Robotics and Autonomous Systems 166 (2023) 104449

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

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

Informed expansion for informative path planning via online distribution learning

Leonardo Zacchini *, Alessandro Ridolfi, Benedetto Allotta

Department of Industrial Engineering (DIEF), University of Florence (UNIFI), via di Santa Marta, 3, Florence, 50139, Italy
Interuniversity Center of Integrated Systems for the Marine Environment (ISME), Genova, 16145, Italy

ARTICLE INFO

Article history:
Received 23 August 2022
Received in revised form 24 November 2022
Accepted 8 May 2023
Available online 16 May 2023

Keywords:
Informative Path Planning
Informed expansion
Mobile robots
Autonomous Underwater Vehicles

ABSTRACT

Mobile robots are essential tools for gathering knowledge of the environment and monitoring areas of interest as well as industrial assets. Informative Path Planning methodologies have been successfully applied making robots able to autonomously acquire information and explore unknown surroundings. Rapidly-exploring Information Gathering approaches have been validated in real-world applications, proving they are the way to go when aiming for Information Gathering tasks. In fact, RIG can plan paths for robots with several degrees of freedom and rapidly explore complex workspaces by using the state-of-the-art Voronoi-biased expansion. Nevertheless, it is an efficient solution when most of the area is unknown but its effectiveness decreases as the exploration/gathering evolves. This paper introduces an innovative informed expansion for IG tasks that combines the Kernel Density Estimation technique and a rejection sampling algorithm. By learning online the distribution of the acquired information (i.e., the discovered map), the proposed methodology generates samples in the unexplored regions of the workspace, and thus steers the tree toward the most promising areas. Realistic simulations and an experimental campaign, conducted in the underwater robotics domain, provide a proof-of-concept validation for the developed informed expansion methodology and demonstrate that it enhances the performance of the RIG algorithm.

1. Introduction and background

In the last years, mobile robots have emerged as fundamental tools for gathering knowledge of the environment, frequently inspecting areas of interest, and performing complex tasks in hazardous scenarios, reducing the costs of monitoring campaigns and, most importantly, the risks for humans. In this context, innovative breakthrough solutions for pushing the boundaries of robots autonomy have been developed. As the level of autonomy grows, mobile robots are increasingly being used not only by researchers but also by science users, industries, and non-trained people in a wide variety of applications, ranging from search and rescue, industrial assets inspections, explorations and monitoring [1–5].

Therefore, in all domains, i.e., ground [6], aerial [7], and marine [8,9], mobile robots are tasked to autonomously acquire information about the environment. To tackle the challenging IG task, several Informative Path Planning (IPP) methodologies have been proposed in the last few years. Despite the different strategies employed, these solutions aim at planning safe and feasible paths for the robot that permit the collection of as much information as possible during the survey/inspection with a given payload sensor. When operating in unknown or rapidly changing scenarios, IPP online solutions that do not rely on a priori knowledge, consider the sensor measurements, and react to the perceived environment shall be employed [2,10]. In particular, in such conditions, IPP algorithms have to compute and update optimal or sub-optimal inspection paths, according to a certain cost/gain function, given the data available at each time step. The planned paths must comply with the robot motion constraints. Additionally, given the limited computational resources generally available onboard on mobile robots and the constrained planning time imposed by real-time applications, IPP algorithms should be computationally efficient.

IPP solutions are generally classified as myopic, i.e., greedy solutions that only evaluate the next robot pose, and non-myopic, where the effects of more actions are predicted. Regarding the former class, various candidates selection criteria could be utilized [6,11,12]. Although they have shown remarkable results [4,13], they require a greedy heuristic for deterministically selecting the Next Best Viewpoint (NBV) that influences the outcomes [14]. On the other hand, non-myopic solutions, by predicting the effects of several possible actions, can provide

* Corresponding author at: Department of Industrial Engineering (DIEF), University of Florence (UNIFI), via di Santa Marta, 3, Florence, 50139, Italy.
E-mail addresses: leonardo.zacchini@unifi.it (L. Zacchini), alessandro.ridolfi@unifi.it (A. Ridolfi).

https://doi.org/10.1016/j.robot.2023.104449

better results than myopic counterparts and can escape from local minima [15]. These latter class of algorithms can generally be categorized as: Evolutionary Algorithm (EA)-based [16,17], Graph-based [18,19] and Random tree-based [15,20]. Inspired by natural evolution processes, EAs have successfully been employed in various scenarios. Nevertheless, they either discretize the workspace or need a fixed endpoint and have a high computational burden. Similarly, Graph-based methods utilize a discretized environment, whose density has a huge impact on the achievable results. Sampling-based algorithms overcome the aforementioned drawbacks. Inspired by the asymptotically optimal Rapidly-exploring Random Tree (RRT*), Rapidly-exploring Random Graph (RRG), and Probabilistic RoadMap (PRM*), they can solve the informative planning problem using iterative sampling, outperforming other approaches [14]. Such algorithms, namely RIG [14], operate in the continuous space, implement the motion constraints easily, and, thanks to the limited computational burden, can run onboard in real-time on mobile robots without requiring dedicated hardware. Besides, IG solutions based on RRT algorithm, by efficiently and rapidly exploring the workspace, can find satisfying solutions even with a high dimensional environment and can deal with robots with several degrees of freedom. Consequently, RRT-based IG methodologies emerged as the best approach for several real-world applications. Thus, this work's focus has shifted to such a class of algorithms, and, for the sake of simplicity, the term RIG will be used to denote this category in the following.

In [2], the quadruped robot ANYmal autonomously mapped an industrial structure by using an RIG algorithm for planning the survey. In the context of aerial robotics, Papachristos [21] and Bircher [7], just to mention a few, made use of an RIG strategy for guiding Micro Aerial Vehicles (MAV) explorations of indoor areas. Similarly, in [22], the task of simultaneously exploring and object searching with MAVs was tackled. In detail, at each time step, an RRT was expanded and the utility function used to select the best path considered both the area exploration and the localization of objects of interest, detected through a Convolutional Neural Network (CNN). Finally, the authors developed and tested an RIG-based inspection methodology for the underwater domain [9]. The solution was used to make an Autonomous Underwater Vehicle (AUV) able to autonomously inspect unknown underwater areas and was validated with real experiments at sea.

Although different scenarios and applications were tackled, the aforementioned works highlight that the RIG approaches can be the way to go when aiming for autonomously exploring and gathering knowledge of the environment. They can be easily applied to different scenarios, plan feasible paths and consider various sensors. As stated above, these solutions were inspired by the RRT algorithm, and its optimal version RRT*, initially designed for motion planning tasks [23]. Starting from the robot actual pose, a tree, whose nodes represent robot configurations, is expanded by iteratively sampling new configurations in the workspace. Then, the closest node to the sampled configuration is found. The tree is expanded by propagating the closest node toward the newly sampled configuration according to the robot motion constraints. Finally, the expected information that can be acquired by the novel generated node is computed by using a sensor acquisition model and a utility function. The path composed as a sequence of connected nodes, i.e., the tree branch, expected to acquire more information, according to the utility function, is selected as the best solution. The strategy is usually applied in a receding horizon manner by executing only the first node, i.e., the NBV.

Of paramount importance, is the ability of RRT-based algorithms to explore rapidly and efficiently the workspace thanks to the Voronoi-biased expansion strategy. When a new configuration is sampled, a uniform distribution over the workspace is employed, and by selecting the closest node for the expansion, tree nodes that correspond to larger Voronoi regions are more likely to be selected. As shown in [24], such a strategy ensures that RRT-based algorithms well explore the workspace better than a naive solution where nodes are randomly selected for the expansion. Thus, the sampling policy of the expansion phase plays a fundamental role in generating well-spawned nodes and finding good solutions.

In IG applications, the limited computational resources generally available on mobile robots can make the information gain estimate the most computational burden part and the performance bottleneck of the algorithm. Indeed, by spending a considerable amount of time on information gain calculation, the tree expansion within a given computational time slot, imposed by real-time needs, is limited, largely affecting the performance of the RIG [25].

Consequently, three ideas could be pursued to enhance the performance of these solutions. The first two are represented by efficient algorithms that allow to keep and update already generated nodes and faster evaluation methodologies enabling the algorithm to create more configurations. They can be achieved with efficient and optimized implementations and/or by exploiting information gain learning-based approximations. Alternatively, informed expansion strategies that guide the tree toward the most promising areas can be used. While some efforts have already been made for the two former approaches, e.g., [15,26,27], the latter is still an open point.

In the context of motion planning, several efforts have been made to develop innovative strategies for sampling new configurations to enhance the performance of RRT algorithms. For instance, a common strategy for steering the tree toward the solution is randomly considering the goal configuration as a sample. Thus, a trade-off between the exploration of the workspace and the rapidity of obtaining a solution to the task can be achieved. Then, in [28], an informed expansion strategy was proposed. A subset of states that can improve a solution was defined and used to sample the new configurations, enhancing the performance of the algorithm in finding the short paths.

Turning to IG applications, the tree exploration properties are of utmost importance. Evaluating the information gain of a configuration is time expensive, and considering the limited computational time of real-world applications, it is crucial to utilize the available resources for the most promising nodes. Thus, generating well-spawned configurations is pivotal during the tree expansion. Voronoi-based expansion, which has been utilized in various works [7,9,21], by sampling using a uniform distribution, is an efficient solution when most of the area is unknown, but it loses its effectiveness as the exploration goes on since to generate a branch that leads the robot to gather new information, more nodes shall be generated [29].

To overcome this limitation, informed sampling methodologies have recently been investigated. Frontier points were exploited in the novel solution for 3D reconstructions proposed in [25]. By exploiting surface frontiers points, complete and accurate reconstructions were achieved. However, the method required a connected surface and created a set of unconnected configurations, needing a separate path-planner to guide the robot to the NBV.

A sampling strategy to bias the exploration to the most informative regions was proposed in [30,31]. In detail, the authors proposed a sampling-based planning algorithm that allowed a multi-robot target localization and tracking. The solution, tested only in simulation, utilized a complex strategy and is based on prior assumptions such as static landmarks. Besides, it requires considerable efforts to make it applicable to real assets and suitable for dealing with exploration tasks.

For what concerns the exploitation of online learning strategies, interesting approaches have been investigated recently.

