Ocean Engineering 355 (2026) 125170

Contents lists available at ScienceDirect

Ocean Engineering

journal homepage: www.elsevier.com/locate/oceaneng

Multi-session perception-aware coverage path planning for active semantic SLAM and automatic change detection

Alessandro Bucci [a,b,*], Alessandro Ridolfi [a,b]

[a] Department of Industrial Engineering, University of Florence, via di Santa Marta 3, Florence, 50139, Italy
[b] Interuniversity Center of Integrated Systems for the Marine Environment (ISME), www.isme.unige.it, Italy

ARTICLE INFO

Keywords:
Marine robotics
Change detection
Coverage path planning
Semantic SLAM

ABSTRACT

This paper introduces a novel framework for multi-session, perception-aware coverage path planning integrated with active semantic Simultaneous Localization and Mapping (SLAM) and automatic change detection. The goal is to enhance autonomous robotic exploration in dynamic environments by combining semantic understanding with adaptive path planning for long-term monitoring. The proposed approach consists of three tightly integrated components. First, a semantic-informed inspection planner uses Kernel Density Estimation (KDE) to prioritize exploration of semantically significant regions. Second, an active semantic SLAM module builds a semantic map incrementally, providing real-time feedback to refine the inspection path. Third, a multi-session change detection strategy compares current and previous semantic data to identify and localize environmental changes. Together, these components allow the robot to intelligently adapt its exploration strategy over time, focusing on areas of interest and reacting to environmental dynamics. The framework is validated through both simulation and real-world experiments, demonstrating improved coverage efficiency, mapping accuracy, and change detection robustness compared to traditional methods. While applied to buoy detection, the system is broadly applicable to long-term robotic tasks such as environmental monitoring, infrastructure inspection, mine counter-measure operations, and disaster response, or other scenarios that demand adaptability and semantic awareness in complex, evolving environments.

1. Introduction

Autonomous underwater robots have become increasingly crucial for marine exploration, environmental monitoring, and underwater inspection tasks. These robots are equipped with various sensors, such as sonar and cameras, to enable them to map, navigate, and interact with the underwater environment. A critical component of such operations is Simultaneous Localization And Mapping (SLAM), which allows the robot to create and update a map of the underwater environment while estimating its position within that map (Bucci et al., 2024; Rahman et al., 2018). While traditional SLAM approaches have predominantly relied on geometric data, semantic SLAM has emerged as an essential advancement, enabling robots to incorporate semantic information such as the recognition of underwater objects, structures, and hazards, further enhancing their situational awareness and operational effectiveness (Kostavelis and Gasteratos, 2015).
While semantic SLAM offers a more comprehensive understanding of the environment, a critical challenge in underwater robotics is efficient coverage path planning (Galceran and Carreras, 2013; Guo et al., 2021). This involves determining the optimal path for the robot to explore an area, ensuring full coverage for tasks such as environmental surveying or object inspection (Pairet et al., 2022; Sodhi et al., 2019; Ma et al., 2024; Yao et al., 2021). Given the large and often complex nature of underwater environments, developing coverage strategies that balance time efficiency, path optimization, and sensor data collection is crucial. However, traditional coverage planning approaches may fall short when applied to dynamic or evolving environments, especially in the context of underwater systems where conditions such as tides, currents, and marine life can alter the environment over time.
Consequently, in underwater robotics, the need for multi-session coverage planning is paramount. As these robots are deployed over extended periods for long-term environmental monitoring, revisiting previously explored areas for change detection becomes essential (Jang et al., 2021; Burguera and Bonin-Font, 2018; Larsen et al., 2023). Underwater environments are highly dynamic, and changes such as the movement of objects, growth of marine life, or alterations in the underwater terrain require the robot to identify and adapt to these modifications. Thus, the integration of automatic change detection is a critical aspect of the

* Corresponding author.
E-mail addresses: alessandro.bucci@unifi.it (A. Bucci), alessandro.ridolfi@unifi.it (A. Ridolfi).

https://doi.org/10.1016/j.oceaneng.2026.125170
Received 4 April 2025; Received in revised form 16 January 2026; Accepted 17 March 2026
Available online 25 March 2026

robot's exploration strategy. Revisiting areas previously explored during earlier sessions introduces the challenge of efficiently planning paths that maximize perception and allow for accurate identification of environmental changes.

While numerous object detection algorithms (Yu et al., 2021; Zhang et al., 2022; Huang et al., 2023)have been developed for underwater robotic applications, ranging from traditional methods to deep learning-based approaches, there remains a notable gap in the area of underwater change detection. Despite the growing interest in underwater mapping and monitoring, few algorithms have been specifically tailored for detecting changes over time in underwater scenes. This highlights a critical need for dedicated research and development in underwater change detection methodologies, which are essential for long-term monitoring, ecological assessments, and infrastructure inspection.

This paper proposes a novel framework for multi-session perception-aware coverage path planning for autonomous underwater robots, aimed at improving active semantic SLAM and automatic change detection. By combining perception-driven coverage planning with multi-session revisiting strategies, our approach enables autonomous underwater robots to adaptively plan paths that enhance the detection and updating of semantic knowledge in dynamic environments. The proposed method is designed to address the challenges of underwater exploration, where environmental conditions can drastically change over time, by refining exploration strategies that prioritize both environmental coverage and the monitoring of changes.

The contributions of this paper are twofold. First, we introduce a robust method for perception-aware coverage path planning that incorporates both geometric and semantic data for underwater robots. One of the main advancements of this work is the unified integration of perception-aware planning, semantic SLAM, and target localization within a single framework specifically designed for underwater robotics. Unlike existing methods that treat these components separately, our approach jointly optimizes coverage and semantic understanding. Second, we extend this framework to multi-session exploration, providing an effective solution for automatic change detection in dynamic underwater environments. While existing change detection techniques are often limited in scope and not embedded within a coverage framework, our method addresses this gap through a dedicated multi-session strategy for detecting and localizing environmental changes over time. The proposed approach not only enhances the efficiency of path planning but also improves the robot's ability to detect, adapt to, and monitor environmental changes, making it an ideal solution for long-term underwater exploration and monitoring. Although the framework has been tested using colored buoys as Objects of Potential Interest (OPIs), it can be easily extended to other, more practical targets for localization and monitoring. In such cases, the primary modifications would concern the Automatic Target Recognition (ATR) algorithm to accommodate different object types or detection needs.

The paper is organized as follows: state-of-the-art in informative path planning and active semantic SLAM are detailed in Section 2, whereas Section 3 outlines the preliminaries and the necessary notations and Section 4 is dedicated to the employed framework to accomplish the desired monitoring mission. Section 5 treats the validation of the strategy in simulated environment, whereas experimental results are reported in Section 6. Finally, Section 7 draws conclusions.

2. Related works and main contributions

Mobile robots, whether operating on the ground (Isler et al., 2016), in the air (Bircher et al., 2016), or in marine environments (Smith et al., 2011; Zacchini et al., 2022), are often tasked with autonomously gathering information about their surroundings. To address the challenges of this Information Gathering (IG) task, various Informative Path Planning (IPP) techniques have been developed in recent years. Despite the different approaches used, these methods share a common goal: to plan safe and feasible paths that allow the robot to collect as much information as possible with the available sensor payload during a survey or inspection. In dynamic or unknown environments, online IPP solutions-those that do not rely on pre-existing knowledge-are crucial. These solutions utilize real-time sensor data and adapt to the environment as it is perceived (Wang et al., 2020; Paull et al., 2012) . Specifically, IPP algorithms in such scenarios must continuously compute and update optimal or near-optimal inspection paths based on the data available at each moment. These paths must adhere to the robot's motion constraints. Moreover, since mobile robots typically have limited computational resources and are often subject to time constraints in real-time applications, IPP algorithms must be efficient in terms of computation. Sampling-based algorithms are the state-of-the-art methods for path planning, due to the absence of a fixed endpoint and the reduced computational burden. Such algorithms, as the Rapidly-exploring Information Gathering (RIG) (Hollinger and Sukhatme, 2014), operate in continuous space, easily incorporate motion constraints, and are computationally efficient, making it suitable for real-time operation on mobile robots without the need of specialized hardware. Additionally, IG solutions based on the Rapidly-exploring Random Tree (RRT) algorithm can efficiently explore large workspaces, even in high-dimensional environments, and handle robots with multiple degrees of freedom. As a result, RRT-based IG approaches have become a leading choice for many real-world applications.

Turning to the SLAM strategies, in the underwater domain, two key works have inspired the development of factor graph approaches, and have been taken as reference for this work. Most of the approaches exploit Incremental Smoothing and Mapping 2 (iSAM2) (Kaess et al., 2008, 2012), an incremental smoothing and mapping solution for graph-based optimization problems. The algorithm in Westman and Kaess (2019) generates constraints between subsequent poses for each SONAR image association and integrates these constraints with vehicle odometry factors in a pose graph optimization framework. Meanwhile, Franchi et al. (2021) uses Ultra-Short BaseLine (USBL) measurements as constraints within the onboard navigation filter, treating the localization task as a Maximum A Posteriori (MAP) estimation problem. Additionally, other graph-based SLAM methods have been proposed to fuse data from both navigation and perception sensors, including acoustic and optical sensors. In Fallon et al. (2013), a graph-based SLAM solution is applied to an Autonomous Underwater Vehicle (AUV) for mine countermeasures and vehicle localization. The graph is initialized using a pose node from Global Positioning System (GPS) when the AUV is on the sea surface, and Doppler Velocity Log (DVL) and Inertial Measurement Unit (IMU)-based dead reckoning estimates along with SONAR images are employed to perform a non-linear least squares optimization when the vehicle is submerged. In Huang and Kaess (2015), with the aim of recovering 3D scene structures from multiple 2D SONAR images while simultaneously localizing the SONAR system, an acoustic structure-from-motion algorithm is introduced.

Turning to underwater semantic mapping and SLAM algorithms references can be found in Himri et al. (2018), Guerneve et al. (2017). In Himri et al. (2018) the authors have developed a localization and mapping system that utilizes a 3D object recognition pipeline, enabling the creation of maps at the object level. The real-time 3D object detection capability was also shown to enhance map accuracy, particularly when closing trajectory loops during SLAM. In Guerneve et al. (2017), a semantic mapping pipeline for underwater environments is proposed, which integrates multibeam SONAR data with a collection of Computer-Aided Design (CAD) models representing the structures of interest. This approach generates a simplified 3D model of the environment, and can be applied both in real-time and during post-processing.

In this paper an intelligent strategy for autonomous inspection of underwater environment and detection of OPIs has been developed. The main contribution is related to the proposed framework, which requires a minimal human intervention, thanks to the employment of the information gathered by the AUV during its mission. The two main actors of the proposed inspection approach, i.e. the perception-aware coverage path planning with geometric and semantic data and the extension to multi-session exploration with automatic change detection, represents the main advancements contained in this work. Regarding the first theme, a new method for coverage path planning that enhances underwater robots' ability to plan efficient exploration paths, incorporating both geometric (spatial) and semantic (contextual or environmental) data is proposed. While most traditional path planning methods primarily focus on the geometric aspect of the environment, overlooking the broader environmental context, which is crucial in unknown real-world scenarios, in the presented work the combination of both geometric and semantic data ensures that the robot adapts its path planning according to the complexity of the underwater world, improving the overall efficiency and relevance of the exploration. Indeed, by adding semantic data, the approach recognizes that certain regions within the environment may have more importance and require more attention.

Turning to the second contribution, it necessary to highlight that it extends the standard approach to multi-session exploration, which is important for long-term, recurring missions. This part of the work focuses on enabling automatic change detection in dynamic underwater environments, an area of significant interest for ongoing monitoring. In particular, traditional exploration systems often focus on a single mission session where the robot maps the environment once. However, in the context of underwater environments, periodic exploration is needed because these ecosystems change over time. With multi-session exploration, the robot revisits the same area at different times to track changes in the environment. The change detection system automatically compares the new data to previously collected data to identify what has changed-whether it's environmental shifts, the appearance of new objects, or changes in marine life. In this paper, we will focus on changes related to new and/or removed objects in the marine environment. This can be crucial, in an extended vision, for monitoring the health of ecosystems or detecting disturbances like illegal fishing, pollution, or even climate change effects. Thus, the robot's path planning method is adjusted to adapt to changes that may have occurred since the last exploration session. This contribution offers a continuous, automated solution for monitoring dynamic underwater environments. It provides a scalable and efficient way to track changes over time, detect anomalies, and respond appropriately, making it ideal for long-term monitoring tasks such as conservation, marine research, or infrastructure inspection.

When combined, these two contributions offer a powerful and flexible approach for underwater exploration. The perception-aware path planning improves the robot's ability to autonomously navigate and map complex environments, while the multi-session framework ensures that the robot can adapt to and monitor long-term changes. This dual approach enhances both the robot's ability to cover large areas efficiently and its capacity to detect, interpret, and adapt to changes over time, making it well-suited for critical applications like underwater monitoring, environmental protection, and autonomous research in marine environments.

3. Preliminaries and notation

3.1. Preliminaries on SLAM algorithms

Considering unknown state variables $X = \{x_1, x_2, ..., x_M\}$, which include poses and landmarks, based on the measurements $Z = \{z_1, z_2, ..., z_N\}$, a navigation and mapping problem can be stated. The objective of the MAP estimator is to find the state configuration X that maximizes the posterior probability $p(X|Z)$, which is correlated to the likelihood of the states X given the measurements Z. Consequently, it can be easily obtained that

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

where $l(z_i|X)$ is the likelihood distribution. The nonlinear problem can be tackled using standard techniques like the Gauss-Newton or Levenberg-Marquardt algorithms. These methods iteratively approach the solution by solving a linearized version of the nonlinear system at each step. More information can be found in Grisetti et al. (2020), Dellaert and Kaess (2017).

Defining the state of the system at instant i as a pose belonging to the special Euclidean group SE(3), it is possible to asses that the pose T_{x_i} can be mathematically expressed as:

$$T_{x_i} = \begin{bmatrix} R_i & \boldsymbol{t}_i \\ \mathbf{0}^{1\times 3} & 1 \end{bmatrix} \tag{2}$$

where $R_i \in \mathrm{SO}(3)$ is the rotation matrix and $\boldsymbol{t}_i \in \mathbb{R}^3$ represents the translation vector. For ease of explanation T_{x_i} is usually described through the vector $\begin{bmatrix} X_{x_i} & Y_{x_i} & Z_{x_i} & \phi_{x_i} & \theta_{x_i} & \psi_{x_i} \end{bmatrix} \in \mathbb{R}^6$, where can be easily distinguished the translation and the rotation parts respectively.

