After validating the solution through realistic simulations, an experimental campaign at sea was conducted in Stromboli Island (Messina), Italy. In conclusion, an algorithm, which works with the Poisson surface reconstruction method to obtain a smooth seabed surface, for mesh creation has been developed.

# Pose-graph underwater Simultaneous Localization And Mapping for autonomous monitoring by means of optical and acoustic sensors

**Alessandro Bucci***
Department of Industrial Engineering
University of Florence
via di Santa Marta 3
50139, Florence, Italy
Interuniversity Center of
Integrated Systems for the
Marine Environment (ISME)

**Alessandro Ridolfi**
Department of Industrial Engineering
University of Florence
via di Santa Marta 3
50139, Florence, Italy
Interuniversity Center of
Integrated Systems for the
Marine Environment (ISME)

**Benedetto Allotta**
Department of Industrial Engineering
University of Florence
via di Santa Marta 3
50139, Florence, Italy
Interuniversity Center of
Integrated Systems for the
Marine Environment (ISME)

## 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 (GNSSs) 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, thanks to the onboard sensors, 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 absence of visual features. After validating the solution through realistic simulations, an experimental campaign at sea was conducted in Stromboli Island (Messina), Italy. In conclusion, an algorithm, which works with the Poisson surface reconstruction method to obtain a smooth seabed surface, for mesh creation has been developed.

*Corresponding author, email address: alessandro.bucci@unifi.it

# 1 Introduction

From geology to exploration and surveillance of archaeological sites and from 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 umbilical cable, acts as a constant connection providing power and communications, and specialised operators are thus able to control the vehicle using the feedback forwarded by the on-board 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 (Prats et al., 2012), (Ferri et al., 2017), 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), and 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 and 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 estimate (Julier and Uhlmann, 2001), (Castellanos et al., 2004), as the estimated covariance from the algorithm can violate the theoretical achievable lower bounds (Dissanayake et al., 2001), (Huang and 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). As shown previously, very few works still exist on underwater SLAM fusing data from a monocular camera and a DVL. Despite that, fusing an optical and an acoustic sensor in a MAP-based framework can take advantages from 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., 2023), (Bucci et al., 2021).
The paper is organized as follows: state-of-the-art in 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 peculiarity of the proposed SLAM strategy are reported in Section 5. While navigation results obtained from simulated environment and from an experimental campaign are reported respectively in Section 6 and 7, an analysis of the mapping capabilities are 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 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. Secondly, by clearly exposing the structure of the problem, reflection on factorial graphs offers many opportunities to improve the performance of key algorithms. Many classical algorithms can be viewed as the application of the elimination algorithm to a particular type of factorial graph. Still, this algorithm is 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. Thirdly, 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 non-linear 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), (Kaess et al., 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) proposes 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 (USBL) measurements are exploited as observations within the on-board navigation filter, where the localization task is solved as a MAP estimation problem. Both these solutions rely on Incremental Smoothing and Mapping 2 (iSAM2), which is the last evolution of the incremental smoothing and mapping solution developed in Georgia Tech Smoothing And Mapping (GTSAM). 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 (GPS), a non-linear 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 motion algorithm for recovering 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 Oriented FAST and Rotated BRIEF (ORB) feature points, with features such as rotational scale invariance and fast detection. ORB-SLAM2 (Mur-Artal and 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) addresses a visual mapping method for precise camera trajectory estimation and 3D reconstruction of underwater ship hull surface using a monocular camera as the primary sensor. (Du et al., 2017) proposes 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 in (Rahman et al., 2018) and (Rahman et al., 2018). Data coming from a mechanical scanning SONAR, a stereo camera, and proprioceptive inertial sensors are fused in a tightly coupled non-linear 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 a 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. A camera is also employed to bound drifts of odometry fused by a DVL, IMU and pressure Depth Sensor (DS) (Kim and Eustice, 2013). Fiducial markers are also integrated into a visual SLAM framework with DVL, IMU, and DS in (Westman and Kaess, 2018).

# 3 Maximum A Posteriori estimation

A navigation and mapping problem is a problem where the unknown state variables $X = \{x_1, x_2, ..., x_M\}$ constituted of poses and landmarks has to be determined given the measurements $Z = \{z_1, z_2, ..., z_N\}$. 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 Eq. 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 $\Sigma_i$ 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), (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), (Westman and Kaess, 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:

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

where $R_i \in \mathrm{SO}(3)$ is the rotation matrix and $\boldsymbol{t}_i \in \mathbb{R}^3$ represents the translation vector. Defining the set of poses at time $k$ with $\mathcal{X}_k$, such that $\mathcal{X}_k = \{T_{x_i}\}_{i=0,1,\dots,k}$, it is possible to define the optimization problem and, in particular, Eq. 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 $\Sigma_{i,j}$, Eq. 4 becomes:

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

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 $\begin{bmatrix} X_{x_i} & Y_{x_i} & Z_{x_i} & \phi_{x_i} & \theta_{x_i} & \psi_{x_i} \end{bmatrix} \in \mathbb{R}^6$ and the measurement function becomes

$$f_{ij}(x_i, x_j) = \begin{bmatrix} X_{x_{i,j}} & Y_{x_{i,j}} & Z_{x_{i,j}} & \phi_{x_{i,j}} & \theta_{x_{i,j}} & \psi_{x_{i,j}} \end{bmatrix}^\top \tag{7}$$

In contrast, for a measurement $z_i$ that indicates a local information on the state $x_i$ with covariance $\Sigma_i$, Eq. 4 is

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

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

$$f_i(x_i) = \begin{bmatrix} X_{x_i} & Y_{x_i} & Z_{x_i} & \phi_{x_i} & \theta_{x_i} & \psi_{x_i} \end{bmatrix}^\top \tag{9}$$

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;
- 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;
- 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 $\begin{bmatrix} X_{x_i} & Y_{x_i} & Z_{x_i} & \phi_{x_i} & \theta_{x_i} & \psi_{x_i} \end{bmatrix} \in \mathbb{R}^6$ that encodes the state at the generic instant. Mathematically, at time $k$, the optimization problem can be written as

