remote sensing

MDPI

Article

Active SLAM for Autonomous Underwater Exploration

Narcís Palomeras [1,*], Marc Carreras [1] and Juan Andrade-Cetto [2]

[1] Computer Vision and Robotics Institute (VICOROB), Universitat de Girona, 17003 Girona, Spain; marc.carreras@udg.edu

[2] Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Llorens Artigas 4-6, 08028 Barcelona, Spain; cetto@iri.upc.edu

* Correspondence: npalomer@eia.udg.edu

Received: 25 October 2019; Accepted: 27 November 2019; Published: 28 November 2019

Abstract: Exploration of a complex underwater environment without an a priori map is beyond the state of the art for autonomous underwater vehicles (AUVs). Despite several efforts regarding simultaneous localization and mapping (SLAM) and view planning, there is no exploration framework, tailored to underwater vehicles, that faces exploration combining mapping, active localization, and view planning in a unified way. We propose an exploration framework, based on an active SLAM strategy, that combines three main elements: a view planner, an iterative closest point algorithm (ICP)-based pose-graph SLAM algorithm, and an action selection mechanism that makes use of the joint map and state entropy reduction. To demonstrate the benefits of the active SLAM strategy, several tests were conducted with the Girona 500 AUV, both in simulation and in the real world. The article shows how the proposed framework makes it possible to plan exploratory trajectories that keep the vehicle's uncertainty bounded; thus, creating more consistent maps.

Keywords: autonomous underwater vehicles; robot exploration; active SLAM; view planning

1. Introduction

Most underwater robotics applications, whether scientific, industrial, or safety related, that are conducted in complex environments for which an a priori map is not available, are nowadays carried out either by divers or with remotely operated vehicles. To carry out these tasks autonomously, the underwater vehicle must have at least the capabilities of mapping, localization, and planning. Moreover, these three capabilities must work together in a coordinated exploration framework. By autonomous exploration we mean the ability for the system to decide the best trajectory to fully cover the explored scene whilst keeping the vehicle correctly localized. These two problems must be solved jointly.

In the literature there are several methods that provide optimal view planning for objects or scene reconstruction whilst the mapping sensor is accurately positioned [1]. However, in the underwater and the underground domains the absence of absolute positioning systems, the low reliability of communications, and the bad visibility, make it difficult to have the vehicle well localized during the exploration session [2], adding positional drift over time.

One way to overcome this problem is to solve the coverage and localization problems jointly, optimizing for both tasks at the same time. To endow an autonomous underwater robot with these capabilities, two algorithms are fundamental: a view planner that drives the robot to cover the area to be explored and a simultaneous localization and mapping (SLAM) algorithm capable of keeping the robot well localized. However, these two algorithms have different and sometimes contradictory objectives: while the view planner tries to discover new viewpoints of the scene, the localization algorithm seeks to reduce the uncertainty in the vehicle's position by revisiting already known viewpoints.

In a previous article [3] we presented a probabilistic next-best-view planner for a hover-capable autonomous underwater vehicle (AUV), that allowed for mapping complex environments without an a priori model of the scene. In the present article, we propose a method that combines our previous next-best-view planner with an active SLAM method to jointly solve the autonomous exploration problem. Active SLAM frameworks [4] study the combined problem of SLAM with deciding where to move next in order to build the map as efficiently as possible. The proposed method consists of the following steps (see Figure 1). The autonomous vehicle moves to a viewpoint and senses the environment using a mapping sensor; then, this data is used to correct both the robot position and the position of all previous viewpoints using a pose-graph algorithm. Next, the data gathered at each viewpoint is combined in a single octree that represents the world. A view planner calculates a set of candidate viewpoints to continue the exploration using this world representation. Thereafter, an action selection mechanism estimates the entropy reduction for each candidate, taking into account the map information provided by this viewpoint (i.e., the reduction of the entropy on the map), and the information provided by the viewpoint in order to close some loop with the viewpoints previously seen (i.e., the reduction of the entropy in the state of the vehicle). The candidate who optimally reduces the combined entropy on the map and the state of the vehicle is selected. If the termination criterion (i.e., coverage, number of scans, elapsed time, etc.) is not reached, a path planner computes an obstacle-free route to the selected viewpoint; the vehicle is commanded to follow it, and once reached, the process is repeated. Once any of the above mentioned termination criteria are reached, the exploration concludes and the vehicle returns to its initial position by planning an obstacle-free path.

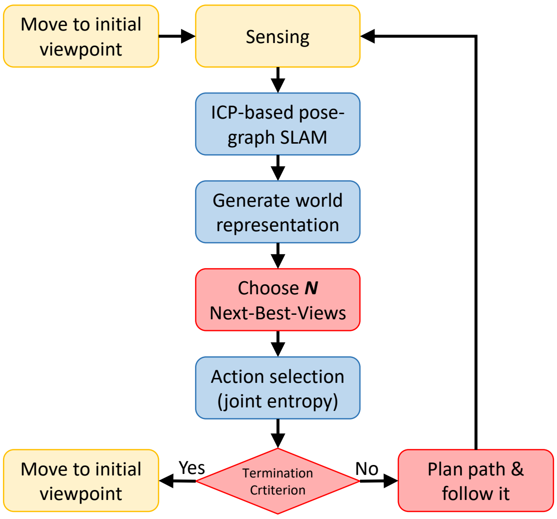

Move to initial viewpoint
Sensing
ICP-based pose-graph SLAM
Generate world representation
Choose *N* Next-Best-Views
Action selection (joint entropy)
Move to initial viewpoint
Yes
Termination Crtiterion
No
Plan path & follow it

Figure 1. Exploration framework steps. Robot primitives are shown in yellow, steps related to active simultaneous localization and mapping (SLAM) are shown in blue, and steps related to planning are represented in red.

In our proposal, the pose-graph SLAM method is responsible for precisely maintaining a good estimate of the vehicle trajectory and the location of each viewpoint while the mapping module recomputes the aggregated octree representation of the scene, merging the scans gathered at each viewpoint.

The proposed exploration method was devised for an AUV equipped with a multibeam sonar mounted on a pan and tilt device able to gather dense 3D point clouds, but it could be easily adapted to systems with other range sensors, such as laser scanners or stereo cameras. Although getting a scan moving a multibeam sonar with a pan and tilt system takes a few seconds, the quality of the point cloud obtained is much better if both are correctly synchronized, than the one that would be obtained with the usual technique in the state of the art, i.e., moving the robot with the sensor rigidly attached to it, since the uncertainty in the vehicle's motion is greater than that in the positioner's motion and the time to complete a scan is reduced to a fraction.

The pose-graph algorithm takes as input, the odometry measured by the robot as it navigates from one viewpoint to another and the 3D point cloud gathered at each viewpoint. The algorithm computes motion constraints between the two robot locations using the iterative closest point algorithm (ICP) [5], a common approach for underwater robot navigation [6,7]. However, one of the main drawbacks of using ICP is the difficulty in assessing the quality of the computed motion constraint. In this paper we propose a novel, closed-form formulation based on the first order error propagation. When the function $f(\cdot)$ to study is known explicitly, its Jacobian could be obtained by taking all the partial derivatives of $f(\cdot)$. However, in the ICP case there is a minimization that defines an implicit function between the data and the results [8]. So, to calculate its Jacobian, $f(\cdot)$ being a non explicit function, we propose to use the implicit function theorem.

The action selection mechanism chooses the viewpoint that most reduces the entropy. However, if after analyzing all the candidates proposed by the view planner, the smallest vehicle uncertainty exceeds a predefined threshold, the action selection mechanism may decide to return to an already visited viewpoint to keep its uncertainty under this threshold.

The interface with the view planner is the following. The input is a world representation, in the form of an octree, and the current robot position. The output is a set of candidate viewpoints that maximize the world exploration. For each viewpoint the view planner must provide its position and orientation and the number of unknown cells expected to be observed from it. Any view planner capable of providing this interface could be used within this framework.

To evaluate the proposed exploration system several tests have been conducted using the Girona 500 AUV [9], both in simulation and in real scenarios. Results demonstrate the benefits of using an active SLAM strategy. The accuracy of the pose-graph SLAM, that implements the new closed-form method to estimate the uncertainty in the ICP registers, has been analyzed and the action selection mechanism that allows to keep a low uncertainty in the vehicle state while maximizing the coverage.

The proposed framework endows AUVs with the necessary capabilities to undertake mapping and inspection missions in a priori, unknown, complex scenarios as it limits their drift, and consequently, allows to create more consistent models with a greater degree of coverage. In addition, the framework is generic enough to be adapted to other vehicles, sensors, domains or view planners than those discussed here.