3.2. Mapping and OctoMap preliminaries

The mapping approach used in this work is based on the occupancy grid mapping method, which has proven to be a robust way to represent the environment. Occupancy grid mapping has been widely applied in marine robotics for various tasks, such as collision detection, obstacle avoidance (Youakim et al., 2020), environment mapping (Franchi et al., 2020; Teixeira et al., 2016; Ho et al., 2018), and path planning (Hernández et al., 2019; Vidal et al., 2020). In this framework, given the robot's pose T_{x_i} and its measurements $z_{1:t}$ up to time t, the occupancy grid aims to construct the map $\mathcal{M}$ based on those observations. The classic occupancy grid theory, as used in this work, incorporates the assumptions of grid cells independence and the Markovian hypothesis. Given these premises, the OctoMap framework (Hornung et al., 2013; Yguel et al., 2008) has been employed to develop the 3D probabilistic occupancy mapping solution. This framework allows for modeling free, occupied, and unmapped areas, which is essential for planning and exploration in unknown environments. It also enables the update of the probabilistic occupancy of each grid cell using the following update rule, within a Bayesian context:

$$\frac{p\left(m_i|z_{1:t}\right)}{1-p\left(m_i|z_{1:t}\right)} = \frac{p\left(m_i|z_t\right)}{1-p\left(m_i|z_t\right)} \frac{p\left(m_i|z_{1:t-1}\right)}{1-p\left(m_i|z_{1:t-1}\right)} \frac{1-p\left(m_i\right)}{p\left(m_i\right)}. \tag{3}$$

It is possible to define the log-odds ratio as the logarithm of the ratio between a probability and its complementary probability, as $l(\cdot) = \log \frac{p(\cdot)}{1-p(\cdot)}$. Following this definition, it is possible to derive that the update formulation for the occupancy grid problem can be modeled as:

$$l\left(m_i|z_{1:t}\right) = l\left(m_i|z_t\right) + l\left(m_i|z_{1:t-1}\right) - l\left(m_i\right) \tag{4}$$

where $l\left(m_i|z_t\right)$ represents the Inverse Sensor Model (ISM), $l\left(m_i|z_{1:t-1}\right)$ is the recursive term updated during each iteration and $l\left(m_i\right)$ is the prior assumption. In addition to the update formulation given in Eq. (4), the standard version of OctoMap assumes a non-informative prior $p\left(m_i\right) = 0.5$. This assumption is sensible when mapping in entirely unknown environments, as it reflects an equal uncertainty about whether each grid cell is occupied or free before any measurements are performed.

3.3. Motion planning formulation

In many underwater applications, AUVs typically perform surveys at a constant depth or altitude. Vehicles which are hydrostatically stable in terms of roll and pitch dynamics are largely employed in underwater inspection tasks. Since the missions in this work do not cause significant changes in these dynamics, the AUV maintains nearly zero roll and pitch angles, with minimal fluctuations. Furthermore, to conserve energy, AUVs primarily travel along the surge axis and only rotate around the heave axis, meaning they adjust their heading without significant vertical motion. As a result, the kinematic motion of the AUV on a horizontal plane can be described with a simple model:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} u\cos\psi \\ u\sin\psi \\ r \end{bmatrix} \tag{5}$$

where x and y represent the position of the vehicle on the North, East (NE) plane and ψ the heading angle. u and r are the control input to the system, i.e. the surge speed and the angular rate respectively. Kinematic models use geometric equations to describe the relationship between the vehicle's positions and velocities, while dynamic models incorporate the forces and torques that influence the motion. As a result, dynamic models offer a more precise representation of the AUV's motion constraints. However, for the purposes of this work, which focuses on addressing the IPP problem, only the kinematic model is utilized.

The motion of the AUV is described using the kinematic model from Eq. (5), with the configuration space represented as $C = \mathrm{SE}(2)$, meaning that a configuration ξ belongs to SE(2). The AUV's kinematic motion is modeled using the Dubins vehicle model, which represents the shortest path between two configurations as a combination of circular arcs with maximum curvature and straight line segments. The Dubins vehicle model is especially useful for planning tasks because it helps generate new configurations, similar to how the tree is expanded in RRT-based solutions, and also functions as a steering mechanism during the rewiring process.

4. Developed framework

The proposed inspection solution, illustrated in Fig. 1, consists of five components: the Mission Manager, the Mapping Module, the IPP Module, the Guidance, Navigation and Control (GNC) Module (which comprises the Semantic SLAM Module) and the Saver/Loader Module. These modules operate onboard the AUV and are integrated within the Robot Operating System (ROS) framework (Quigley et al., 2009). Additionally, a Graphical User Interface (GUI) was developed to run on a laptop computer, enabling an operator to communicate and interface with the AUV. The GUI provides an easy way to set the area to be inspected, which is then transmitted to the AUV via the radio or Wi-Fi link.

The Mission Manager utilizes the defined limits of the inspection area to constrain the workspace of the other modules. Upon receiving the start command from the GUI or when the vehicle reaches the each viewpoint, it activates the planner contained in the IPP module to compute the Next Best Viewpoint (NBV). Additionally, the Mission Manager monitors the progress of the coverage.

The Mapping Module is responsible for creating and updating an occupancy map. It uses the AUV's estimated pose provided by the SLAM algorithm, the DVL measurements and the collected optical images as inputs. When a new image is captured or a new DVL data is available, the module generate a 3D point cloud thanks to the front-end part and update the occupancy map with the back-end part. A detailed explanation of the mapping process can be found in Section 4.2.

The IPP module, by using the RIG algorithm outlined in Section 4.3, addresses a maximization problem by expanding a tree of possible views from the robot's current configuration. At this point, the proposed informed expansion strategy is applied. The discovered semantic map is used to train the Kernel Density Estimation (KDE) estimator online. The RIG algorithm then leverages the learned distribution to guide the vehicle toward the most promising areas, generating nodes using the informed sampling policy and weighting them in function of the task to be accomplished.

The GNC Module comprises the algorithms dedicated to guidance, navigation, and control and that are employed by the AUV to estimate its position and follow the planned route. This module will be analyzed in Section 4.1 for what concern the navigation framework. Indeed, a Semantic SLAM algorithm has been developed and employed for the vehicle pose estimation and the OPIs detection and localization.

Finally, the Saver/Loader Module is employed to save data and to transfer them from the first inspection mission to the second change detection mission. In particular, this module is firstly dedicated to save the inspection area boundaries decided by the user through the GUI and the semantically-annotated map computed by the vehicle. During the change detection mission, the module load the inspection area directly to the vehicle to be sure that the AUV will inspect the same area and the semantic occupancy map to perform the comparison for possible changes detection.

4.1. Semantic SLAM for target localization and vehicle navigation

It is necessary to introduce how to mathematically model the factors used to represent the measurement constraints in solving the autonomous navigation and mapping problem. Drawing inspiration from Westman and Kaess (2018, 2019), the factors outlined below have been utilized. The data from the available onboard sensors has been encoded as measurement factors to constrain the optimization process, which has to be solved to evaluate the MAP estimate. The following factors have been incorporated (see Fig. 2):

- a relative 4D pose-to-pose constraint on x, y, and z translation and yaw rotation, is established using the measurements from the DVL and the yaw estimated thanks to the attitude estimation filter (Allotta et al., 2015; Allotta et al., 2016) operating on IMU and Fiber Optic Gyroscope (FOG) measurements. This constraint is encoded with the measurement $o_{i-1,i}$ and associated covariance $\Sigma_{o_{i-1,i}}$;
- a unary constraint on pitch and roll rotations, estimated through the attitude estimation filter and represented with the measurement r_i and the covariance Σ_{r_i};
- a unary 1D constraint on z translation exploiting the Depth Sensor (DS) measurements $z_i \in \mathbb{R}$ with covariance Σ_{z_i};
- a unary 2D constraint on x and y translation thanks to the GPS observations $\boldsymbol{g}_i \in \mathbb{R}^2$ with covariance Σ_{g_i};
- a camera-based constraint between the vehicle 6D pose (i.e., position and orientation of the AUV) and the landmark 3D position for each OPI detected with the ATR algorithm applied to multiple images.

The implemented approach adds a new state only when at least one observation from GPS, DVL, or DS is available. The link between adjacent nodes is maintained by collapsing the relative motion XYZ-Y in a single compound constraint, where simple Dead Reckoning (DR) is performed between the two consecutive nodes with the last acquired DVL measurements. Representing the T_{x_i} 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

$$
\begin{aligned}
\mathcal{X}_k^* \quad = \quad & \underset{X}{\operatorname{argmax}} \sum_{i=1}^{k-1} \\
& \left(\| m_{XYZ-Y}(x_{i-1}, x_i) \ominus o_{i-1,i} \|^2_{\Sigma_{o_{i-1,i}}} + \| m_{RP}(x_i) \ominus r_i \|^2_{\Sigma_{r_i}} \right) + \\
& + \sum_{i \in \mathcal{Z}} \| m_Z(x_i) - z_i \|^2_{\Sigma_{z_i}} + \\
& + \sum_{i \in \mathcal{G}} \| m_{XY}(x_i) - \boldsymbol{g}_i \|^2_{\Sigma_{g_i}} + \\
& + \sum_{j \in \mathcal{LM}, i \in C} \rho \left(\| \boldsymbol{p}_{ij} - \pi_i(T_{x_i} \boldsymbol{P}_{x_j}) \|^2_{\Sigma_{lm_i}} \right) + \\
& + \| T_{x_0} \ominus T_{x_{prior}} \|^2_{\Sigma_0}
\end{aligned} \tag{6}
$$

where $m_{XYZ-Y}(x_{i-1}, x_i) = \begin{bmatrix} X_{x_{i-1,i}} & Y_{x_{i-1,i}} & Z_{x_{i-1,i}} & \psi_{x_{i-1,i}} \end{bmatrix}^\top$, $m_{RP}(x_i) = \begin{bmatrix} \phi_{x_i} & \theta_{x_i} \end{bmatrix}^\top$, $m_Z(x_i) = \begin{bmatrix} Z_{x_i} \end{bmatrix}$, and $m_{XY}(x_i) = \begin{bmatrix} X_{x_i} & Y_{x_i} \end{bmatrix}^\top$ are the measurement functions, $o_{i-1,i}$, r_i, z_i, and $\boldsymbol{g}_i$ are the measured values and $\Sigma_{o_{i-1,i}}$, Σ_{r_i}, Σ_{z_i}, and Σ_{g_i} are the covariances associated to the previously introduced factors. Thanks to the features extracted from the ATR algorithm applied to optical images, it is possible to optimize landmark locations $\boldsymbol{P}_{x_j} \in \mathbb{R}^3$ and related poses $T_{x_i} \in \mathrm{SE}(3)$ minimizing the reprojection error with respect to the projected objects $\boldsymbol{p}_{ij} \in \mathbb{R}^2$ on the image plane. To define the function to be optimized in the form as in Eq. (6), the Huber robust cost function $\rho(\cdot)$ is employed, where $\pi_i(\cdot)$ represents the projection function and Σ_{lm_i} is the covariance matrix on the image plane. The set of pose nodes

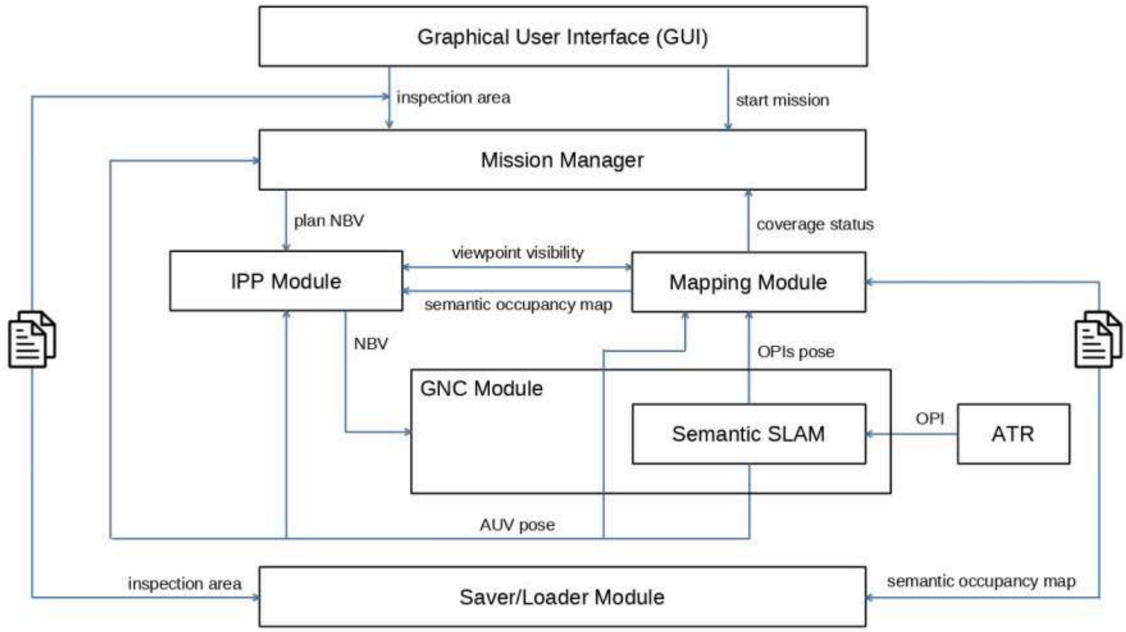

Fig. 1. The developed inspection framework.

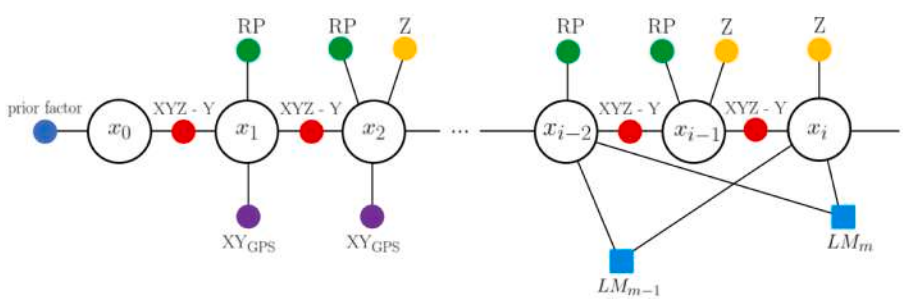

Fig. 2. Example of the factor graph at the iteration i constrained with vision-based landmarks and all the onboard sensors. While circles stand for constraints related to the robot proprioceptive sensors, squares are employed for landmark-based constraints. XYZ-Y represents the constraint on x, y, and z translation and yaw rotation, RP represents the constraint on pitch and roll rotations, Z represents the constraint on z translation and XY_{GPS} represents the constraint on x and y translation thanks to the GPS and LM represents the constraint between the vehicle pose and a landmark.

represented as $\mathcal{Z}$, and $\mathcal{G}$ are respectively the ones for which DS, and GPS measurements are available, and $\mathcal{LM}$ is the set of nodes from which a landmark is seen. Finally, with the aim of anchoring the state evolution to a global reference system, $T_{x_{prior}}$ is employed as the prior constraint on the first pose T_{x_0}. Regarding the landmark measurement generation process, the ATR findings were used for updating the graph and constraining the vehicle poses to the detected landmark positions. This ATR system uses (CNNs) to identify the OPIs, providing the predicted object bounding boxes with an associated confidence level. The center of the bounding box has been employed to compute $\boldsymbol{p}_{ij}$ and the confidence has been used to evaluate Σ_{lm_i}. For further details on the CNN-based ATR system used, the reader can refer to Zacchini et al. (2020).

The proposed Semantic SLAM framework is described in Fig. 3 and is summarized in Algorithm 1. The algorithm begins with the initialization step, where a prior constraint is added to the AUV's pose T_{x_0} using available GPS, IMU, and DS data. It then proceeds through a loop over time step i, where it continually updates the pose graph. At each iteration, the system attempts to detect semantic landmarks using ATR on optical images. If a detection z_{ATR} is obtained, the algorithm checks whether it corresponds to a new landmark or an already-known one. While for new landmarks, a new map element is added, for known landmarks, loop closure is performed to refine the map and correct drift. The updated constraints are incorporated into the graph, which is then optimized to compute the best estimate. The updated AUV pose T_{x_i} and any relevant landmark positions $\boldsymbol{P}_{x_j}$ are then retrieved from the optimized graph and returned. This process enables consistent and accurate localization and semantic mapping through continuous refinement and incorporation of both new observations and loop closures.