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

$\{m_{XYZ-Y}(\cdot), o_{i-1,i}, \Sigma_{o_{i-1,i}}\}$, $\{m_{RP}(\cdot), r_i, \Sigma_{r_i}\}$, $\{m_Z(\cdot), z_i, \Sigma_{z_i}\}$, $\{m_{XY}(\cdot), \boldsymbol{g}_i, \Sigma_{g_i}\}$, $\{m_{XYZ-RPY}(\cdot), p_{i,j}, \Sigma_{o_{i,j}}\}$ are the measurement functions, the measured values and covariances associated to 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, resepectively, $z_i \in \mathbb{R}$ is the depth measurement, $\boldsymbol{g}_i \in \mathbb{R}^2$ is the GNSS measurement. The measurement functions are:

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

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 $T_{x_i} \in \mathrm{SE}(3)$ 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 $\begin{bmatrix} x_{ij} & y_{ij} & z_{ij} \end{bmatrix}^\top$ are the coordinates of the point. The cost function to be minimized can be defined as:

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

where $\rho(\cdot)$ is the Huber robust cost function and $\Sigma_{lm_i}$ is the covariance matrix associated to the scale at which the keypoint $i$ was detected. While $\mathcal{Z}$, $\mathcal{G}$ and $\mathcal{C}$ 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 (Fig. 1).

In terms of implementation, the GTSAM library (Dellaert, 2012) has been used as the back-end for the localization solution. Further information can be found in (Kaess et al., 2008), (Kaess et al., 2012). iSAM2, which is the latest evolution of the incremental smoothing and mapping solution developed in GTSAM, allows only the typical small subset of variables affected by a new measurement, i.e., the measurement function and associated covariances, to be identified and updated, thus limiting the computational load of the

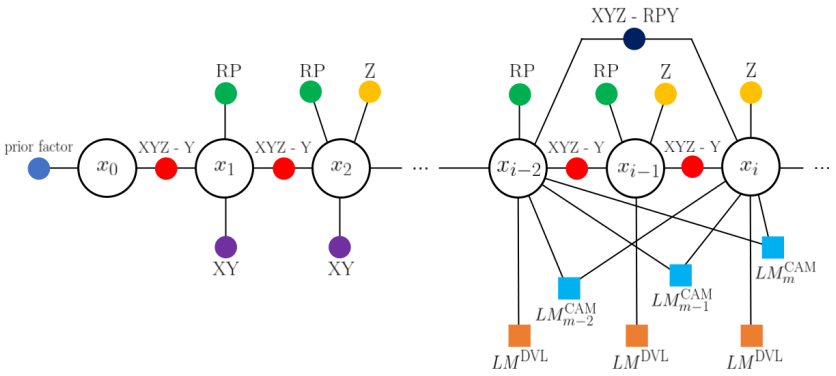


Figure 1: Example of the factor graph at the iteration $i$ constrained with vision-based landmarks and all the onboard sensors.

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 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 necessary to highlight 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. It is necessary to notice that 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, 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 in (Mur-Artal et al., 2015), (Mur-Artal and Tardós, 2017). Furthermore,

following the results reported in (Zacchini et al., 2019), (Bucci et al., 2022), where accurate comparisons between several feature detectors are explained, ORB feature detector has been chosen as the preferable solution instead of Scale Invariant Feature Transform (SIFT), Speeded Up Robust Features (SURF) and Accelerated-KAZE (AKAZE). 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 $\boldsymbol{t}_{c,b} = \begin{bmatrix} t^x_{c,b} & t^y_{c,b} & t^z_{c,b} \end{bmatrix}^\top$ and a rotation matrix $R^b_c = R_z(\psi^b_c)R_y(\theta^b_c)R_x(\phi^b_c)$. It is based on comparing the trajectories estimated through the DVL and 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 the 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. Firstly, the closed-form solution is found by computing the trajectory alignment transformation with translational component on the $xy$-plane of the trajectory estimated with the DVL and the camera and with rotational component computed with respect to the perpendicular axis to this plane. Given the DVL-based positions $\{\boldsymbol{p}^{DVL}_i\}^N_{i=1}$ and the camera-based positions $\{\boldsymbol{p}^{CAM}_i\}^N_{i=1}$, it is necessary to determine the optimal similarity transformation $S^* = \{s^*, R^{b\,*}_c, \boldsymbol{t}^*_{c,b}\} = \{s^*, \psi^{b\,*}_c, t^{x\,*}_{c,b}, t^{y\,*}_{c,b}\}$ that satisfies the minimization problem reported in Eq. 15.

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

where it is necessary to suppose that

$$R^b_c = R_z\left(\psi^b_c\right) \tag{16}$$

$$\boldsymbol{t}_{c,b} = \begin{bmatrix} t^x_{c,b} & t^y_{c,b} & 0 \end{bmatrix}^\top \tag{17}$$

The solution of 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 non-linear least squares problems of the form:

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & \frac{1}{2}\sum_i \rho_i\left(\|f_i(x_{i_1}, ..., x_{i_k})\|^2\right). \\ l_j \leq x_j \leq u_j & \end{aligned} \tag{18}$$

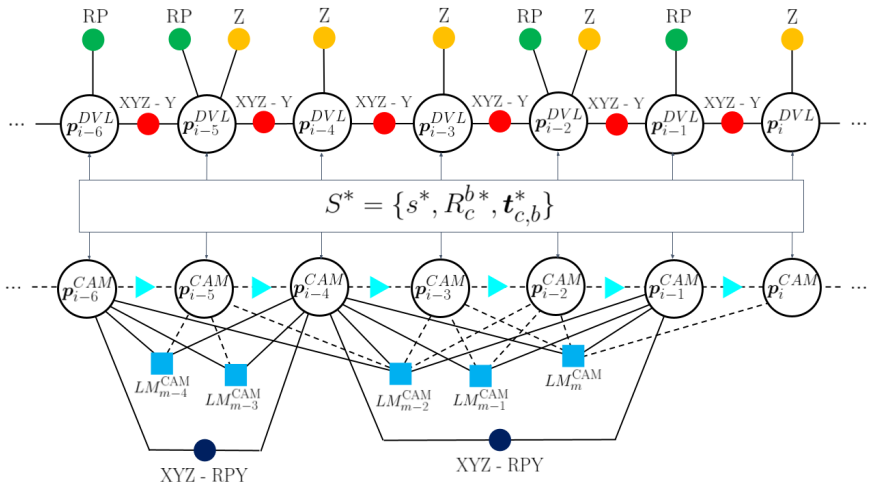


Figure 2: Comparison of the two factor graphs (e.g., the DVL-based graph on the top and the camera-based graph on the bottom of the image) employed for the scale factor ambiguity resolution. The dashed lines in the bottom graph are the edges which are not reported in the whole graph. For ease of reading, a one-to-one association between the two graphs is considered.