In [32], Gaussian Processes (GPs) were used to predict the most informative areas for sampling. Then, in [33], extension of [34], a fast GP-based occupancy mapping solution was proposed. In addition, based on that representation, the authors developed an algorithm for driving the robot toward the most promising region. Nevertheless, a selection criterion for greedy information gain-based exploration was considered.

Finally, a similar approach to the one proposed in this paper was introduced in [29]. To enhance the exploration performance of IG algorithms, Schmid and colleagues introduced an innovative learning methodology to a sampling-based exploration paradigm. In detail, conditional variational autoencoders were exploited to directly learn from a standard occupancy map an informed distribution, accounting for information gain, kinematic feasibility, collision avoidance, and dynamics-based motion cost. Nevertheless, the complex methodology was designed only for local planning purposes. Thus, the authors developed a two-stage global–local planning approach, where for the former, a standard algorithm was implemented.

Contribution

In this paper, an intelligent tree expansion for IG is addressed. The main contribution to the state of the art is represented by a novel-informed expansion that steers the tree toward unexplored areas. The here-introduced methodology learns online the distribution of the acquired data (in the context of this research, corresponds to discovered map), which is used to sample new configurations in unexplored regions. It combines the KDE technique [35] for the learning stage and a rejection sampling algorithm for biasing the RIG expansion. The KDE is a non-parametric method to estimate, given a set of samples, the probability density function of a random variable using kernels as weights. It is a powerful solution that does not require a high computational burden. It has found extensive use in statistics and machine learning [36,37], but, to the best of the authors' knowledge, its exploitation in the context of IPP has yet to be deepened.

The developed approach allows using a single-stage planning paradigm for exploration tasks that can run on compact robots and do not require additional assumptions. The solution is validated through both simulations and real experiments conducted in the context of underwater robotics, the research field of the authors. It is worth emphasizing that the informed expansion methodology has general validity and can find applications in various robotics domains. In detail, the developed RIG algorithm was utilized for making an AUV able to autonomously conduct underwater inspections in an unknown area with a Forward-Looking Sonar (FLS). The performance of the RIG solution using the novel informed expansion strategy is compared with the state-of-the-art RRT sampling policy based on a uniform distribution. The comparisons demonstrate that the proposed strategy enhances the performance of the RIG algorithm.

The remainder of the paper is organized as follows. Section 2 introduces the IG fundamental concepts and the notation used in this work. Section 3 firstly presents the developed IG system and the RIG algorithm. Then, it describes the learning solution for estimating the distribution of the discovered map, and details the realized informed expansion. The proposed methodology is validated with realistic simulations reported in Section 4. The solution was used to make a compact AUV able to autonomously explore an area of interest. In addition, the IG system was tested in a real experimental campaign, detailed in Section 5. Finally, Section 6 concludes the paper and discusses avenues for future works.

2. Information gathering planning preliminaries

In this section, the IG task is formulated and useful preliminary concepts are introduced. Sampling-based IPP algorithms try to compute the path $\mathcal{P}^\star$ that allows the robot to maximize the information gathered $\mathcal{G}^\star_{\mathcal{P}^\star}$, which corresponds to solve the following optimization problem:

$$\begin{aligned} \max_{\mathcal{P}\in\mathcal{W}} \quad & \mathcal{G}_{\mathcal{P}} \\ \text{s.t.} \quad & \mathcal{P} = \{\xi_i\} \\ & \xi_i = f(\xi_{i-1}, u_{i-1}) \\ & \xi \in \mathcal{W}\ , \end{aligned} \tag{1}$$

where a path $\mathcal{P}$ is of a sequence of robot configurations $\{\xi_i\}$, $\mathcal{W}$ represents the workspace.

Regarding the robot motion model f, dynamic or kinematic models could be employed. While kinematic models only use geometric equations that relate the vehicle positions and velocities, dynamic models consider the force and torques that create the motion. In the context of this work, to enable the use of a rewiring strategy, which consists of a routine that checks whether a new node could improve the cost of neighbor nodes (described in detail in the following Section), a kinematic model has been favored. In fact, a steering function, which returns the optimum path between two states, is needed. When using dynamic motion constraints, computing a steering function means addressing a two-point Boundary Value Problem (BVP). That is, it corresponds to solving a differential equation under certain boundary conditions [38], which is generally a difficult problem. Therefore, in this work, the Dubins vehicle model, which is of particular interest in planning tasks since it can be used for both generating new configurations and works as a steering function for the rewiring routine, has been utilized. More details regarding the AUV motion models can be found in the Appendix.

When operating in unknown environments, IPP strategies are usually applied in a receding horizon fashion, meaning that the maximization problem in Eq. (1) is solved, at each step, according to the available data. Then, only the first move is performed, new data are acquired, and the process is iteratively repeated. Generally speaking, the IG solution aims at computing and updating online a path for the robot so that it can explore the workspace $\mathcal{W}$, initially assumed unknown.

To conduct autonomous exploration, in order to monitor the IG process and safely navigate in unknown environments, robots create a map of the environment that reflects its structure. Occupancy grid mapping paradigms, developed as a robust representation of the surrounding, has shown remarkable results in the last years in several robotic domains and applications. In the context of this research, the mapping solution described in [9], which makes use of the well-known Octomap framework [39], was considered. According to the occupancy mapping theory, the workspace is partitioned in 3D independent cells m_i, so for the map m holds $m = \{m_i\}$. Each cell m_i has a probability P_o of being occupied, whose value can determine the cell status, i.e., unmapped, free, or occupied. Thus, the goal of the IG methodology can be considered as the problem of guiding the robot so that each cell is classified as free or occupied.

Occupancy mapping paradigms allow to evaluate the information gathered by a configuration through a ray casting routine. By knowing the sensor Field of View (FoV) and range, the set of rays $\mathcal{R}_\mathfrak{r}$ is determined. Each ray $\mathfrak{r}$ ends when it reaches the maximum sensor range, the limit of the map or it hits an occupied voxel. Thus, the expected information obtainable from a robot configuration ξ is computed as:

$$\mathcal{I}_\xi = \sum_{\forall \mathfrak{r}\in\mathcal{R}_\mathfrak{r}} \ \sum_{\forall x\in\mathcal{X}_\mathfrak{r}} I(x), \tag{2}$$

where $I(x)$ denotes the Volumetric Information (VI) contained in a voxel x, $\mathcal{X}_{\mathfrak{r}}$ represents the set set of voxels traversed by the ray $\mathfrak{r}$. The information gain for a path $\mathcal{P}$ should consider all the voxels $\mathcal{X}_{\mathcal{P}}$ that can be mapped by following the path $\mathcal{P}$; in a compact notation:

$$\mathcal{G}_{\mathcal{P}} = \sum_{\forall x \in \mathcal{X}_{\mathcal{P}}} I(x). \tag{3}$$

Regarding the VI formulation, occupancy mapping paradigms allow considering the map uncertainty. In this research, an entropy-based formulation was exploited:

$$I(x) = -P_o(x)\ln\left(P_o(x)\right) - \bar{P}_o(x)\ln\left(\bar{P}_o(x)\right), \tag{4}$$

where $\bar{P}_o(x) = 1 - P_o(x)$. It is worth mentioning that, in such formulation, unmapped voxels, which due to the non-informative prior assumption of the mapping framework have an occupancy probability of $P_o(x) = 0.5$, match the highest *VI* value.

3. Informed information gathering planning

In this Section, the novel methodology for biasing the tree expansion toward unexplored areas is introduced. Firstly, the utilized framework for autonomous IG and the developed RIG algorithm are presented. Then, the focus shifts on the innovative informed expansion methodology to steer the tree toward the promising areas. The realized strategy for learning the distribution of the discovered map is based on the KDE technique. New configurations can then be sampled according to the learned distribution, biasing the tree expansion toward unexplored areas.

3.1. Receding-horizon information gathering framework

The developed receding-horizon IG system is detailed in this section. The system block diagram is depicted in Fig. 1. The Mission Manager monitors the robot status and supervises the exploration/inspection survey. It requires to the IPP module to compute the IG path $\mathcal{P}$. To this end, the IPP module, by utilizing the RIG algorithm described in the next section, solves the maximization problem of Eq. (1) by expanding from the robot actual configuration a tree of possible views. Here, the proposed informed expansion strategy comes into play. As described in the following, the discovered map is used to train online the KDE estimator. Then, the RIG algorithm utilizes the learned distribution to steer the tree toward the most promising areas by generating as many nodes as possible, drawn with the informed sampling policy, within the assigned computational time slot. The expected information gain for each view, and so for each path, is computed as reported in Section 2 through a ray casting procedure on the map, built and updated by the Mapping system utilizing the measurements of the payload sensor and the estimated robot pose. Once the best IG path $\mathcal{P}$ has been found, the first configuration that composes the path is selected as the NBV. The robot navigates toward the NBV, and when it reaches the designed configuration, the Mission Manager closes the loop by triggering the IPP module again. The previously computed IG path is used to initialize the new tree expansion, realizing the receding-horizon manner. Finally, the Guidance, Navigation and Control (GNC) module includes strategies and algorithms that the robot uses to estimate its position and track the planned IG path.

3.2. RIG algorithm

The RRT-based IG algorithm used by the IPP module is detailed in Algorithm 1. It takes as input the robot actual configuration ξ_t and the best path computed at the previous call $\mathcal{P}_{t-1}$ (if available). A new tree containing ξ_t is created (line 4, Algorithm 1), and, if the algorithm had already been activated, the remainder of the previously computed path is evaluated on the updated map and it is added to the tree (lines 6–8, Algorithm 1). Then, given the computational maximum time, the algorithm expands the tree (lines 10–21, Algorithm 1). The expansion consists of two phases: a generation step, where a new node is created and added to the tree, and then a rewiring step, where possible connections that can maximize the information gathering are tested. The RIG algorithm proposed in Algorithm 1 allows running the generation and the rewiring procedures in parallel. At each step, it generates a new node ξ_n and computes its visibility (i.e., the set of observable voxels) (line 14, Algorithm 1), while it runs the rewiring routine for finding the best connections for the node created at previous iteration ξ_p (line 15, Algorithm 1). The here proposed algorithm, in contrast to classic formulations [7,9,14], allows running the two routines, i.e., **generateNode()** and **rewireNode(ξ)**, concurrently, reducing the time required for a step of the algorithm, and thus increasing the number of nodes that can be generated within the maximum computational time slot.