The rest of the paper is structured as follows. First, several works related to view planning, localization and mapping, and exploration frameworks, specially those applied to underwater vehicles, are reviewed in Section 2. Next, Section 3 defines the pose-graph SLAM algorithm and the action selection mechanism that make up the proposed active SLAM. The closed-form solution to estimate the ICP error propagation is also presented in this section. Results for each algorithm and for the whole framework are reported in Section 4 before conclusions.

2. Related Work

The objective of this work was to develop an exploration framework so that AUVs have the capacity to explore a priori, unknown, complex scenarios. To this end, we propose to combine SLAM and view planning techniques with an action selection mechanism to create an active SLAM framework for exploration tailored to AUVs. Relevant works related to SLAM, view planning, and exploration in different domains are presented below.

The first works about the autonomous exploration of an unknown environment date back to the 70s with the well known art gallery problem [10]. However, it was not until the seminal work of Yamauchi [11] that autonomous exploration got more attention from the robotics community. Yamauchi proposed a view planner, based on a frontier-based approach, that has been the starting point for many other works in the view planning field. Whaite and Ferrie [12] presented the first uncertainty driven strategy for acquiring 3D-models of objects. However, they do not consider the uncertainty in the sensor pose. Despite some exploration solutions that claim to be robust against these uncertainties in the vehicle/sensor pose estimation [13], when absolute positioning sensors are not available, it is necessary to localize the vehicle while mapping the environment to avoid corrupted maps.

The concept that an exploring robot has different actions to perform and it must choose the one that produces the maximum information was proposed by Feder et al. [14]. Bourgault et al. [15] also addressed the problem of maximizing the accuracy of the map building process during exploration by adaptively selecting control actions that maximize localization accuracy. The core of their framework consists of maximizing the information gain on the map while minimizing the uncertainty of both vehicle position and map features in the SLAM process. This concept has been exploited by many later publications, including ours. Sim and Roy [16] showed the problem of using relative entropy as an object function. They point out that, although it is possible to quickly reduce the uncertainty of a distribution in some dimensions using this metric, in other dimensions there is no information gain.

Almost all the proposals presented during the late 90s and early 00s were based on the fact that the environment contains landmarks that can be uniquely determined during mapping. In contrast to this, Stachniss et al. [17] presented a decision-theoretic framework that makes no assumptions about distinguishable landmarks. This framework simultaneously considers the uncertainty in the map and in the pose of the vehicle to evaluate potential actions using an efficient Rao–Blackwellized particle filter. In Vidal et al., [18], the joint robot and map entropy reduction problem is faced using robot and map cross correlations obtained by an extended Kalman filter (EKF) that implements a visual SLAM. Valencia et al.'s work [19] was similar to Stachniss et al.'s but used a pose-graph SLAM (i.e., a global approach able to optimize the whole trajectory) instead of a particle filter (i.e., a filter approach that optimizes only the current state). Additionally, their proposal takes into account the cost of long action sequences during the selection of candidates using the same information metrics to keep the robot localized during the path execution. Although these works have focused on mobile robots, exploring in most cases 2D scenes, there are also a good number of articles where exploration is performed by autonomous aerial vehicles in 3D environments [20,21]. However, like the exploration algorithms for 3D object reconstruction [22,23], the localization of these systems is normally solved and the methods focus mainly on the view planning and not on the active localization.

For the underwater domain, to the best of the authors knowledge, there are no works that combine localization, mapping, and exploration. In fact, most of the explorations carried out nowadays by AUVs are limited to follow a predefined mission that consists of flying over the sea bottom at a safe distance while acquiring data [24]. It is possible to find works focused on view planning and others focused on SLAM but none that combine both techniques in a 3D scenario without a priori knowledge. Moreover, as several authors have stated [17,19], a straightforward combination of a view planner and a SLAM strategy is not sufficient to ensure a drift-free map with a high degree of coverage, which is why we propose an active SLAM solution. One of the few methods in which an AUV autonomously explores a complex structure was proposed by Englot [25]. The method is focused on sensor coverage and uses a sampling-based strategy to guide the vehicle. However, a model of the object to map is required. We have also presented several works regarding underwater exploration. The first one was an iterative framework [26], in which, starting from a low resolution a priori model, an exploration path is computed off-line an executed on-line in parallel with a SLAM algorithm that keeps the vehicle drift-bounded. The process is repeated until the desired resolution and coverage are reached. Despite the fact that the method combines an off-line view planning and SLAM, it does not check if the resulting trajectory will allow the SLAM algorithm to keep the localization uncertainty bounded, enforcing only some overlap between viewpoints. We have also presented solutions that do not require a preliminary map, but that do not localize the vehicle while exploring the 2D [27] or 3D [3] scene, nor use the robot state uncertainty to drive the exploration.

Underwater SLAM has attracted more attention from researchers than underwater exploration. Several works present solutions about how to map a complex underwater structures while localizing the vehicle with respect to it. A good example of this are the works of Vanmiddlesworth et al. [6] and

Teixeira et al. [7] that use an ICP-based pose-graph SLAM algorithm. To decide if the ICP registers were valid, they used a fitness score that represents the normalized sum of squared distances between corresponding points in the register. Nevertheless, how the uncertainty of each register was estimated was not discussed by either. This problem affects all domains, not only the underwater one. Common solutions to estimate this uncertainty are based on Monte Carlo algorithms, that can not be used online due to their high computational cost [28], or covariance estimation methods that rely on the Hessian objective function [29].

3. Methodology

The proposed active SLAM strategy contains a pose-graph localization algorithm that uses scan matching to compute the registers between scans, and an action selection mechanism that makes use of the joint map and state entropy to choose the next-best-viewpoint to explore from a set of viewpoints proposed by a view planner. As shown in Figure 2, the robot provides the scan (i.e., a point cloud) and the odometry between viewpoints to the pose-graph algorithm. A new position in the localization graph is created, and using the ICP algorithm, the current scan is registered with all previously gathered scans that may overlap with it (see Section 3.1). The uncertainty for each register is estimated (see Section 3.2), and finally, the pose-graph is optimized. With the optimized position for each viewpoint, a low-resolution world model (i.e., an octree) is built, merging all the scans. This representation is used by the view planner to calculate a set of candidate viewpoints and by the path planner to obtain a free path from the current position to the next-best-viewpoint.

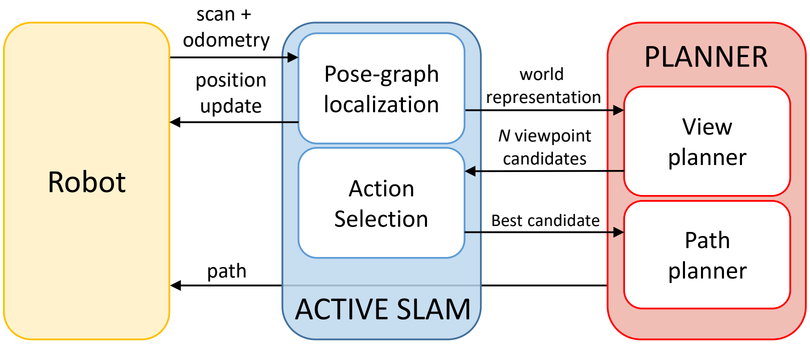


Figure 2. Relation between robot, active SLAM, and planning modules.

This article focuses on the active SLAM part of the system, whereas the view planner used is the one reported in [3]. Note, however, that this exploration framework could use any view planner capable of computing a set of candidate viewpoints that further explore the scene and the number of unknown cells observable from them, given an octree that represents the already explored region.

From the set of candidate viewpoints computed by the view planner, the action selection mechanism calculates the decrease in entropy that would be achieved by observing the scene from each of them. First, it computes the mutual information gain by closing a loop between each candidate and any previously visited viewpoint in which there is overlap. To compute this information gain, the uncertainty of the previously-visited viewpoints that can be obtained from the pose-graph algorithm, is used. The resulting state entropy reduction is combined with the map entropy reduction calculated using the number of unknown cells observable from the candidate viewpoint, as detailed in Section 3.3. The viewpoint with the greatest state-map entropy reduction is then chosen. A rapidly-exploring random tree star (RRT*) path planner is used to drive the robot from its position to the desired viewpoint [30].

The following subsections explain the details for the ICP-based pose-graph algorithm, the way in which the uncertainty of each ICP register is computed, and how the action selection mechanism computes the map and the state entropy reduction and combines them to select the next viewpoint to visit.

