Chapter

Progressive Underwater Exploration with a Corridor-Based Navigation System

Mario Alberto Jordan

Abstract

The present work focuses on the exploration of underwater environments by means of autonomous submarines like AUVs using vision-based navigation. An approach called Corridor SLAM (C-SLAM) was developed for this purpose. It implements a global exploration strategy that consists of first creating a trunk corridor on the seabed and then branching as far as possible in different directions to increase the explored region. The system guarantees the safe return of the vehicle to the starting point by taking into account a metric of the corridor lengths that are related to their energy autonomy. Experimental trials in a basin with underwater scenarios demonstrated the feasibility of the approach.

Keywords: monocular active SLAM, vision-based navigation, dense mapping, feature-based SLAM, corridor network, path planning, guidance system, self-similarity, sunlight caustics

1. Introduction

Vision-based SLAM systems are much appreciated nowadays in a broad spectrum of applications in robotics, autonomous navigation, and guidance systems, in airborne, underwater, and terrestrial environments [1–4]. A global comparison shows that SLAM applications underwater are less abundant (see some state-of-the-art works) [5–10].

In particular, underwater navigation generally excludes any form of global positioning system such as in other environments, so vision is a sound alternative, provided visibility is good enough. On the other hand, in turbid waters it is possible, to some extent, to navigate at low altitude on the seabed. However, in these extreme cases, the texture of the seabed may appear voluminous, which implies visual occlusions and collision probabilities. Conversely, shallow waters, rapid changes in the natural illumination, for example, due to the sunlight caustic on the floor, shadows, and flashes, can seriously damage the photometric properties of the images. In such extreme cases, the estimation of the vehicle position can be so impaired with the possibility of vehicle loss.

These characteristics of the underwater environment pose major challenges to the success of navigation, especially if it takes place in unknown regions. These challenges relate not only to robust estimations of the vehicle position and surrounding cartography [11] but also to the ability of the guidance system to avoid possible collisions at low altitudes.

In this work we will present an approach for the autonomous exploration of unknown regions of the seabed by means of a navigating system connected to a decision-making process hosted in the submarine. This vision-based approach is called Corridor SLAM (C-SLAM), because it combines SLAM techniques with a strategy to build a corridor over the seabed. The original concept was described in [12]. This document presents a generalization of previous work centered on a robust network of corridors. A variant of the basic concept uses active SLAM for exploring and building a grid map (see recent works [13–15]), but they do not enclose in any way the concept of path optimality as here, which proposes continuous paths with viewpoints associated.

A relatively close idea to the concept of corridor can be found in the “teach and repeat” method, [16], to follow a path in unfamiliar outdoor environment, such as in the exploration on the surface of Mars by a rover. However, the basic method does not include an autonomous exploration, as the first path construction is given remotely by a human operator with the aid of a camera at the rover front. Once an a priori feasible path has been defined, the vehicle can “repeat” the path that was previously “learned.”

The main objective of our approach is to make the right decisions to build robust pathways to explore and configure them in a network. At the same time, the system is able to bring the vehicle back to the starting point from any position in the explored area. The robustness properties of the network are conceived in the sense that there are preferential directions to explore and are such that one allows a successful return.

2. C-SLAM-based autonomous navigation

Figure 1 illustrates the complete structure of the proposal for an autonomous vision-based system. It consists of three nested feedback loops, called adaptive control loop, reference adjustment loop, and dense mapping loop. They are described in details below.

The adaptive control loop generates the control actions u necessary to achieve the prescribed reference path $\hat{\eta}_{ref}$ and the reference speed $\hat{\dot{\eta}}_{ref}$. Both are provided by the guidance system. The path following will attenuate the influence of sea currents

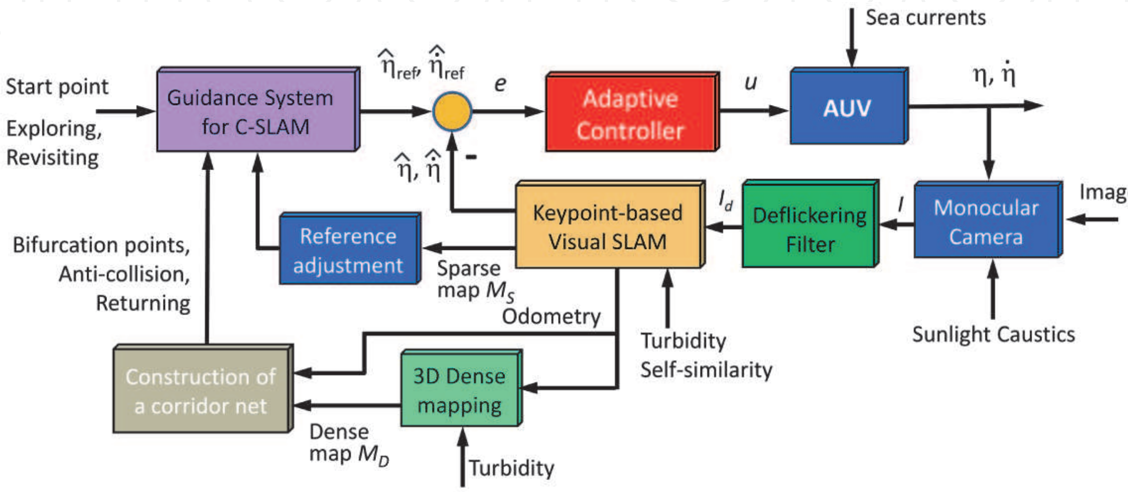

Figure 1.
C-SLAM structure. Three-loop-based approach to underwater exploration.

that push the vehicle off the corridor. The adaptive law circumvents the requirement to make available an AUV model which is generally complex due to its 6 degrees of freedom (see [17, 18]). Additionally, the action of marine currents and a variable speed of the AUV make hydrodynamic resistance forces very difficult to estimate and include in the dynamic model. For these reasons, controllers with self-adjustment of coefficients and dynamics adaptation seem to be much more effective than fixed controllers.

In the adaptive control loop feedback, there is the vision system that basically takes images of the seabed with the vehicle in motion to estimate the position η and rate $\dot{\eta}$ of the AUV. The estimates are termed $\hat{\eta}$ and $\hat{\dot{\eta}}$, respectively. The monocular camera onboard takes images in a sequence called I. The line of vision is oriented forward with a certain inclination towards the ground. In case of sunlight caustics on the seafloor [19], the quality of the images may be significantly improved by an online deflickering filter [20–22]. The filter attempts to retrieve the original photometric properties of the frame. Its output is the frame sequence I_d. A feature-based SLAM algorithm estimates the position of the vehicle and a sparse depth map M_S of the floor. Here, texture characteristics of the ground like self-similarity and water turbidity will generally difficult the self-localization and map estimation. Therefore they are considered as disturbances [23, 24].

The reference adjustment loop has the function of reformulating the references $\hat{\eta}_{\mathrm{ref}}$ and $\hat{\dot{\eta}}_{\mathrm{ref}}$ according to a new scale factor of the estimated relief. This is necessary due to the variability of the estimated sparse map when revisiting the corridor site.

The outer loop of the C-SLAM, namely, the dense mapping loop, is dedicated to the construction of the corridor network, providing alternatives for bifurcating the path during the exploration. The feedback consists of the navigation system, a block for estimating a dense mapping M_D of the terrain, and a block for path planning and bifurcation managing. These blocks will process the odometry information provided by the inner feedback in order to update the topography of the corridors. The dense map helps the guidance system to implement the future path to follow and to avoid collisions.

A typical navigation route consists in part of stretches between relevant points or landmarks. For example, a stretch that links a starting point with a bifurcation point, or one that covers an entire branch, or one between a bifurcation point and a point of no return. During the exploration, the path is created step by step, in which the vehicle direction is regularly changed towards preferential points that should ensure the continuity of the future path.