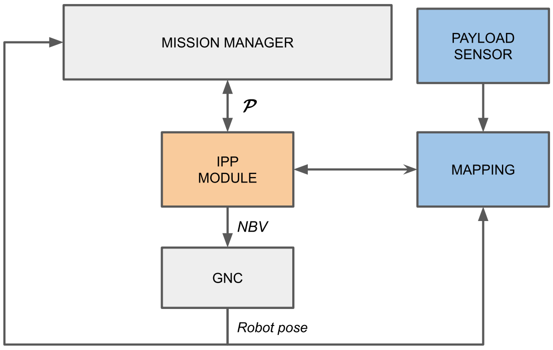


Fig. 1. Block diagram of the IG system architecture. The arrows depict the data workflow. The Mission Manager supervises the survey by monitoring the robot's pose and triggers the IPP module to plan the IG path, exploiting the map created by the Mapping module. To this end, the robot estimated pose is used to update the map with new measurements provided by the utilized sensor, represented by the Payload Sensor block. Given the computed path $\mathcal{P}$, the NBV is selected as the first configuration and set as the goal configuration for the robot that uses the guidance, navigation, and control algorithms (GNC block) to estimate its pose and track the path.

The former procedure is detailed in Function generateNode: a configuration ξ_s is sampled (line 4, Function generateNode), and a new node is computed by expanding the closest tree node ξ_n toward the direction of the sampled configuration (lines 5–6, Function generateNode). It is worth mentioning that the sampling strategy is of utmost importance. RIG solutions usually exploit a uniform distribution over the workspace that leads to a Voronoi-biased expansion that guarantees that the tree explores the workspace rapidly and efficiently (see [24]). In fact, nodes corresponding to larger Voronoi regions are more likely to be selected for the expansion. Finally, once the node pose ξ_{new} has been calculated, if the motion between ξ_n and ξ_{new} is safe and feasible (line 7, Function generateNode), its visibility can be evaluated (line 8, Function generateNode). Due to the submodularity of IPP objective functions [40], which arises from the spatial correlation of the measurement locations (i.e., the tree nodes), to allow the use of the rewiring routine, the function **computeVisibility**(ξ) computes the set of visible voxels, which depends only on the robot pose.

Regarding the rewiring routine, it is summarized in Function rewireNode. Briefly, given a node ξ, the set of the nearest neighbors is retrieved (line 5, Function rewireNode). Then, the rewiring strategy firstly looks for the node parent that maximizes the gain (lines 6–15, Function rewireNode), and then, evaluates whether the new node could improve the expected gain of other paths, corresponding to tree branches (lines 16–26, Function rewireNode).

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 (function **computeGain(·, ·)** in Function rewireNode):

$$\mathcal{G}_{\mathcal{P}_\xi} = \mathcal{G}_{\mathcal{P}} e^{\lambda_\varphi(\Delta\psi(\xi_0,\xi))} e^{\lambda_d(\text{ distance } (\xi_0,\xi))}, \tag{5}$$

where $\mathcal{G}_{\mathcal{P}}$ is calculated according to Eq. (3), $\lambda_\varphi \in R$ and $\lambda_d \in R$ are penalization factors for curvy and long paths, respectively.

The RIG algorithm ends selecting the most promising path $\mathcal{P}_t$, and the first node becomes the NBV ξ_{t+1} (lines 22–23, Algorithm 1).

Algorithm 1 RIG algorithm

```
 1: INPUT:Robot configuration $\xi_t$, Previous best path $\mathcal{P}_{t-1}$
 2: OUTPUT: Next best configuration $\xi_{t+1}$
 3: ITERATION:
 4: Initialize a new tree $\mathcal{T}$ with $\xi_0 = \xi_t$
 5: Initialize $\xi_p = Null$
 6: if $\mathcal{P}_{t-1} != Null$ then
 7:     addPath($\mathcal{T}$, $\mathcal{P}_{t-1}$)
 8:     updatePathGain($\mathcal{P}_{t-1}$)
 9: end if
10: while $time < time_{max}$ do
11:     if $\xi_p$ = Null then
12:         $\xi_n \leftarrow$ generateNode()
13:     else
14:         $\xi_n \leftarrow$ generateNode()
15:         rewireNode($\xi_p$)
16:     end if
17:     if nodeIsValid($\xi_p$) then
18:         addNode($\mathcal{T}$, $\xi_p$)
19:     end if
20:     $\xi_p \leftarrow \xi_n$
21: end while
22: $\mathcal{P}_t \leftarrow$ getBestPath($\mathcal{T}$)
23: $\xi_{t+1} \leftarrow$ getNBV($\mathcal{P}_t$ )
24: Delete$\mathcal{T}$
25: RETURN: $\xi_{t+1}$
```

Function generateNode(·)

```
INPUT:
OUTPUT: New configuration $\xi_{new}$
FUNCTION:
    $\xi_s \leftarrow$ sampleNewConfiguration()
    $\xi_n \leftarrow$ getNearest($\mathcal{T}$, $\xi_s$)
    $\xi_{new} \leftarrow$ randomPropagation($\xi_s$, $\xi_n$)
    if ( isMotionValid($\xi_{new}$, $\xi_n$)) then
        computeVisibility($\xi_{new}$)
        setNodeValidity($\xi_{new}$, $True$)
        return $\xi_{new}$
    else
        generateNode()
    end if
```

Function rewireNode(ξ)

```
 1: INPUT: $\xi$
 2: OUTPUT:
 3: FUNCTION:
 4:     $g_b \leftarrow$ getGain($\xi$)
 5:     $\mathcal{V}_p \leftarrow$ getNeighbors($\mathcal{T}$, $\xi$)
 6:     for all $\xi_p \in \mathcal{V}_p$ do
 7:         if (isMotionValid( $\xi_p$, $\xi$)) then
 8:             $g \leftarrow$ computeGain($\xi_p$, $\xi$)
 9:             if isGainBetterThan($g$, $g_b$) then
10:                 setParent($\xi$, $\xi_p$)
11:                 setGain($\xi$, $g$)
12:                 $g_b \leftarrow g$
13:             end if
14:         end if
15:     end for
16:     for all $\xi_c \in \mathcal{V}_p$ do
17:         if ( isMotionValid($\xi$, $\xi_c$)) then
18:             $g_c \leftarrow$ getGain($\xi_c$)
19:             $g \leftarrow$ computeGain($\xi$, $\xi_c$)
20:             if isGainBetterThan($g$, $g_c$) then
21:                 setParent($\xi_c$, $\xi$)
22:                 setGain($\xi_c$, $g$)
23:                 updateChildrenGain($\xi_c$)
24:             end if
25:         end if
26:     end for
27:     return
```

3.3. KDE for learning the discovered map distribution

In the context of motion planning, solutions for improving the planning results (shorter paths in less computational time) based on defining the sampling set of most promising configurations for achieving an informed tree expansion have been proposed, as in [28] among the others. However, in the motion planning task, the endpoint is established and its knowledge is of utmost importance. Consequently, such solutions cannot be extended to the challenging IG task, where the robot must decide where to go to explore the scenario (maximize the knowledge of the environment).

In this research work, a novel RIG expansion strategy aware of the already acquired data is proposed. The solution is based on learning online the density function of the discovered map and utilizing this knowledge to guide the tree expansion toward the most promising areas. The learning stage makes use of the KDE technique, a non-parametric method to estimate the probability density function of independent and identically distributed samples, that has found broad use in statistics and machine learning.

In particular, let $(s_1, s_2, \ldots, s_n)$ be independent and identically distributed samples drawn from some distribution with an unknown density function f at any given point s. The KDE estimator allows to estimate the value of $f(s)$ as:

$$\widehat{f}_h(s) = \frac{1}{n}\sum_{i=1}^{n} K_h\left(s - s_i\right) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{s - s_i}{h}\right), \tag{6}$$

where K is the kernel, which in the context of this research a Gaussian Kernel was considered, and h is the bandwidth, a tuning parameter.

The KDE technique is a simple yet powerful method that can be exploited for various purposes. It does not require a high computational burden or long training time. Thus, it can be run online even on mobile robots with limited hardware capabilities.

As far as this work is concerned, as discussed above, the KDE technique is used to learn the distribution of the discovered map. In detail, considering an occupancy mapping paradigm, by utilizing as samples the instantiated map cells $\{m_i\}_{i=1}^{n}$, i.e. the discovered map, the value of density function of a point $p \in \mathcal{R}^3$ can be estimated:

$$\widehat{f}_h(p) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{p - p_{m_i}}{h}\right). \tag{7}$$

where $p_{m_i} \in \mathcal{R}^3$ denotes the position of the cell m_i, and K is a Gaussian Kernel.

Consequently, thanks to Eq. (7), the robot can estimate the density function of the discovered map. In particular, the higher the value of $\widehat{f}_h(p)$, the closer the point p is to an already explored area. As detailed in the next section, Eq. (7) is used with a rejection sampling algorithm to generate samples in non-explored regions.

3.4. Informed tree expansion toward unexplored regions

The proposed KDE technique for learning the distribution of the map provides the opportunity to develop an informed map-aware tree expansion strategy that is more likely to find paths that steer the robot toward non-explored regions than by using the standard Voronoi approach.

A sampling strategy shall be designed. Given a distribution, samples are typically drawn by using different techniques so that they are distributed accordingly. As this work is concerned, the goal is to generate samples that do not belong to the given distribution. In particular, since Eq. (7) estimates the distribution of the discovered map, a sampling policy that aims at generating observations in the non-explored regions is required.

To this end, the estimated probability density function given by Eq. (7) was employed for implementing a rejection sampling algorithm [41]. The rejection sampling algorithm is a basic sampling approach that allows drawing random numbers from various distributions. It was used since Eq. (7) only estimates $\widehat{f}_h(p)$, and an explicit form of f is not available, and thus, approaches such as the inverse transform sampling cannot be exploited.