3.1. ICP-Based Pose-Graph Algorithm

When no absolute navigation sensors are available, the localization of the AUV is based on a dead reckoning filter that integrates measurements from on-board navigation sensors. These sensors are: a Doppler velocity log (DVL) that measures the AUV linear velocity, a pressure sensor that measures the vehicle's depth, and an attitude and heading reference system (AHRS) that provides orientations and angular velocities. For the experimental part we used the AUV Girona 500 [9], with 4 controllable DoFs—x, y, z, and *yaw* (ψ), whilst being very stable in *roll* and *pitch*. Of the 6 DoFs, the z term can be directly measured from a high accuracy pressure sensor (i.e., $< 0.01\%$ of the range) and a tactical grade AHRS provides *roll* and *pitch* measurements with accuracies over $0.1°$. However, there is no on-board sensor able to provide drift-free measurements for x, y, and ψ once the AUV is submerged. A compass could be used to mitigate the drift in heading but this can induce further problems when working close to structures which are common in many inspection-like missions [7].

The drift in x, y, and ψ can corrupt the world model and cause the robot to end up colliding with some obstacles in the environment. To avoid that, an online, ICP-based pose-graph SLAM algorithm is implemented. This algorithm estimates the vehicle position and orientation at each viewpoint (i.e., $\mathbf{x}_t = [x_0, y_0, \psi_0, \ldots x_n, y_n, \psi_n]^T$). The difference between the position estimated by the dead reckoning filter and the pose-graph SLAM is used to correct the former. The procedure shown in Figure 3 is executed every time that the vehicle reaches a viewpoint and gathers a new scan.

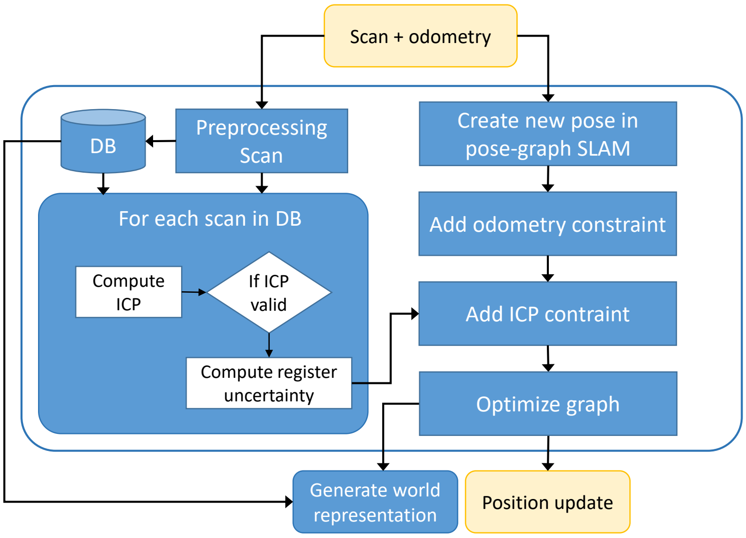


Figure 3. Iterative closest point algorithm (ICP)-based pose-graph SLAM algorithm, including the ICP registration (left) and the creation and optimization of the pose-graph (right).

The ICP-based pose-graph algorithm uses as input, a scan, containing the gathered point cloud, the odometry from the previous viewpoint ($odom = [\Delta_x, \Delta_y, \Delta_\psi]$) according to the dead reckoning filter running in the AUV, and the values that are not going to be optimized ([z, ϕ, θ]). The ICP registration pipeline filters the point cloud using a statistical outlier removal filter and downsamples it using a grid filter. Next, normals are computed. The resulting filtered point cloud with normals (Λ_i) is stored.

To check if the last scan gathered from viewpoint vp_n may overlap with any previous scan gathered from viewpoint vp_i, where $i \in (0 \ldots n-1)$, the overlap of the mapping sensor FoV between these two viewpoints, is computed. If the overlap is over a threshold, each local point cloud Λ_i and Λ_n are transformed to the world frame (i.e., ${}^wT_i$ and ${}^wT_n$). Next, the ICP algorithm is applied and a rigid transformation (${}^iA_n$) is obtained. Because all the localization is carried out in SE2, the ICP algorithm is restricted to estimate ${}^iA_n$ also in SE2. The composition of the original transformation between the viewpoints and the result of the ICP register (i.e., ${}^{i}\zeta_n = ({}^{w}T_i)^{-1}\,{}^{w}T_n \oplus^{i} A_n$) is added in the pose-graph as a loop closure between vp_i and vp_n. The scan matching pipeline was implemented using the point cloud library (PCL) [31].

In parallel with the scan registration, the pose-graph is created. The first viewpoint (vp_0) is used to fix the initial pose in the graph. Subsequent viewpoints generate new poses in the pose-graph connecting each new viewpoint to the last one using the odometry provided by the dead reckoning filter. The following state augmentation equation is used:

$$\mathbf{x}_{k+1} = \begin{cases} \bar{x}_k = x_{k-1} + \cos(\psi_{k-1})\Delta x - \sin(\psi_{k-1})\Delta y \\ \bar{y}_k = y_{k-1} + \sin(\psi_{k-1})\Delta x + \cos(\psi_{k-1})\Delta y \\ \bar{\psi}_k = \psi_{k-1} \oplus \Delta\psi \end{cases} . \tag{1}$$

All the constraints resulting from the scan matching pipeline are also added in the pose-graph connecting the current pose with all loop closing poses. Once all the constraints have been set, the graph is optimized and the position obtained for the last pose is sent back to the dead reckoning filter to correct the estimate of the AUV current position. The Ceres library [32] was used to implement the graph optimization.

3.2. ICP Error Propagation Estimation

After matching two scans using the ICP algorithm, the rigid transformation that better aligns them is used to create a new constraint in the pose-graph together with an estimation of its uncertainty. ICP algorithms do not compute this uncertainty and only provide some metrics, such as the sum of squared distances from the source point cloud to the target, to assess how good the matching is. Since the ICP algorithm consists of minimizing an error function iteratively, it is not possible to express it with an explicit function, and therefore calculate its Jacobian to propagate the error in the data to obtain this uncertainty. For this reason, we propose using the implicit function theorem [33] to estimate how the uncertainty in the matched point clouds is propagated.

Given a system

$$F(x_0, \ldots, x_n, y_0, \ldots, y_m) = 0, \tag{2}$$

the implicit function theorem states that, under a mild condition on the partial derivatives (with respect to the y_i) at a point, the variables y_i are differentiable functions of the x_j in some neighborhood of the point. Because these functions can generally not be expressed in closed form, they have to be implicitly defined by (2).

The implicit function theorem assumes that $F(\mathbf{x}, \mathbf{y})$ is the continuous and partial derivative in a neighborhood (x_0, y_0), such that

$$\begin{aligned} F(x_0, y_0) &= 0 \\ \frac{\partial F}{\partial \mathbf{y}}(x_0, y_0) &\neq 0. \end{aligned}$$

Then, there exists a neighborhood of (x_0, y_0) in which there is an implicit function $\mathbf{y} = f(\mathbf{x})$ such that:

$$\begin{aligned} f(x_0) &= y_0 \\ F(\mathbf{x}, f(\mathbf{x})) &= 0\ \forall \mathbf{x} \text{ near to } x_0 \\ f'(\mathbf{x}) &= -\frac{\frac{\partial F}{\partial \mathbf{x}}(\mathbf{x}, f(\mathbf{x}))}{\frac{\partial F}{\partial \mathbf{y}}(\mathbf{x}, f(\mathbf{x}))}. \end{aligned}$$

Given a reference set of points $\mathbf{y}$, where each point is defined as $y_i = [y_i^x, y_i^y]$, and a second set of points $\mathbf{p}$, where each point is defined as $p_i = [p_i^x, p_i^y]$, the ICP algorithm associates each p_i with the closest $y_i \; \forall p_i \in \mathbf{p}$. In other words, given two point clouds ($\mathbf{y}$ and $\mathbf{p}$) related by

$$\mathbf{y} = R\mathbf{p} + \mathbf{t}, \tag{3}$$

where

$$R = \begin{bmatrix} cos(\psi) & -sin(\psi) \\ sin(\psi) & cos(\psi) \end{bmatrix}$$

is a rotation matrix and $\mathbf{t}$ a translation vector $[t_x, t_y]^T$, the ICP algorithm estimates the transformation values (t_x, t_y, ψ) that minimize the cost function

$$C = \sum_{i=0}^{n} |e_i|^2, \tag{4}$$

where

$$e_i = y_i - Rp_i - \mathbf{t}. \tag{5}$$

Expressing the previous equation using vectors for n points we obtain:

$$C = \left(\mathrm{Y}^T - \mathrm{P}^T\mathrm{ROT}^T - \mathrm{T}^T\right)(\mathrm{Y} - \mathrm{ROT}\;\mathrm{P} - \mathrm{T}) \tag{6}$$

where

$$\mathrm{ROT} = \begin{bmatrix} R & 0_{2x2} & \dots & 0_{2x2} \\ 0_{2x2} & R & \dots & 0_{2x2} \\ \vdots & \vdots & \vdots & \vdots \\ 0_{2x2} & 0_{2x2} & \dots & R \end{bmatrix},$$

$$\mathrm{T} = \left(\mathbf{t}^t \begin{bmatrix} I_{2x2} & I_{2x2} & \dots & I_{2x2} \end{bmatrix}\right)^T,$$

$$\mathrm{Y} = [y_1 \dots y_n]^T,$$

$$\mathrm{P} = [p_1 \dots p_n]^T.$$

To minimize the cost function (6), it has to be partially derived by t_x, t_y, and ψ and made equal to zero. See Appendix A for its development.