According to the main objectives of the exploration, C-SLAM should allow a successful return of the vehicle to the starting point from any distant point of the network. The farthest points are called no-return points. In this case, it is assumed that when the vehicle reaches a point of no return, it possesses at least half of its energy autonomy [25–27]. It is therefore that C-SLAM must permanently estimate points of no return and also ensure that there exists at least one way for the comeback before losing the energy autonomy.

3. Corridor net

The network of submarine corridors is the composition of interconnected strips on the seabed, which covers a certain explored region and allows navigation between different points of it. Topographically the strips are sequences of recognizable sites on M_S and topologically by a branched network that is made up of nodes and bifurcation points.

A node is characterized by a cluster of seabed characteristics. It has a position and a viewpoint associated with the local characteristics. These characteristics are represented by keypoints of the local scene. The position of the cluster is determined by its centroid.

In the exploration, C-SLAM has to decide at each moment the next step of movement that starts from the current node. Each step consists of the identification of the next node, the direction of its centroid, and the speed of navigation to bridge the gap between nodes. The detection and identification of a node is a process that can involve many steps, during which a node is tracked and classified among other nodes in a merit order list.

In order to choose a suitable node among many possible ones, the vision system evaluates the most outstanding characteristics of the scene and then decides the optimal direction towards the corresponding node. The optimization of this direction results from the maximization of a cost function of the quantifiable requirements. Similarly, the rate is properly determined based on a previously specified vehicle cruising speed. Thus, the speed can be regulated to go up and down according to the curvature of the path. This construction procedure is repeated in the next step permanently during the exploration.

Cluster tracking involves localization on both a scattered local map and a dense map. The estimation of distances between nodes of any network branch is decisive for the planning of revisits according to the available margins of autonomy of the vehicle.

It is worth noting that estimated distances on the maps are simply evaluated on the particular scale of the monocular SLAM method, which does not necessarily coincide with the actual scale. This would not represent an obstacle to C-SLAM as long as the estimated distance between the starting point and any point of no return on the network is exactly related to half the energy autonomy of the vehicle. Some approaches define another type of scaled topology, achieved by means of stereo vision or scanner [8, 10, 27].

Bifurcation points are commonly distributed along each branch. There are two different sets of bifurcation points. The first set includes divergent points at which the paths physically bifurcate. The second set covers the so-called open bifurcation points, which were identified and ranked in a list of suitable landmarks for a future deviation of the path. The ranking position is issued in accordance to a criterion of robustness.

While each new corridor is simply added to the network as a new branch, existing corridors are updated in each new revisiting. Each round trip implies that the vehicle starts with plenty of energy autonomy. This indicates the dimensions of the explored region. Clearly, with the deployment of multi-robots, the scan will be faster [28, 29] but will not necessarily cover a larger region than with this proposal.

Since the camera is monocular, safe return is practically implemented by guiding the vehicle in reverse. This is because the system might not identify all the nodes in the 3D landscape in the way back, i.e., when navigating from the opposite direction. However, with the camera in reverse, the system is able to recognize the seabed features on the path straightforwardly. In reverse motion, all nodes are revisited but from back to front.

4. Construction of a corridor net

Before starting a round trip in the network of corridors, C-SLAM must make the decision to advance the exploration or revisit explored branches of the network. Commonly, both decisions are involved with the intent to expand the explored region. In addition, the system defines a suitable ground altitude for navigation with the end of achieving good visibility conditions.

For the next discussion, **Figure 2** is taken as support.

The first corridor and each new section of the corridor network are constructed according to an optimum criterion. This criterion takes into account the most outstanding characteristics of the seabed in the form of keypoints (shortened KP), which can be categorically recognized over time and preferably from different points of view.

Autonomous navigation begins with a commissioning phase (CP) to adjust the controller and initialize SLAM techniques for scattered and dense mapping. To achieve these objectives, the vehicle is forced to travel fast short distances, zigzagging over the bottom with a rather erratic course. In this way, images can reflect the changes in texture and parallax that are necessary for a visual odometry. This, in turn, produces a stable estimation of the vehicle localization at the beginning of the exploration.

Once odometric data η and $\dot{\eta}$ are considered reliable, the adaptation of the controller coefficients is carried out by providing persistent and rich-in-frequencies changes in the reference position $\hat{\eta}_{ref}$ and rate $\dot{\hat{\eta}}_{ref}$ in the main degrees of freedom. To this end, it is convenient for $\hat{\eta}_{ref}$ to continue in time the previous erratic motion. Additionally, the rate $\dot{\hat{\eta}}_{ref}$ is defined as the sum of the cruising rate and a random sequence that causes soft accelerations and brakings along the trajectory. Once the convergence of the controller coefficients has been attained, the exploration starts at the current position. This position is marked as the starting point (shortened SP) and fixed for future trips.

Figure 2 (left) illustrates the construction of a corridor path from SP up to an end point (shortened EP), for instance, the estimated no-return point. The system analyzes the texture characteristics in the image sequence in order to detect robust clusters (C). Once they are recognized, the system tracks them as they pass in front of the camera. With a set of robust clusters, the system is able to triangulate frame by frame the pose of the camera.

Usually, raw information for robust cluster detection is provided by the feature-based SLAM method employed in the vision system. Certainly, for self-localization, the SLAM method constantly defines keyframes (KF) with the most stable keypoints in the image, i.e., with those that have been followed up so far. Thus, C-SLAM has to group the robust keypoints in clusters and finally perform an

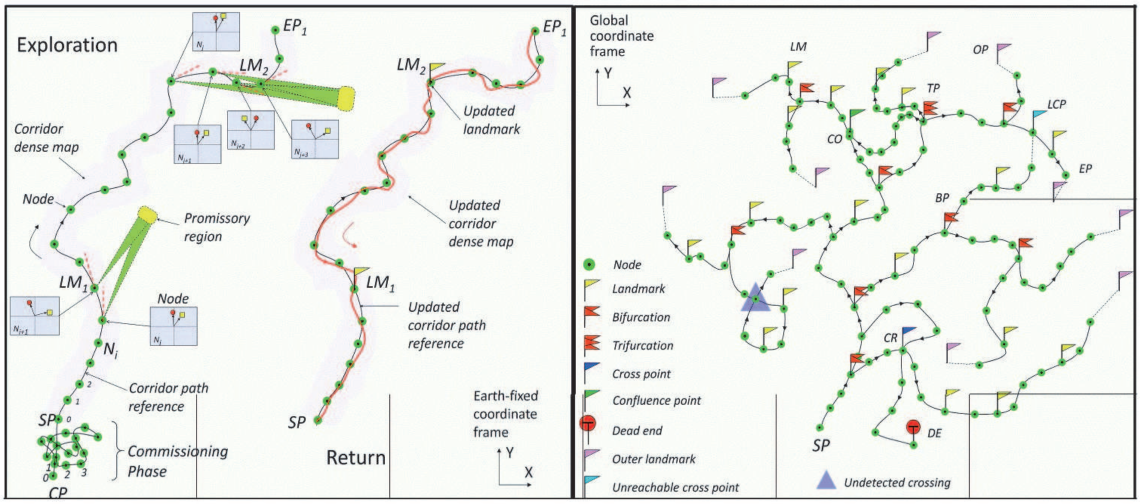

Figure 2.
Round trip during exploration (left). Corridor network topology (right).

evaluation of cluster set. In this way, C-SLAM can define nodes (N) and eventually bifurcation points (BP).

The topology of the corridor network usually contains a diversity of elements that are described in **Figure 2** (right), shortened SP, EP, KP, KF, C, N, LM, and BP. Additionally there are trifurcation points (TF) which are created from BPs: confluence points (CO) that occur between two paths that meet at a point; outer points (OP) which are detected landmarks or nodes but due to energy reasons are unreachable; dead-end points (DP) at which the vehicle must turn by force due to the absence of Ns or LMs in the horizon; and, finally, crosspoints (CR) at which a path crosses another path and these can or cannot be detected by the vision system.