The developed map-aware sampling procedure utilizes a simple workflow, reported in Function kdeSampling. It samples a configuration ξ_s with a uniform distribution over the workspace (denoted as $\mathcal{U}(\mathcal{W})$), and a value $u \in \mathcal{U}[0, \max\{\widehat{f}_h(m_i)\}_{i=1}^{n_d}]$, where $\{(m_i)\}_{i=1}^{n_d}$ denote the map discovered cells and $\mathcal{U}$ denotes the uniform distribution. Then, $\widehat{f}_h(\xi_s)$ is evaluated, and if $u > \widehat{f}_h(\xi_s)$, which means that the configuration ξ_s does not belong to the distribution f, i.e., it is in an unexplored region, the configuration ξ_s is accepted, otherwise it is rejected and process is repeated until a good candidate is found.

Function kdeSampling($\widehat{f}_h$)

```
INPUT: Learned map distribution $\widehat{f}_h$
OUTPUT: Configuration $\xi$
FUNCTION:
    Initialize $\xi = Null$
    while $\xi_s$ = Null do
        $\xi_s \sim \mathcal{U}(\mathcal{W})$
        $u \sim \mathcal{U}[0, \max\{\widehat{f}_h(m_i)\}_{i=1}^{n_d}]$
        if $u > \widehat{f}_h(\xi_s)$ then
            $\xi \leftarrow \xi_s$
        end if
    end while
    return $\xi$
```

This algorithm allows the robot, which has learned the distribution of the discovered map, to sample configurations in the most promising areas for gathering knowledge of the environment. Such configurations can be used to steer the RIG expansion.

To this end, the function **sampleNewConfiguration**() (line 4, Function generateNode), which in IG applications usually exploits only a uniform distribution, has been modified as follows. At each call, the function samples a value $p_s \sim \mathcal{U}[0, 1]$. Then, if p_s is smaller than a pre-defined threshold $t_s \in [0, 1]$, ξ_s is generated using the KDE-based methodology so that it biases the tree toward the unknown region. Otherwise, it is randomly sampled with a uniform distribution over the workspace. For the sake of clarity, the procedure is summarized in Function informedSampling.

Finally, by replacing the function **sampleNewConfiguration** with informedSampling, the RIG informed expansion is realized.

It is worth highlighting that the threshold t_s allows modifying the behavior of the expansion procedure. While small values promote the use of the standard uniform distribution, which leads to the Voronoi-bias and thus a rapid exploration of the workspace, with a threshold close to 1, the here proposed informed expansion is more likely to be used, and the tree is steered toward the most promising areas. By selecting intermediate values, a trade-off between the two behavior can be achieved.

Function informedSampling($\widehat{f}_h$)

```
INPUT: Learned map distribution $\widehat{f}_h$
OUTPUT: Configuration $\xi$
FUNCTION:
    $p_s \sim \mathcal{U}[0, 1]$
    if $p_s < t_s$ then
        $\xi \leftarrow kdeSampling(\widehat{f}_h)$
    else
        $\xi \sim \mathcal{U}(\mathcal{W})$
    end if
    return $\xi$
```

4. Validation

To demonstrate the proposed informed tree expansion methodology it has been applied to the underwater domain, the main research field of the authors, with the aim of providing a proof-of-concept field implementation. The IG planning system was used to make a compact AUV able to autonomously explore an area of interest. In detail, the task of inspecting the sea bottom with an FLS guaranteeing to cover adequate portions of the area was addressed.

Regarding the FLS-based mapping solution, a detailed description of the system exploited in this work can be found in [9]. Briefly, the mapping solution is based on the Octomap framework and allows to create a reconstruction of the seabed from FLS images.

Firstly, the KDE-based strategy for learning the map distribution and sampling configurations in unexplored regions was validated. Seabed reconstructions with a resolution of 0.5 m were simulated. Three examples are depicted on the left side of Fig. 2. A visualization of the estimated map distributions by means of the KDE technique is reported in the middle column. Regarding the utilized KDE estimator, the mlpack library was used [42]. In detail, a Gaussian kernel was employed, and the bandwidth was set to 0.5, corresponding to the map resolution. Finally, the histogram of 5k samples generated by using the above-described rejection sampling algorithm (Function kdeSampling) is reported on the right side of Fig. 2. The histograms show that the proposed methodology can sample new configurations in areas where the robot has not gathered data yet, and thus can be used to steer the RIG algorithm.

Then, the IG system described Section 3 was validated with realistic simulations performed through the Unmanned Underwater Vehicle Simulator (UUV Simulator) [43]. The simulation setup is as follows. A dynamic model of FeelHippo AUV, the robot utilized for the experimental campaign and described in the next section,

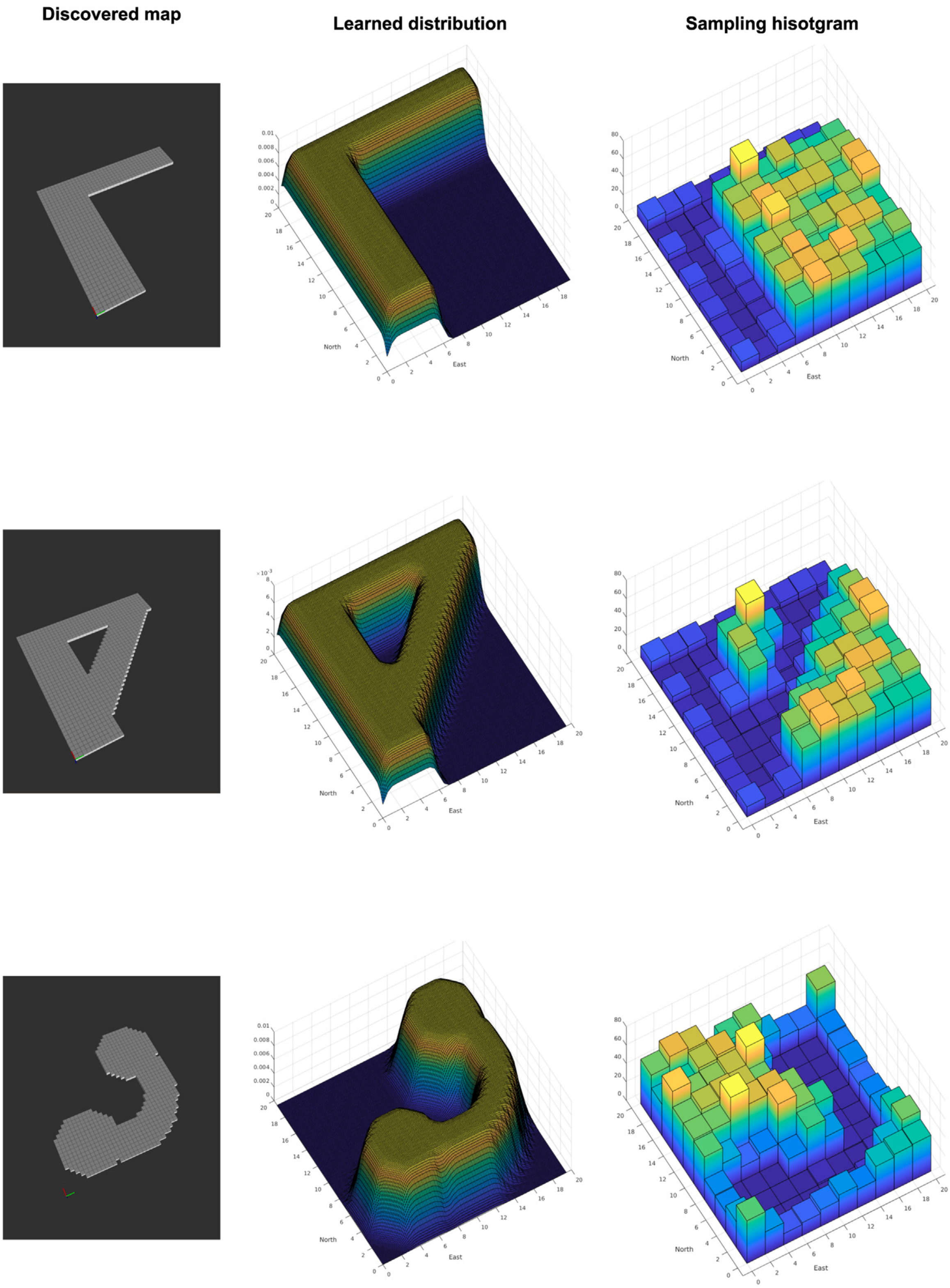


Fig. 2. On the left side, the occupancy map of the simulated seabeds, whose distributions were estimated with the KDE approach (middle column). On the right side, the histogram of 5k samples drawn in the unexplored region.

was implemented and the BlueView M900 2D FLS, mounted on the vehicle, was simulated. For the sake of completeness, the sensor has a horizontal FoV 130° and a vertical FoV of 20°, and it is mounted in front of the vehicle with a tilt angle of 30° w.r.t. the horizontal plane. The range was set to 10 m. To make the simulations as realistic as possible, the AUV was tasked to inspect an area of 36 × 34 m, as the one utilized for the experimental campaign. Besides, underwater seabed inspections are usually conducted with the vehicle flying at a constant altitude, here 2 meters from the seabed. Thus, a robot pose is described by the 2D position and the heading angle, i.e., $\xi = (x, y, \psi)$. Further information about the robot motion models can be found in the Appendix.

For what concerns the RIG algorithm, it was implemented using the Open Motion Planning Library (OMPL) [44] in C++ and was integrated with the FeelHippo AUV software. The maximum computing time for the algorithm was set to 2 s.

The simulation study was performed on a laptop fitted with 16 GB RAM, an Intel Core i7-8750H processor (CPU), and an Nvidia GeForce GTX 1050 Ti card (GPU). It is worth highlighting that the developed planning methodology was executed by exploiting only the CPU.

Regarding the computing time slot, it has been empirically selected. The planning time is a trade-off between the proposed framework's performance and the online computation constraints. Increasing the computational time allows the algorithm to expand the tree more, i.e., creating and evaluating more possible inspection paths. On the other hand, when selecting the computational time for real-world applications, to avoid the robot stopping waiting for a new path to be computed, which in the context of marine robotics leads to an unnecessary waste of energy, the computational time shall be constrained. Finally, it is worth noting that the proposed learning stage was performed before the planning routine and was fast enough not to decrease the number of nodes generated in the tree. Thus, the computing time slot is almost entirely utilized for creating and evaluating as many nodes as possible.