The expression $\rho_i\left(\|f_i(x_{i_1},...,x_{i_k})\|^2\right)$ 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 $\{x_{i_1},...,x_{i_k}\}$. $l_j$ 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).

### 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 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 are 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

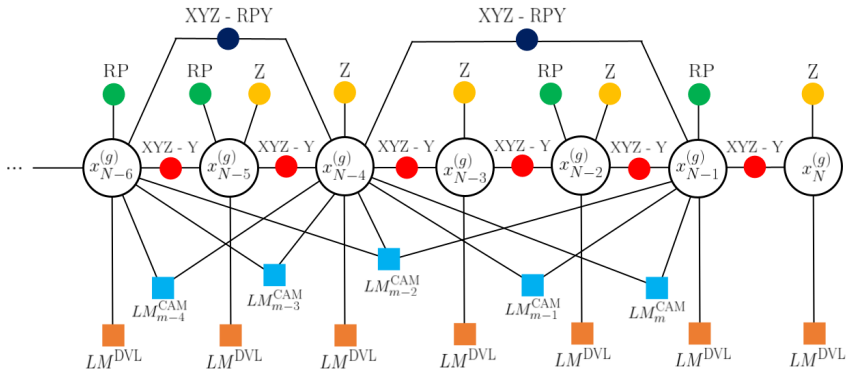


Figure 3: Last nodes of the factor graph $g$ constrained with vision-based landmarks and all the onboard sensors.

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.

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 Fig. 3 and Fig. 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 $x^{(g)}_k$, 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 $x^{(g)}_N$, 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 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.

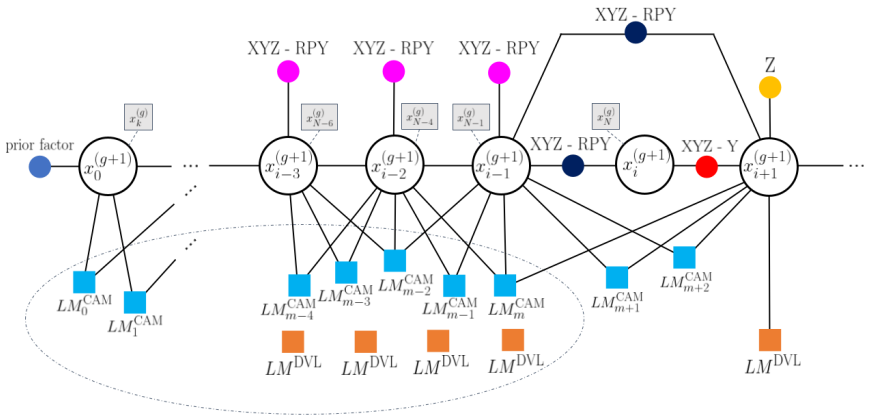


Figure 4: First nodes of the factor graph $g+1$ after the employment of the keyframe reset procedure. The values in the grey boxes represent the corresponding states taken from the previous factor graph $g$ and transferred to the actual graph $g+1$.

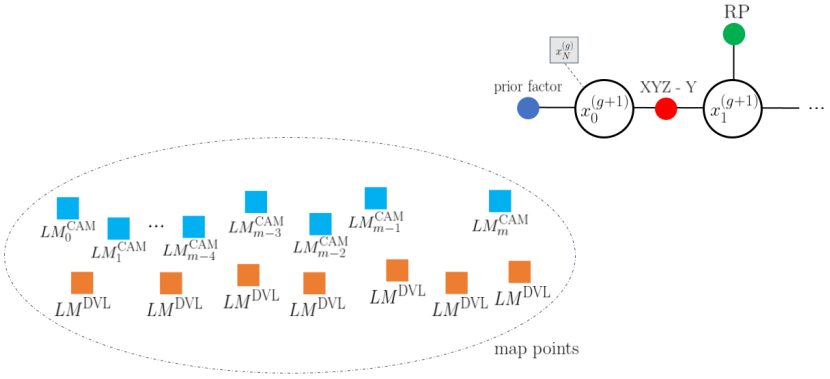


Figure 5: First nodes of the factor graph $g+1$ after the employment of the global reset procedure. The values in the grey boxes represent the corresponding states taken from the previous factor graph $g$ and transferred to the actual graph $g+1$.

Considering the global reset procedure and referring to Fig. 3 and Fig. 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 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.

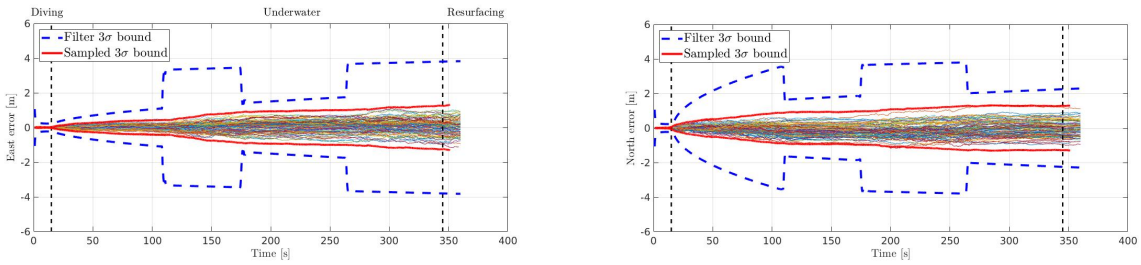


Figure 6: East and North position estimation errors versus their $3\sigma$ bounds obtained from 100 simulation analysis with the SLAM algorithm. The $\sigma$ values are computed as the square-root of the corresponding diagonal element of the estimated covariance matrix.

# 6 Navigation results in simulated environment

To validate the developed DVL and camera-based SLAM algorithm, realistic simulations were performed by means of the Unmanned Underwater Vehicle Simulator (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.