$$\Phi = \begin{bmatrix} \frac{\partial C}{\partial t_x} \\ \\ \frac{\partial C}{\partial t_y} \\ \\ \frac{\partial C}{\partial \psi} \end{bmatrix} = 0. \tag{7}$$

The resulting Φ is an implicit function that represents the minimum cost of the ICP algorithm and fulfills the conditions for applying the implicit function theorem. To apply it, the partial derivatives ($\partial \mathbf{x}$) of Φ with respect to the known variables (i.e., the set of points P and Y) and the partial derivatives ($\partial \mathbf{y}$) with respect the unknown variables (i.e., t_x, t_y, and ψ) must be computed:

$$\Phi' = -\left(\frac{\partial \Phi}{\partial \mathbf{y}}\right)^{-1} \frac{\partial \Phi}{\partial \mathbf{x}}, \tag{8}$$

where

$$\frac{\partial \Phi}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial C}{\partial t_x,\partial p_0^x} & \frac{\partial C}{\partial t_x,\partial p_0^y} & \cdots & \frac{\partial C}{\partial t_x,\partial y_n^x} & \frac{\partial C}{\partial t_x,\partial y_n^y} \\ \frac{\partial C}{\partial t_y,\partial p_0^x} & \frac{\partial C}{\partial t_y,\partial p_0^y} & \cdots & \frac{\partial C}{\partial t_y,\partial y_n^x} & \frac{\partial C}{\partial t_y,\partial y_n^y} \\ \frac{\partial C}{\partial \psi,\partial p_0^x} & \frac{\partial C}{\partial \psi,\partial p_0^y} & \cdots & \frac{\partial C}{\partial \psi,\partial y_n^x} & \frac{\partial C}{\partial \psi,\partial y_n^y} \end{bmatrix}$$

and

$$\frac{\partial \Phi}{\partial \mathbf{y}} = \begin{bmatrix} \frac{\partial C}{\partial t_x \partial t_x} & \frac{\partial C}{\partial t_x \partial t_y} & \frac{\partial C}{\partial t_x \partial \psi} \\ \frac{\partial C}{\partial t_y \partial t_x} & \frac{\partial C}{\partial t_y \partial t_y} & \frac{\partial C}{\partial t_y \partial \psi} \\ \frac{\partial C}{\partial t_x \partial \psi} & \frac{\partial C}{\partial t_y \partial \psi} & \frac{\partial C}{\partial \psi \partial \psi} \end{bmatrix}.$$

Therefore, the error in the point clouds P and Y (i.e., σ_P and σ_Y) can be propagated according to Φ'. It is worth noting that to compute Φ', only the transformation $[t_x, t_y, \psi]$ and the association $p_i \rightarrow y_i$, both resulting from the ICP algorithm, are required.

$$\Sigma_y = \Phi' \begin{bmatrix} \sigma_P^2 \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \sigma_Y^2 \mathbf{I} \end{bmatrix} \Phi'^T. \tag{9}$$

3.3. Joint Map and State Entropy

As discussed in the introduction, the simple combination of a SLAM algorithm and a view planner is not sufficient to ensure a consistent exploration. For this reason we included a third element that decides which viewpoint, of all those proposed by the view planner, should be visited next. This action-selection mechanism combines the information gained from the exploration of new regions with the vehicle state estimation improvement resulting from revisiting known locations. This measure is known as the map and state joint entropy.

If all motions and observations are given, the map and state joint entropy can be expressed as the state entropy plus an average of the entropy for all infinite maps resulting from all infinite states x weighted by the probability of each state [17]:

$$\mathbf{H}(x, m|u, z) = \mathbf{H}(x|u, z) + \int_x p(x|u, z)\mathbf{H}(m|x, u, z)dx.$$

This equation can be approximated as:

$$\mathbf{H}(x, m|u, z) \approx \mathbf{H}(x|u, z) + \alpha(p(x|u, z))\mathbf{H}(m|\mu_x, u, z),$$

where instead of computing the map entropy averaged for all infinite possible maps, we compute it only for the mean pose estimates μ_x. The $\alpha(p(x|u, z))$ factor is added to scale the map entropy depending on the probability distribution.

Because we are only interested on the entropy reduction, the following equation can be used instead:

$$\Delta\mathbf{H}(x, m|u, z) \approx \Delta\mathbf{H}(x|u, z) + \alpha(p(x|u, z))\Delta\mathbf{H}(m|\mu_x, u, z). \tag{10}$$

Therefore, the action selection mechanism evaluate the reduction of joint entropy in these two terms for each of the candidate viewpoints and select the best one.

3.3.1. State Entropy

To compute the reduction in the state entropy, we assume

$$\Delta\mathbf{H}(x|u, z) = \begin{cases} \mathcal{I}_{ik} & \text{if } \exists \text{ loop}_{ik} \\ 0 & \text{otherwise} \end{cases}, \tag{11}$$

being

$$\mathcal{I}_{ik} = \frac{1}{2} \ln \frac{|\mathbf{S}_{ik}|}{|\mathbf{\Sigma}_y|}, \tag{12}$$

where $\mathbf{\Sigma}_y$ is the sensor registration covariance and $\mathbf{S}_{ik}$ is the innovation covariance computed as

$$\mathbf{S}_{ik} = \mathbf{\Sigma}_y + [\mathbf{H}_i \mathbf{H}_k] \begin{bmatrix} \mathbf{\Sigma_{ii}} & \mathbf{\Sigma_{ik}} \\ \mathbf{\Sigma_{ik}} & \mathbf{\Sigma_{kk}} \end{bmatrix} [\mathbf{H}_i \mathbf{H}_k]^T. \tag{13}$$

$\mathbf{H}_i$ and $\mathbf{H}_k$ are the Jacobians of h with respect to poses i and k evaluated at the state means μ_i and μ_k (see Appendix B), $\mathbf{\Sigma}_{ii}$ and $\mathbf{\Sigma}_{kk}$ are the marginal covariances of vp_i and vp_k respectively, and $\mathbf{\Sigma}_{ik}$ is their cross correlation [34]. The h function relates an observation $z_{ik} = (\Delta x, \Delta y, \Delta \psi)$, Δx, Δy, and $\Delta \psi$ being the distances between states x_i and x_k in world coordinates. They can be expressed as:

$$h_{ik}(\mathbf{x}) \begin{cases} \cos(\psi_i)(x_k - x_i) + \sin(\psi_i)(y_k - y_i) \\ \sin(\psi_i)(x_i - x_k) + \cos(\psi_i)(y_k - y_i) \\ \psi_k - \psi_i. \end{cases} \tag{14}$$

The value of $\mathbf{\Sigma}_y$ cannot be known in advance using (9). Therefore, it is assumed that for the same sensor, configuration. and scene, $\mathbf{\Sigma}_y$ will be similar to previously computed ones and an average of them is used.

Because $\mathbf{S}_{ik}$ depends on the number of loops that can be closed, the sensor FoV overlap described in Section 3.1 is used to check how many loop closures exist between the analyzed viewpoint and any of the viewpoints already visited. Therefore, matrices in (13) increase its dimensions according to the number of loops that can be closed.

3.3.2. Map Entropy

To measure $\mathbf{H}(m|x,u,z)$ using an octree, according to the common independence assumption about the cells, the entropy of a map m is the sum over the entropy values of all cells [17]:

$$\mathbf{H}(m|x,u,z) = \; -l^2 \sum_{c \in m} (p(c) \ln(p(c)) + (1 - p(c)) \ln(1 - p(c))),$$

where c is the cell probability and l the length of the side of a cell.

If the view planner used provides the number of unknown cells in m that are observable from each proposed viewpoint (i.e., $N(m|z)$), the map entropy reduction $\Delta\mathbf{H}(m|z)$ can be approximated by:

$$\Delta\mathbf{H}(m|z) \approx -l^2 \cdot N(m|z). \tag{15}$$

The factor accounting for the path probability distribution $\alpha(p(x|u,z)$ in (10) has an intuitive meaning: explorations made from well-located viewpoints produce more accurate maps than those made from uncertain locations. In fact, if we add new scans to the map from viewpoints where the robot has large values of marginal covariance, this can increase the entropy of the map as it is possible to register point clouds that, although being similar to each other, are clearly very distant. The inverse of the determinant of the current state marginal covariance is used to weight the map entropy reduction:

$$\alpha(p(x|u,z)) = \alpha \frac{1}{|\Sigma_{kk}|}. \tag{16}$$

The parameter α can be used to fine tune the weight of the map with respect to the state.