It can frequently happen that, due to the existence of COs and CRs, many possible path alternatives are established to return from any point or to join two distant points in the network. Loops can also be generated during the expansion (see **Figure 2**, right).

As mentioned previously, the topological space is subject to an unknown and variable cartographic scale that produces accumulative odometric errors. This means that corridors can cross others without this being reflected in the global map. However, this does not invalidate the objective of C-SLAM, to return safely to SP.

It is expected that after reiterated revisits along different corridors, the number of loop closures carried out by the basic SLAM method will increase and the network will progressively take on a more real form. Simultaneously, the scale of the map will become more uniform, and the used metric will increasingly gain in accuracy.

The construction of the network is mainly based on a criterion to define a metric in the topological space. This is discussed below.

5. Nodes, landmarks, and keyframes

C-SLAM sets up a node to accurately synchronize future movement steps. For example, the direction to which the vehicle is to be driven in the short term in order to reach the next node is precisely defined at the current node. In addition, the speed of the vehicle on this direction is also defined here. All this gives the reference position $\hat{\eta}_{\mathrm{ref}}$ and rate $\hat{\dot{\eta}}_{\mathrm{ref}}$, which are used in the adaptive control system as inputs. Many other steps are synchronized at the current node as detailed below.

When C-SLAM decides to expand the network with a new branch, it searches for a landmark in the current corridor and then deflects the path in the indicated direction.

Nodes and landmarks are gestated in a common optimization process. Optimization involves maximization of a cost function with one optimum and many suboptimal results. The optimum outcome defines the first place in a ranking list, followed orderly by the suboptimal outcomes. Nodes and landmarks are selected from this ranking list.

The direction that defines $\hat{\eta}_{\mathrm{ref}}$ and $\hat{\dot{\eta}}_{\mathrm{ref}}$ to the next node is given by the optimal result. On the other hand, the best solution in the suboptimal set is reserved for a landmark.

Unlike a node whose next direction is defined in each step, a landmark can demand many steps. A landmark is a node whose high position on the merit list ranks it for a successful future branching. Therefore, not all the nodes become landmarks.

Finally, there is a subtle difference between a node and a keyframe generated from any SLAM technique. A node is the most robust cluster in the current keyframe, and in general its appearance frequency is generally lower than the frequency of a keyframe.

5.1 Selection of optimal directions and landmarks

During exploration, keyframes are the basis for selecting nodes. For every few keyframes of the sequence generated by the active SLAM algorithm, C-SLAM chooses the current keyframe and defines a node. The frequency of the node appearance depends physically on the visibility conditions and is commonly fixed in the commissioning phase.

On the other side, the selection of a path direction from the current node to navigate to the next (yet not defined) node follows a particular criterion. Also the selection of a promissory landmark for a future bifurcation is subject to the same criterion.

It is assumed that many keypoint clusters are faced at the same time in the ranking process in order to establish an order of merit. When the score of a particular cluster exceeds a threshold value, it is then considered as a potential landmark. Clearly, many clusters remain in the vision field for a time, so that they may continuously be tracked during their latency periods. Some of them will become landmarks.

After a landmark is lost in the field of vision, it will remain inactive with its last score and its location on the map. While in the exploration phase landmarks are generated, in a revisiting period they are consolidated, or even separated. Eventually, a landmark becomes a bifurcation point when C-SLAM decides to branch out to explore new regions.

The proposed criterion is based on a cost that is defined as a linear combination of quantifiable requirements to be met by each cluster in its latency period. Thus

$$V_i = \lambda_1 f_C + \lambda_2 T_C + \lambda_3 \Delta\theta + \lambda_4 \delta^{-1} \tag{1}$$

where V_i is a cost for one keypoint cluster named C_i, given the weights λ_j with $0 \leq \lambda_j \leq 1$ and $\Sigma_j \lambda_j = 1$. The requirements for C_i are described and symbolized as

- High density of keypoints in a cluster: factor f_C
- Continuous traceability of a cluster: period T_C
- Robustness against the change of points of view: span angle $\Delta\theta$
- Alignment of a cluster on the heading direction: factor δ^{-1}

The cost weights will be defined previously according to the emphasis placed on some particular aspects of robust navigation. Since all requirement variables will be normalized to their expected highest values, a trivial choice $\lambda_j = 0.25$ is generally satisfactory. The simple election $\lambda_1 = 1$ should support a continuous navigation; however in this case zigzagging on the pathway should not be ruled out.

Since some detected landmarks are far away from the vehicle position, one should check whether these will be attainable with the current autonomy range of the vehicle. If they are unreachable, they will be removed from the ranking list or marked as OPs. To this goal a metric is required for the topological space of the corridor network.

5.2 Cost parameters

5.2.1 Density of keypoints in the cluster f_{C}

The key assumption to calculate the first parameter f_C is that visual features are very volatile in harsh underwater environments. Therefore, dense clusters may be thought to be more resistant to long-term disappearance than dispersed clusters. The first step to determine f_C is to group keypoints of the current keyframe in clusters. For this, for example, k-means clustering can be used in combination with an efficient initialization method. For the time k of cluster appearance, the following is calculated:

$$f_C(k) = \frac{N_{KP}(C_i)}{\max N_{KP}} \tag{2}$$

where i identifies the cluster of the set and N_{KP} is the number of KPs in C_i. Thus, the parameter can be determined recursively by

$$f_C = \frac{(k-1)f_C(k-1) + f_C(k)}{k} \tag{3}$$

Since the set of selected robust keypoints in the SLAM algorithm is generally sparse, clustering operation should not be excessively time-consuming.

Statistically, the maximum expected number of N_{KP} can be estimated in the commissioning phase as a function of the harshness of underwater environments. For example, poor visibility, self-similarity of the ground, caustic sunlight, shadows, flares, etc. are factors that contribute to reduce the number of robust keypoints.

5.2.2 Period for continuous traceability T_{C}

The second parameter is related to the traceability of C_i. The longer C_i is detected, the more feasible it will be to achieve continuity of navigation and a safe return. The parameter is estimated in each keyframe as follows:

$$T_C = \frac{N_{KF}(C_i)}{\max N_{KF}} \tag{4}$$

where N_{KF} is the number of consecutive KFs in which C_i appears approximately at the same position and almost with a similar density. To assess the similarity of the N_{KF} clusters, a Euclidean norm can be used. In addition, attention should be paid to changes in f_C and cluster midpoint. N_{KF} is estimated by thresholding all these variables properly.

Finally, the maximum number of consecutive keyframes depends to some extent on vehicle speed, altitude, camera tilt, and field of vision, among other factors. As a reference for $\max N_{KF}$ one can take the number of KFs required to cover a distance of 10 times the length of visibility in the line of sight.

5.2.3 Span angle $\Delta\theta$ *for viewpoint range*

The third parameter $\Delta\theta$ relates the robustness of C_i with the variation of viewpoints. A fairly large angle would be necessary for a cluster to be unambiguously detected from different points of view. $\Delta\theta$ applies in these two important cases: first when clusters are aligned in front of the vehicle and second when they are aligned to one side of the vehicle (see example in **Figure 2**, left, for landmarks LM1 and LM2). Thus, it results in

$$\Delta\theta = \frac{\text{span}\ \{\theta_j(C_i)\}}{\max\ \text{span}\{\theta_j\}} \tag{5}$$

where the symbol span{.} means the difference between the maximum and minimum value of the sequence θ_j during the tracking of C_i. In turn, θ_j is the angle between the line that joins the camera with the cluster midpoint and the camera optical axis.

A large value of $\Delta\theta$ suggests a more robust link between the nodes and, on the other hand, in the case of landmarks, a more successful bifurcation.