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, depth sensor, DVL and camera. To increase adherence to the real dataset, 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 UUV Simulator, which has traveled a rectangular path at a fixed depth of 2 m. A Monte Carlo simulation with 100 iterations has been performed. The position errors and the estimated $3\sigma$ bounds along the East and North directions are reported in Fig. 6. The covariance trend follows the trajectory described by the vehicle. Still, the SLAM algorithm, due to the presence of visual landmarks which constrains the vehicle position, provides an elliptic $3\sigma$ bound with major axis perpendicular to the direction followed by the vehicle. Despite its particular shape, the $3\sigma$ 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 the previous section, the estimated resurfacing position has been compared with the theoretical first GNSS fix and its $3\sigma$ bound. The resurfacing positions estimated in all the Monte Carlo simulations fall inside the $3\sigma$ bound, guaranteeing reasonable estimations. Furthermore, it is possible to compare the $3\sigma$ bound estimation obtained from the filter and the $3\sigma$ 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 (Fig. 7).

Analyzing the results obtained from the lawnmower survey at a constant depth of 5 meters and comparing the estimated trajectory with the ground truth provided by the simulator, it is possible to notice that the divergence over time of the navigation error is reduced (see Fig. 8). Indeed, even if a global loop closure on the visual keyframes is not performed, the presence of the highly accurate DVL measurements can maintain a low estimation error drift. Furthermore, Fig. 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

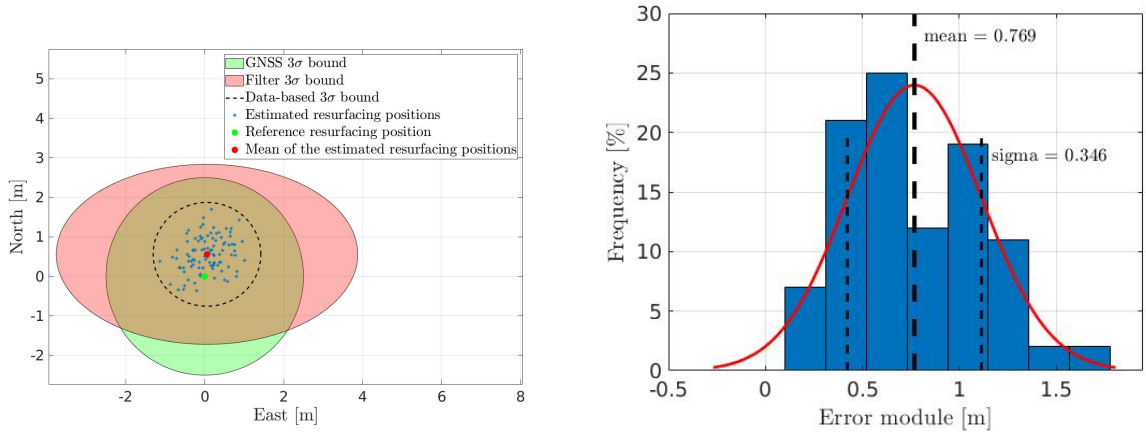


Figure 7: On the left, the estimated resurfacing positions versus the theoretical GNSS fix position obtained from 100 simulation analysis for the SLAM algorithm. On the right, histograms containing the estimated resurfacing position errors obtained from 100 simulation analysis for the SLAM algorithm.

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 |

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. Fig. 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. Consequently, the seabed reconstruction capabilities of the developed algorithm are analyzed in Section 8, where the employed post-processing strategies are described.

# 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 along a pre-programmed 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 reference the trajectory and the map.
FeelHippo AUV (see Fig. 10) is a compact vehicle capable of performing missions in shallow waters. The main features of FeelHippo AUV are reported in Tab. 1. Furthermore, the sensors available on board are listed as follows:

- U-blox 7P precision GNSS;

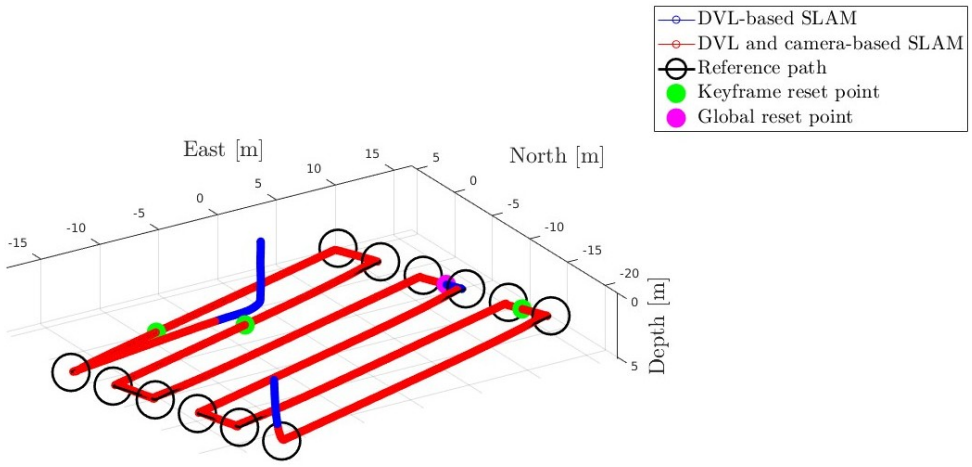


Figure 8: 3D plot of the estimated trajectory in the NED reference system, where the reset points and the areas where vision is not working are highlighted.

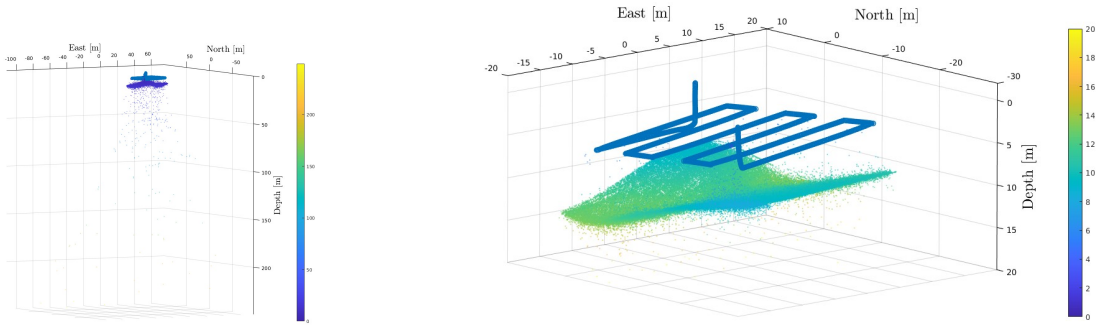


Figure 9: Representation of the point cloud and the travelled trajectory estimated through the SLAM algorithm. While on the top image the entire point cloud is reported and, due to the presence of outliers, the depth scale is too extended, on the bottom image a zoom on the region of interest is performed.