4. Results

Results for all the algorithms discussed in the methodology section are presented here using data either simulated or obtained with the Girona 500 AUV equipped with a multibeam sensor mounted on a pan and tilt device (see Figure 4). Experiments have been divided in two parts for the paper: the former discusses the ICP error propagation estimation and shows some results regarding the ICP-based pose-graph SLAM algorithm; the later presents the benefits of using the proposed active SLAM strategy for exploring a scene.

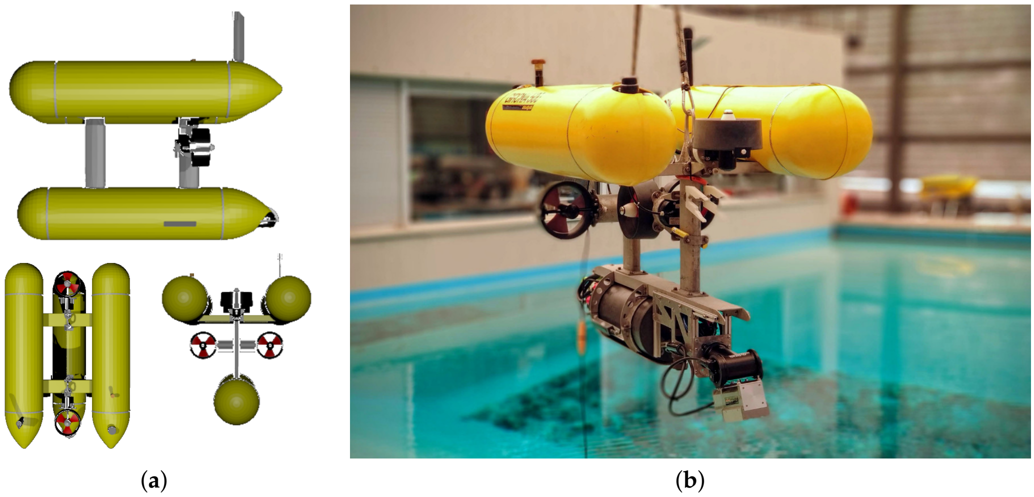

(**a**) (**b**)

Figure 4. (**a**) Lateral, top, and front views of Girona 500 AUV. The vehicle consists of three torpedo-shaped hulls 0.3 m in diameter. Its overall dimensions are 1 m high, 1 m wide, and 1.5 m long and it weighs less than 200 kg. (**b**) Girona 500 AUV equipped with a multibeam sonar mounted over a pan and tilt positioner.

4.1. Vehicle Localization and Error Propagation Estimation

To estimate the uncertainty for each ICP register, the uncertainty in the point clouds to register is propagated using the closed form equation presented in (9). To test the accuracy of the proposed method, it was compared against a classic Monte Carlo approach using synthetic data consisting of two point clouds with a standard variation of $\sigma = 0.02$ (see Figure 5). After 1000 Monte Carlo simulations, the resulting uncertainty was comparable to the one computed with the proposed method, although the later was slightly more optimistic (see Figure 6). One reason why this may happen is that (9) assumes that the transformation resulting from applying ICP is always the one that best registers the two point clouds, when in reality does not always happen (e.g., due to a bad initialization). Nevertheless, the proposed method provides a closed-form solution which provides similar results to the Monte Carlo method in real time, based solely on the uncertainty of the point clouds and the ICP result (i.e., the transformation (t_x, t_y, ψ) and the association between the source and target point clouds).

Regarding to the pose-graph localization using the AUV odometry and the ICP registers, we first tested the proposed method in simulation, in order to have a ground truth to compare it to, and later with real data consisting of multibeam sonar scans gathered by the Girona 500 AUV in a harbor.

Using the Gazebo simulator [35] and a realistic dynamic model for the Girona 500 AUV, a circular trajectory around a structure of interest (see Figure 7a) was performed. Figure 7b shows the ground truth trajectory; the odometry obtained by the dead-reckoning filter running in the AUV which had noise that was added in the velocity and the orientation measurements; and the optimized trajectory obtained using the proposed pose-graph SLAM algorithm. It is clear that the optimized trajectory is closer to the ground-truth, especially when it is possible to close loops with the initial positions, as shown in Figure 8.

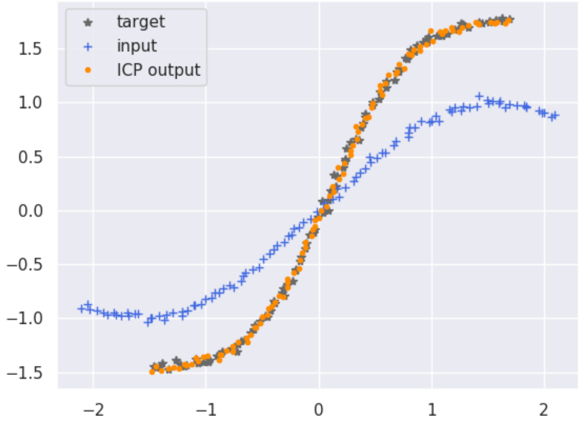


Figure 5. Input point cloud, represented by "+" dots, has to be aligned with the target point cloud, represented by "*" dots. After computing the transformation (t_x, t_y, ψ) using the ICP algorithm and applying it to the input point cloud, the ICP output point cloud, represented by "·," is obtained.

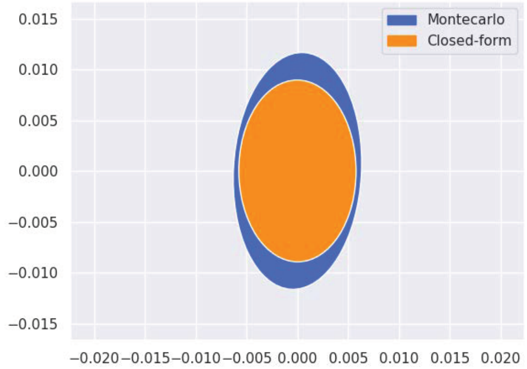
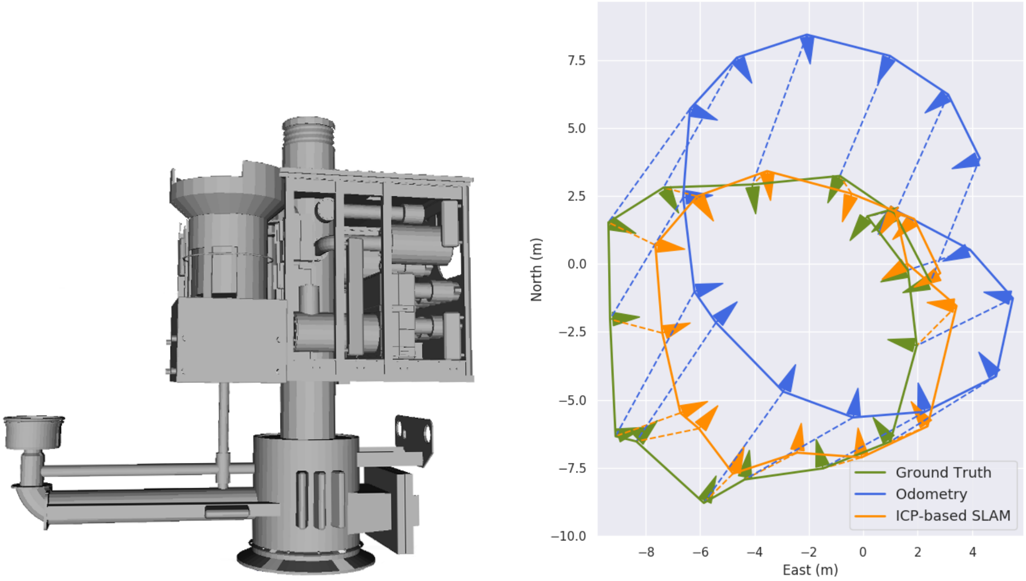


Figure 6. Comparison of ICP register covariance estimation between a Monte Carlo solution, in blue, and the proposed closed-form method, in orange.



(**a**) (**b**)

Figure 7. (**a**) Oil and Gas structure around which the trajectory was executed. (**b**) Dead reckoning odometry and ICP-based pose-graph SLAM trajectories compared to ground truth trajectory. The pose-graph SLAM and the odometry error with respect to the dead reckoning are shown in Figure 8.