The Georgia Tech Smoothing And Mapping (GTSAM) library has been employed as the back-end for the localization and mapping solution. The latest version of GTSAM, known as iSAM2, updates only the small subset of variables that are influenced by a new measurement, leading to an incremental smoothing and mapping strategy. During each iteration, the latest acquired measurements and their associated covariances are included in the graph, thereby reducing the computational load of the estimation process and providing a balance between accuracy and efficiency. More details can be found in Kaess et al. (2008, 2012), Dellaert (2012).

4.2. Semantically annotated occupancy mapping

The OctoMap framework (Hornung et al., 2013), which serves as the foundation for this study, has been extended to implement the semantic occupancy mapping strategy. In its original form, OctoMap represents

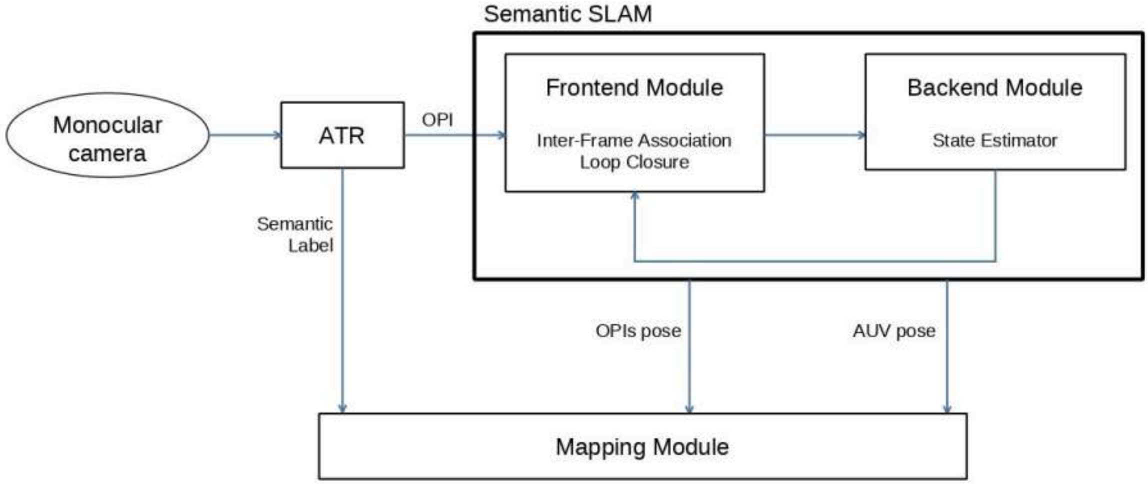

Fig. 3. Overview of the proposed Semantic SLAM framework.

Algorithm 1: Pose-graph semantic SLAM algorithm.

INPUT: Graph $\mathcal{G}^*_{i-1}$, Sensor measurements $m(\cdot)$
OUTPUT: Optimized AUV pose $T^*_{x_i}$, Optimized landmark pose $\boldsymbol{P}^*_{x_j}$
`*/ Initialization /*`
Add prior constraint to AUV pose T_{x_0} with GPS, IMU and DS;
for $i = 1,2,\ldots,N$ **do**
  `*/ Graph update /*`
  $z_{ATR} \leftarrow$ **performATR**( );
  if z_{ATR} *is not* $Null$ **then**
    if ***isNewLandmark***(z_{ATR}) **then**
      $m_{LM}(\cdot) \leftarrow$ **addNewLandmark**(z_{ATR});
    else
      $m_{LM}(\cdot) \leftarrow$ **performLoopClosure**(z_{ATR});
  $\mathcal{G}_t \leftarrow$ **addConstraints**$(\mathcal{G}^*_{i-1}, m(\cdot), m_{LM}(\cdot))$;
  addInitialEstimate$(\mathcal{G}_i)$;
  $\mathcal{G}^*_i \leftarrow$ **optimizeGraph**$(\mathcal{G}_i)$;
  $T^*_{x_i}, \boldsymbol{P}^*_{x_j} \leftarrow$ **retrieveState**$(\mathcal{G}^*_i)$;
  RETURN $T^*_{x_i}, \boldsymbol{P}^*_{x_j}$

free and occupied regions through nodes (commonly known as voxels), each of which corresponds to a cubic volume of space. It uses an octree data structure to efficiently update the likelihood of a voxel being occupied, based on a Bayesian method. Each voxel has a probability P_o indicating the likelihood of being occupied, which is adjusted using sensor measurements. To generate the semantic occupancy map, semantic information have been integrated into the voxels. Specifically, each voxel was extended to also store a probability P_a that represents the accuracy of belonging to an OPI in both the discovery mission and the change detection mission. Regarding the change detection mission, the probability P_e describes the accuracy of each voxel of belonging to a confirmed, new or removed OPI during the change detection mission, (i.e., describes the change status). Therefore, each voxel now contains:

- the probability P_o of being occupied, that allows to categorize the voxel into unexplored, free and occupied;
- the probability P_a representing the accuracy of belonging to an OPI, that allows to categorize the occupied voxel into environment and OPI;
- the probability P_e of being an OPI identified during the change detection mission, that allows to distinguish occupied voxels between environment, confirmed OPI, new OPI and removed OPI.

To achieve this, an update procedure needs to be established. First, it is important to note that, based on the foundational assumptions of the occupancy grid theory, P_a can be updated according to the Bayesian rule described in Hornung et al. (2013). Then, using the log-odds ratio definition, a streamlined and computationally efficient update formulation can be applied:

$$l_a\left(m_i|z^{ATR}_{1:t}\right) = l_a\left(m_i|z^{ATR}_{1:t-1}\right) + l_a\left(m_i|z^{ATR}_t\right) - l_a\left(m_i\right) \tag{7}$$

where z^{ATR} denotes the ATR system outputs during the discovery mission and the change detection mission.
Since the mapping strategy is designed for mapping environments which are a-priori unknown, a non-informative assumption has to be employed for P_a initialization, i.e. $P_a\left(m_i\right) = 0.5$, and consequently, $l_a\left(m_i\right) = 0$.

4.2.1. Discovery mission

The proposed mapping strategy aims to create a semantic model of the environment, designed to identify areas likely to contain OPIs, or to be utilized by a fully autonomous AUV. The proposed Semantic Mapping framework for the discovery mission is described in Fig. 5. The classification of occupied voxels is based on the probability P^d_a, i.e. the probability P_a for the discovery mission. Low values of P^d_a correspond to occupied-environment voxels, while values of P^d_a close to one indicate occupied-OPI voxels. Intermediate values classify the voxels as occupied-uncertain, meaning these voxels have a significant chance of containing an OPI but have not been definitively classified yet. For clarity, Fig. 4 provides a summary of the voxel classification scheme.
Therefore, each voxel is characterized by two probabilities:

- the probability P^d_o of being occupied;
- the probability P^d_a representing the accuracy of belonging to an OPI.

The occupancy of the sea bottom voxels is updated through the mapping solution outlined in Bucci et al. (2023), which relies on an DVL-based reconstruction methodology, while the probability P^d_a is updated using the results from the Semantic SLAM solution.

Each bounding box can be precisely located in relation to the body reference frame thanks to the SLAM algorithm previously described. For each ATR detection, a point cloud is generated by dividing the bounding box into cells and using the center of each cell. To clarify, the center of each cell, is represented as a point in the cloud. Finally, the ISM for updating P_a is determined by assigning to each point the probability of it belonging to an OPI, which is calculated using a Gaussian distribution as shown below:

$$P^d_a\left(m_i|z^{ATR}_t\right) \approx p_c e^{-\frac{1}{2}\left(\frac{\left|m_i-\pi_i(T_{x_i}\boldsymbol{P}_j)\right|}{\sigma_c}\right)^2}, \tag{8}$$

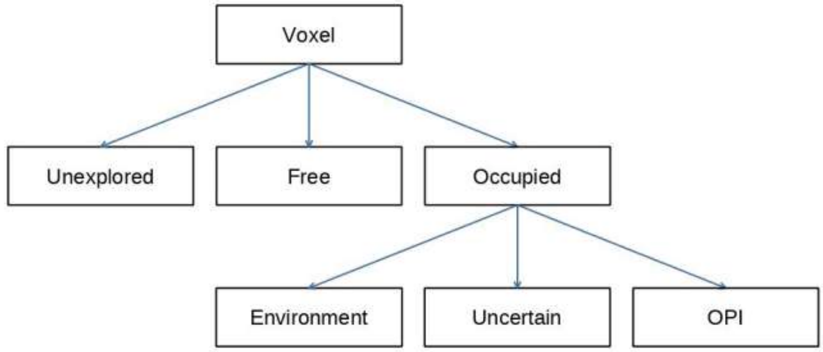

Fig. 4. The voxel classification policy employed to generate the semantically annotated map during the discovery mission enhances the standard OctoMap label map by assigning a probability to each occupied voxel, indicating the probability of it belonging to an OPI. This probability is updated based on the outputs coming from the CNN-based ATR system.

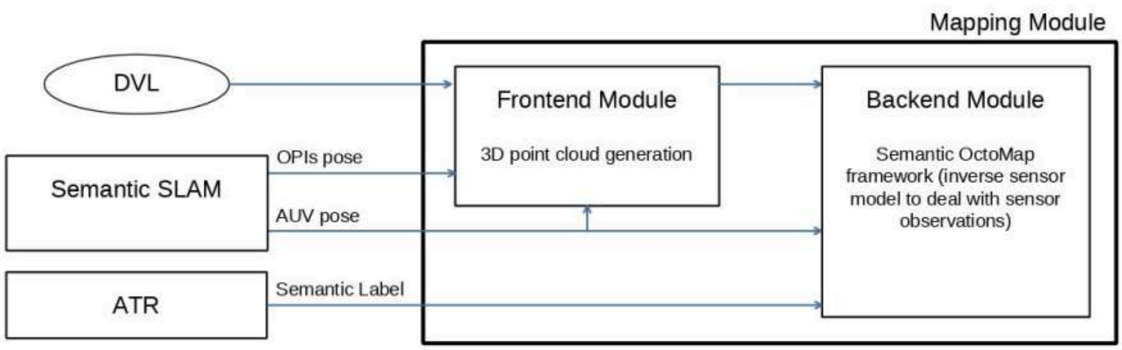

Fig. 5. Overview of the proposed semantic mapping module for the discovery mission.

where $\pi_i(T_{x_i}\boldsymbol{P}_j)$ denotes the center of the detected OPI's projected on the image plane, p_c the computed confidence level, and σ_c the constant variance.

It is important to note that the noise and the reduced visibility in the optical images make OPI recognition a challenging task. During an inspection survey, the AUV may observe the same OPI from various angles, and these variations can lead to missed or incorrect detections, but thanks to the SLAM algorithm, which includes a global loop closure strategy, these wrong associations can be avoided. Despite this, for each image where the ATR system does not detect targets, the voxels are updated with a probability $P_a^d\left(m_i|z_t^{ATR}\right)$ equal to 0.45, in order to reduce the probability of these voxels to belong to an OPI. This approach was implemented to mitigate the impact of incorrect detections on the build semantic map and to improve the precision of the OPI localization.

4.2.2. Change detection mission

The proposed mapping strategy aims to develop a semantic model of the environment status to identify the areas where some changes have been occurred to be used by the AUV to optimize its second inspection. The proposed Semantic Mapping framework for the change detection mission is described in Fig. 7. The classification of occupied voxels relies on the probability P_e^{cd}. Low P_e^{cd} values indicate occupied-environment-seabottom voxels, while values closer to 1 suggest occupied voxels with one of the possible label. A detailed summary of the voxel classification can be found in Fig. 6. To obtain this classification, each voxel must also necessarily be categorized with the labels used during the discovery mission (i.e., unexplored, free, occupied-environment, occupied-uncertain, or occupied-OPI). Therefore, each voxel is characterized by three probabilities:

- the probability P_o^{cd} of being occupied;
- the probability P_a^{cd} representing the accuracy of belonging to an OPI during the change detection mission;
- the probability P_e^{cd} representing the accuracy of its status (i.e., of being a new (N), confirmed (C) or removed (R) OPI during the change detection mission).

As in the discovery mission, the occupancy of voxels is updated through the mapping approach presented in Bucci et al. (2023), which uses a DVL-based reconstruction method, while the probability P_e^{cd} is updated via the comparison of the saved map with the outputs of the Semantic SLAM technique applied during the change detection mission. For each detected OPI, a point cloud is generated as in the discovery mission. The probability P_e^{cd} is assigned to each point, reflecting the likelihood of each voxel being part of a previously detected OPI, and it is computed after an evaluation of the status of each voxel. To quantitatively estimate if a voxel belong to an OPI detected during the discovery mission or to a new OPI, all the voxels with $P_a^{cd} > 0.7$ detected during the discovery mission have been employed to build clusters. The centroid of each cluster has been considered as the estimated position of each OPI and compared with the output of the semantic SLAM algorithm to evaluate the status of each voxel with potential OPI. As comparison metrics, the distance along the North and East directions from the cluster's centroid to the voxel under consideration has been calculated.

To provide a statistical evaluation of an object to be a confirmed object, the following considerations have been done. When two 3D objects composed of voxels are compared and each voxel has an associated probability P_a^d of belonging to an OPI detected during the discovery mission and P_a^{cd} of belonging to an OPI detected during the change detection mission, it is possible to compute the probability that these two objects are the same object. In order to model the likelihood that the entire set of points in one object has the same overall occupation pattern as the other object, it is possible to compute the likelihood of matching occupation

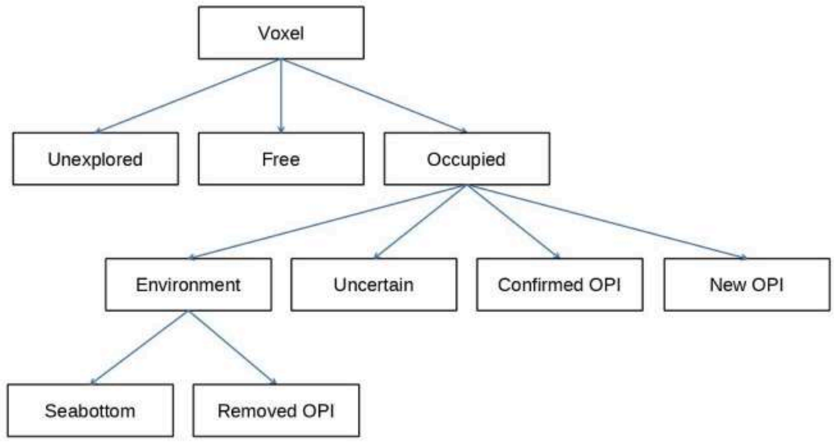

Fig. 6. The voxel classification policy used to create the semantically annotated map during the change detection mission. In this case, each occupied voxel is updated with the probability of belonging to a new, confirmed or removed OPI with respect to the map obtained from the discovery mission. This probability is obtained comparing the outputs of a CNN-based ATR system with the results coming from the discovery mission.