This setup allowed to provide a proof-of-concept validation of the informed RIG expansion. Fig. 3 highlights the effectiveness of the developed learning-sampling methodology. On the top, is a snap of one of the performed surveys using the informed expansion strategy. In detail, the green lines delimit the inspection area. The gray voxels depict the discovered sea bottom, the yellow lines represent the generated tree, and the blue line highlights the selected best path. On the left-bottom side, a representation of the discovered map distribution learned online by the robot using the KDE technique. Such a distribution is employed by the sampling algorithm to generate samples in unexplored regions. For the sake of clarity, the histogram of 1k samples generated accordingly is reported on the right-bottom side. The learning-sampling method was used to generate the RIG tree, reported in yellow on the top side of the image, which is clearly steered toward the most promising areas.

Turning to quantitative analysis, three configurations of the algorithm were tested. The Voronoi-based expansion (the standard RRT strategy) was compared with the hereby proposed informed expansion that utilizes the KDE methodology for estimating the discovered map. In addition, a balanced solution that makes use of both the Voronoi and the informed expansions was investigated. With regard to the informed sampling strategy (Function informedSampling), such configurations are achieved by considering the thresholds $t_s = 0$, $t_s = 1$, and $t_s = 0.5$, respectively.

Being the IG system fully probabilistic, for each algorithm configuration, denoted as *RIG - Voronoi* ($t_s = 0$), *RIG - Informed* ($t_s = 1$), and *RIG - Balanced*($t_s = 0.5$), 10 simulations with random starting AUV positions and orientation were performed, and the performance was evaluated in terms of the path length required to achieved the 80%, 90%, and 95% of coverage of the area. The mean values of the 10 runs were considered. The results are summarized in Table 1.

Thanks to the developed IG system, the AUV managed to autonomously acquire the data over the area of interest. None of the runs failed and the 95% of coverage was achieved.

Table 1
Simulation outcomes.

(a) Results for 80% of coverage

Method	Coverage level 80%	
	Path length mean [m]	Path length standard deviation [m]
RIG - Voronoi	85.69	12.88
RIG - Informed	79.63	10.23
RIG - Balanced	75.15	11.99

(b) Results for 90% of coverage

Method	Coverage level 90%	
	Path length mean [m]	Path length standard deviation [m]
RIG - Voronoi	116.98	21.04
RIG - Informed	99.18	10.68
RIG - Balanced	100.34	7.67

(c) Results for 95% of coverage

Method	Coverage level 95%	
	Path length mean [m]	Path length standard deviation [m]
RIG - Voronoi	140.00	26.93
RIG - Informed	120.71	20.31
RIG - Balanced	118.54	12.58

The proposed informed expansion enhances the performance of the RIG algorithm. The considered coverage levels are achieved with shorter paths when the KDE methodology is used for generating the samples for expanding the tree toward unexplored regions than exploiting the state-of-the-art Voronoi bias.

Table 1 shows that the best solution is using a balanced expansion strategy. That is, the planning algorithm has an equal probability (50-50 chance) of using either the Voronoi-biased or the Informed expansion at each iteration. Thus, a trade-off between the exploration of the workspace and a guided expansion toward the goal, i.e., unexplored regions, is exploited. Using only the here proposed KDE-based methodology for sampling new configuration improves the performance compared to the Voronoi bias. However, only focusing on the expansion toward new areas prevents the tree from exploring the space. Thus, it limits the IG algorithm performance.

In fact, as shown in Table 1, the Balanced and Informed strategies showed similar performance, better than the standard Voronoi approach, in covering the 80% and 90% of the area of interest. The mean length is significantly decreased. The *RIG - Informed* achieved a reduction of about 6 m and 16 m (7% and 13.7% of the total mean path length) than the *RIG - Voronoi* for covering the 80% and 90% of the area, respectively. The *RIG - Informed* showed the best results. It reduced the path length of about 10 m and 16 m, 11.7% and 13.7% of the mean path length.

The higher the target coverage level, the more the RIG benefits from the developed Informed expansion solution. In fact, when the 95% of coverage is demanded, which is the most challenging task, the KDE-based sampling strategy leads to a mean path length reduction of about 20 m (15% of the total mean path length, see Table 1) with respect to the state-of-the-art Voronoi strategy, proving the advantages of the proposed expansion methodology.

For the sake of completeness, for each algorithm configuration, an example of the tree expansion is reported. In detail, Fig. 4 reports the use of the Voronoi-biased expansion, while Fig. 5 shows the effect of the proposed Informed strategy. Finally, Fig. 6 depicts the balanced approach and highlights its impact. This latter solution combines the benefits of the aforementioned strategies. It drives the tree toward the unexplored regions while maintaining a fast exploration of the workspace. These images help to

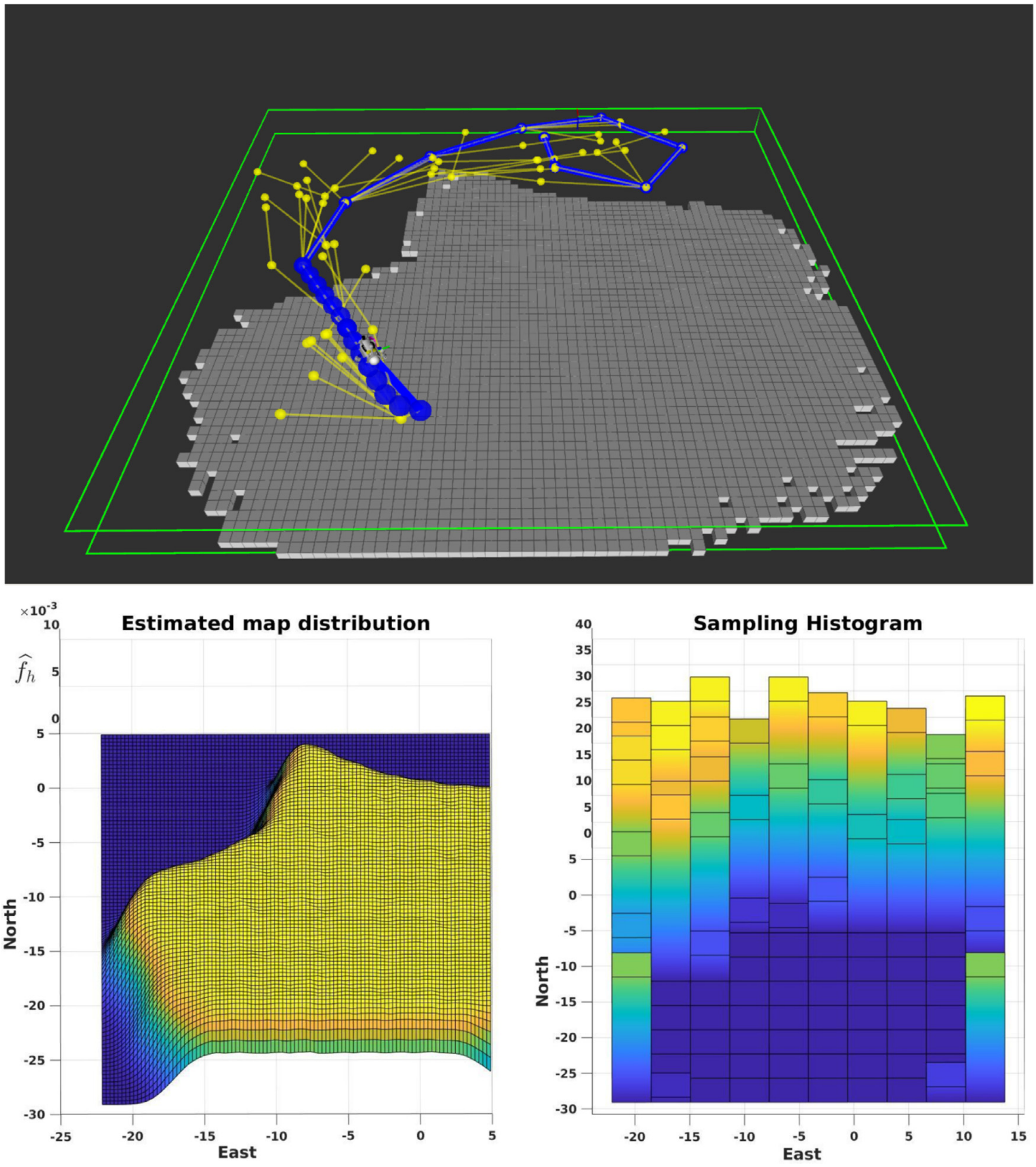


Fig. 3. The proposed learning-sampling methodology utilized to develop an informed expansion. On the top, a snap collected during one of the survey depicting the inspection area (delimited by the green lines) and the discovered map, reported in gray. The map distribution learned online by the robot with the KDE strategy is shown on the left-bottom side. On the right, the histogram of 1k samples generated in the non-explored area thanks to the learning-sampling methodology is reported. The RIG informed expansion is realized by means of this learning-sampling, which allows to steer the tree (shown in yellow) toward most promising regions. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

represent and analyze the effect of the presented solutions. As discussed above, by using a uniform distribution for generating new samples, the Voronoi-biased expansion (Fig. 4) ensures that the tree rapidly covers the area of interest. However, it does not consider the already discovered map, and since it could generate non-informative branches, it could lead to longer paths. In fact, as shown in Fig. 4, generated inspection tree (in yellow) is well-distributed over the area of interest, but it has various branches over the already discovered map that do not allow the robot to acquire any information. On the other hand, the here proposed informed expansion methodology can overcome this limitation. As depicted in Fig. 5, it steers the tree toward new areas, focusing the RIG algorithm on looking in the most promising areas. Nevertheless, by using only the informed strategy, the RIG algorithm generates greedy paths that do not explore the workspace and head only toward unexplored regions. The Balanced configuration takes advantage of two strategies. Fig. 6 highlights its effects and helps understand the achieved results. It allows the tree to explore rapidly and efficiently the workspace and simultaneously generate samples in the most informative areas, enhancing the performance of the RIG algorithm. Fig. 6 depicts this behavior: the generated tree, shown in yellow, is well-distributed over the area of interest, and the branches are steered toward the unexplored regions. It is worth noting that this does not hold for the state-of-the-art Voronoi-based expansion, where various branches of the tree end on non-informative regions, see Fig. 4.