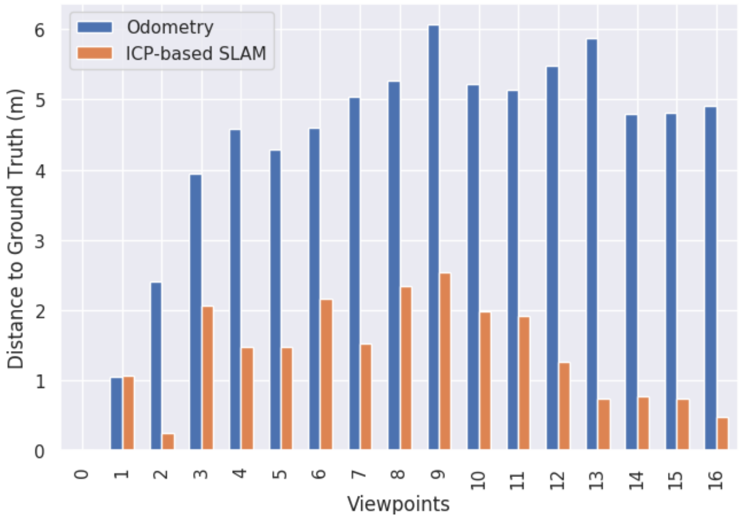
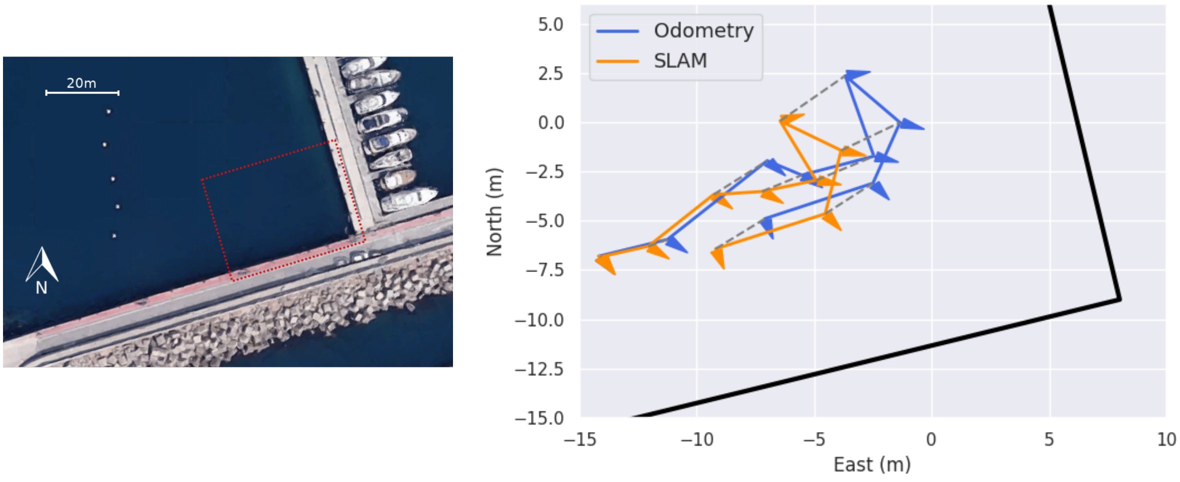


Figure 8. Distance between estimated position for each viewpoint and the ground-truth. The dead reckoning odometry is clearly drifting apart, while the optimized trajectory reduces the error significantly, especially when loops with initial viewpoints (i.e., vp_{13} to vp_{16}) are closed.

Figure 9 shows the odometry gathered by Girona 500 AUV while performing an autonomous exploration in the Sant Feliu de Guixols harbour compared with the trajectory optimized using the pose-graph SLAM. Although it is difficult to obtain quantitative results using real data, it can be seen that when all scans are combined into a single point cloud (see Figure 10), the optimized trajectory generates a more consistent map than the odometry.



(**a**) (**b**)

Figure 9. (**a**) Google maps image from Sant Feliu de Guixols harbour (lat. 41.7774, lon. 3.0343) where Girona 500 AUV executed an exploration mission. (**b**) Resulting trajectories where the dead reckoning odometry is shown in blue and the result of the ICP-based pose-graph SLAM in orange.

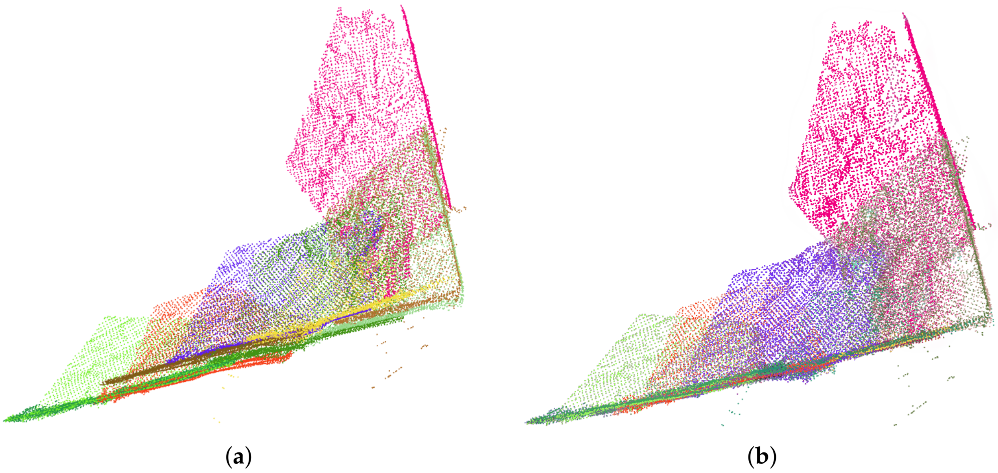


Figure 10. Point clouds gathered by Girona 500 AUV and placed in the position given by the odometry (**a**), or the ICP-based pose-graph algorithm (**b**). It can be appreciated how in (**a**), some walls appear duplicated, while in (**b**) the map is more consistent.

4.2. Active SLAM

To test the active SLAM strategy, a mission in a real wave breaker structure was simulated. Figures 11 and 12 compare the performance of a view planner when the best candidate, according to the planner himself, is selected or when the same view planner is used to propose a set of candidates and the action selection mechanism selects the viewpoint that best reduces the joint entropy.

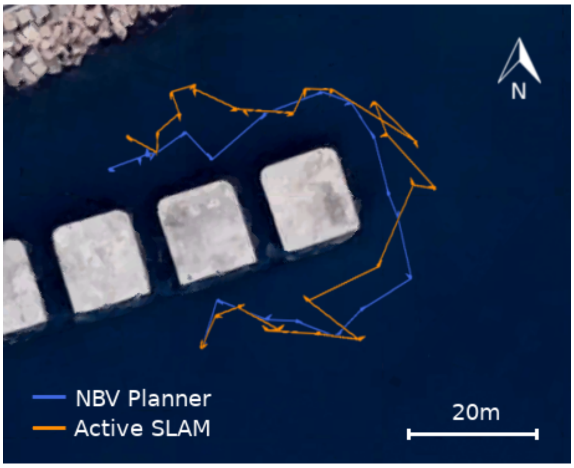


Figure 11. Trajectories executed using the view planner and a pose-graph-SLAM in blue, and the proposed active-SLAM strategy in orange, overlaid to the real scenario taken from Google maps (lat. 41.77754, lon. 3.03783).

Using the proposed exploration framework some more scans are required to be collected (i.e., in this particular example 25 instead of 23) to explore a similar area. However, the uncertainty in the vehicle state is several orders of magnitude smaller (see Figure 12). This is of great importance because if the uncertainty in the vehicle position is large, the new scans incorporated in the world can cause it to lose consistency, and therefore, the exploration to fail or even cause the vehicle to collide with an obstacle. It is worth noting that neither of the two alternatives is able to explore the entire scene as there are regions that simply cannot be explored (e.g., inside the blocs).

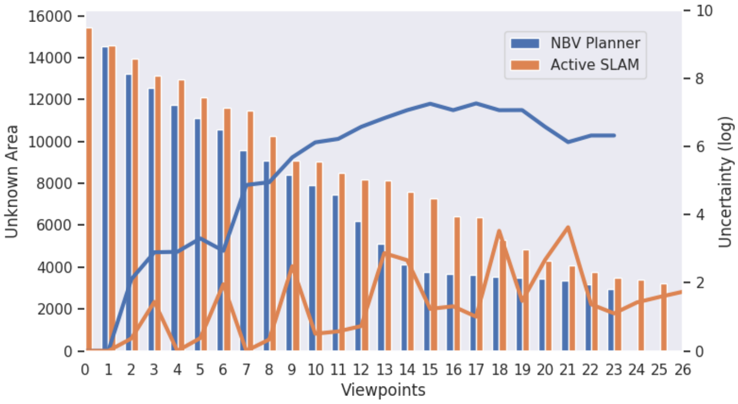


Figure 12. Bars represent the unknown area in m^3, while lines represent the vehicle's state uncertainty at each viewpoint computed as $\log(det(\Sigma_i) + 1)$.