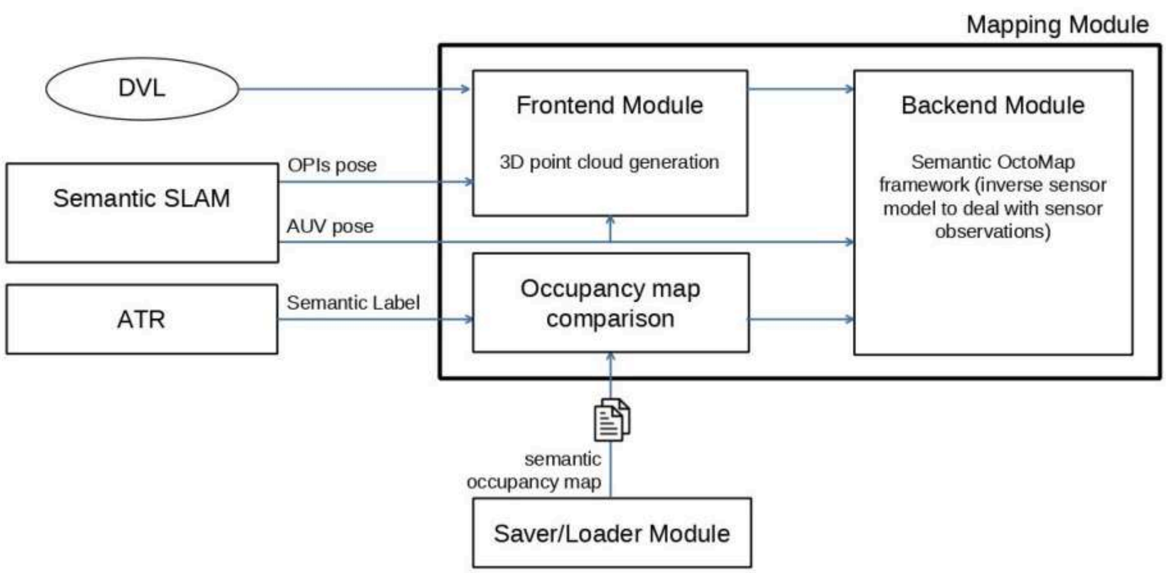

Fig. 7. Overview of the proposed semantic mapping module for the change detection mission.

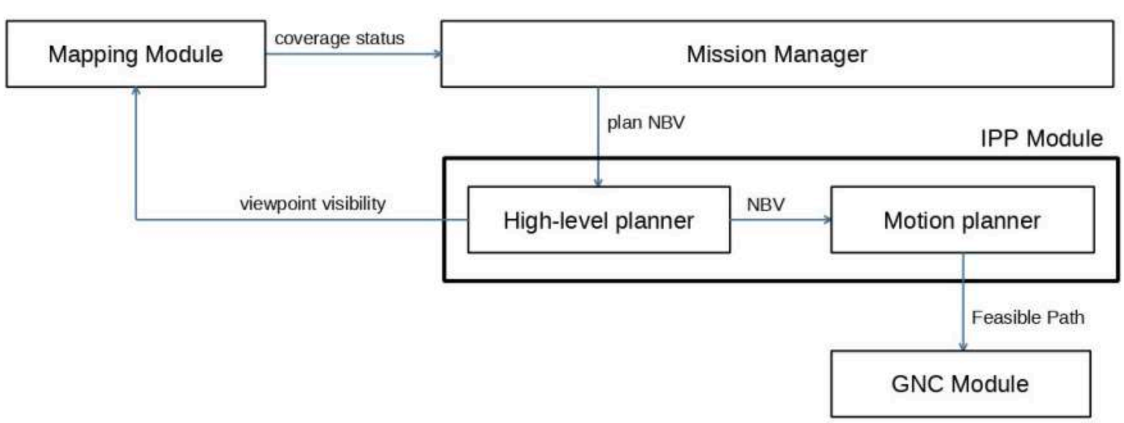

Fig. 8. Overview of the proposed IPP module.

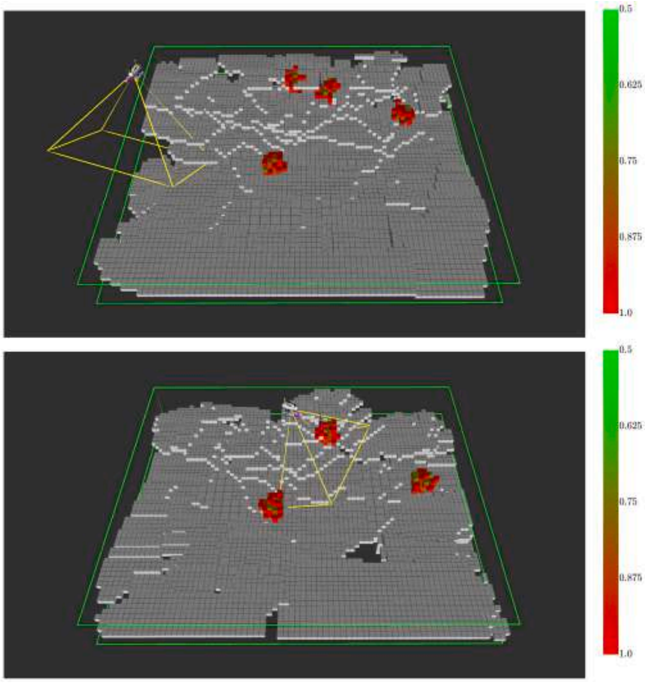

Fig. 9. Semantically annotated occupancy maps performed during a discovery mission (on the Top) and during a change detection mission (on the Bottom). The color indicates the accuracy of the voxel to belong to an OPI.

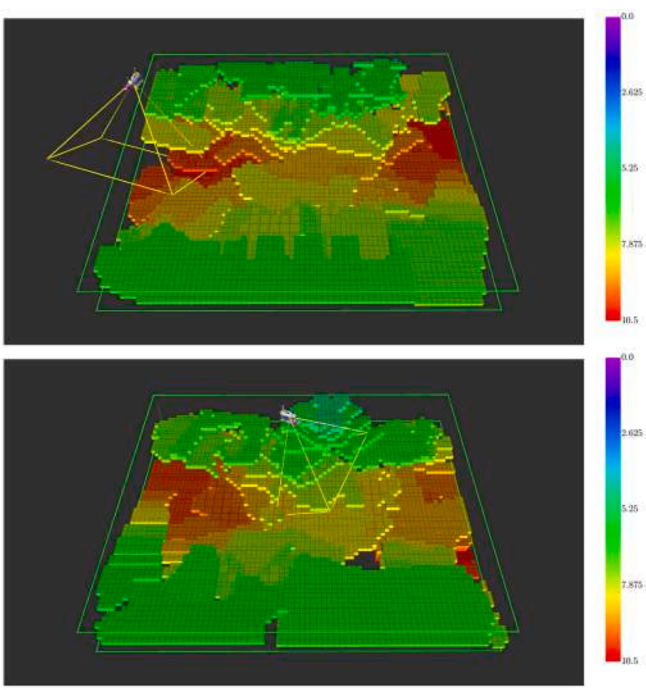

Fig. 10. Occupancy maps performed during a discovery mission (on the Top) and during a change detection mission (on the Bottom). The color indicates the depth of each point of the seabottom.

patterns across the entire object based on the voxel-wise probabilities. In this case, the idea would be to compute the probability distribution over all occupation patterns for both objects and compare them. For each corresponding voxel pair, the total probability of matching the occupation state can be computed. The combined probability of being a confirmed object for each voxel would be:

$$p(C_i) = P_a^d P_a^{cd} + \left(1 - P_a^d\right)\left(1 - P_a^{cd}\right) - P_a^d\left(1 - P_a^{cd}\right) + \left(1 - P_a^d\right)P_a^{cd} \tag{9}$$

where $P_a^d P_a^{cd}$, $(1 - P_a^d)(1 - P_a^{cd})$ and $P_a^d(1 - P_a^{cd}) + (1 - P_a^d)P_a^{cd}$ are respectively the probabilities that both points are occupied, both points are not occupied, and one is occupied and the other is not.
The total probability is updated through the log-odds, i.e.

$$l(C) = \sum_{i=1}^{n} l\left(C_i\right) = \sum_{i=1}^{n} \log \frac{p\left(C_i\right)}{1 - p\left(C_i\right)} \tag{10}$$

The probability $p(C)$ can be retrieved as

$$p(C) = \frac{1}{1 + e^{-l(C)}}. \tag{11}$$

This represents the probability that, for each corresponding point, the two objects have the same occupation state.
To provide a statistical evaluation of an object to be a removed object, the following considerations have been done. Assuming that the voxels are independent, then the accuracy of the object being removed can be related the product of the probabilities of the points containing an object during the discovery mission. Given a set of voxels with a probability P_a^d during the discovery mission, the probability that each voxel is not containing an OPI is $1 - P_a^d$. The probability of the entire object being removed during the change detection mission can be computed as the product of the probabilities of each point being not an OPI. Consequently, the probability of the object being removed is the complement of this probability.

$$p(R_i) = 1 - P_a^d \tag{12}$$

Retrieving the log-odds of $p(R_i)$, it is possible to determine the accuracy of a removed object:

$$l(R) = 1 - \sum_{i=1}^{n} l(R_i) \tag{13}$$

Reverting from the log-odds, the probability $p(R)$ can be found.
Finally, to provide a statistical evaluation of an object to be a new object, similar considerations to the removed OPI case have been done. The probability of the entire object being a new object appeared between the discovery and the change detection mission can be computed as the product of the probabilities of each point being an OPI during the change detection mission. The probability of the object being added is equal to this probability.

$$P(N_i) = P_e^{cd} \tag{14}$$

As for the removed OPIs, thanks to the log-odds of $p(N_i)$, the accuracy of a detected new OPI can be defined as:

$$l(N) = \sum_{i=1}^{n} l(N_i) \tag{15}$$

From $l(N)$, the accuracy $p(N)$ can be easily determined.
As for the discovery mission, in cases where the ATR system does not detect any targets, the voxels are updated with a default probability $P_a^{cd} = 0.45$ and considered as seabottom, reducing the likelihood of these voxels to be associated to a different status.

4.3. Coverage solution

An IPP algorithm was developed to guarantee the coverage of the entire area to be inspected using specified sensors, which, in this study, is an optical camera for the semantic part and an acoustic sensor, as the DVL, for the occupancy term. Specifically, the IPP algorithm is responsible for calculating and adapting the inspection survey in real-time, using the computed semantic map as feedback to maintain adequate coverage levels and accurately achieve the desired tasks (i.e., the OPIs inspection and detection during the discovery mission and the modification search during the change detection mission). To achieve this, a RIG was

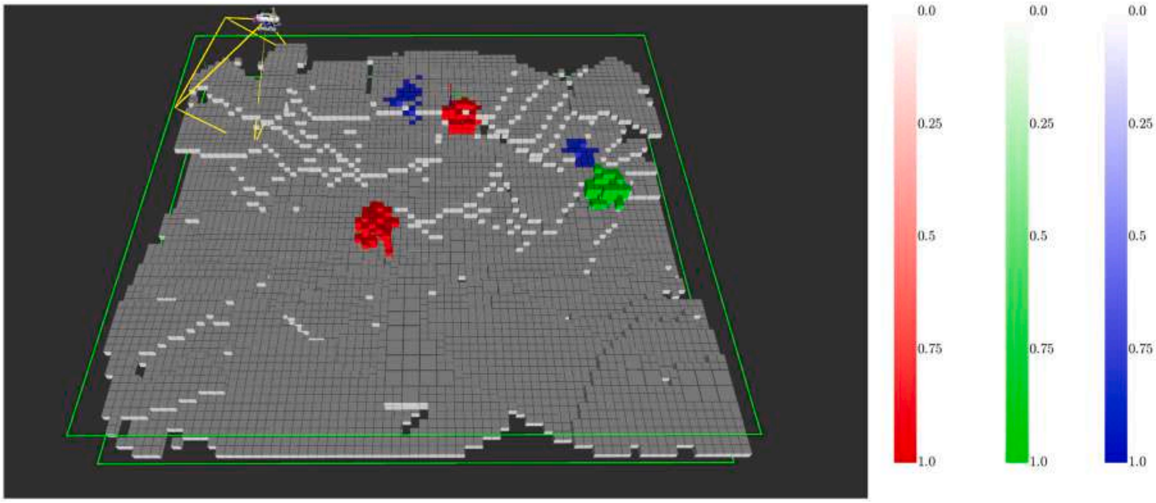

Fig. 11. Semantically annotated change detection map performed during the change detection mission. The color indicates the accuracy of the voxel to be a confirmed (in red), removed (in blue) or new (in green) OPI.

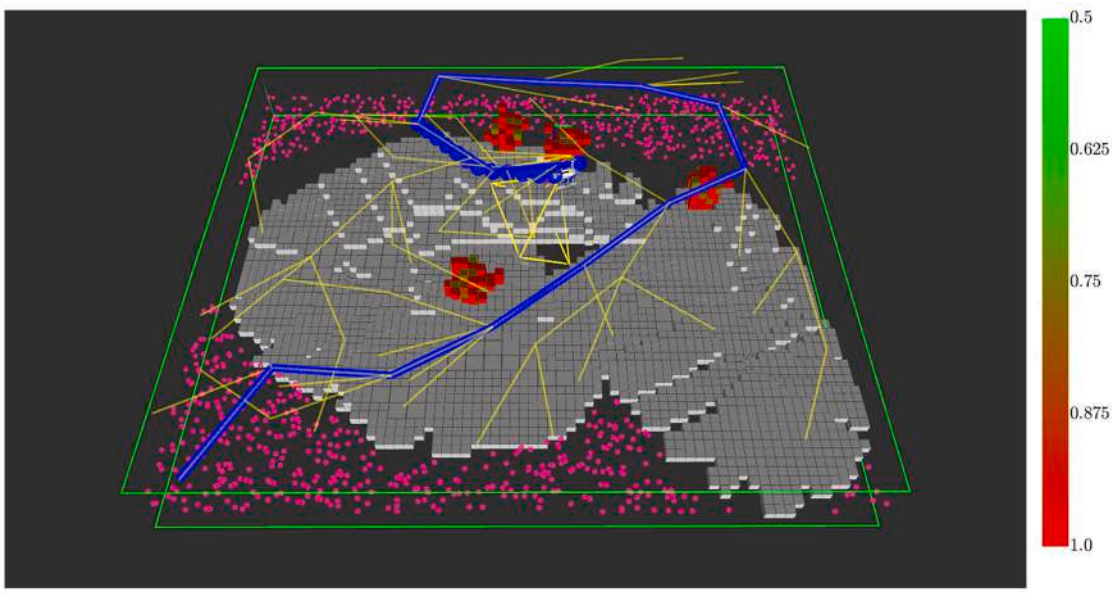

Fig. 12. The snapshot illustrates the impact of the proposed gain function. The semantic map is utilized in real-time by the AUV, with information about the OPIs being leveraged for planning the survey. By incorporating the target uncertainty-aware gain function, the planner chose a path (in blue) that directs the AUV towards the coverage task while also enabling the re-observation of the areas with the highest uncertainty. For clarity, the green lines outline the inspection area, and the generated view planning tree is shown in yellow.

designed, utilizing the standard RRT expansion and requiring specific routines (Zacchini et al., 2023). More specifically, the RIG algorithm iteratively solves the following optimization problem online:

$$\begin{aligned} \max_{\mathcal{B}} \quad & \mathcal{G}_{\mathcal{B}} \\ \text{s.t.} \quad & \mathcal{B} = \{\xi_i\}_{i=0}^{N_p} \\ & \xi_i = f\left(\xi_{i-1}\right) \\ & \xi \in \mathcal{W} \end{aligned} \tag{16}$$

where $\mathcal{G}_{\mathcal{B}}$ is the objective function, $\mathcal{B}$ is a branch that has to be evaluated and is constituted of a sequence of N_p possible AUV configurations $\{\xi_i\}$, ξ_0 denotes the AUV configuration at the current timestamp, and $\mathcal{W}$ is the space of the admissible configurations. By considering a survey with constant altitude, the configurations that can be reached by the AUV can be described as a 2D position in the North, East, Down (NED) frame (Fossen, 1994), i.e. $\begin{bmatrix} X & Y \end{bmatrix}$, and heading ψ. Consequently, the AUV state can be represented with the vector $\xi = \begin{bmatrix} X & Y & \psi \end{bmatrix}$. The state evolution function $f(\cdot)$ provides the relation between two consecutive vehicle configurations ξ_i and ξ_{i-1}, that, as reported in Section 3.3, is described by using the Dubins curves. $\mathcal{G}_{\mathcal{B}}$ is the function to be optimized and it is different for the discovery and the change detection mission. The IPP Module framework is reported in Fig. 8.