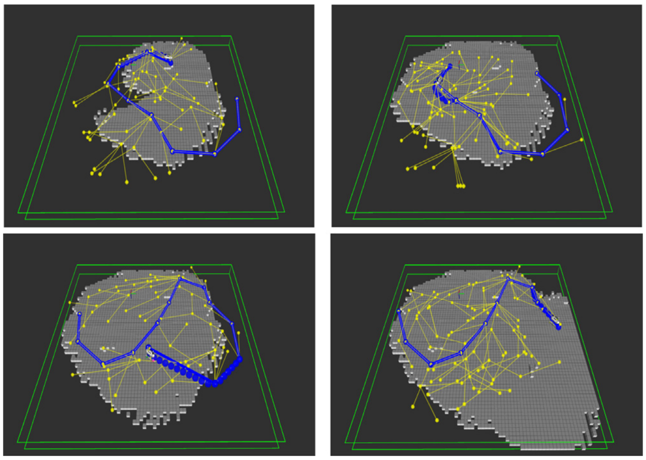


Fig. 4. Examples of the generated inspection tree (in yellow) by using the Voronoi-bias during the expansion. Each yellow point represents an AUV configuration. The green lines delimit the inspection area, while the blue one depicts the selected inspection path, i.e., the best branch. The Voronoi bias ensures that the tree rapidly explores the workspace. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

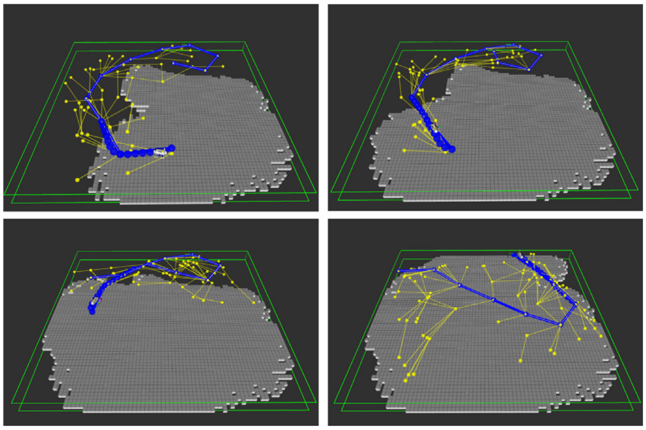


Fig. 5. The tree expansion when only the KDE-based Informed strategy is exploited. The tree is biased through non-explored regions.

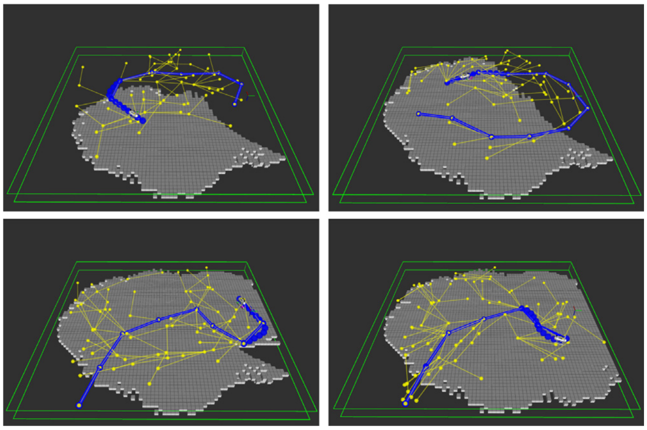


Fig. 6. Four snapshots depicting the tree generated using the balanced expansion approach. By randomly exploiting both the Voronoi and the Informed bias, it enhanced the performance of conducted autonomous coverage surveys.

Table 2
FeelHippo AUV main features.

Weight [kg]	35
Dimensions [mm]	600 × 640 × 500
Controlled DOFs	5
Thrusters	6
Maximum depth [m]	35
Maximum longitudinal speed [m/s]	1
Battery life [h]	4

5. Experimental results

An experimental campaign for testing and validating the proposed IG system and the informed RIG algorithm was conducted in July 2021 at the Naval Support and Experimentation Center (Centro di Supporto e Sperimentazione Navale - CSSN) basin in La Spezia, Italy. The robot utilized for sea trials was FeelHippo AUV [45], a compact and lightweight vehicle developed by the Department of Industrial Engineering of the University of Florence (UNIFI DIEF), Italy. FeelHippo AUV main characteristics are summarized in Table 2, while a hardware overview is reported in Fig. 7. In detail, it is equipped with the following suite of sensors:

- 7th Generation Intel i7 U-series Processor (2.60 GHz) (main computer);
- NVIDIA Jetson Nano (payload computer);
- U-blox 7P precision Global Navigation Satellite System (GNSS);
- Orientus Advanced Navigation Attitude and Heading Reference System (AHRS);
- KVH DSP 1760 single-axis high precision Fiber Optic Gyroscope (FOG);
- Nortek DVL1000 Doppler Velocity Log (DVL), measuring linear velocity and acting as Depth Sensor (DS);
- EvoLogics S2CR 18/34 acoustic modem;
- Teledyne BlueView M900 2D FLS.

The main computer (Intel processor), was used to run online the IG planning system and KDE-based learning technique, implemented in C++ using the OMPL and the mlpack libraries. Regarding the GNC strategies utilized by FeelHippo AUV, depicted during the conducted experimental campaign in Fig. 8, the interest reader can find detailed information in [46–48]. Briefly, for clarity, in the context of this research work, the AUV localization relied on the following navigation strategy. The non-linear observer proposed in [47] was used to compute the vehicle attitude exploiting IMU and FOG data. Then, the attitude estimate was used with DVL measurements in a dead reckoning fashion to calculate the AUV position, initialized with GNSS measurements when the vehicle was on the sea surface. As shown in [46,47, 49], this strategy can achieve accurate AUV localization by using high-grade sensors.

Since conducting experimental sea trials is challenging, and time and cost expansive, two configurations of the view planning algorithm were tested during the experimental campaign. The *RIG - Balanced*, that fuses the Voronoi-bias with the developed informed strategy and emerged as the best solution during the validation process, was compared with the *RIG - Voronoi* that utilizes the state-of-the-art Voronoi-based expansion. The algorithm maximum computing time was 2 s.

Thus, 5 coverage surveys of an area of 36 × 34 m for each configuration were conducted. The surveys were performed autonomously by FeelHippo AUV utilizing the BlueView M900 2D FLS, which was mounted in front of the vehicle with a tilt angle of 30° w.r.t. the horizontal plane and whose range was set to 10 m. As for the simulations, the surveys were evaluated in terms

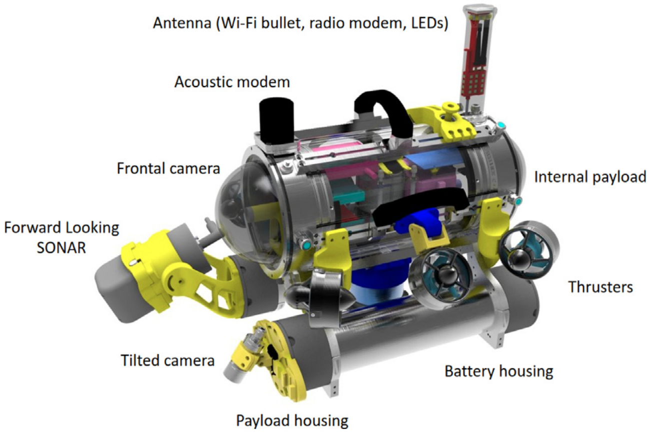


Fig. 7. FeelHippo AUV hardware overview.

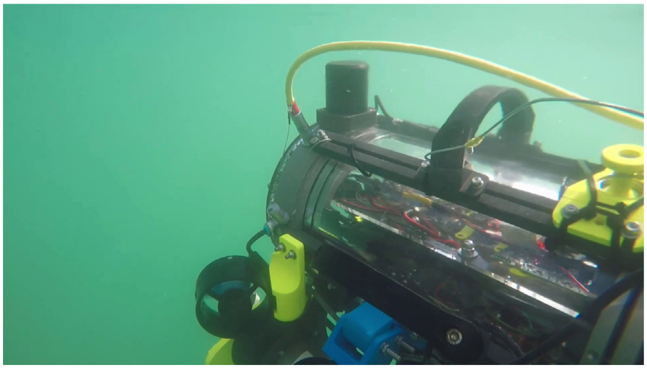


Fig. 8. FeelHippo AUV during the conducted experimental campaign, autonomously performing an underwater survey thanks to the developed IG system.

Table 3
Experimental results.

Method	Path length mean [m]	Path length standard deviation [m]
(a) Results for 80% of coverage		
	Coverage level 80%	
RIG - Voronoi	85.49	5.91
RIG - Balanced	67.12	11.58
(b) Results for 90% of coverage		
	Coverage level 90%	
RIG - Voronoi	123.56	21.63
RIG - Balanced	91.76	14.84
(c) Results for 95% of coverage		
	Coverage level 95%	
RIG - Voronoi	145.08	5.93
RIG - Balanced	126.68	11.00

of path length required to achieved the 80%, 90%, and 95% of coverage of the area, Table 3 reports the mean value as well as the standard deviation of the 5 trials.

Firstly, it is worth highlighting that by using the proposed IG strategy, the AUV monitors the acquired data and can ensure to gather information over the area of interest without knowing the scenario in advance, avoiding expensive repeated attempts. Besides, a skilled operator to design the AUV mission is not required: the robot actively considers as feedbacks the measurements of the payload sensor and decides the best actions to fulfill the task. Thus, an AUV exploiting the developed IG system simplifies the data gathering process and could be an excellent solution for science users and non-trained people for exploring the environment.

Being fully probabilistic, the results can vary. As expected from the simulations, the *RIG - Balanced* outperformed the state-of-the-art *RIG - Voronoi* in all the benchmarks, see Table 3. By mixing the uniform distribution and the map-aware sampling policy, the balanced configuration guides the expansion of the tree toward unexplored regions while guaranteeing it well-explores the workspace. In fact, thanks to the proposed KDE-based methodology, the vehicle learns where to sample new configurations in the most promising areas and enhances the RIG algorithm performance. The *RIG - Balanced* reached a mean path length reduction of 18.38 m, 31.80 m, and 18.40 m (21.5%, 25.7%, and 12.7% of the total mean path length) for the 80%, 90%, and 95% of coverage, respectively. These results are in accordance with the validation experiments reported in Table 1. The more the area is covered, the more challenging the task, and the more the RIG benefits from the proposed expansion methodology.