Figure 10: FeelHippo AUV before an on-field underwater mission.

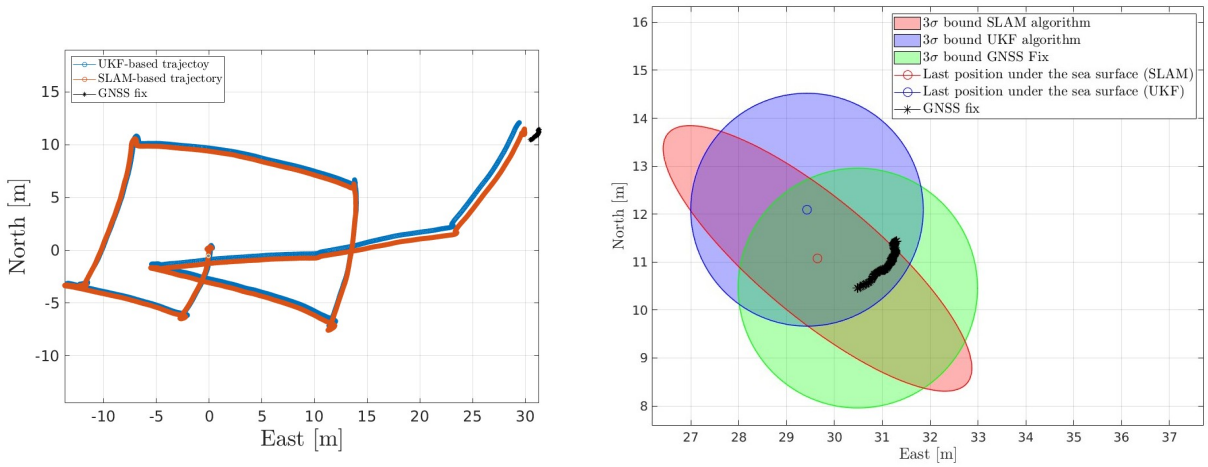


Figure 11: On the left, comparison between the trajectories estimated with the SLAM algorithm and the UKF algorithm during the mission accomplished in Stromboli Island, Messina (Italy). A ground truth, when the vehicle was under the sea surface, is not available, but the first GNSS fix when the vehicle resurfaces can be employed as reference to evaluate the resurfacing error. On the right, $3\sigma$ bound of the last positions under the sea surface estimated with the SLAM and UKF algorithms and the first GNSS fix measurement with its accuracy $3\sigma$ bound.

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 |

- Orientus Advanced Navigation IMU;
- KVH DSP 1760 single-axis high precision Fiber Optic Gyroscope (FOG);
- 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 in (Bucci et al., 2023). The position resurfacing error values and covariances have been evaluated on the North-East plane. Fig. 11 reports the estimated trajectories and an analysis of the resurfacing errors with their $3\sigma$ bound. From Tab. 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.

To evaluate the agreement between estimation errors and estimated uncertainty, the $3\sigma$ bounds during the resurfacing phase are presented. This is summarized in Fig. 11, where the $3\sigma$ 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 the attention on the SLAM algorithm and its mapping capabilities, Fig. 12 reports the SLAM-based estimated trajectory and the generated point cloud. It is possible to evaluate the algorithm mapping

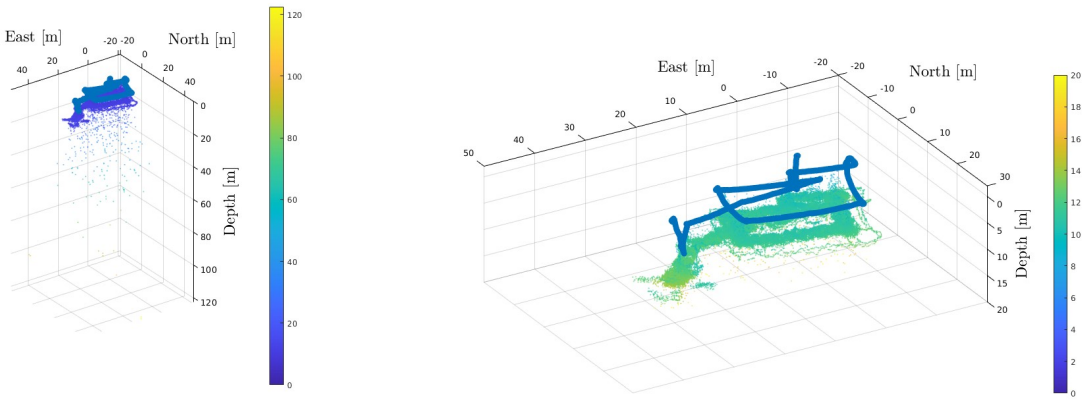


Figure 12: Representation of the point cloud and the travelled trajectory estimated through the SLAM algorithm during the mission in Stromboli Island, Messina (Italy). While on the top image the entire point cloud is reported and, due to the presence of outliers, the depth scale is too extended, on the bottom image a zoom on the region of interest is performed.