Algorithm 2 outlines the RRT-based IG algorithm used in the IPP module. It begins with the robot's current configuration ξ_t and optionally the best path from the previous iteration $\mathcal{P}_{t-1}$. A new tree is initialized with ξ_t, and if a prior path exists, its valid remaining segments are added to the tree using the updated map. The core of the algorithm lies in its tree expansion loop, which runs within a fixed computational time. This expansion is split into two parallel phases:

- Node generation: a new node ξ_n is sampled and its visibility is computed.
- Node rewiring: the previous node ξ_p is optimized by evaluating better parent-child connections to maximize information gain.

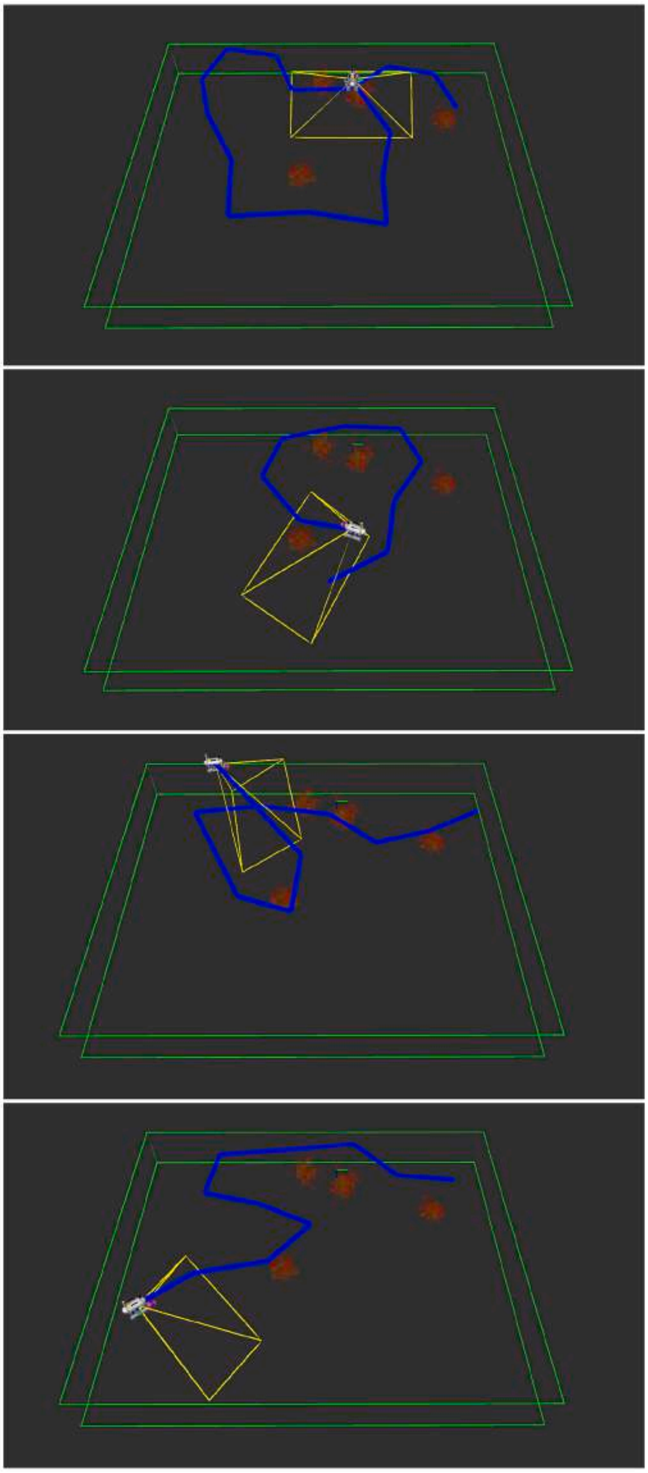

Fig. 13. These snapshots illustrate the impact of the proposed gain function. The previously identified OPIs (shown in transparent red-to-green colors) are used to inform the planning of the initial phase of the survey during the change detection mission. By utilizing the uncertainty-aware gain function, the planner chose a path (represented in blue) that guides the AUV towards the coverage task while also ensuring that the most uncertain regions are revisited. For clarity, the green lines outline the inspection area.

The **generateNode**( ) function handles the creation of new nodes for tree expansion. It starts by sampling a configuration ξ_s and extending the nearest existing node ξ_n toward it to generate a new node ξ_{n+1} (Algorithm 3). This process uses a uniform sampling strategy, which results in a Voronoi-biased expansion, promoting efficient and broad exploration of the workspace. If the motion from ξ_n to ξ_{n+1} is valid (i.e., safe and feasible), the node's visibility-the set of voxels observable from that pose-is computed. This visibility evaluation is essential for the rewiring step and is made efficient by the submodular nature of the IPP objective function, meaning the value of information is dependent only on the robot's current pose due to spatial measurement correlations.

The algorithm presents a novel informed expansion strategy that guides the tree toward unexplored areas and that is employed in **sampleNewConfiguration**( ). The proposed method learns the distribution of acquired data-interpreted here as the discovered map-online and leverages this information to sample new configurations in regions that have not yet been explored. It integrates KDE during the learning phase and employs a rejection sampling algorithm to bias the tree expansion.

The rewiring routine is described in Function **rewireNode**( ) (see Algorithm 4). Given a node ξ, the algorithm first retrieves its set of nearest neighbors. It then performs a two-step optimization: first, it searches among the neighbors for the best parent, i.e., the node that maximizes the information gain if selected as the parent of ξ. Second, it evaluates whether ξ can improve the gain of other nodes in the neighborhood by becoming their new parent, potentially enhancing the expected information gain of entire subtrees. To this end, once a node has been associated to a path its gain is computed by considering the voxels already seen along the path and employing the rewiring functions that will be described in the following for both the discovery and the change detection missions.

Algorithm 2: IPP algorithm.

INPUT: AUV configuration ξ_t, Previous best path $\mathcal{P}_{t-1}$
OUTPUT: Next best view configuration ξ_{t+1}
```
*/ Initialization /*
```
Initialize a new tree $\mathcal{T}$ with $\xi_0 = \xi_t$;
Initialize $\xi_p = Null$;
if $\mathcal{P}_{t-1}$ *is not* $Null$ **then**
  addPath$(\mathcal{P}_{t-1}, \mathcal{T})$;
  updatePathGain$(\mathcal{P}_{t-1})$;
```
*/ New node generation and evaluation /*
```
while $\Delta t < \delta$ **do**
  if $\xi_p = Null$ **then**
    $\xi_n \leftarrow$ **generateNode**( );
  else
    $\xi_n \leftarrow$ **generateNode**( );
    rewireNode(ξ_p);
  if ***nodeIsValid***(ξ_p) **then**
    addNode$(\xi_p, \mathcal{T})$;
  $\xi_p \leftarrow \xi_n$;
```
*/ Best path and NBV update /*
```
$\mathcal{P}_t \leftarrow$ **getBestPath**$(\mathcal{T})$;
$\xi_{t+1} \leftarrow$ **getNBV**$(\mathcal{P}_t)$;
Delete $\mathcal{T}$;
RETURN ξ_{t+1}

4.3.1. Discovery mission

By employing an occupancy mapping strategy, like the one previously outlined in Section 4.2.1, the objective function $\mathcal{G}_{\mathcal{B}}$ can be evaluated using a straightforward ray-casting procedure to highlight all the illumined voxels from each pose that can be assumed by the AUV. Given all the configurations ξ in $\mathcal{B}$, the available occupancy information can

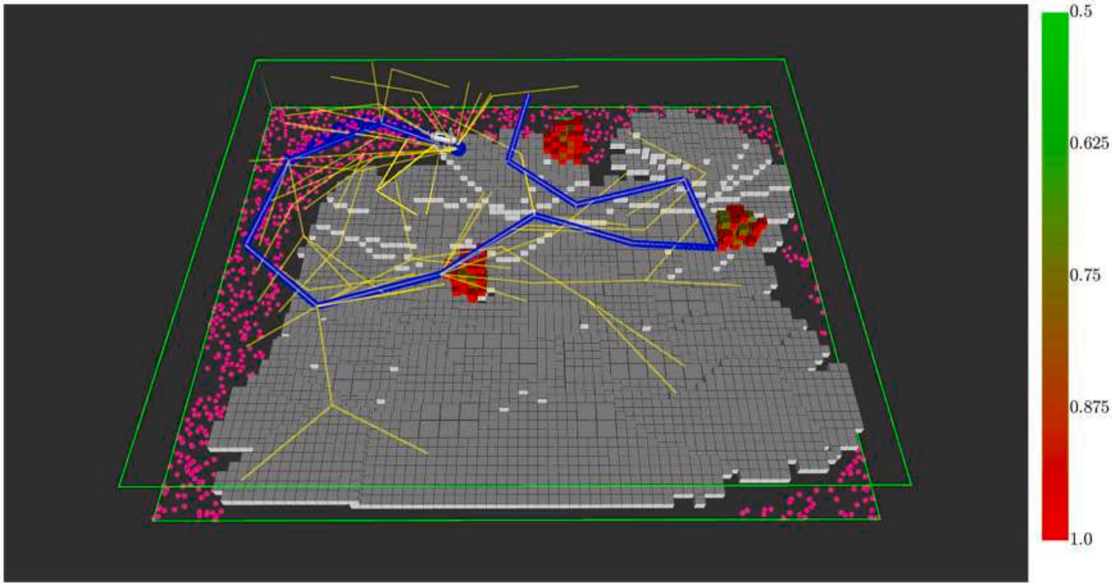

Fig. 14. The snapshot illustrates the impact of the proposed gain function. The AUV uses the updated semantic map in real-time, leveraging the knowledge of the OPIs to plan the survey for the second phase of the change detection mission. By applying the target uncertainty-aware gain function, the planner selects a path (shown in blue) that guides the AUV towards the coverage task while also revisiting the most uncertain regions. For clarity, the green lines outline the inspection area, and the generated view planning tree is highlighted in yellow.

Fig. 15. FeelHippo AUV before an on-field underwater mission.

Algorithm 3: Function generateNode( ).

$\xi_s \leftarrow$ **sampleNewConfiguration**( );
$\xi_n \leftarrow$ **getNearest**$(\xi_s, \mathcal{T})$;
$\xi_{n+1} \leftarrow$ **randomPropagation**(ξ_s, ξ_n);
if ***isMotionValid***(ξ_n, ξ_{n+1}) **then**
  computeVisibility(ξ_{n+1});
  setNodeValidity(ξ_{n+1}, True);
  RETURN ξ_{n+1}
else
  generateNode();

Fig. 16. Examples of the detected yellow buoy.

be estimated as follows:

$$\mathcal{G}_{\mathcal{B}} = \sum_{\forall \xi \in \mathcal{B}} \sum_{\forall m \in \mathcal{M}_{\xi}} I(x) \tag{17}$$

where $I(x) = -P_o(x)\ln P_o(x) - \bar{P}_o(x)\ln \bar{P}_o(x)$ with $\bar{P}_o(x) = 1 - P_o(x)$ as the complementary probability. $I(x)$ represents the entropy of the voxel m, which belongs to $\mathcal{M}_{\xi}$, i.e. the set of the voxels which are visible from the configuration ξ. Consequently, the value of $\mathcal{G}_{\mathcal{B}}$ can be evaluated summing all configurations that constitute the considered branch $\mathcal{B}$.
However, by focusing solely on the information related to the exploration of unknown environment as in Eq. (17), the planning algorithm aims to compute the shortest path to cover the area. It does not take in consideration the OPI search or the active goal of inspecting the surrounding environment trough a semantic map. The object uncertainty has to ba taken into account by the RIG planner, and consequently the cost function to be optimized has to be adjusted in order to minimize the number of voxels labeled as occupied-uncertain (and, consequently, to maximize the voxels classified as free, or occupied with environment

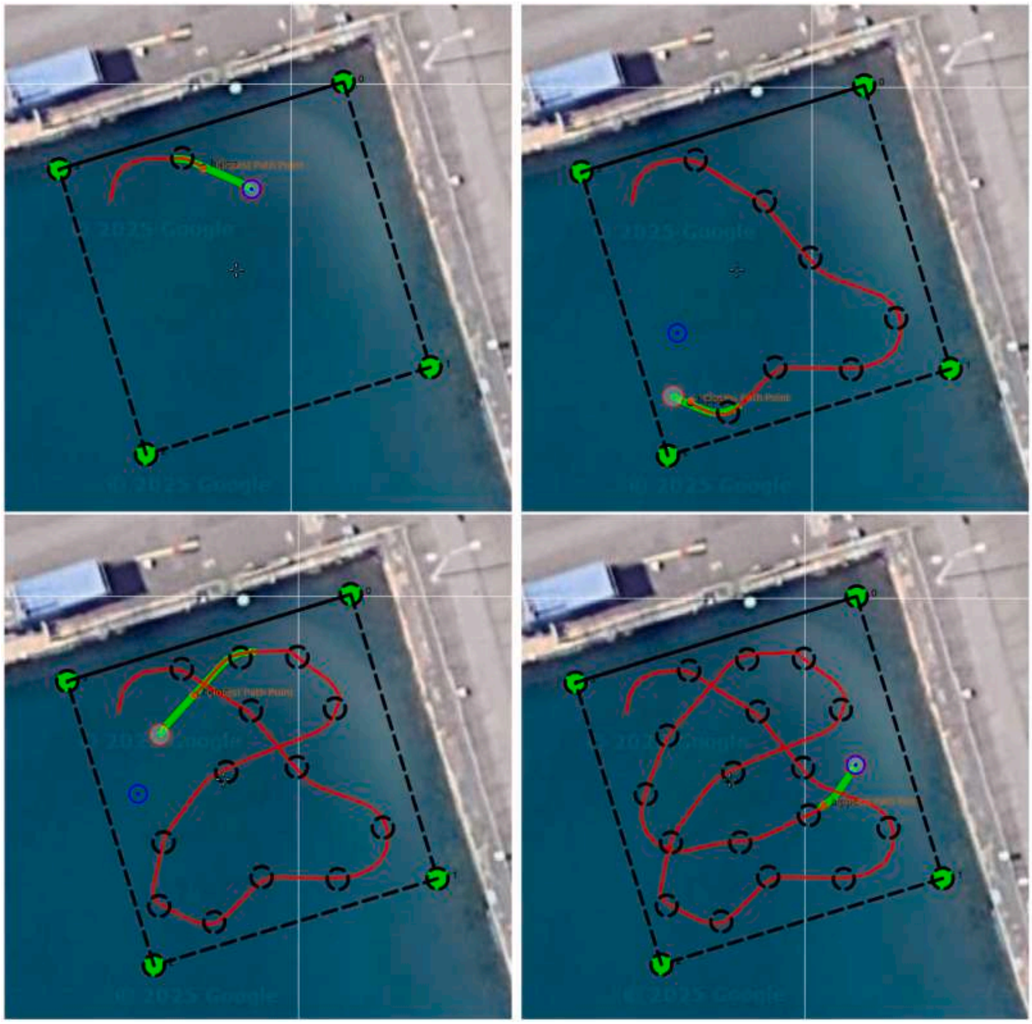

Fig. 17. Various snapshots from one of the autonomous discovery surveys performed by the FeelHippo AUV during the experimental campaign in La Spezia, Italy, are presented. Utilizing the developed coverage framework, the vehicle successfully surveyed the seabed within the target area, outlined by the four green points and four black dashed lines, with the objective of locating the OPIs. The route guiding the AUV to the NBV is shown in green, while the estimated path traveled by the AUV is marked in red.