The maximal span of θ_j is set equal to half of the horizontal field of vision of the camera, which specifies the tracking of a cluster that is initially seen far away in front of the vehicle and disappears from the field of vision sideways.

5.2.4 Course alienation factor δ^{-1}

The fourth parameter δ^{-1} plays a key role in the construction of the corridor. When the corridor is built step by step, one would normally prefer to keep the real course along a straight line rather than change the course. This is because the control of the vehicle along a line is generally more precise than on curves. Similarly, revisits on straight stretches are more immune to loop-closing failures than on curved stretches.

Therefore, in the case of nodes, δ^{-1} should generally produce a higher cost than in the case of landmarks that are reserved for bifurcations. Hence, landmarks are generally created on one side of the path, while nodes are rather in a straight line.

Since the cost (Eq. (1)) contains terms that must be maximized, δ^{-1} reflects the opposite of the alienation parameter here called β. Therefore, δ^{-1} should indicate how much the transverse separation of a cluster from the route is δ^{-1} assessed in the horizontal navigation plane. To this end, both the optical axis and the line that joins the camera with the midpoint of the cluster are projected onto the navigation plane. Between these lines, the angle β_j is obtained. A third line is defined, which connects the camera with the point furthest to the right of the vision field. Projecting this line on the navigation plane results in the angle called β^* with the projected optical axis.

Hence, it results in geometric mean

$$\delta^{-1} = \frac{1}{N}\sqrt{\sum\nolimits_j \left(\frac{tan\ \beta^*}{tan\ \beta_j}\right)^2} \tag{6}$$

where N is the number of assessments j during cluster tracking.

5.3 Application of the selective criterion

The inclusion of a cluster C_i in the merit order list during its tracking is subject to

$$V_i > V^* \tag{7}$$

where V^* is a global threshold given for the cluster score. The merit order list is updated keyframe by keyframe. Hence, if C_i is already in the list, its score is simply actualized.

If V^* is set too high, many landmarks might be created, and the network could be unnecessarily too dense. The optimal number of bifurcations per corridor should satisfy a general rule that the distance between two consecutive BPs is spanned by about 20 nodes. On the other hand, if the branches are too scattered, V^* could be too high.

When a new node is created, C-SLAM immediately evaluates the next most promising direction to lead the vehicle into the unknown environment with the goal of exploration. Therefore, the highest score refers to the cluster with which the reference path $\hat{\eta}_{ref}$ should be continued.

In addition, C-SLAM checks whether the second highest score corresponds to an active cluster and in which case decides to set a landmark. Another more cautious strategy is to set a landmark when the second highest cluster just disappears from the list.

Landmarks can be also consolidated in the return and in every passage on any explored corridor. On the other side, the landmark cost may decrease so much that it is to be removed from the list. In this way, landmarks are permanently evaluated up to the moment they are employed to bifurcate into the corridor. At this moment, they change their status from LM to BP and disappear from the list forever.

In a changing environment, the adaptation of Ns, LMs, and BPs after a long period is necessary. This allows a renewal of the corridor network as needed (cf. [7, 30]).

During assiduous navigation, the topology of the network is outlined like a tree of branches. Each branch contains its identification in the tree, the sequence of nodes arranged in one direction, the landmarks, the bifurcation points, and any other particular element that should be important for decision-making.

6. Expansion of the network

Extending the boundaries of the corridor network requires C-SLAM to implement certain policy-makings, which goes beyond the corridor network building.

A first policy implies the definition of an appropriate metric to extend the lengths of the corridor network as much as possible according to the available energy autonomy.

A second policy supports the decision-making to choose bifurcations in order to multiply the number of branches. In this case, revisitings are of secondary importance as they take place only when they are needed as bridges to create new branches.

A third and final policy concerns the optimal scheduling of paths between two points of the network. In a situation with many alternatives to get a connection between sites, the system will search for the one with the best energy efficiency.

6.1 Metric

The metric space represented by the corridor network depends on an unknown scale which also changes over time due to cumulative odometric errors. This will affect, above all, the global map, which is the composition of many local maps with their own scale. As scale variations are generally small from map to map, the metrics are similar.

In order to maximize the length of a corridor without compromising the safety of the way back, it is essential to have a reliable metric. Since safety and energy autonomy are closely related, the metric must express the energy margin that the vehicle possesses at any time and position in the network (cf. [24, 25]).

It is clear that due to the challenging environment, the energy used to connect two points is often not the same in both directions. This will undoubtedly depend on sea currents, vehicle speed changes, travel breaks, and zigzagging on the reference route. Thus, for example, real-time detection of the no-return points based on the battery state of charge may not be reliable enough in unforeseeable circumstances. For this reason, a more powerful detection of no-return points is developed here on an empirical basis.

To this end, a function for the estimation of the energy margin is proposed. It adds up the energy consumption until the full autonomy is completed. The function can be evaluated at any time, especially in the decision to return.

The idea behind the proposal is that almost all the energy available in navigation is intended to move the vehicle. Therefore, an approach that is based solely on the energy of motion along the travelled path seems to be quite rational.

The first one defines the consumption of energy as a set of possible cases:

$$E\left(\hat{\dot{\eta}}, \hat{\eta}\right) = \left\{ \int_{C_i} \hat{\dot{\eta}}^2 (\hat{\eta}) d\hat{\eta} \right\} \tag{8}$$

where $\{.\}$ is a symbol for a set of cases and $\int_{C_i}\_d\hat{\eta}$ describes a curvilinear integral along the path i that spans the way from the SP up to the point where the autonomy is completed (see **Figure 3** to support the concept). Similarly, the distance covered by the vehicle on this path i until all available energy is consumed is defined as a set of cases:

$$d(\hat{\eta}) = \left\{ \int_{C_i} d\hat{\eta} \right\} \tag{9}$$

It should be noted that all routes cover each shape, both straight and curved ones.

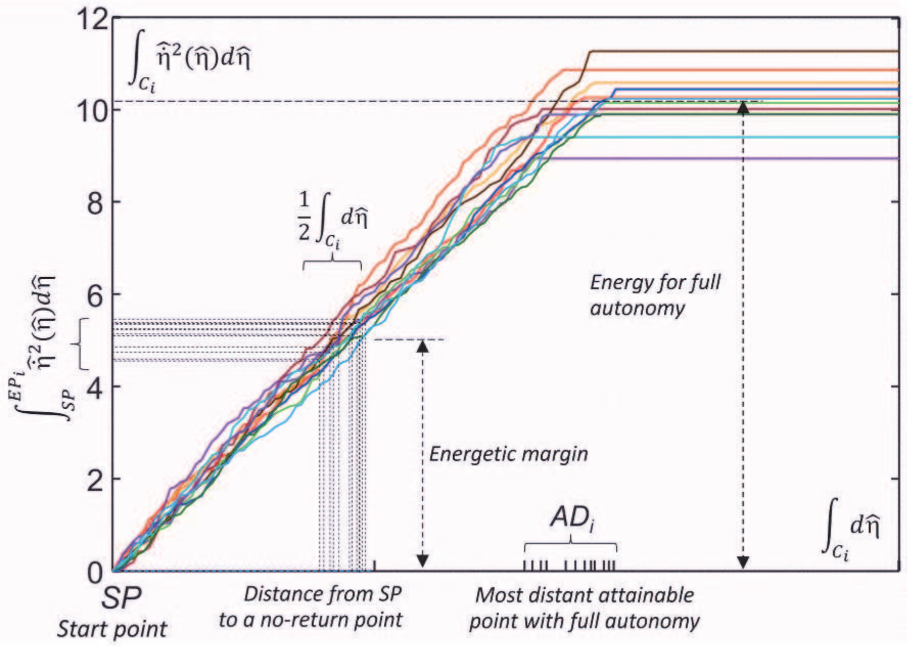

Figure 3.
Energy margin to reach total energy consumption. Numerical identification of no-return points on different paths.