It is worth highlighting that the proposed informed expansion strategy, based on learning the discovered map distribution, was run online on the main computer mounted on FeelHippo AUV. It does not require dedicated hardware and can be used even on compact and lightweight robots.

In conclusion, the conducted experimental campaign demonstrates, through several trials, the effectiveness of the novel informed expansion methodology that can be easily implemented and used on a real system acting in unknown environments.

6. Conclusions and future trends

This paper proposes an informed tree expansion for IG tasks. While in the context of motion planning several solutions that define the regions for sampling the most promising configurations and steering the tree toward the goal have been developed in the last years, a goal-oriented RIG expansion has not been investigated yet. In this work, for the first time to the best of the author's knowledge, a novel informed expansion strategy that steers the tree toward unexplored areas by learning online the distribution of the discovered map is proposed.

In particular, the KDE technique, which has found a large use in statistics and machine learning, is exploited in the learning stage to estimate the distribution density function of the acquired data, which, in the context of this work, is the discovered occupancy map. The estimated density function is combined with a rejection sampling algorithm, allowing to draw new samples in the unexplored regions of the workspace. The learning-sampling methodology realizes an informed expansion: by biasing the expansion with the configurations sampled with the proposed strategy the tree is steered toward the most promising regions where new information can be gathered. That is, the robot learns where to sample the configurations for guiding the RIG algorithm, improving the performance, i.e., reducing the traveled path for exploring the workspace.

A proof-of-concept validation was provided with realistic simulations as well as an experimental campaign, conducted in the context of underwater robotics, the research field of the authors. The developed RIG algorithm was utilized for making an AUV able to autonomously conduct underwater inspections in an unknown area with an FLS. The performance of the RIG solution using the novel informed expansion strategy is compared with the state-of-the-art RRT sampling policy based on a uniform distribution, which leads to a Voronoi-biased expansion.

Simulations as well as real experimental results, collected during sea trials carried out in July 2021 at the CSSN basin in La Spezia, Italy, demonstrate that the proposed strategy enhances the performance of the developed RIG algorithm. In particular, the results highlight that the state-of-the-art Vornoi-based expansion was outperformed by the informed approach. However, mixing the two strategies emerged as the best configuration. In fact, using both the uniform distribution and the map-aware sampling policy offers a good trade-off between a goal-guided expansion and the exploration of the workspace.

The developed informed expansion strategy paves the way to several future trends. Firstly, the here proposed methodology could be applied to different domains and scenarios. In fact, it could improve the exploration capabilities of aerial and ground robots in various applications. Then, the integration of the informed expansion strategy with an object detection solution for exploration and simultaneous object search tasks could be investigated. Besides, the learning strategy for estimating the distribution of the discovered map can be utilized for a fast information gain evaluation procedure [50] or developing a collision checking routine for validating the safety of planned paths.

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.

Data availability

The authors do not have permission to share data.

Acknowledgments

The research leading to these results has also been supported by the European project EUMarineRobots, which received funding from the European Unions Horizon 2020 research and innovation program under grant agreement N° 731103, as well as the H2020 European project PASSport, which received funding from the European GNSS Agency (GSA), under grant agreement N° 101004234 (H2020-SPACE-EGNSS-2020).

Appendix. AUV motion modeling for planning

Generally, the pose of an AUV in 3D is expressed by means of six variables, and its motion can be expressed through kinematic and dynamic models. According to [51], by considering the AUV as a rigid body, the AUV pose with respect to the North-East-Down (NED) frame ($\langle \mathrm{N} \rangle$) is expressed with $\boldsymbol{\eta} = \left[{}^{N}\boldsymbol{\eta}_1^\top \ \boldsymbol{\eta}_2^\top\right]^\top \in \mathbb{R}^6$, where ${}^{N}\boldsymbol{\eta}_1 \in \mathbb{R}^3$ indicates the position of the vehicle with respect to the NED frame and $\boldsymbol{\eta}_2 \in \mathbb{R}^3$ its orientation. The AUV linear and angular velocities with respect to the body reference frame ($\langle b \rangle$) are represented with ${}^{b}\boldsymbol{\nu} = \left[{}^{b}\boldsymbol{\nu}_1^\top \ {}^{b}\boldsymbol{\nu}_2^\top\right]^\top$, where ${}^{b}\boldsymbol{\nu}_1^\top = [u \ v \ w]$, denotes the linear velocities and ${}^{b}\boldsymbol{\nu}_2^\top = [p \ q \ r]$ are the angular counterparts along the axes of the body frame, namely surge, sway, and heave. Thus, the vehicle kinematic model is expressed as:

$$\begin{pmatrix} {}^{N}\dot{\boldsymbol{\eta}}_1 \\ \dot{\boldsymbol{\eta}}_2 \end{pmatrix} = \begin{bmatrix} J_1\left(\boldsymbol{\eta}_2\right) & 0_{3\times 3} \\ 0_{3\times 3} & J_2\left(\boldsymbol{\eta}_2\right) \end{bmatrix} \begin{pmatrix} {}^{b}\boldsymbol{\nu}_1 \\ {}^{b}\boldsymbol{\nu}_2 \end{pmatrix}, \tag{8}$$

where

$$J_1\left(\boldsymbol{\eta}_2\right) = {}^{N}R_b^N = R_z(\psi)R_y(\theta)R_x(\phi)$$
$$= \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi c_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix}, \tag{9}$$

and

$$J_2\left(\boldsymbol{\eta}_2\right) = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & \frac{s_\phi}{c_\theta} & \frac{c_\phi}{c_\theta} \end{bmatrix}. \tag{10}$$

In a compact notation, Eq. (8) can be expressed as:

$$\dot{\boldsymbol{\eta}} = J(\boldsymbol{\eta})\boldsymbol{\nu}, \tag{11}$$

where

$$J(\boldsymbol{\eta}) = \begin{bmatrix} J_1\left(\boldsymbol{\eta}_2\right) & 0_{3\times 3} \\ 0_{3\times 3} & J_2\left(\boldsymbol{\eta}_2\right) \end{bmatrix}. \tag{12}$$

In several applications, AUVs usually conduct surveys at constant depth or altitude. Vehicles such as FeelHippo AUV have the roll and pitch dynamics hydrostatically stable, and given that the missions considered in this work do not excite these degrees of freedom, the AUV navigates with roll and pitch angles almost zero with negligible variations. Additionally, to reduce energy consumption, FeelHippo AUV usually moves about the surge axis and only rotates about the heave axis, i.e., it only changes its orientation. As a consequence of these statements, the kinematic motion on a horizontal plane of FeelHippo AUV can be represented as a simple car-like vehicle:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} u\cos(\psi) \\ u\sin(\psi) \\ r \end{bmatrix}. \tag{13}$$

While kinematic models only use geometric equations that relate the vehicle positions and velocities, dynamic models consider the force and torques that create the motion. As a consequence, this latter class of models describes the AUV motion constraints more accurately. Further information about these models can be found in [51]. As far as this work is concerned, since the IPP problem was tackled, only the kinematic model was considered. In fact, to allow the use of a rewiring strategy, which consists of a routine that checks whether a new node could improve the cost of neighbor nodes, a steering function, which returns the optimum path between two states, is needed. When using motion constraints, computing a steering function means addressing a two-point BVP. That is, it corresponds to solving a differential equation under certain boundary conditions [38], which is generally a difficult problem.

Describing the AUV motion with kinematic model of Eq. (13), the configuration space ($\mathcal{C}$) is $\mathcal{C} = SE(2)$, i.e., a configuration $\xi \in SE(2)$. Assuming that the vehicle navigates with a constant surge speed u, and by considering that it has a maximum turning rate r_{max}, which defines a minimum turning radius R_{min}, the AUV kinematic motion can be described with the Dubins vehicle model. In particular, by using this model, the shortest path between two configurations consists of circular arcs of maximum curvature and straight line segments. Thus, the shortest path can be obtained by combining three possible maneuvers: straight (S), right turn (R), and left turn (L). Thus, the shortest path will always be at least one of the six combinations: RSR, RSL, LSR, LSL, RLR, LRL. The Dubins vehicle model is of particular interest in planning tasks since it can be used for both generating new configurations, which in RRT-based solutions corresponds to expanding the tree, and works as a steering function for the rewiring routine.

In conclusion, as a consequence of these considerations, the robot motion was modeled by using the Dubins vehicle model.

References

[1] F. Colas, S. Mahesh, F. Pomerleau, M. Liu, R. Siegwart, 3D path planning and execution for search and rescue ground robots, in: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2013, pp. 722–727.

[2] Y. Wang, M. Ramezani, M. Fallon, Actively mapping industrial structures with information gain-based planning on a quadruped robot, in: 2020 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2020, pp. 8609–8615.

[3] M. Dharmadhikari, T. Dang, L. Solanka, J. Loje, H. Nguyen, N. Khedekar, K. Alexis, Motion primitives-based path planning for fast and agile exploration using aerial robots, in: 2020 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2020, pp. 179–185.

[4] E. Vidal, N. Palomeras, K. Istenič, N. Gracias, M. Carreras, Multisensor online 3D view planning for autonomous underwater exploration, J. Field Robotics 37 (6) (2020) 1123–1147.

[5] L. Zacchini, M. Franchi, V. Manzari, M. Pagliai, N. Secciani, A. Topini, M. Stifani, A. Ridolfi, Forward-looking sonar CNN-based automatic target recognition: an experimental campaign with FeelHippo AUV, in: 2020 IEEE/OES Autonomous Underwater Vehicles Symposium, AUV, IEEE, 2020, pp. 1–6.

[6] S. Isler, R. Sabzevari, J. Delmerico, D. Scaramuzza, An information gain formulation for active volumetric 3D reconstruction, in: 2016 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2016, pp. 3477–3484.

[7] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, R. Siegwart, Receding horizon "next-best-view" planner for 3d exploration, in: 2016 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2016, pp. 1462–1468.

[8] R.N. Smith, M. Schwager, S.L. Smith, B.H. Jones, D. Rus, G.S. Sukhatme, Persistent ocean monitoring with underwater gliders: Adapting sampling resolution, J. Field Robotics 28 (5) (2011) 714–741.

[9] L. Zacchini, M. Franchi, A. Ridolfi, Sensor-driven autonomous underwater inspections: A receding-horizon RRT-based view planning solution for AUVs, J. Field Robotics 39 (5) (2022) 499–527.