Algorithm 4: Function rewireNode(ξ).

$g_b \leftarrow$ **getGain**(ξ);
$\mathcal{V} \leftarrow$ **getNeighbors**($\xi, \mathcal{T}$);
for $\forall \xi_p \in \mathcal{V}$ **do**
  if ***isMotionValid***(ξ_p, ξ) **then**
    $g \leftarrow$ **computeGain**(ξ_p, ξ);
    if $g < g_b$ **then**
      setParent(ξ, ξ_p);
      setGain(ξ, g);
      $g_b \leftarrow g$;
for $\forall \xi_c \in \mathcal{V}$ **do**
  if ***isMotionValid***(ξ, ξ_c) **then**
    $g_c \leftarrow$ **getGain**(ξ_c);
    $g \leftarrow$ **computeGain**(ξ, ξ_c);
    if $g < g_c$ **then**
      setParent(ξ_c, ξ);
      setGain(ξ_c, g);
      updateChildrenGain(ξ_c);
RETURN

or OPI label). It is important to highlight that the voxel uncertainty regarding the presence of an OPI can be represented by its entropy:

$$I_a^d(x) = -P_a^d(x)\ln P_a^d(x) - \bar{P}_a^d(x)\ln \bar{P}_a^d(x) \tag{18}$$

where $\bar{P}_a^d(x) = 1 - P_a^d(x)$.
Thus, the voxel-based OPI entropy cost evaluated along a branch $\mathcal{B}$ can be computed as:

$$\mathcal{G}_{\mathcal{B}_a}^d = \sum_{\forall m \in \mathcal{M}_{\mathcal{B}}} I_a^d(x). \tag{19}$$

The exploration-oriented function $\mathcal{G}_{\mathcal{B}}$ previously introduced in Eq. (17) needs to be combined with this target uncertainty-based cost function to define a branch gain model that take under consideration both the tasks to be accomplished. In the context of this research, $\mathcal{G}_{\mathcal{B}_a}^d$ was used to encourage branches that guide the AUV towards unexplored areas while reducing the target localization uncertainty. Therefore, the complete branch gain has been defined as:

$$\mathcal{L}_{\mathcal{B}} = \mathcal{G}_{\mathcal{B}} e^{\lambda_a \mathcal{G}_{\mathcal{B}_a}^d} \tag{20}$$

where $\lambda_a > 0 \in \mathbb{R}$ is a tuning parameter that permits to balance between the tasks of inspecting the area to be explored and detecting the OPIs, in order to reach an optimal trade-off between both these objectives.
Indeed, while $\mathcal{G}_{\mathcal{B}}$ enables the algorithm to choose the path that directs

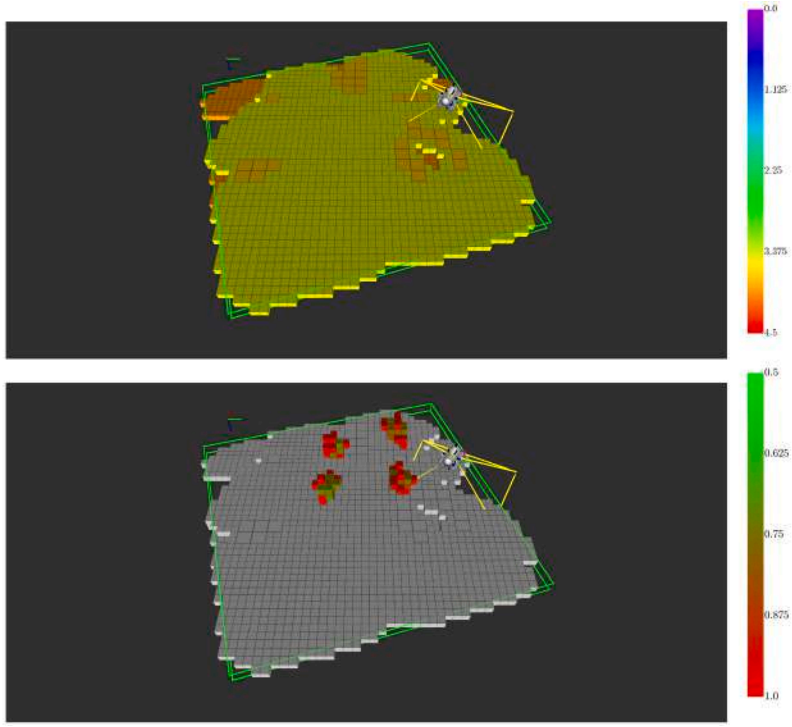

Fig. 18. Occupancy maps performed during a discovery mission. On the Top, the depth map, where the color indicates the depth of each point of the seabottom, on the Bottom, the semantic map, where the color indicates the accuracy of the voxel to belong to an OPI.

the vehicle to non-inspected regions in the shortest time, the cost function $\mathcal{L}_{\mathcal{B}}$ encourages paths that not only explore the area but also reduce the localization uncertainty of the OPIs thanks to the influence of the components proportional to $\mathcal{G}^{d}_{\mathcal{B}_a}$. It is important to note that $\mathcal{L}_{\mathcal{B}}$ may result in longer inspection surveys, as improving the localization of the OPIs might necessitate the vehicle re-observing areas that have already been explored.

4.3.2. Change detection mission

For what concern the change detection mission, the objective function $\mathcal{G}_{\mathcal{B}}$ can be evaluated as in the discovery mission. Indeed, the overall environment has to be entirely mapped for the second time to find the presence of possible changes. As in the discovery mission, the uncertainty related to OPIs along a branch can be estimated as:

$$\mathcal{G}^{cd}_{\mathcal{B}_a} = \sum_{\forall m \in \mathcal{M}_{\mathcal{B}}} I^{cd}_{a}(x). \tag{21}$$

where $I^{cd}_{a}(x) = -P^{cd}_{a}(x) \ln P^{cd}_{a}(x) - \bar{P^{cd}_{a}}(x) \ln \bar{P^{cd}_{a}}(x)$. Furthermore, the voxel accuracy obtained from the discovery mission is employed to computed $\mathcal{G}^{\dagger}_{\mathcal{B}_a}$ to move the vehicle through branches that firstly inspect areas where during the discovery mission OPIs were found. In this case, branches are not privileged through voxel uncertainty, but through their accuracy in containing an OPI. Consequently, $\mathcal{G}^{d}_{\mathcal{B}_a}$ is computed as

$$\mathcal{G}^{\dagger}_{\mathcal{B}_a} = \sum_{\forall m \in \mathcal{M}_{\mathcal{B}}} I^{\dagger}_{a}(x). \tag{22}$$

where $I^{\dagger}_{a}(x) = P^{d}_{a}(x) \ln P^{d}_{a}(x)$. Both these components can be combined with the exploration function $\mathcal{G}_{\mathcal{B}}$ to define a suitable branch gain model. Therefore, the branch gain is formulated as:

$$\mathcal{L}_{\mathcal{B}} = \mathcal{G}_{\mathcal{B}} e^{\lambda_a \left((1-\theta)\mathcal{G}^{\dagger}_{\mathcal{B}_a} + \mathcal{G}^{cd}_{\mathcal{B}_a} \right)} \tag{23}$$

where $\lambda_a > 0 \in \mathbb{R}$ is a tuning parameter that controls the influence of target uncertainty, balancing the trade-off between exploration and OPI localization and $\theta \in [0, 1]$ is the ratio between the inspected area and the total area to be monitored. While $\mathcal{G}_{\mathcal{B}}$ helps the algorithm choose the shortest path to unexplored regions, $\mathcal{L}_{\mathcal{B}}$ encourages paths that firstly inspect areas where OPIs where detected during the discovery mission and, during the second part of the mission, reduce OPI localization uncertainty. As previously, it's important to note that, also in this case, the function $\mathcal{L}_{\mathcal{B}}$ might lead to longer inspection surveys, as improving OPI localization may require the vehicle to revisit areas that have already been explored.

5. Validation

The proposed inspection strategy has been initially validated using simulations. The framework underwent validation through a comprehensive comparison with realistic simulations conducted using the Unmanned Underwater Vehicle Simulator (UUV Simulator), where a digital twin of FeelHippo AUV, chosen as the test platform and detailed in Section 6, has been employed. Since sea trials are both time-consuming and costly, this simulation-based validation played a critical role. The results presented in this Section aimed at validating and fine-tuning the developed framework. During the experimental campaign, the coverage solution has been tested using the parameters derived from this validation process.

To assess the mapping performance within the simulation environment, both the DVL and the camera were replicated using the simulator's capabilities. The simulations were built on the dynamic model of the FeelHippo AUV, which was implemented in the UUV Simulator including all the available onboard sensors. In particular, the four DVL beams were simulated by employing the geometry of the sensor and by adding noise to the measured values, resulting in errors in the velocity measurements and in the beam 3D positioning. Similarly, the camera was modeled with noise affecting the pixel positions in the captured images, which influences the estimation of both the vehicle and landmark positions. The data used for modeling all the other sensors were chosen based on the specifications of the devices installed on the FeelHippo AUV.

The IG framework and the mapping solution previously described were used in conjunction with the ATR strategy outlined in Zacchini et al.

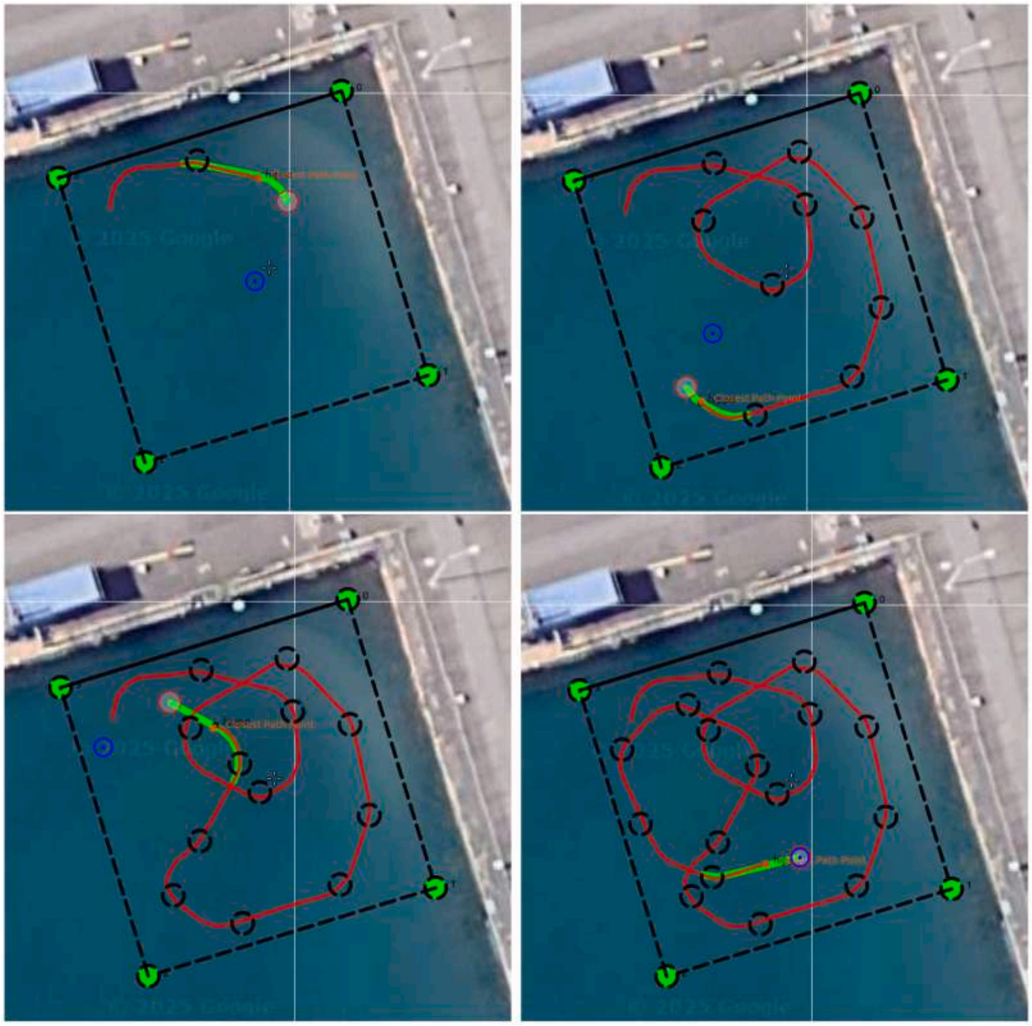

Fig. 19. Various snapshots from one of the autonomous change detection surveys carried out by the FeelHippo AUV during the experimental campaign in La Spezia, Italy, are shown. Using the developed coverage framework, the vehicle successfully examined the seabed within the target area, marked by four green points and four black dashed lines, with the initial goal of locating the previously identified OPIs. The route guiding the AUV to the NBV is shown in green, while the estimated path followed by the AUV is indicated in red.

(2020). Specifically, the Single Shot MultiBox Detector (SSD) Mobilenet V2 network architecture (Liu et al., 2016) was chosen for detecting the OPIs in the optical images. To set up a suitable simulation environment, four objects, resembling spheres with radius of 0.5 m were placed both on the sea floor and floating. To validate and assess the performance of the proposed methodology, several inspections were carried out and four OPIs positioned at known locations, with their coordinates in the NED frame listed in Table 1. The autonomous missions were conducted to inspect an area of approximately 30×36 m with a constant altitude of 4 meters above the seabed.

The semantic mapping module was designed to generate an occupancy grid map of the covered area, which serves as real-time feedback for the high-level planning module. As previously mentioned, the motion planner module accounted for the AUV kinematic constraints using Dubins curves with a turning radius of 3 meters. The high-level planner had 2 s to determine the NBV. It employed the algorithm outlined in Section 4.3 to grow a random tree starting from the AUV's initial position. The branch that was expected to gather the most information was chosen, and the first node became the NBV. Subsequently, the FeelHippo AUV followed the path generated by the motion planner. Once the NBV was reached, the high-level planner built a new random tree, using the previous best solution as the starting point.

Table 1
Simulation setup: OPI positions in the NED reference frame during the discovery and the change detection missions.

	Discovery mission	Change detection mission
OPI 1	$\begin{bmatrix} 4 & 1 & 5.5 \end{bmatrix}$	$\begin{bmatrix} 4 & 1 & 5.5 \end{bmatrix}$
OPI 2	$\begin{bmatrix} 2 & 10 & 7 \end{bmatrix}$	—
OPI 3	$\begin{bmatrix} 6 & -3 & 5.5 \end{bmatrix}$	—
OPI 4	$\begin{bmatrix} -6 & -5 & 6.5 \end{bmatrix}$	$\begin{bmatrix} -6 & -5 & 6.5 \end{bmatrix}$
OPI 5	—	$\begin{bmatrix} -3 & 11 & 6 \end{bmatrix}$

5.1. Navigation and mapping performance

To quantitatively evaluate the accuracy of OPI localization, Euclidean clusters were extracted from the voxels with $P_a > 0.7$, when the AUV reached the 95% coverage threshold. From each cluster the centroid has been extracted and it has been taken as the estimated position of the closest OPI and compared with the output of the semantic SLAM algorithm and with the ground truth of the simulation. At the end of the survey, these regions were used to identify the areas most likely to contain the OPIs. The results of one simulation are summarized in Table 2.