Using the the same view planner and pose-graph SLAM algorithm but imposing only a certain degree of overlap between views instead of using the proposed action selection mechanism, 23 loops are closed along the exploration, while with the active SLAM strategy, the number of loops increases up to 44.

Figure 13 shows the joint entropy reduction value per viewpoint truncated at 100 units. In this figure, the map and the state entropy values, calculated according (10), are represented in different colors to show the contributions of the state and the map in the joint entropy. Looking at Figures 12 and 13 it is easy to understand the vehicle's behavior: when the vehicle uncertainty is small, the action-selection mechanism tries to reduce the map uncertainty scanning by as many unknown cells as possible, while when the vehicle uncertainty is large, the action-selection mechanism aims to reduce this uncertainty by closing loops with previously visited viewpoints, preferably with those with little uncertainty. This behavior appears naturally without any need of adjustment and ensures more reliable explorations, reducing the vehicle's uncertainty when necessary and maximizing the exploration whenever possible.

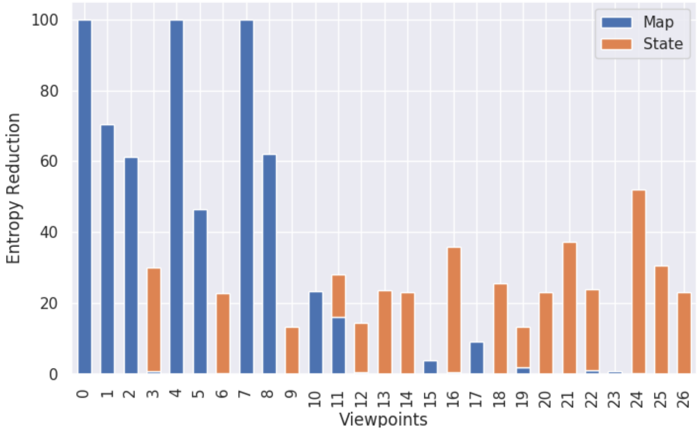


Figure 13. Each bar represents the reduction of the joint entropy truncated to 100 units. In blue is shown the part of the joint entropy reduction that comes from the map, and in orange the part that comes from the state of the robot.

5. Conclusions

A new exploration framework based on an active SLAM strategy is proposed herein. The system was tailored to AUVs and was conceived to be used in complex environments, with a strong 3D component, where there is no a priori map. The methodology employed combines a view planner and a pose-graph based localization and mapping algorithm with an action selection mechanism.

This mechanism decides the next viewpoint to visit taking into account which will be able to reduce the entropy in the map more and in the state of the vehicle jointly. The paper is focused on the localization and mapping algorithm and in the action-selection policy. The former was implemented using a standard pose-graph approach and using the ICP algorithm to register the point clouds gathered from different viewpoints. To compute the error propagation in the ICP algorithm, a closed-form solution was proposed. This solution allows one to compute a realistic covariance for each ICP register in real time. Compared to a Monte Carlo simulation, the method produces similar but slightly optimistic results. The proposed pose-graph SLAM pipeline was tested with data obtained through simulations and with data collected with the Girona 500 AUV in a harbor. As expected, in both situations, the trajectory and the world model obtained, once having applied the optimization, showed better correspondence with reality than the ones obtained only with the AUV odometry. The use of an action-selection mechanism led to the natural emergence of a new exploration behavior in the robot. The fact of weighing the exploration (i.e., examining unknown regions) and the improvement of the robot state (i.e., revisiting already known positions), using in both cases, the entropy reduction, causes the following behavior: whenever the uncertainty in the vehicle state is small, the active SLAM maximizes the exploration of new regions, while when this uncertainty is large, the vehicle tries to reduce it by revisiting known regions with small uncertainty. When possible, the action-selection mechanism chooses viewpoints in which both of the above goals apply.

The proposed framework is generic enough as it can be used in other domains or easily adapted to other scanning sensors, Moreover, it can make use of any view planner that takes as input, the robot location and an octree with the current exploration state and from this is able to generate a set of valid viewpoint candidates to further explore the environment.

As a future work, the two main limitations that have been identified when testing the active SLAM framework with different scenarios should be addressed. On the one hand, if the scene does not have enough features, the ICP algorithm is not capable of registering the point clouds, or, in the worst case, registers them poorly (e.g., when registering two flat surfaces). To solve this problem we propose extracting the 3D features of the point clouds first, to make a first coarse alignment, and only if this succeeds, completing the registration with the ICP algorithm [36]. A second problem has been observed when the vehicle has to travel long distances close to obstacles in order to move from one viewpoint to another. Since the vehicle position is only corrected in the viewpoints, this can cause the robot to collide with an obstacle. To avoid this, we propose developing a motion planning algorithm that ensures that the vehicle state uncertainty never exceeds a certain threshold during the entire route. Therefore, the vehicle should plan a path in which it is possible to close enough loops with previously visited regions to keep this threshold.

Author Contributions: Conceptualization, methodology, software, validation, and investigation, N.P.; investigation, supervision, and funding acquisition, J.A.-C.; resources, funding acquisition, and supervision, M.C.

Funding: This research was funded by the Spanish State Research Agency through grants DPI2015-73978-JIN and DPI2017-89564-P, and the María de Maeztu Seal of Excellence to IRI (MDM-2016-0656).

Acknowledgments: Thanks to the Institut de Robòtica i Informàtica Industrial that is a Joint Research Center of the Spanish Council for Scientific Research (CSIC) and the Technical University of Catalonia (UPC) who hosted one of us in a postdoctoral stay, under the supervision of Juan Andrade-Cetto, where most of this work was done.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Implicit Function Theorem for the ICP Cost Function

Development of partial derivatives $\frac{\partial C}{\partial t_x}$, $\frac{\partial C}{\partial t_y}$ and $\frac{\partial C}{\partial \psi}$ of Equation (7):

$$\begin{aligned}
\frac{\partial C}{\partial t_x} &= 2nt_x - 2\sum_{i=0}^{n} y_i^x + 2\sum_{i=0}^{n} p_i^x cos(\psi) - p_i^y sin(\psi) \\
\frac{\partial C}{\partial t_y} &= 2nt_y - 2\sum_{i=0}^{n} y_i^y + 2\sum_{i=0}^{n} p_i^x sin(\psi) + p_i^y cos(\psi) \\
\frac{\partial C}{\partial \psi} &= 2\sum_{i=0}^{n} p_i^x (y_i^x sin(\psi) - y_i^y cos(\psi)) + p_i^y (y_i^x cos(\psi) + y_i^y sin(\psi)) + \\
& 2\sum_{i=0}^{n} ty(p_i^x cos(\psi) - p_i^y sin(\psi)) - t_x(p_i^x sin(\psi) + p_i^y cos(\psi))
\end{aligned}$$

Development of Section 3.2:

$$\frac{\partial \Phi}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial C}{\partial t_x, \partial p_0^x} & \frac{\partial C}{\partial t_x, \partial p_0^y} & \cdots & \frac{\partial C}{\partial t_x, \partial y_n^x} & \frac{\partial C}{\partial t_x, \partial y_n^y} \\ \frac{\partial C}{\partial t_y, \partial p_0^x} & \frac{\partial C}{\partial t_y, \partial p_0^y} & \cdots & \frac{\partial C}{\partial t_y, \partial y_n^x} & \frac{\partial C}{\partial t_y, \partial y_n^y} \\ \frac{\partial C}{\partial \psi, \partial p_0^x} & \frac{\partial C}{\partial \psi, \partial p_0^y} & \cdots & \frac{\partial C}{\partial \psi, \partial y_n^x} & \frac{\partial C}{\partial \psi, \partial y_n^y} \end{bmatrix}$$

$$\frac{\partial \Phi}{\partial \mathbf{y}} = \begin{bmatrix} \frac{\partial C}{\partial t_x \partial t_x} & \frac{\partial C}{\partial t_x \partial t_y} & \frac{\partial C}{\partial t_x \partial \psi} \\ \frac{\partial C}{\partial t_y \partial t_x} & \frac{\partial C}{\partial t_y \partial t_y} & \frac{\partial C}{\partial t_y \partial \psi} \\ \frac{\partial C}{\partial t_x \partial \psi} & \frac{\partial C}{\partial t_y \partial \psi} & \frac{\partial C}{\partial \psi \partial \psi} \end{bmatrix}$$