Figure 3 shows numerical simulation results wherein the distance travelled by a vehicle over different paths is computed after accomplishing full autonomy. In the experiments the vehicle acquires a cruise velocity, but this is slightly changed around this value at random. The resulting ground truth statistics must be available before applying C-SLAM but are updated during the construction of the corridor network.

In order for C-SLAM to decide the time point for the comeback on a new corridor, a norm based on the average on the sets in Eqs. (8) and (9) is applied. The Euclidean norm can be used to obtain the energy limits called E_{EP} and travelled distance d_{EP}:

$$\|E_{EP}\|_2 = \frac{1}{N}\sqrt{\sum\nolimits_i \left(\int_{SP}^{EP_i} \dot{\hat{\eta}}^2(\hat{\eta}) d\hat{\eta}\right)^2} \; and \; \|d_{EP}\|_2 = \frac{1}{2N}\sqrt{\sum\nolimits_i \left(\int_{C_i} d\hat{\eta}\right)^2} \tag{10}$$

subject to the conditions for safe navigating on the path *i*

$$\|d_{EP}\|_2 \geq \int_{C_i} d\hat{\eta} \tag{11}$$

$$\|E_{EP}\|_2 \geq \int_{C_i} \dot{\hat{\eta}}^2(\hat{\eta}) d\hat{\eta} \tag{12}$$

where the equality of either of the two equations will mean the identification of the end point EPi for this path. N is the number of trips in the statistics that cover a distance d_i for any route from SP. It is very important to frame the confidence of the norm in the risk of loss of the vehicle. In this sense, the most unfavorable case might be more appropriated. In this case, it is valid

$$\|E_{EP}\|_{max} = \underset{C_i}{max} \left\{\int_{SP}^{EP_i} \dot{\hat{\eta}}^2(\hat{\eta}) d\hat{\eta}\right\} \; and \; \|d_{EP}\|_{min} = \underset{C_i}{min} \left\{\frac{1}{2}\int_{C_i} d\hat{\eta}\right\} \tag{13}$$

and so the conditions for the detection of a no-return point are as follows:

$$\|d_{EP}\|_{min} \geq \int_{C_i} d\hat{\eta} \tag{14}$$

$$\|E_{EP}\|_{max} \geq \int_{C_i} \dot{\hat{\eta}}^2(\hat{\eta}) d\hat{\eta} \tag{15}$$

where again the equality of either of the two equations will mean the identification of the end point EPi for this path *i*.

In order to continuously adapt the norm to the environment, the set (Eqs. (8) and (9)) must be updated on each round trip. It can be noticed that the update can be applied both in a new corridor and in a revisiting case as well, regardless of the path chosen.

6.2 Planning of new corridors

The way in which C-SLAM progressively explores and increases branches can be supported by different criteria.

6.2.1 Method I

The first method is based on the criterion that for fast expansion of new branches, occurrences of revisitings of old branches should be reduced as much as possible. To this end, the number of revisited stretches in old branches and their distances to SP

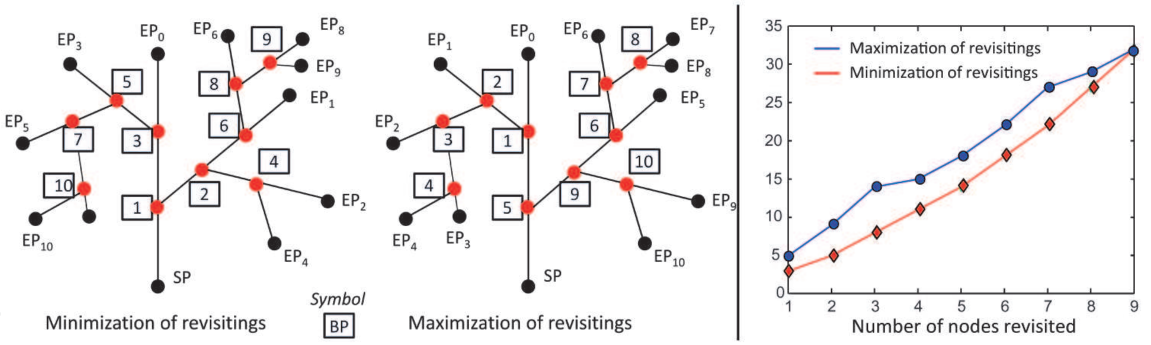

Figure 4.
Generation of BPs according to different criteria: the fastest branch expansion (left) and the slowest branch expansion (middle). Number of revisited stretches during network expansion: optimal expansion (red) and a suboptimal expansion (blue).

should be minimized during the sequence of round trips. Here, the rectification of the path $\int_{SP}^{BP_i} d\hat{\eta}$ is employed. It is evident that to reach a certain BP on an old branch, the vehicle must unfailingly revisit some first stretches on this branch.

Figure 4, right and middle, illustrates a case study of an expanded network in two different forms.

Once the trunk corridor is created, the proposed method starts from SP and searches for the closest LM and splits into a new branch until it culminates in an EP. At this stage, this LM becomes a BP. In doing so, the approach minimizes the length of all revisited stretches. Any other branch generation is suboptimal. The worst suboptimal generation, on the contrary, expands the network by starting each round trip from SP to the most distant LM. In fact, this option produces a maximization of the revisited stretches.

Even when the optimal branch generation expands the network faster than other options, all methods end up with the same number of revisited stretches. This fact is reflected in **Figure 4** (right). Therefore, from the point of view of the total energy consumed, the difference is short and medium term only.

6.2.2 Method II

The optimal solution described above has a theoretical rather than a practical value. The disadvantage of method I is that the way BPs are used has no connection with the merit order list. For that reason, vehicle safety might be compromised. When making decisions about where to branch, trust in the list is the only support for secure expansion.

Therefore, the main focus of the second method is to build up a solid network in which the path from every EPi to SP maximizes the trust on the BPs on the pathway. This means that the LMs chosen for bifurcation have the maximal score on this pathway. The number of revisited stretches over time is described by a curve that lies between the two extreme curves in **Figure 4** (right). This means that in the long term, the new curve will also converge to the same extreme value.

Many other strategies can be applied besides the other two methods. For instance, one could be specifically interested in developing the branches to the left (or to the right) of the trunk line or in creating a solid statistic for the newly explored branch before continuing with the exploration.

6.3 Path scheduling

Starting from the conception of a fully interconnected corridor network, multiple alternatives can be considered to obtain a connection between two distant

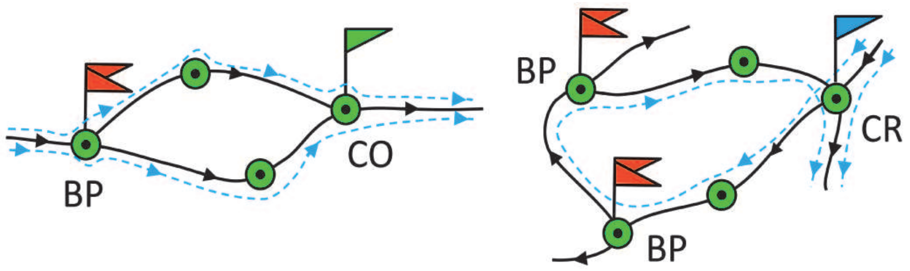

Figure 5.
Bypass (left) and loop (right) in the network topology.

points of the network (see, for example, in **Figure 3**, right, the bypass shaped by the points BP-CO and BP-TP-CO). In such situations, C-SLAM should be able to seek that path that involves the best energy efficiency. For that purpose C-SLAM must count with odometric information of each stretch and the energy consumption cost per unit of length.

Once the network has converged in its expansion and especially after a large number of revisits, it is common to count with the presence of bypasses and loops. Thereupon these elements will become part of the topology of the network.