In terms of localization accuracy, all the OPI positions were estimated with a maximum error of about 1 meter. Additionally, the

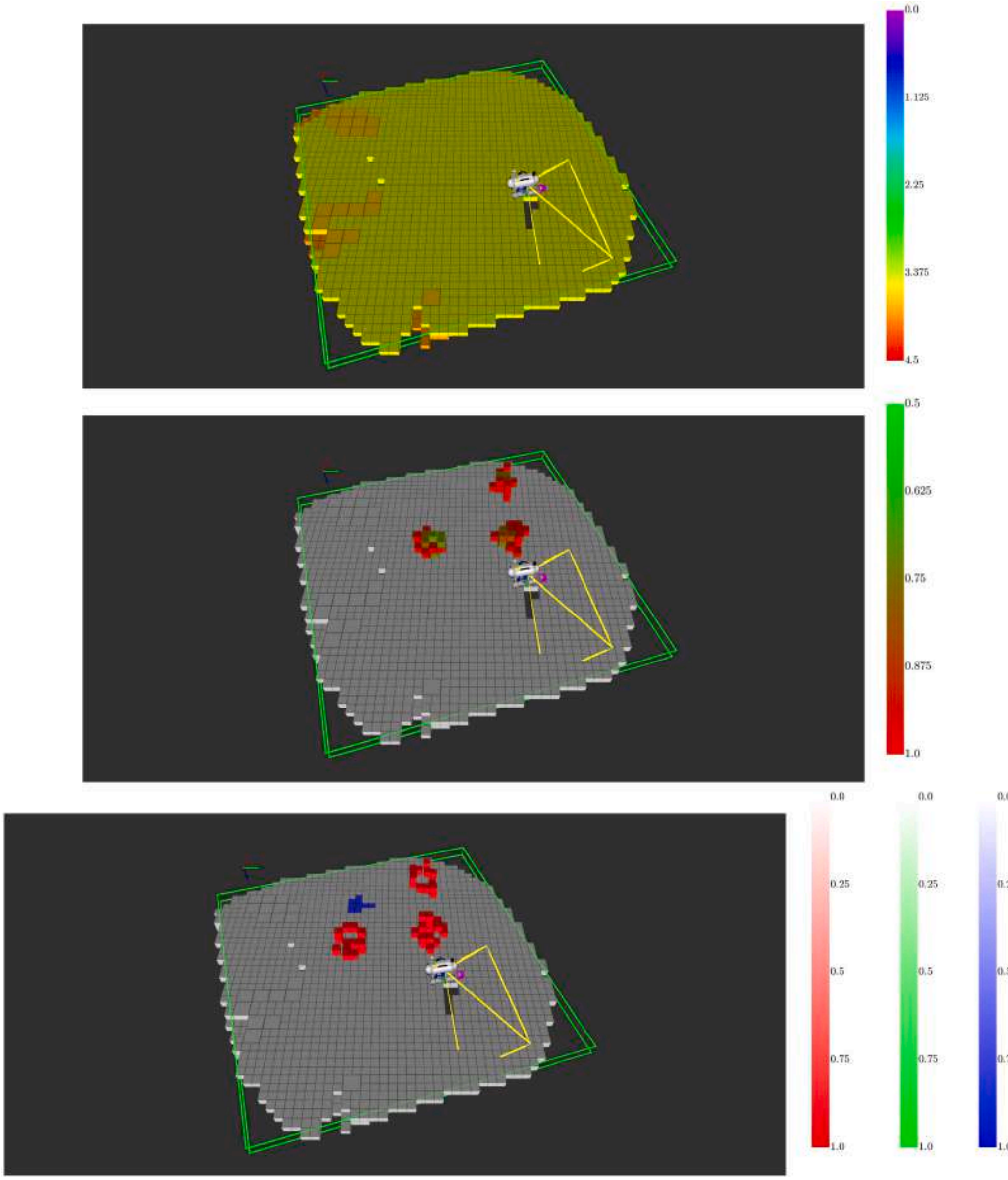

Fig. 20. Occupancy maps performed during a change detection mission. On the Top, the depth map, where the color indicates the depth of each point of the seabottom, in the Middle, the semantic map, where the color indicates the accuracy of the voxel to belong to an OPI, on the Bottom, the semantic change detection map, where the color indicates the status of the voxel to be a confirmed, removed or new OPI.

semantic map ensured that the OPIs were accurately located. Given the dimensions of the targets and the maximum distance of the cluster points from the centroid, the OPIs were within the defined regions, which is crucial for various applications. For example, in Mine Countermeasure (MCM) operations for planning safe routes and interventions, or in archaeological campaigns to design more precise inspections, this level of accuracy is essential.

Fig. 9 shows the semantic maps reconstructed during the two surveys (i.e., the discovery mission and the change detection mission), where occupied voxels in the seabottom are shown in gray, and colored cubes are associated to voxels which have a probability greater than 0.7 of belonging to an OPI. The color transitions from green to red as the probability increases. This semantic map is valuable for operators carrying out the tasks mentioned above or can even be used online by the robot.

Fig. 10 shows the depth maps obtained from the two surveys, where the occupied environment voxels are colored as a function of their depth values.

5.2. Change detection analysis

To validate and assess the performance of the proposed methodology, one change detection mission was carried out to inspect four OPIs positioned at known locations, with their coordinates in the NED frame listed in Table 1. The autonomous coverage survey was conducted in the same condition of the discovery mission, i.e. constant altitude of 4 meters and same limits to the area to be inspected. The semantically annotated map previously saved and gathered during the discovery mission is employed by the vehicle as reference and comparison to evaluate the status of each voxel which had an accuracy over 0.7 to belong to an OPI. The results of the change detection mission are reported in Fig. 11. It shows the semantic map reconstructed during the change detection mission, where occupied voxels in the environment are shown in gray, and colored cubes represent voxels with a probability of belonging to a confirmed, new or removed OPI with accuracy greater than 0.7. Table 3 summarizes these results, showing the status probabilities of each cluster computed with the Eqs. (10), (13), and (15).

Table 2
Outcomes of one simulation regarding the OPI localization capabilities during the discovery and the change detection missions.

		Discovery mission	Change detection mission
OPI 1	SLAM-based localization	$[3.687 \quad 0.943 \quad 5.534]$	$[3.784 \quad 1.132 \quad 5.493]$
	SLAM-based covariance	$[0.23 \quad 0.282 \quad 0.651]$	$[0.341 \quad 0.231 \quad 0.538]$
	OctoMap centroid	$[3.779 \quad 1.131 \quad 5.37]$	$[4.186 \quad 0.897 \quad 5.524]$
	OctoMap covariance	$[0.38 \quad 0.404 \quad 0.338]$	$[0.33 \quad 0.326 \quad 0.347]$
OPI 2	SLAM-based localization	$[2.314 \quad 10.121 \quad 7.231]$	—
	SLAM-based covariance	$[0.322 \quad 0.263 \quad 0.511]$	—
	OctoMap centroid	$[2.206 \quad 9.798 \quad 6.837]$	—
	OctoMap covariance	$[0.262 \quad 0.32 \quad 0.37]$	—
OPI 3	SLAM-based localization	$[6.034 \quad -2.845 \quad 5.554]$	—
	SLAM-based covariance	$[0.343 \quad 0.157 \quad 0.576]$	—
	OctoMap centroid	$[6.22 \quad 9.898 \quad 5.614]$	—
	OctoMap covariance	$[0.314 \quad 0.275 \quad 0.333]$	—
OPI 4	SLAM-based localization	$[-6.127 \quad -4.863 \quad 6.467]$	$[-6.241 \quad -4.757 \quad 6.523]$
	SLAM-based covariance	$[0.263 \quad 0.156 \quad 0.468]$	$[0.184 \quad 0.203 \quad 0.536]$
	OctoMap centroid	$[-6.227 \quad -4.872 \quad 6.579]$	$[-5.86 \quad -4.748 \quad 6.573]$
	OctoMap covariance	$[0.264 \quad 0.38 \quad 0.263]$	$[0.326 \quad 0.179 \quad 0.376]$
OPI 5	SLAM-based localization	—	$[-3.312 \quad 10.743 \quad 5.834]$
	SLAM-based covariance	—	$[0.342 \quad 0.267 \quad 0.634]$
	OctoMap centroid	—	$[-2.971 \quad 10.836 \quad 6.131]$
	OctoMap covariance	—	$[0.28 \quad 0.224 \quad 0.256]$

Table 3
Outcomes of one simulation regarding OPI status detection.

	Detected status	Probability
OPI 1	confirmed	$p(C) = 0.96$
OPI 2	removed	$p(R) = 0.94$
OPI 3	removed	$p(R) = 0.95$
OPI 4	confirmed	$p(C) = 0.93$
OPI 5	new	$p(N) = 0.95$

Table 4
Outcomes of four simulations in terms of traveled path distance during the discovery mission.

		Mean path length [m]	Covariance [m]
Coverage level 90 %	Cost function Eq. (20)	424.42	25.82
	Cost function Eq. (17)	383.13	29.03
Coverage level 95 %	Cost function Eq. (20)	482.89	41.11
	Cost function Eq. (17)	437.54	39.63

5.3. Coverage policy evaluation

In terms of the coverage task that has to be accomplished, although both the uncertainty-aware gain functions (Eqs. (20) and (23)) taking slightly longer paths if compared to the exploration-focused approach like the one obtained from Eq. (17), the AUV is still able to cover the area of interest effectively.

During the autonomous discovery surveys, the ATR system successfully detected all four OPIs, and the semantic map was continuously adapted to the new acquisitions. As previously mentioned, the gain function $\mathcal{L}_B$ (Eq. (20)) encourages the AUV to navigate toward unexplored areas while also reducing localization uncertainty for the targets. Fig. 12 illustrates this behavior. Instead of opting for the shortest path which lead to accomplish the task of completing the coverage, the planning algorithm favored a slightly longer path (depicted in blue), which allowed the vehicle to revisit areas with greatest uncertainty in voxel status recognition. This approach helped the vehicle first re-examine the targets to reduce their localization uncertainty before finishing the coverage task. Table 4 summarizes the results of the simulations, where a comparison of the traveled distance to achieve the 90% and 95% of the coverage were considered by employing the cost function reported in Eq. (20), which takes into account the ATR outputs, and the cost function defined in Eq. (17).

Table 5
Outcomes of four simulations in terms of traveled path distance during the change detection mission.

		Mean path length [m]	Covariance [m]
OPIs change detection	Cost function Eq. (23)	55.21	6.23
	Cost function Eq. (17)	194.29	24.92
Coverage level 90 %	Cost function Eq. (23)	457.62	32.52
	Cost function Eq. (17)	390.24	25.72
Coverage level 95 %	Cost function Eq. (23)	506.21	37.34
	Cost function Eq. (17)	438.46	31.87

Similarly, during the autonomous change detection missions, the ATR system successfully detected all three OPIs (one OPI has been removed to evaluate all the possible cases of change detection), and the semantically annotated map was continuously updated and compared with the semantic map acquired during the discovery mission. In this case, the gain function $\mathcal{L}_B$ (Eq. (23)) conducts the AUV to follow paths that firstly search the previously detected OPIs and then reduce localization uncertainty of the detected targets (new and confirmed), while at the same time it completes the coverage of the area. Figs. 13 and 14 illustrate these behaviors. Also in this case, the IG algorithm allows the vehicle to cover areas which has been visited to re-examine the targets and reduce their localization uncertainty before finishing the coverage task. Table 5 summarizes the results of the simulations, where the traveled distance to reaches all the points where the OPIs were found during the discovery mission and to cover the 90% and 95% of the inspection area were considered. These results have been obtained by employing the cost function reported in Eq. (23), which takes into account the ATR outputs and the comparison with the saved semantically annotated map, and the cost function defined in Eq. (17).

Analyzing the first rows of Table 5, it is necessary to highlight the differences between the traveled distance by the vehicle to inspect the zones where it detects OPIs during the discovery mission. Indeed, only by employing the cost function defined in Eq. (23) the vehicle is biased to inspect firstly the areas where OPIs were discovered. On the contrary, Eq. (17) determines the vehicle only to focus on the coverage task and the information related to the positions of the OPIs is not exploited.

Table 6
FeelHippo AUV available sensors.

GPS	U-blox 7P
IMU	Advanced Navigation Orientus
single-axis FOG	KVH DSP 1760
DVL-DS	Nortek DVL1000
optical camera	Microsoft Lifecam Cinema

6. Experimental results

The proposed strategy has been tested by employing FeelHippo AUV (Allotta et al., 2017), depicted in Fig. 15, a lightweight and compact vehicle developed by the Department of Industrial Engineering at the University of Florence (UNIFI DIEF). The main sensors of FeelHippo AUV are listed in Table 6, focusing the attention on the ones necessary to perform navigation and mapping tasks.

The proposed inspection strategy was experimentally validated using data collected at the Naval Support and Experimentation Center (Centro di Supporto e Sperimentazione Navale) (CSSN) experimental basin in La Spezia, Italy. During the trials, the main onboard computer was tasked with executing the IG planning system, which integrates the KDE-based learning technique, and the SLAM algorithm for autonomous navigation and mapping. In parallel, the payload computer was dedicated to the execution of the ATR module. The control loop frequency of the navigation and guidance system was set to 10 Hz, providing sufficient responsiveness for real-time control. This frequency was found to be fully compatible with the computational requirements of the IG planning algorithm, allowing the system to update the planned trajectory and map information without introducing delays or bottlenecks in the inspection workflow.

To evaluate the system's performance in detecting and localizing OPIs, four colored buoys were positioned underwater at varying depths. During the discovery mission, all four buoys were present. In contrast, for the change detection mission, one buoy was intentionally removed in order to assess the system's capability to identify changes in the environment and update the semantic map accordingly.

In both missions, the ATR module, based on the SSD MobileNet V2 network, was responsible for object detection and recognition. This network, optimized for real-time inference on embedded systems, was deployed on an NVIDIA Jetson Nano module integrated into the FeelHippo AUV. The system maintained an operational frame rate of up to 4 fps, enabling reliable perception in underwater conditions (see Fig. 16). Furthermore, to enable accurate target localization, camera calibration was performed prior to the experiments using a standard checkerboard-based intrinsic and extrinsic calibration procedure, followed by verification underwater to account for refraction effects at the air-water interface.

The experimental campaign was conducted within a controlled area approximately 20×20 meters in size, characterized by a relatively flat seabed. The FeelHippo AUV performed multiple surveys at a constant altitude of 3 meters, systematically exploring the area to detect buoys and inspect the environment.

The experimental results confirm that the proposed strategy effectively guided the AUV throughout the inspection task. As shown in Fig. 17, the policy derived from Eq. (20) demonstrated the desired behavior: the AUV dynamically replanned its path in response to feedback from the semantic mapping module, enabling it to revisit areas where OPIs had previously been detected. This adaptive behavior enhances the reliability of semantic map construction, as evidenced by the occupancy and semantic maps presented in Fig. 18.