Appendix B. Joint Map and State Entropy

$$[\mathbf{H}_i \mathbf{H}_k] = \begin{bmatrix} -\cos(\psi_i) & -\sin(\psi_i) & \sin(\psi_i)(x_i - x_k) + \cos(\psi_i)(y_k - y_i) & \cos(\psi_i) & \sin(\psi_i) & 0 \\ \sin(\psi_i) & -\cos(\psi_i) & \cos(\psi_i)(x_i - x_k) + \sin(\psi_i)(y_i - y_k) & -\sin(\psi_i) & \cos(\psi_i) & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \end{bmatrix}$$

References

1. Scott, W.R.; Roth, G.; Rivest, J.F. View planning for automated three-dimensional object reconstruction and inspection. *ACM Comput. Surv. (CSUR)* **2003**, *35*, 64–96. [CrossRef]
2. WillowGarage. DARPA Subterranean Challenge. 2019. Available online: https://www.subtchallenge.com (accessed on 28 November 2019).
3. Palomeras, N.; Hurtós, N.; Vidal, E.; Carreras, M. Autonomous Exploration of Complex Underwater Environments Using a Probabilistic Next-Best-View Planner. *IEEE Robot. Autom. Lett.* **2019**, *4*, 1619–1625. [CrossRef]
4. Valencia, R.; Andrade-Cetto, J. Active Pose SLAM. In *Mapping, Planning and Exploration with Pose SLAM*; Springer Tracts in Advanced Robotics; Springer, Ed.; Springer: New York, NY, USA, 2018; Chapter 4, pp. 89–108. [CrossRef]
5. Rusinkiewicz, S.; Levoy, M. Efficient variants of the ICP algorithm. In Proceedings of the Third International Conference on 3-D Digital Imaging and Modeling, Quebec City, QC, Canada, 28 May–1 June 2001; pp. 145–152. [CrossRef]
6. VanMiddlesworth, M.; Kaess, M.; Hover, F.; Leonard, J. Mapping 3D Underwater Environments with Smoothed Submaps. In *Field and Service Robotics*; Springer Tracts in Advanced Robotics; Mejias L., Corke P., Roberts J., Eds.; Springer: Cham, Switzerland; Volume 105. [CrossRef]
7. Teixeira, P.V.; Kaess, M.; Hover, F.S.; Leonard, J.J. Underwater inspection using sonar-based volumetric submaps. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems, (IROS 2016), Daejeon, Korea, 9–14 October 2016; pp. 4288–4295.

8. Clarke, J.C. *Applications of Sequence Geometry to Visual Motion*; Technical Report; Department of Engineering Science, University of Oxford: Oxford, UK, 1998.
9. Ribas, D.; Palomeras, N.; Ridao, P.; Carreras, M.; Mallios, A. Girona 500 auv: From survey to intervention. *IEEE/ASME Trans. Mechatron.* **2012**, *17*, 46–53. [CrossRef]
10. O'Rourke, J. *Art Gallery Theorems and Algorithms*; Oxford University Press: Oxford, UK, 1987.
11. Yamauchi, B. A frontier-based approach for autonomous exploration. In Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97. 'Towards New Computational Principles for Robotics and Automation', Monterey, CA, USA, 10–11 July 1997; pp. 146–151. [CrossRef]
12. Whaite, P.; Ferrie, F.P. Autonomous exploration: Driven by uncertainty. *IEEE Trans. Pattern Anal. Mach. Intell.* **1997**, *19*, 193–205. [CrossRef]
13. Ko, J.; Stewart, B.; Fox, D.; Konolige, K.; Limketkai, B. A practical, decision-theoretic approach to multi-robot mapping and exploration. In Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453), Las Vegas, NV, USA, 27–31 October 2003; Volume 4, pp. 3232–3238. [CrossRef]
14. Feder, H.J.S.; Leonard, J.J.; Smith, C.M. Adaptive Mobile Robot Navigation and Mapping. *Int. J. Robot. Res.* **1999**, *18*, 650–668. [CrossRef]
15. Bourgault, F.; Makarenko, A.A.; Williams, S.B.; Grocholsky, B.; Durrant-Whyte, H.F. Information based adaptive robotic exploration. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Lausanne, Switzerland, 30 September–4 October 2002; Volume 1, pp. 540–545. [CrossRef]
16. Sim, R.; Roy, N. Global A-Optimal Robot Exploration in SLAM. In Proceedings of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain, 18–22 April 2005; pp. 661–666. [CrossRef]
17. Stachniss, C.; Grisetti, G.; Burgard, W. Information gain-based exploration using Rao-Blackwellized particle filters. In Proceedings of the Robotics: Science and Systems I, Cambridge, MA, USA, 8–11 June 2005; pp. 65–72.
18. Vidal-Calleja, T.; Sanfeliu, A.; Andrade-Cetto, J. Guiding and Localising in Real-Time a Mobile Robot with a Monocular Camera in Non-Flat Terrains. In Proceedings of the 6th IFAC Symposium on Intelligent Autonomous Vehicles (IAV), Toulouse, France, 3–5 September 2007; pp. 1–6.
19. Valencia, R.; Valls Miró, J.; Dissanayake, G.; Andrade-Cetto, J. Active Pose SLAM. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Algarve, Portugal, 7–12 October 2012; pp. 1885–1891. [CrossRef]
20. Yoder, L.; Scherer, S., Autonomous Exploration for Infrastructure Modeling with a Micro Aerial Vehicle. In *Field and Service Robotics: Results of the 10th International Conference*; Wettergreen, D.S., Barfoot, T.D., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 427–440._28. [CrossRef]
21. Song, S.; Jo, S. Online inspection path planning for autonomous 3D modeling using a micro-aerial vehicle. In Proceedings of the IEEE International Conference on Robotics and Automation, Singapore, 29 May–3 June 2017; pp. 6217–6224. [CrossRef]
22. Vasquez-Gomez, J.I.; Sucar, L.E.; Murrieta-Cid, R. View/state planning for three-dimensional object reconstruction under uncertainty. *Auton. Robot.* **2017**, *41*, 89–109. [CrossRef]
23. Daudelin, J.; Campbell, M. An Adaptable, Probabilistic, Next-Best View Algorithm for Reconstruction of Unknown 3-D Objects. *IEEE Robot. Autom. Lett.* **2017**, *2*, 1540–1547. [CrossRef]
24. Yoerger, D.R.; Jakuba, M.; Bradley, A.M.; Bingham, B. Techniques for Deep Sea Near Bottom Survey Using an Autonomous Underwater Vehicle. *Int. J. Robot. Res.* **2007**, *26*, 41–54. [CrossRef]
25. Englot, B.J. Sampling-based coverage path planning for complex 3D structures. Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2012.
26. Palomeras, N.; Hurtos, N.; Carreras, M.; Ridao, P. Autonomous mapping of underwater 3D structures: From view planning to execution. *IEEE Robot. Autom. Lett.* **2018**, *3*, 1965–1971. [CrossRef]
27. Vidal, E.; Palomeras, N.; Istenič, K.; Hernández, J.; Carreras, M. Two-Dimensional Frontier-Based Viewpoint Generation for Exploring and Mapping Underwater Environments. *Sensors* **2019**, *19*, 1460. [CrossRef] [PubMed]
28. Iversen, T.M.; Buch, A.G.; Kraft, D. Prediction of ICP pose uncertainties using Monte Carlo simulation with synthetic depth images. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 4640–4647.

29. Censi, A. An accurate closed-form estimate of ICP's covariance. In Proceedings of the 2007 IEEE International Conference on Robotics and Automation, Roma, Italy, 10–14 April 2007; pp. 3167–3172. [CrossRef]
30. Şucan, I.A.; Moll, M.; Kavraki, L.E. The Open Motion Planning Library. *IEEE Robot. Autom. Mag.* **2012**, *19*, 72–82. [CrossRef]
31. Rusu, R.B.; Cousins, S. 3D is here: Point Cloud Library (PCL). In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 1–4. [CrossRef]
32. Agarwal, S.; Mierle, K. Ceres Solver. Available online: http://ceres-solver.org (accessed on 28 November 2019).
33. Krantz, S.G.; Parks, H.R. *The Implicit Function Theorem*; Springer: New York, NY, USA, 2013. [CrossRef]
34. Ila, V.; Porta, J.M.; Andrade-Cetto, J. Information-Based Compact Pose SLAM. *IEEE Trans. Robot.* **2010**, *26*, 78–93. [CrossRef]
35. Koenig, N.; Howard, A. Design and Use Paradigms for Gazebo, An Open-Source Multi-Robot Simulator. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Sendai, Japan, 28 September–2 October 2004; pp. 2149–2154.
36. Palomer, A.; Ridao, P.; Ribas, D. Inspection of an underwater structure using point-cloud SLAM with an AUV and a laser scanner. *J. Field Robot.* **2019**, *36*, 1333–1344. [CrossRef]