[10] E. Galceran, M. Carreras, A survey on coverage path planning for robotics, Robot. Auton. Syst. 61 (12) (2013) 1258–1276.

[11] J.I. Vasquez-Gomez, L.E. Sucar, R. Murrieta-Cid, View planning for 3D object reconstruction with a mobile manipulator robot, in: 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2014, pp. 4227–4233.

[12] S. Kriegel, C. Rink, T. Bodenmüller, M. Suppa, Efficient next-best-scan planning for autonomous 3D surface reconstruction of unknown objects, J. Real-Time Image Process. 10 (4) (2015) 611–631.

[13] R. Marchant, F. Ramos, Bayesian optimisation for intelligent environmental monitoring, in: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2012, pp. 2242–2249.

[14] G.A. Hollinger, G.S. Sukhatme, Sampling-based robotic information gathering algorithms, Int. J. Robot. Res. 33 (9) (2014) 1271–1287.

[15] L. Schmid, M. Pantic, R. Khanna, L. Ott, R. Siegwart, J. Nieto, An efficient sampling-based method for online informative path planning in unknown environments, IEEE Robot. Autom. Lett. 5 (2) (2020) 1500–1507.

[16] M. Popović, T. Vidal-Calleja, J.J. Chung, J. Nieto, R. Siegwart, Informative path planning for active field mapping under localization uncertainty, in: 2020 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2020, pp. 10751–10757.

[17] C. Xiong, D. Chen, D. Lu, Z. Zeng, L. Lian, Path planning of multiple autonomous marine vehicles for adaptive sampling using Voronoi-based ant colony optimization, Robot. Auton. Syst. 115 (2019) 90–103.

[18] J. Binney, G.S. Sukhatme, Branch and bound for informative path planning, in: 2012 IEEE International Conference on Robotics and Automation, IEEE, 2012, pp. 2147–2154.

[19] K.-C. Ma, L. Liu, H.K. Heidarsson, G.S. Sukhatme, Data-driven learning and planning for environmental sampling, J. Field Robotics 35 (5) (2018) 643–661.

[20] A. Viseras, D. Shutin, L. Merino, Robotic active information gathering for spatial field reconstruction with rapidly-exploring random trees and online learning of Gaussian processes, Sensors 19 (5) (2019) 1016.

[21] C. Papachristos, S. Khattak, K. Alexis, Uncertainty-aware receding horizon exploration and mapping using aerial robots, in: 2017 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2017, pp. 4568–4575.

[22] T. Dang, C. Papachristos, K. Alexis, Autonomous exploration and simultaneous object search using aerial robots, in: 2018 IEEE Aerospace Conference, IEEE, 2018, pp. 1–7.

[23] S. Karaman, E. Frazzoli, Sampling-based algorithms for optimal motion planning, Int. J. Robot. Res. 30 (7) (2011) 846–894.

[24] S.M. LaValle, J.J. Kuffner Jr., Randomized kinodynamic planning, Int. J. Robot. Res. 20 (5) (2001) 378–400.

[25] Y. Kompis, L. Bartolomei, R. Mascaro, L. Teixeira, M. Chli, Informed sampling exploration path planner for 3d reconstruction of large scenes, IEEE Robot. Autom. Lett. 6 (4) (2021) 7893–7900.

[26] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, J. Nieto, Voxblox: Incremental 3D euclidean signed distance fields for on-board mav planning, in: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2017, pp. 1366–1373.

[27] M. Ghaffari Jadidi, J. Valls Miro, G. Dissanayake, Sampling-based incremental information gathering with applications to robotic exploration and environmental monitoring, Int. J. Robot. Res. 38 (6) (2019) 658–685.

[28] J.D. Gammell, S.S. Srinivasa, T.D. Barfoot, Informed RRT*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic, in: 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2014, pp. 2997–3004.

[29] L. Schmid, C. Ni, Y. Zhong, R. Siegwart, O. Andersson, Fast and compute-efficient sampling-based local exploration planning via distribution learning, 2022, arXiv preprint arXiv:2202.13715.

[30] Y. Kantaros, B. Schlotfeldt, N. Atanasov, G.J. Pappas, Asymptotically optimal planning for non-myopic multi-robot information gathering., in: Robotics: Science and Systems, 2019, pp. 22–26.

[31] Y. Kantaros, B. Schlotfeldt, N. Atanasov, G.J. Pappas, Sampling-based planning for non-myopic multi-robot information gathering, Auton. Robots 45 (7) (2021) 1029–1046.

[32] S. Bai, J. Wang, F. Chen, B. Englot, Information-theoretic exploration with Bayesian optimization, in: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2016, pp. 1816–1822.

[33] M. Ghaffari Jadidi, J. Valls Miro, G. Dissanayake, Gaussian processes autonomous mapping and exploration for range-sensing mobile robots, Auton. Robots 42 (2) (2018) 273–290.

[34] M.G. Jadidi, J.V. Miro, G. Dissanayake, Mutual information-based exploration on continuous occupancy maps, in: RSJ International Conference on Intelligent Robots and Systems, IROS, 2015, pp. 6086–6092.

[35] B.A. Turlach, Bandwidth selection in kernel density estimation: A review, in: CORE and Institut de Statistique, Citeseer, 1993.

[36] A. Elgammal, R. Duraiswami, L.S. Davis, Efficient kernel density estimation using the fast gauss transform with applications to color modeling and tracking, IEEE Trans. Pattern Anal. Mach. Intell. 25 (11) (2003) 1499–1504.

[37] Q. Wang, L. Zhou, X. Chen, Kernel density estimation and convolutional neural networks for the recognition of multi-font numbered musical notation, Electronics 11 (21) (2022) 3592.

[38] Y. Li, Z. Littlefield, K.E. Bekris, Asymptotically optimal sampling-based kinodynamic planning, Int. J. Robot. Res. 35 (5) (2016) 528–564.

[39] A. Hornung, K.M. Wurm, M. Bennewitz, C. Stachniss, W. Burgard, OctoMap: An efficient probabilistic 3D mapping framework based on octrees, Auton. Robots 34 (3) (2013) 189–206.

[40] A. Krause, C. Guestrin, Submodularity and its applications in optimized information gathering, ACM Trans. Intell. Syst. Technol. 2 (4) (2011) 1–20.

[41] M.T. Wells, G. Casella, C.P. Robert, Generalized accept-reject sampling schemes, in: A Festschrift for Herman Rubin, Institute of Mathematical Statistics, 2004, pp. 342–347.

[42] R.R. Curtin, M. Edel, M. Lozhnikov, Y. Mentekidis, S. Ghaisas, S. Zhang, Mlpack 3: a fast, flexible machine learning library, J. Open Source Softw. 3 (26) (2018) 726.

[43] M.M.M. Manhães, S.A. Scherer, M. Voss, L.R. Douat, T. Rauschenbach, UUV simulator: A gazebo-based package for underwater intervention and multi-robot simulation, in: OCEANS 2016 MTS/IEEE Monterey, IEEE, 2016, http://dx.doi.org/10.1109/oceans.2016.7761080.

[44] I.A. Şucan, M. Moll, L.E. Kavraki, The open motion planning library, IEEE Robot. Autom. Mag. 19 (4) (2012) 72–82, http://dx.doi.org/10.1109/MRA.2012.2205651, https://ompl.kavrakilab.org (last visit October 2021).

[45] B. Allotta, R. Conti, R. Costanzi, F. Fanelli, J. Gelli, E. Meli, N. Monni, A. Ridolfi, A. Rindi, A low cost autonomous underwater vehicle for patrolling and monitoring, Proc. Inst. Mech. Eng. M 231 (3) (2017) 740–749.

[46] B. Allotta, A. Caiti, R. Costanzi, F. Fanelli, D. Fenucci, E. Meli, A. Ridolfi, A new AUV navigation system exploiting unscented Kalman filter, Ocean Eng. 113 (2016) 121–132.

[47] R. Costanzi, F. Fanelli, N. Monni, A. Ridolfi, B. Allotta, An attitude estimation algorithm for mobile robots under unknown magnetic disturbances, IEEE/ASME Trans. Mechatronics 21 (4) (2016) 1900–1911.

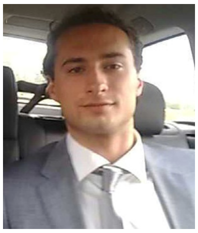
[48] L. Zacchini, V. Calabrò, M. Candeloro, F. Fanelli, A. Ridolfi, F. Dukan, Novel noncontinuous carouseling approaches for MEMS-based north seeking using Kalman filter: Theory, simulations, and preliminary experimental evaluation, IEEE/ASME Trans. Mechatronics 25 (5) (2020) 2437–2448.

[49] M. Franchi, A. Ridolfi, B. Allotta, Underwater navigation with 2D forward looking SONAR: An adaptive unscented Kalman filter-based strategy for AUVs, J. Field Robotics 38 (3) (2021) 355–385.

[50] L. Zacchini, M. Franchi, A. Bucci, N. Secciani, A. Ridolfi, Randomized MPC for view planning in AUV seabed inspections, in: OCEANS 2021: San Diego–Porto, IEEE, 2021, pp. 1–6.

[51] G. Antonelli, Underwater Robots, Vol. 3, Springer, 2014.

Leonardo Zacchini received the M.S. degree in Automation and Control Engineering, and the Ph.D. degree with a thesis focused on Autonomous Inspection Strategies for Underwater Robots from the University of Florence, Italy, in 2018 and 2021, respectively. He is currently a Postdoctoral Researcher in Robotics at the University of Florence, Italy. His research interests include guidance, navigation, and control systems for mobile robots, underwater robotics, robotics exploration, motion planning, and AI for robotics.



Alessandro Ridolfi is a Ph.D. Researcher and Assistant Professor of Machine Theory and Robotics with the School of Engineering, Department of Industrial Engineering, University of Florence, Italy. His current research interests include biorobotics, vehicle dynamics, mechanical systems modeling, robotics, and underwater robotics.

Benedetto Allotta received the Ph.D. degree. He is a Full Professor of Robotics with the School of Engineering, University of Florence, Firenze, Italy, and the Co-Founder and Coordinator of the Laboratory of Mechatronics and Dynamic Modeling. His current research interests include underwater robotics, biorobotics, sensor-based navigation of vehicles, hardware-in-the-loop simulation, and automation in transport system.