During the change detection mission, the policy defined in Eq. (23) also exhibited the expected behavior, as depicted in Fig. 19. The AUV initially prioritized the inspection of regions previously associated with OPIs, followed by a complete area survey. It then revisited the newly detected OPIs to refine their localization. This behavior demonstrates the system's ability to integrate prior semantic information and dynamically adapt to environmental changes. The results, summarized through updated occupancy and semantic maps in Fig. 20, highlight the robustness of the proposed framework in identifying and localizing changes in underwater environments.

Table 7
Outcomes of four experimental results in terms of traveled path distance.

		Mean path length [m]	Covariance [m]
Coverage level 90 %	discovery mission	131.06	16.14
	change detection mission	154.77	9.82
Coverage level 95 %	discovery mission	147.68	10.03
	change detection mission	171.05	16.11

Table 8
Outcomes of one experimental results regarding OPI status detection.

	Discovery mission	Change detection mission	Status	Accuracy
OPI 1	✓	✓	confirmed	$p(C) = 0.92$
OPI 2	✓		removed	$p(R) = 0.95$
OPI 3	✓	✓	confirmed	$p(C) = 0.96$
OPI 4	✓	✓	confirmed	$p(C) = 0.95$

Quantitative metrics related to mission efficiency, in terms of the path length required to achieve 90% and 95% area coverage, are reported in Table 7. Additionally, Table 8 presents the status of each OPI as computed during the change detection phase. These results substantiate the system's capacity for effective environmental monitoring and autonomous change detection, reinforcing its applicability for long-term multi-session underwater inspection tasks.

7. Conclusion and future development

This paper presented a novel approach to multi-session perception-aware coverage path planning for active semantic SLAM, with a focus on automatic change detection. By integrating semantic understanding into the coverage path planning process, we addressed key challenges in autonomous exploration and mapping, enabling more effective and context-aware decision-making for robotic systems. The proposed method was evaluated through both simulations and real-world experiments, demonstrating its effectiveness in dynamic environments. The effectiveness of the proposed framework is supported by complementary evidence obtained from both simulation and real-world experiments, each offering distinct yet reinforcing insights. In simulation, quantitative analyses were conducted to evaluate navigation and mapping performance, including the accuracy and covariance of the estimated positions of OPIs during both discovery and change detection missions, measured against ground truth provided by the simulator. The change detection capability was assessed by comparing OPI positions across missions to compute the probability of correctly classifying each object as confirmed, removed, or newly detected. Furthermore, the coverage policy was rigorously evaluated by analyzing the path lengths required to achieve 90% and 95% area coverage during the discovery phase, as well as the effort needed to detect environmental changes during change detection missions. These simulation results provided valuable insights into the scalability and adaptability of the proposed framework across several simulations. These findings are further reinforced by real-world experimental trials, which validated the system's practicality and robustness under realistic deployment conditions. Empirical measurements confirmed the effectiveness of the framework in terms of path length during both discovery and change detection missions, as well as the probability of correctly determining the status of each OPI. Notably, the real-world data highlighted significant improvements in path efficiency, semantic mapping accuracy, and environmental change detection, thereby demonstrating the method's suitability for long-term autonomous operation in dynamic environments.

Our approach successfully combines perception and planning in a manner that not only facilitates coverage of the environment but also enhances the robot's ability to detect and respond to changes. The method showed promise in handling dynamic and uncertain environments, which is crucial for applications like environmental monitoring, infrastructure inspection, and disaster response.
While the results obtained in this study are promising, several aspects of the proposed system can be improved or extended in future research. Some potential directions for future development include scalability to larger environments and the introduction of acoustic sensors for the data gathering. Regarding the first point, the current approach works well for moderately sized environments, but further optimization is needed for scaling up to larger, more complex areas. Regarding the latter, incorporating real-time data fusion techniques, where data from various sensor modalities (e.g., acoustic and optical sensors) are merged effectively, could lead to more accurate semantic mapping and change detection. This fusion could also enhance the robot's ability to navigate and interact with complex environments more intelligently.

CRediT authorship contribution statement

Alessandro Bucci: Writing – original draft, Visualization, Validation, Software, Methodology, Conceptualization; **Alessandro Ridolfi:** Writing – review & editing, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Allotta, B., Conti, R., Costanzi, R., Fanelli, F., Gelli, J., Meli, E., Monni, N., Ridolfi, A., Rindi, A., 2017. A low cost autonomous underwater vehicle for patrolling and monitoring. Inst. Mech. Eng. Part M J. Eng. Marit. Environ. 231 (3), 740–749. https://doi.org/10.1177/1475090216681354

Allotta, B., Costanzi, R., Fanelli, F., Monni, N., Ridolfi, A., 2015. Single axis FOG aided attitude estimation algorithm for mobile robots. Mechatronics 30, 158–173. https://doi.org/10.1016/j.mechatronics.2015.06.012

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), pp. 2612–2617. https://doi.org/10.1109/ICRA.2016.7487419

Bircher, A., Kamel, M., Alexis, K., Oleynikova, H., Siegwart, R., 2016. Receding horizon next-best-view planner for 3D exploration. In: 2016 IEEE International Conference on Robotics and Automation (ICRA), pp. 1462–1468. https://doi.org/10.1109/ICRA.2016.7487281

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. J. Field Rob. 41 (8), 2543–2563. https://doi.org/10.1002/rob.22375

Bucci, A., Secciani, N., Zacchini, L., Ridolfi, A., 2023. Shallow water bathymetry using a DVL-based mapping strategy. In: OCEANS 2023 - Limerick, pp. 1–6. https://doi.org/10.1109/OCEANSLimerick52467.2023.10244555

Burguera, A.B., Bonin-Font, F., 2018. Towards multi session visual SLAM in underwater environments colonized with posidonia oceanica. In: 2018 IEEE/OES Autonomous Underwater Vehicle Workshop (AUV), pp. 1–7. https://doi.org/10.1109/AUV.2018.8729730

Dellaert, F., 2012. Factor Graphs and GTSAM: A Hands-On Introduction. Technical Report. Georgia Institute of Technology.

Dellaert, F., Kaess, M., 2017. Factor graphs for robot perception. Found. Trends Rob. 6 (1-2), 1–139. https://doi.org/10.1561/2300000043

Fallon, M.F., Folkesson, J., McClelland, H., Leonard, J.J., 2013. Relocating underwater features autonomously using sonar-based SLAM. IEEE J. Oceanic Eng. 38 (3), 500–513. https://doi.org/10.1109/JOE.2012.2235664

Fossen, T., 1994. Guidance and Control of Ocean Vehicles. London, UK, Wiley.

Franchi, M., Bucci, A., Zacchini, L., Ridolfi, A., Bresciani, M., Peralta, G., Costanzi, R., 2021. Maximum a posteriori estimation for AUV localization with USBL measurements. IFAC-PapersOnLine 54 (16), 307–313. https://doi.org/10.1016/j.ifacol.2021.10.109

Franchi, M., Ridolfi, A., Pagliai, M., 2020. A forward-looking sonar and dynamic model-based auv navigation strategy: preliminary validation with FeelHippo AUV. Ocean Eng. 196, 106770. https://doi.org/10.1016/j.oceaneng.2019.106770

Galceran, E., Carreras, M., 2013. A survey on coverage path planning for robotics. Rob. Auton. Syst. 61 (12), 1258–1276. https://doi.org/10.1016/j.robot.2013.09.004

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. https://doi.org/10.3390/robotics9030051

Guerneve, T., Subr, K., Petillot, Y., 2017. Underwater 3D structures as semantic landmarks in SONAR mapping. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 614–619. https://doi.org/10.1109/IROS.2017.8202215

Guo, Y., Liu, H., Fan, X., Lyu, W., 2021. Research progress of path planning methods for autonomous underwater vehicle. Math. Probl. Eng. 2021 (1), 8847863. https://doi.org/10.1155/2021/8847863

Hernández, J.D., Vidal, E., Moll, M., Palomeras, N., Carreras, M., Kavraki, L.E., 2019. Online motion planning for unexplored underwater environments using autonomous underwater vehicles. J. Field Rob. 36 (2), 370–396. https://doi.org/10.1002/rob.21827

Himri, K., Ridao, P., Gracias, N., Palomer, A., Palomeras, N., Pi, R., 2018. Semantic SLAM for an AUV using object recognition from point clouds. IFAC-PapersOnLine 51 (29), 360–365. https://doi.org/10.1016/j.ifacol.2018.09.497

Ho, B.-J., Sodhi, P., Teixeira, P., Hsiao, M., Kusnur, T., Kaess, M., 2018. Virtual occupancy grid map for submap-based pose graph SLAM and planning in 3D environments. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2175–2182. https://doi.org/10.1109/IROS.2018.8594089

Hollinger, G.A., Sukhatme, G.S., 2014. Sampling-based robotic information gathering algorithms. Int. J. Rob. Res. 33, 1271–1287. https://doi.org/10.1177/0278364914533443

Hornung, A., Wurm, K.M., Bennewitz, M., Stachniss, C., Burgard, W., 2013. OctoMap: an efficient probabilistic 3D mapping framework based on octrees. Auton. Rob. 34 (3), 189-206. https://doi.org/10.1007/s10514-012-9321-0

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). IEEE, pp. 758–765. https://doi.org/10.1109/IROS.2015.7353457

Huang, X., Yang, R., Wang, Q., Yu, F., He, B., 2023. A novel method for real-time ATR system of AUV based on Attention-MobileNetV3 network and pixel correction algorithm. Ocean Eng. 270, 113403. https://doi.org/10.1016/j.oceaneng.2022.113403

Isler, S., Sabzevari, R., Delmerico, J., Scaramuzza, D., 2016. An information gain formulation for active volumetric 3d reconstruction. In: 2016 IEEE International Conference on Robotics and Automation (ICRA), pp. 3477–3484. https://doi.org/10.1109/ICRA.2016.7487367

Jang, H., Yoon, S., Kim, A., 2021. Multi-session underwater pose-graph SLAM using inter-session opti-acoustic two-view factor. In: 2021 IEEE International Conference on Robotics and Automation (ICRA), pp. 11668–11674. https://doi.org/10.1109/ICRA48506.2021.9561161

Kaess, M., Johannsson, H., Roberts, R., Ila, V., Leonard, J.J., Dellaert, F., 2012. iSAM2: incremental smoothing and mapping using the Bayes tree. Int. J. Rob. Res. 31 (2), 216–235. https://doi.org/10.1177/0278364911430419

Kaess, M., Ranganathan, A., Dellaert, F., 2008. iSAM: incremental smoothing and mapping. IEEE Trans. Rob. 24 (6), 1365–1378. https://doi.org/10.1109/TRO.2008.2006700

Kostavelis, I., Gasteratos, A., 2015. Semantic mapping for mobile robotics tasks: a survey. Rob. Auton. Syst. 66, 86–103. https://doi.org/10.1016/j.robot.2014.12.006

Larsen, M.K., Pizarro, O., Ludvigsen, M., 2023. Geometric registration of benthic imagery for learning appearance-based place recognition over multiple sessions. In: 2023 IEEE Underwater Technology (UT), pp. 1–10. https://doi.org/10.1109/UT49729.2023.10103451

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C., 2016. SSD: single shot multibox detector. In: Computer Vision – ECCV 2016, pp. 21–37. https://doi.org/10.1007/978-3-319-46448-02

Ma, C., Zou, H., An, X., 2024. A complete coverage path planning approach for an autonomous underwater helicopter in unknown environment based on VFH+ algorithm. J. Mar. Sci. Eng. 12 (3). https://doi.org/10.3390/jmse12030412

PairetÃ., Hernández, J.D., Carreras, M., Petillot, Y., Lahijanian, M., 2022. Online mapping and motion planning under uncertainty for safe navigation in unknown environments. IEEE Trans. Autom. Sci. Eng. 19 (4), 3356–3378. https://doi.org/10.1109/TASE.2021.3118737

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, pp. 2875–2880. https://doi.org/10.1109/IROS.2012.6385838

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y., 2009. ROS: an open-source robot operating system. In: ICRA Workshop on Open Source Software. Vol. 3. Kobe, Japan, p. 5.

Rahman, S., Li, A.Q., Rekleitis, I., 2018. Sonar visual inertial SLAM of underwater structures. In: 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 5190–5196. https://doi.org/10.1109/ICRA.2018.8460545

Smith, R.N., Schwager, M., Smith, S.L., Jones, B.H., Rus, D., Sukhatme, G.S., 2011. Persistent ocean monitoring with underwater gliders: adapting sampling resolution. J. Field Rob. 28 (5), 714–741. https://doi.org/10.1002/rob.20405

Sodhi, P., Ho, B.-J., Kaess, M., 2019. Online and consistent occupancy grid mapping for planning in unknown environments. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 7879–7886. https://doi.org/10.1109/IROS40897.2019.8967991

Teixeira, P.V., Kaess, M., Hover, F.S., Leonard, J.J., 2016. Underwater inspection using sonar-based volumetric submaps. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4288–4295. https://doi.org/10.1109/IROS.2016.7759631

Vidal, E., Palomeras, N., IsteniČ, K., Gracias, N., Carreras, M., 2020. Multisensor online 3D view planning for autonomous underwater exploration. J. Field Rob. 37 (6), 1123–1147. https://doi.org/10.1002/rob.21951

Wang, Y., Ramezani, M., Fallon, M.F., 2020. Actively mapping industrial structures with information gain-based planning on a quadruped robot. 2020 IEEE International Conference on Robotics and Automation (ICRA) , 8609–8615. https://doi.org/10.1109/ICRA40945.2020.9197153

Westman, E., 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., Kaess, M., 2019. Degeneracy-aware imaging sonar simultaneous localization and mapping. IEEE J. Oceanic Eng. 45 (4), 1280–1294. https://doi.org/10.1109/JOE.2019.2924615

Yao, P., Qiu, L., Qi, J., Yang, R., 2021. AUV path planning for coverage search of static target in ocean environment. Ocean Eng. 241, 110050. https://doi.org/10.1016/j.oceaneng.2021.110050

Yguel, M., Aycard, O., Laugier, C., 2008. Update Policy of Dense Maps: Efficient Algorithms and Sparse Representation. Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 23–33. https://doi.org/10.1007/978-3-540-75404-63

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. J. Field Rob. 37 (6), 925–950. https://doi.org/10.1002/rob.21923

Yu, Y., Zhao, J., Gong, Q., Huang, C., Zheng, G., Ma, J., 2021. Real-time underwater maritime object detection in side-scan sonar images based on transformer-YOLOv5. Remote Sens. 13 (18). https://doi.org/10.3390/rs13183555

Zacchini, L., Franchi, M., Ridolfi, A., 2022. Sensor-driven autonomous underwater inspections: a receding-horizon RRT-based view planning solution for AUVs. J. Field Rob. 39 (5), 499–527. https://doi.org/10.1002/rob.22061

Zacchini, L., Ridolfi, A., Allotta, B., 2023. Informed expansion for informative path planning via online distribution learning. Rob. Auton. Syst. 166, 104449. https://doi.org/10.1016/j.robot.2023.104449

Zacchini, L., Ridolfi, A., Topini, A., Secciani, N., Bucci, A., Topini, E., Allotta, B., 2020. Deep Learning for on-board AUV automatic target recognition for optical and acoustic imagery. IFAC-PapersOnLine 53 (2), 14589–14594. https://doi.org/10.1016/j.ifacol.2020.12.1466

Zhang, L., Li, C., Sun, H., 2022. Object detection/tracking toward underwater photographs by remotely operated vehicles (ROVs). Future Gener. Comput. Syst. 126, 163–168. https://doi.org/10.1016/j.future.2021.07.011