In the case of one-way corridors, the existence of a bypass can only occur in the presence of COs or CRs (see **Figure 5**, left). On the other hand, the generation of a loop needs indefectibly at least one CR (see **Figure 5**, right). It is worth noticing that a loop represents a dummy alternative to connect a point with oneself through the loop as illustrated in **Figure 5** (right).

As topological elements of the network, loops are marked to avoid unnecessary energy consumption. For example, in **Figure 5** (right) when a loop is encountered, the path entering the loop is avoided unless the destination is a BP allowing exit from the loop. For the optimization of paths, the criterion consists in searching for the shortest path between two points or for the path that minimizes the energy consumption, or a combination of both. If safety aspects are emphasized, the energy approach is the most suitable for C-SLAM. This option is described in the following.

To find the optimal path, the following cost functional is minimized for a path starting at node N_A and ending at node N_B:

$$J(N_A, N_B) = \underset{\{N_A, N_B; C_k\}}{min} \left\{ \sum_{N_A}^{N_B} \int_{N_i}^{N_{i+1}} \hat{\dot{\eta}}^2 (\hat{\eta}) d\hat{\eta} \right\} \tag{16}$$

where C_k is one of the paths that connects N_A and N_B and the pair N_i and N_{i+1} is the terminal nodes of the stretch in C_k. The energy information on the stretch from N_i to N_{i+1} is the result of an average each time that the vehicle passes on this section. For the minimization in Eq. (16), BPs and COs, even dead-end points, are taken into account along with all nodes. Thus, the minimal value of the sum on every stretch that links N_i and N_{i+1} of C_k represents the optimal path denoted by C_k^*.

To assist Eq. (16), the A* search algorithm has been chosen in this work over other tree search algorithms because of its optimal efficiency [31]. Therefore, the cost is

$$F(N_{i+1}) = g(N_{i+1}) + h(N_{i+1}) \tag{17}$$

where g is the cost of the path from N_A to N_{i+1} and h is a heuristic function that estimates the cost of the cheapest path C_k^* from N_{i+1} to N_B. A* ends when the goal is reached or when there are no paths eligible to be extended. In this work, the cost functions g and h are selected according to the following two terms:

$$F(N_{i+1}) = J(N_A, N_{i+1}) + \underset{\{N_{i+1}, N_B; C_k\}}{min} \left\{ \sum_{N_{i+1}}^{N_B} \int_{N_j}^{N_{j+1}} \hat{\eta}^2(\hat{\eta}) d\hat{\eta} \right\} \qquad (18)$$

The heuristic function $h(N_{i+1})$ is admissible provided that paths with one or more sections travelled in both directions are excluded from the minimization. Thus, branches that do not represent a bypass to the branch on which N_B is located are discarded. Loops are also avoided when used to connect a node to itself.

7. Case study

This section shows some experimental results which are selected to illustrate the viability of the proposal for autonomous navigation based on the proposal. Many of the functions of the approach are tested together according to the C-SLAM structure in **Figure 1**. It is important to note that the case studies are carried out in limited spaces with simulation of light effects and seabed textures. However, the staged environment*s* reflect quite well the difficulties encountered in image processing in real underwater scenarios.

7.1 Environment

In order to provide a ground truth for testing and to achieve acceptable reproducibility of results, many scenarios were set up in a basin with a staged underwater landscape that closely resembled the natural seabed. In this scenario, rocks, gravel, sandbanks, and benthos, among others, predominated wherein a variety of underwater visual effects could be obtained [19, 23] (see **Figure 6**).

The bioactivity of microorganisms was cyclically changing the characteristics of soil texture and water transparency. The scenarios were illuminated by direct and indirect sunlight. However, light disturbances and turbidity were controlled for the

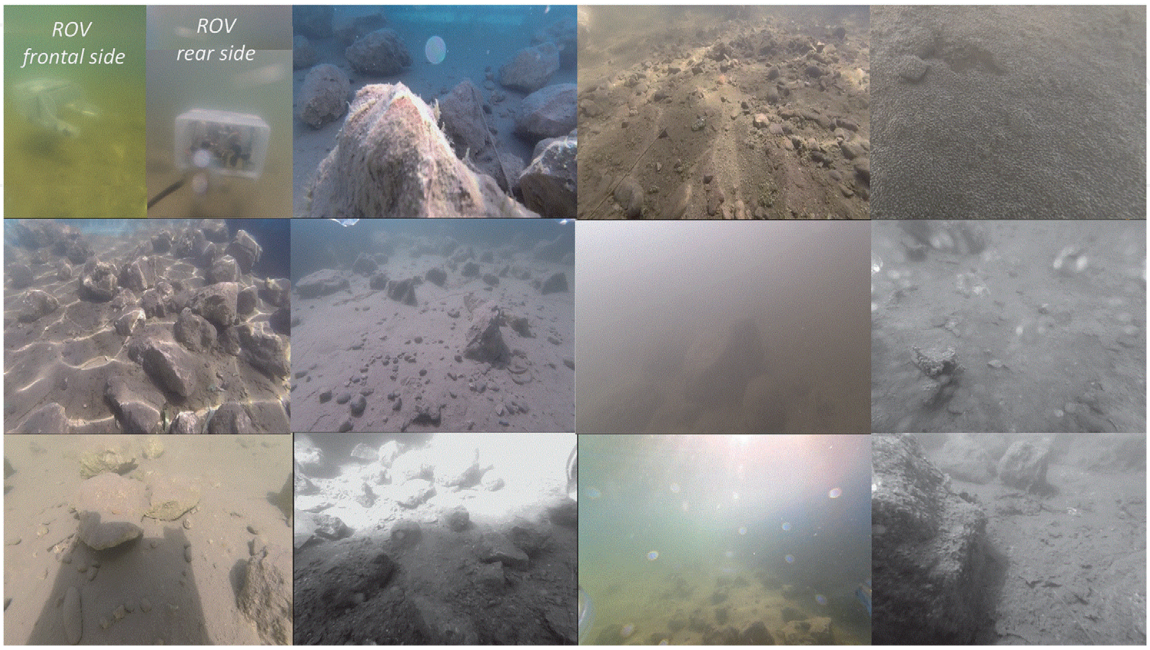

Figure 6.
Stage installations for tests underwater. Diversity of scenarios.

range of tests. For instance, strong or weak sunlight flickers on the ground were generated by agitating the water in two orthogonal directions. In addition, visibility was reduced by discharging silt particles onto the surface that remained suspended for a period of time.

Therefore, a wide variety of case studies could be faithfully reproduced, such as rapid changes in luminance, transition from dark to bright scenarios, blurriness, lens flares, motion blur, self-similarities, glare, and bubbles (see **Figure 6**).

7.2 Hardware

An ROV (model OpenROV v2.8) was used as the platform for the experiments, although this was hydrodynamically reformed with side fins to reduce the motion blur.

Two independent cameras were installed on board, a high-resolution, wide-angle vehicle camera (Genius f100) and a high-performance camera (GoPro Session H4). Both cameras are rolling shutter and operate at different frame rates. To attenuate undesired effects of the rolling shutter mechanism, especially in photometric-based algorithms, a high speed of 120 fps with an image size of 848 × 480 pixels was used for dense mapping, albeit off-line at the end of every round trip.

The vehicle was completely steered by C-SLAM that was programmed and installed in a notebook with GPGPU technology. The bi-directional flow of video and control signals was implemented via cable. The altitude was conveniently fixed in advance to adapt to the visibility of the environment. Altitude control was carried out independently of the adaptive controller by a PI controller. An adaptive speed-gradient controller [17] was used to self-adjust the controller coefficients for the main dynamics of the ROV.