Table 3: Estimated scale factor and rototranslation transform between DVL and camera reference systems.

| Parameter | Initial guess | Value after step 1 | Value after step 2 |
|---|---|---|---|
| $s$ | 0.0 | 5.448 | 5.529 |
| $\phi_c^b$ [deg] | 0.0 | 0.0 | $-0.005$ |
| $\theta_c^b$ [deg] | 90.0 | 90.0 | 89.477 |
| $\psi_c^b$ [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 |

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 has 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 Tab. 3. It is necessary to highlight that the proposed strategy can compensate 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 meters, 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

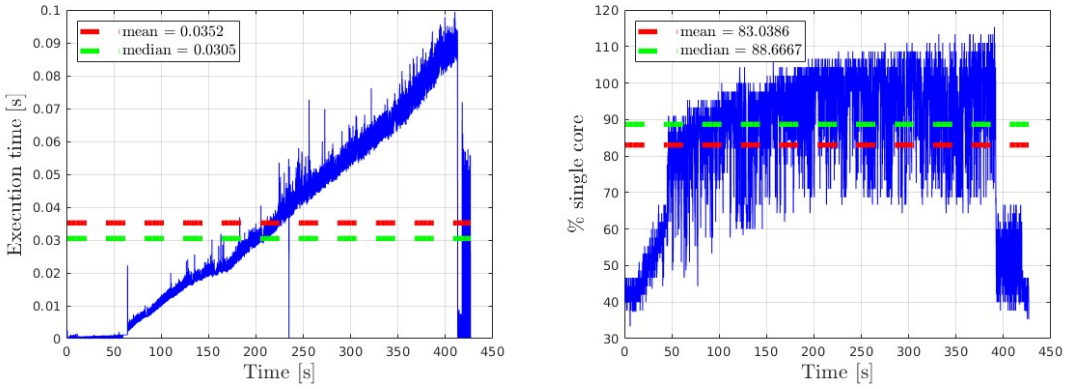


Figure 13: On the left, execution time of the SLAM filter, calculated at each iteration as the sum of the requested time for measurement insertion in the factor graph and for the optimization process. On the right, Central Processing Unit (CPU) burden analysis. In red and green are respectively reported the mean and the median.

Table 4: Mean with the associated covariance and median values of the depth error in presence and in 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 |

process has been considered. For what concern the CPU analysis, the output of the command *top* has been recorded to store the data. The results can be found in Fig. 13. 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 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 (PCL) (Rusu and 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 as the displayed graphs 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

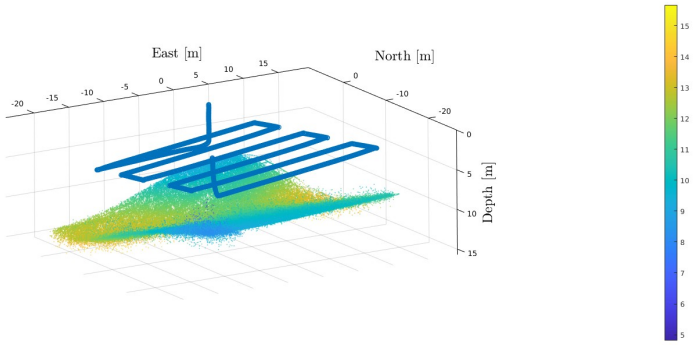


Figure 14: Filtered point cloud and estimated trajectory travelled by the simulated vehicle.

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 set to optimize the reconstruction process. 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 compromise reconstruction speed and estimation quality. The advantages of the Poisson algorithm over other surface fitting methods are numerous. Many implicit methods of surface fitting segment the data into regions for local fitting and further combine these local approximations using blending functions. In contrast, Poisson reconstruction is a global solution that considers all the data simultaneously without resorting to heuristic partitioning or blending. In this way, Poisson reconstruction creates very smooth surfaces that robustly approximate noisy data.

Firstly, considering that in the simulated environment created with UUV Simulator the seabed can be generated with a mathematical function $z = f(x, y)$, it is possible to evaluate the performance of the filtering algorithm. It is also necessary to notice that 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. Fig. 14 reports the 3D filtered point cloud with the estimated trajectory. It is necessary to compare this point cloud with the one directly obtained from the SLAM algorithm and reported in Fig. 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. It is necessary to notice that the outlier points are correctly removed, and the point cloud size is reduced to increase its easiness of management by the reconstruction algorithm (see Fig. 15 and Fig. 16). As can be retrieved from Tab. 4, while the outlier removal process does not influence the mean and the median values due to the high number of points, the covariance associated with the mean value is strongly reduced.

Finally, the 3D point cloud has been processed with the Poisson reconstruction algorithm to build the 3D mesh. Thanks to the chosen reconstruction algorithm, the obtained mesh is smoothed and correctly follows the shape of the simulated sea bottom (see Fig. 17). Some of the results obtained during the mission performed in Stromboli Island, Messina (Italy), in September 2022, are presented. In particular, a 3D reconstruction and a geo-localized map of the sea bottom are reported. The reconstruction comprises around $240k$ points obtained as output from the SLAM algorithm. Firstly, the outlier points have been removed (see Fig. 19), and then, the 3D point cloud has been processed with the Poisson reconstruction algorithm to build the 3D mesh, which is shown in Fig. 20.

The good matching between the reference bathymetry, whose data have mainly funded by the National Research Council and Presidenza del Consiglio dei Ministri–Dipartimento della Protezione Civile, through specific agreement (see Fig. 18), and the estimated 3D reconstruction can also be observed to prove the

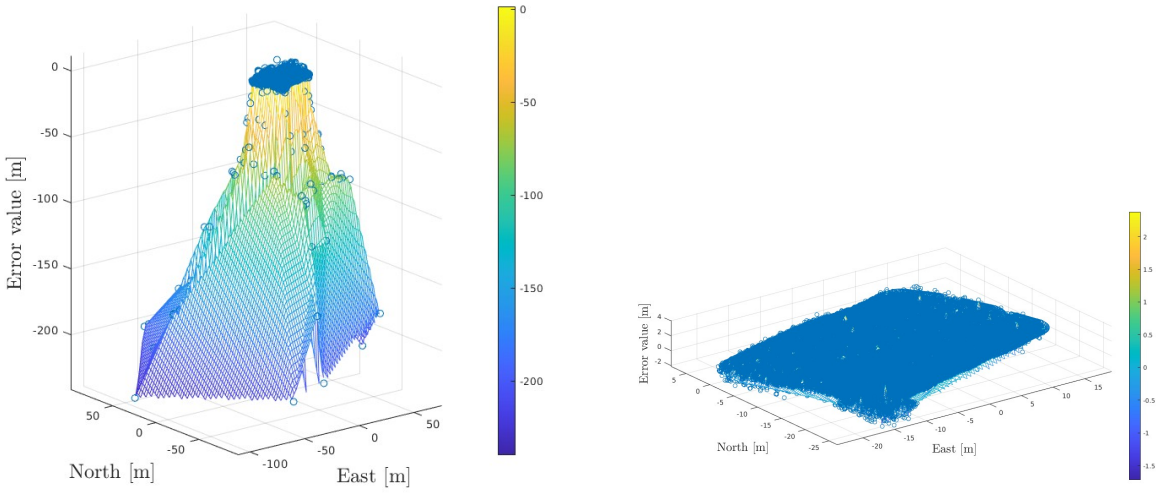


Figure 15: Representation of the error point clouds computed by comparing the reference sea bed function and the estimated point cloud and generation of the estimated error maps before (top image) and after (bottom image) the filtering procedure.

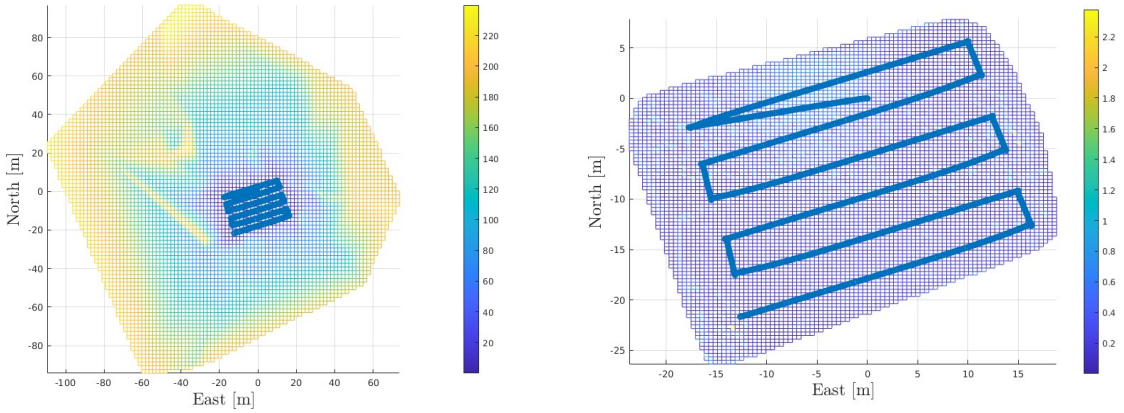


Figure 16: Comparison between the estimated error maps before (top image) and after (bottom image) the filtering procedure with respect to the travelled trajectory by the simulated vehicle.

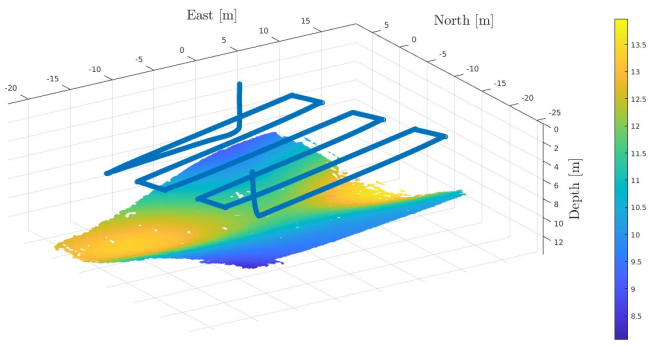


Figure 17: Resulting sea bottom 3D mesh reconstruction and estimated trajectory travelled by the simulated vehicle.

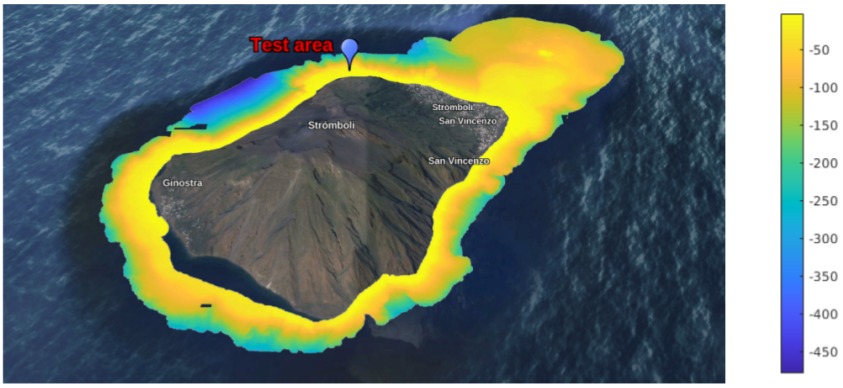


Figure 18: Reference bathymetry of the sea bottom around Stromboli Island, Messina (Italy). The test area, where FeelHippo AUV performed its autonomous mission, is highlighted.

reconstruction's goodness. The provided bathymetry has a horizontal resolution of 5 meters. 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 (Fig. 21).

# 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. Furthermore, progresses on the developed algorithms still needs to be made. Integrating the developed estimation strategy within

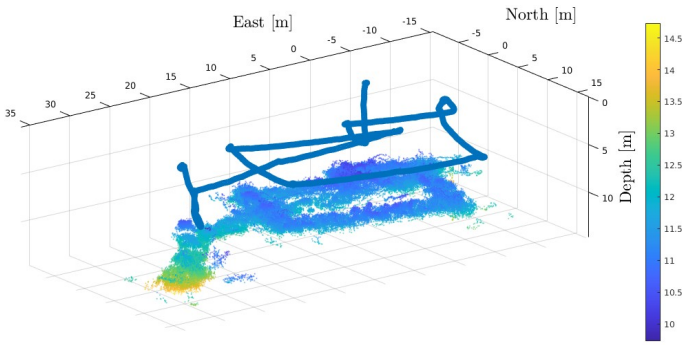


Figure 19: Filtered point cloud and estimated trajectory travelled by the vehicle during the autonomous mission accomplished in Stromboli Island, Messina (Italy).