7.3 Software

Among other functions, C-SLAM uses state-of-the-art free software to localization and mapping. As shown in **Figure 1**, the block for feature-based SLAM was implemented with the algorithm ORB-SLAM [32]. On the other hand, the block for dense mapping was implemented with the photometric-based DSO-SLAM [33] and sometimes with LSD-SLAM [34, 35]. However, there was a significant modification in the implementation, namely, DSO (or LSD-SLAM) was supported with the estimated camera position provided by ORB. This was necessary in general for improving the stability of the mapping in very harsh environments, especially due to the presence of strong sunlight caustics or/and poor visibility [23]. The deflickering filter in this work is based on estimations of sunlight caustic fringes using a feedback of predicted images [20], which employs very high accuracy velocity estimation [36]. Software for guidance, control, and corridor construction was developed specially for C-SLAM.

7.4 Results

One begins with **Figure 7** which illustrates the role deflickering filter plays in the improvement of dense mapping in environments with strong sunlight caustics. Generally, spatiotemporal light changes affect the performance of photometric-based methods seriously. As seen in the heat maps, the camera depths are coherent with the physical environment. Thus, the photometric consistence is preserved after image deflickering.

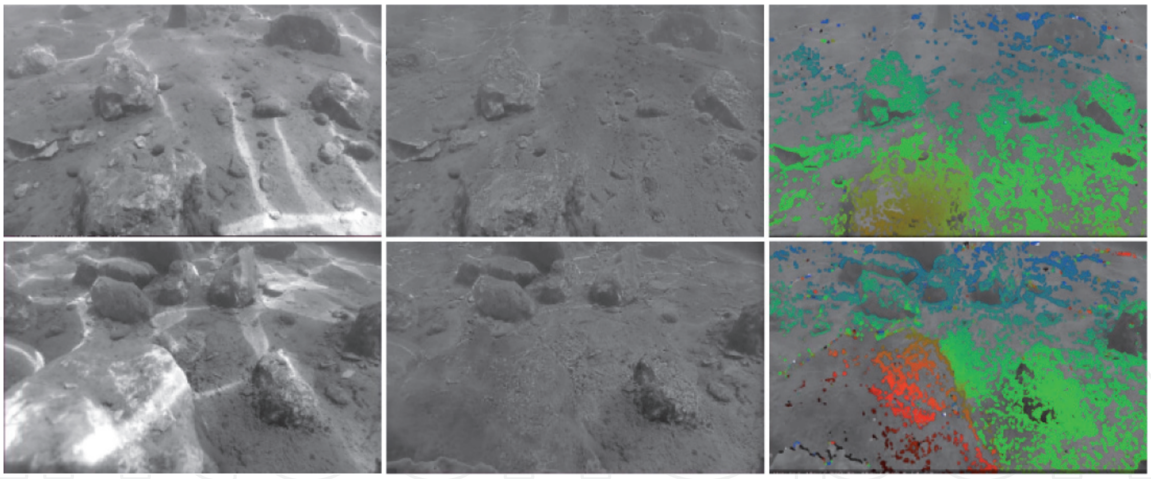

Figure 7.
Dense mapping in two cases after image deflickering. Images with sunlight caustics (left), deflickered images (center) [14], and heat maps for camera depth (right).

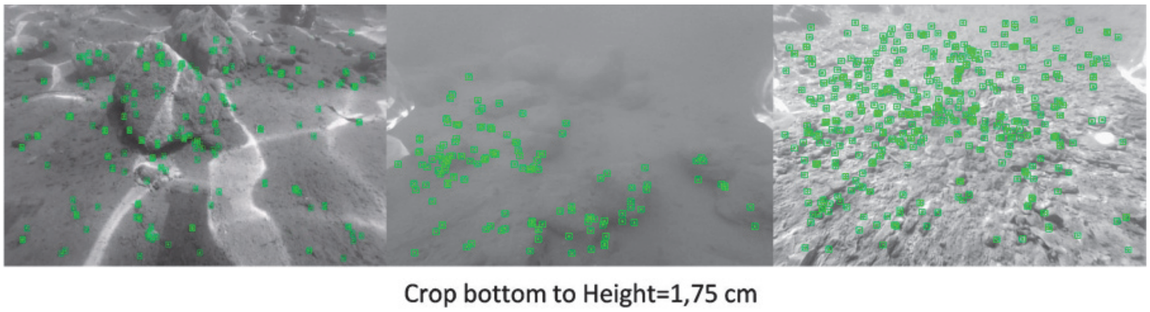

Figure 8.
Selection of robust keypoints in three harsh environments: with strong sunlight caustics (left), low visibility (center), and self-similarities (right).

The situation is totally different in the case of ORB-SLAM, where light disturbances do not affect ORB to the same degree as DSO (or LSD-SLAM) (see **Figure 8**). However, from several experiments, it was concluded that in turbid waters, robust texture features decrease substantially, although the overall performance of ORB does not degrade as much as in the case of DSO (see [8]). Therefore, the filtering of caustic sunlight waves can be omitted in the case of ORB, but not in the case of DSO (or LSD-SLAM). For this reason, the deflickering filter in the C-SLAM structure in **Figure 1** is necessary only for dense mapping ends. Another conclusion was drawn from navigation in self-similar terrains staged in the basin like in the third picture to the right in **Figure 8**.

These terrains provide commonly numerous features in a similar cluster pattern, but ORB-SLAM often losses the track as it is unable to deal with nearby similarities.

An important instance at the beginning of C-SLAM navigation is the start-up phase for initializing SLAM algorithms and adaptive controller parameters. To this end, in the study, the vehicle movements were performed manually providing a zigzag path of the camera. **Figure 9** illustrates the initial process of constituting a dense mapping of the environment with an adequate camera trajectory. From there, the start of the exploration was supplied with good estimates of vehicle position and speed, which in turn, allow the adaptive controller to adjust its coefficients.

Dense methods suffer, more than any other class, from odometric errors [8], which are minimized in the particular case of DSO through a very cumbersome and thorough camera calibration process. In these trials, the combination of direct methods for mapping with ORB-SLAM for tracking increases the accuracy of the global map of the corridor network, even in the case of normal rolling shutter cameras as used here. **Figure 10** illustrates these results in one experiment with good textured floor.

The limited space and relatively short time span for the experiments prevented the use of an energy autonomy metric beyond the context of numerical computer simulations. In this sense, end points located sideways must be at a certain limited distance from the trunk corridor. The frequency of occurrence of nodes was synchronized with the keyframe generation according to the simple strategy, namely, "one keyframe, one node." The direction to the next node was optimized primarily by the maximizing the density of cluster keypoints near the front line of vision to avoid zigzagging of the vehicle. In this way, branches could be extended to a relatively significant length.

An important feature of the monocular C-SLAM is the vehicle return through the corridor in reverse motion as seen in the display provided by ORB-SLAM by means of the symbol "☒⇨," indicating the direction of movement (see **Figure 11**, top). In addition, the track link that exists when the algorithm identifies a connection between nodes is characterized by green segments between them (see **Figure 12**, left).

Reverse movements often caused motion blur due to pulling and cable drag on the floor [37]. Besides, the drag of the ROV backside is much more pronounced than

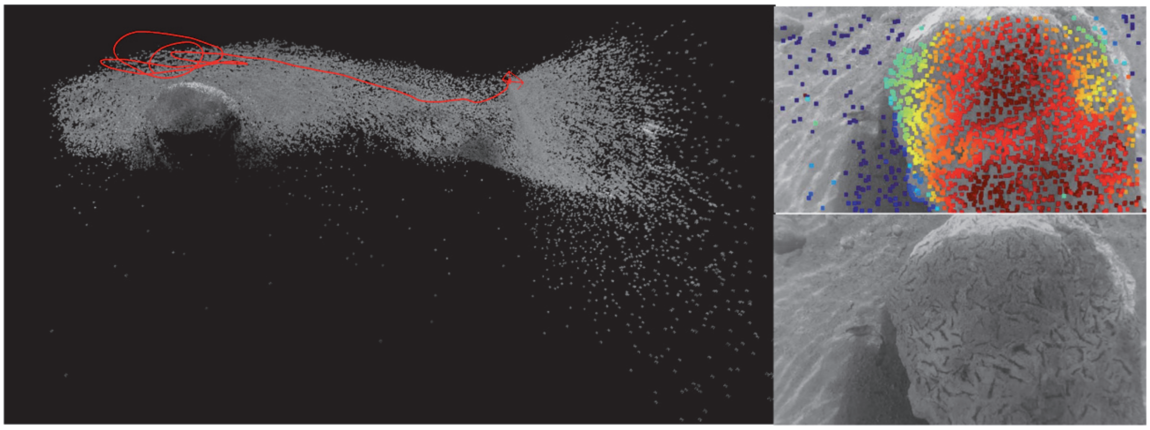

Figure 9.
Commissioning phase to initialize photometric dense mappings. Camera path on dense map (left), camera depth (top-right), and original frame (bottom-right).