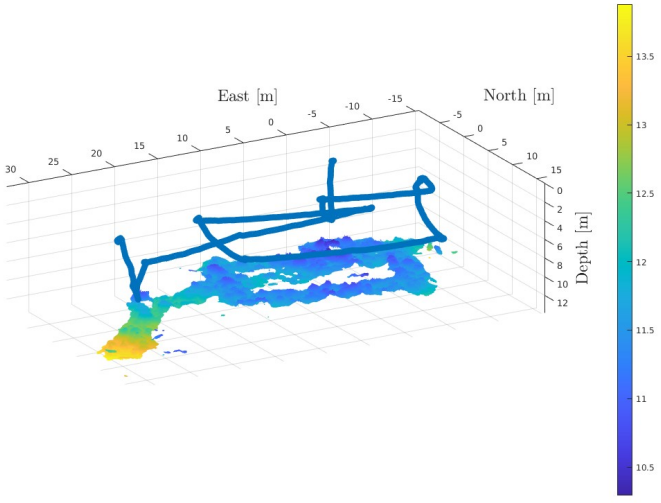


Figure 20: Resulting sea bottom 3D mesh reconstruction and estimated trajectory travelled by the vehicle during the autonomous mission accomplished in Stromboli Island, Messina (Italy).

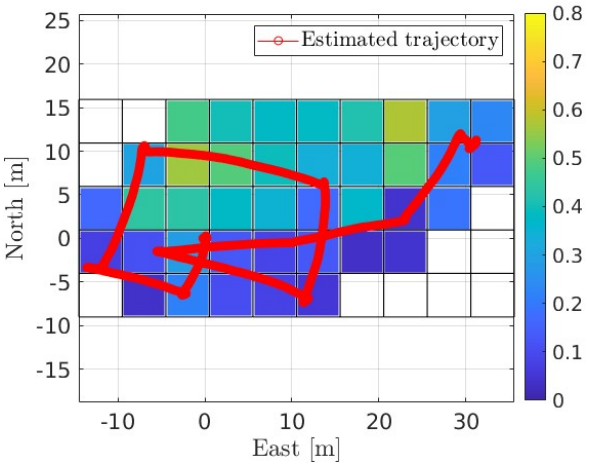


Figure 21: Estimated error bathymetry map with respect to the trajectory travelled by the vehicle during the autonomous mission accomplished in Stromboli Island, Messina (Italy).

the attitude estimator 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), in 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. Finally, from a mapping-based point of view, a multi-vehicle 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.

# 10 Acknowledgement

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 Programme, and with the support of the National Research Council and Presidenza del Consiglio dei Ministri–Dipartimento della Protezione Civile.

## References

Allotta, B., Conti, R., Costanzi, R., Fanelli, F., Gelli, J., Meli, E., Monni, N., Ridolfi, A., and Rindi, A. (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.

Björck, A. (1996). *Numerical Methods for Least Squares Problems*. Society for Industrial and Applied Mathematics.

Bucci, A., Franchi, M., Ridolfi, A., Secciani, N., and 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., and Allotta, B. (2021). Covariance and gain-based federated unscented

kalman filter for acoustic-visual-inertial underwater navigation. In *OCEANS 2021: San Diego – Porto*, pages 1–7.

Bucci, A., Zacchini, L., Franchi, M., Ridolfi, A., and 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., Reid, I., and Leonard, J. J. (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., M. Montiel, J. M., and D. Tardós, J. (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., and Magazzeni, D. (2014). AUV mission control via temporal planning. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6535–6541.

Castellanos, J. A., Neira, J., and Tardós, J. D. (2004). Limits to the consistency of ekf-based slam. *IFAC Proceedings Volumes*, 37(8):716–721. IFAC/EURON Symposium on Intelligent Autonomous Vehicles, Lisbon, Portugal, 5-7 July 2004.

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

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

Dellaert, F. and 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. and 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., and 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., and Qu, F. (2017). View-based underwater SLAM using a stereo camera. In *OCEANS 2017 - Aberdeen*, pages 1–6.

Fallon, M. F., Folkesson, J., McClelland, H., and 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., and Djapic, V. (2017). Multi-domain robotics competitions: The CMRE experience from SAUC-E to the European Robotics League Emergency Robots. In *OCEANS 2017 - Aberdeen*, pages 1–7.

Forster, C., Carlone, L., Dellaert, F., and 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., and Costanzi, R. (2021). Maximum A Posteriori estimation for AUV localization with USBL measurements. *IFAC-PapersOnLine*, 54(16):307–313. 13th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles CAMS 2021.

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

Hong, S. and 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. and Dissanayake, G. (2007). Convergence and consistency analysis for extended kalman filter based slam. *IEEE Transactions on Robotics*, 23(5):1036–1049.

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

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

Kaess, M., Johannsson, H., Roberts, R., Ila, V., Leonard, J. J., and 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., and Dellaert, F. (2008). iSAM: Incremental smoothing and mapping. *IEEE Transactions on Robotics*, 24(6):1365–1378.

Kazhdan, M., Bolitho, M., and Hoppe, H. (2006). Poisson Surface Reconstruction. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, page 61–70.

Kim, A. and 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., and Tardós, J. D. (2015). ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics*, 31(5):1147–1163.

Mur-Artal, R. and 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. and 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*, pages 1042–1049.

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

Prats, M., Ribas, D., Palomeras, N., García, J. C., Nannen, V., Wirth, S., Fernández, J. J., Beltrán, J. P., Campos, R., Ridao, P., Sanz, P. J., Oliver, G., Carreras, M., Gracias, N., Marín, R., and Ortiz, A. (2012). Reconfigurable AUV for intervention missions: a case study on underwater object recovery. *Intelligent Service Robotics*, 5:19–31.

Rahman, S., Li, A. Q., and Rekleitis, I. (2018). Sonar Visual Inertial SLAM of Underwater Structures. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5190–5196.

Rahman, S., Li, A. Q., and Rekleitis, I. M. (2018). SVIn2: Sonar Visual-Inertial SLAM with Loop Closure for Underwater Navigation. *ArXiv*, abs/1810.03200.

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

Rusu, R. B., Marton, Z. C., Blodow, N., Dolha, M., and 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., and Carreras, M. (2020). Multisensor online 3D view planning for autonomous underwater exploration. *Journal of Field Robotics*, 37(6):1123–1147.

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

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

Westman, E. and 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., and 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., and Ridolfi, A. (2019). Mono visual odometry for Autonomous Underwater Vehicles navigation. In *2019 MTS/IEEE Oceans, Marseille, France*.

Zhang, Y., Zhang, T., and Huang, S. (2018). Comparison of ekf based slam and optimization based slam algorithms. In *2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pages 1308–1313.

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