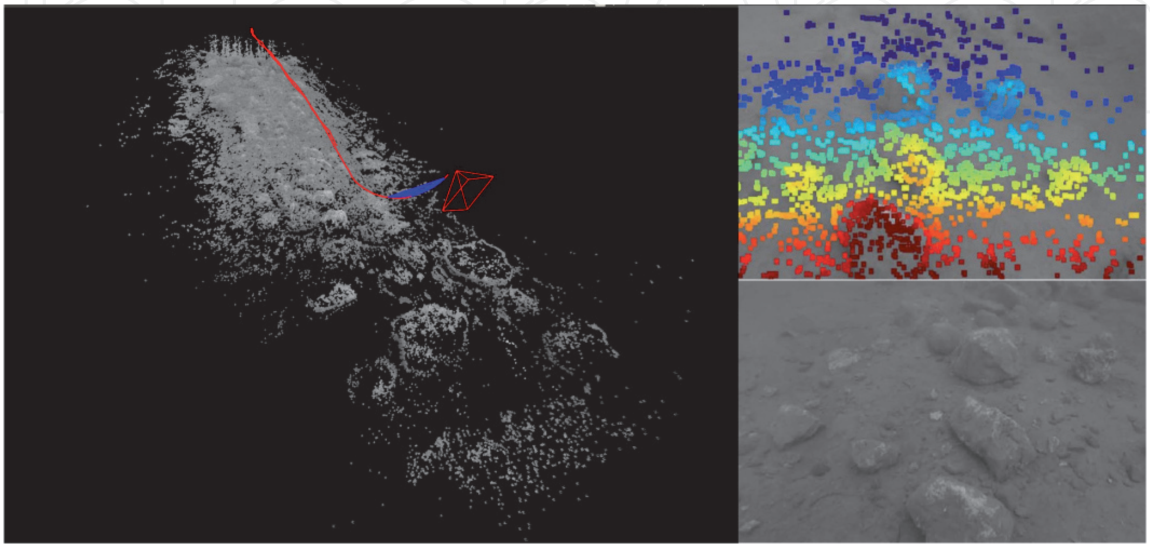

Figure 10.
Estimated camera path and dense map. Map and path (left), heat map of camera depth (top-right), and original frame (bottom-right).

in the forward displacement. All this made it necessary to reduce the cruise velocity to minimize large heading perturbations that would cause track loss. In the following, two case studies illustrate the C-SLAM performance under these circumstances.

Figure 11 shows a corridor network that was constructed under the criterion of fast expansion of new branches, i.e., minimizing of revisits in short and medium terms. Here, three bifurcation points were implemented in four steps. First, the system creates the trunk corridor and returns to SP. Accordingly, in the next round trip, the vehicle is led to the nearest BP and forced to bifurcate to create the first branch, and thereafter it returns to SP again. This routine was repeated for the second and third BP.

It is notorious that the way back through the trunk corridor from the BP3 to the SP does not completely agree with the way out. However the position track was never lost, which demonstrates the robustness of the system against changes of points of views. The differences in trajectories were due to many causes, including control tracking errors, cable tugs, and drag disturbances [37].

The diversity of terrain texture shown in **Figure 11** (bottom) and their impact in the performance and robustness of C-SLAM are noticeable. For instance, at BP$_1$, the terrain is bulky, so the contrast is high and the keypoints are robust. On the other hand, in BP2 and BP3, the terrain was acquiring an increasingly self-similar appearance, and the keypoint clusters were becoming increasingly volatile. In some trips in reverse through this self-similar zone, the vehicle briefly lost its tracked position.

Figure 12 illustrates a more sophisticated experiment in which branches were allowed to occur on both sides of the trunk corridor. Over time, in addition to the formation of branches, confluence of paths and way crossings were also appearing. The picture on the left shows the network from above and the interconnection of nodes. On the other side, the picture on the right displays the network nodes and

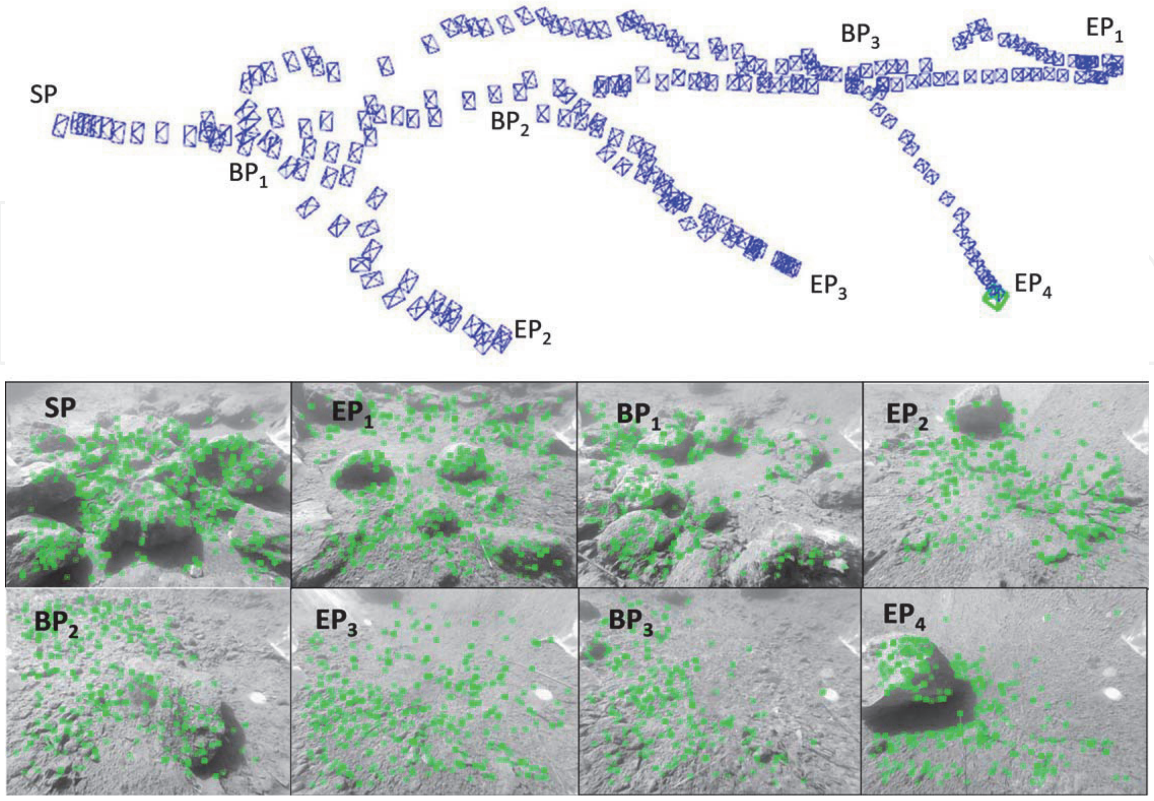

Figure 11.
Corridor network construction with three branches (top). Camera-taken scenes at the corresponding SP, BPs, and EPs (bottom).

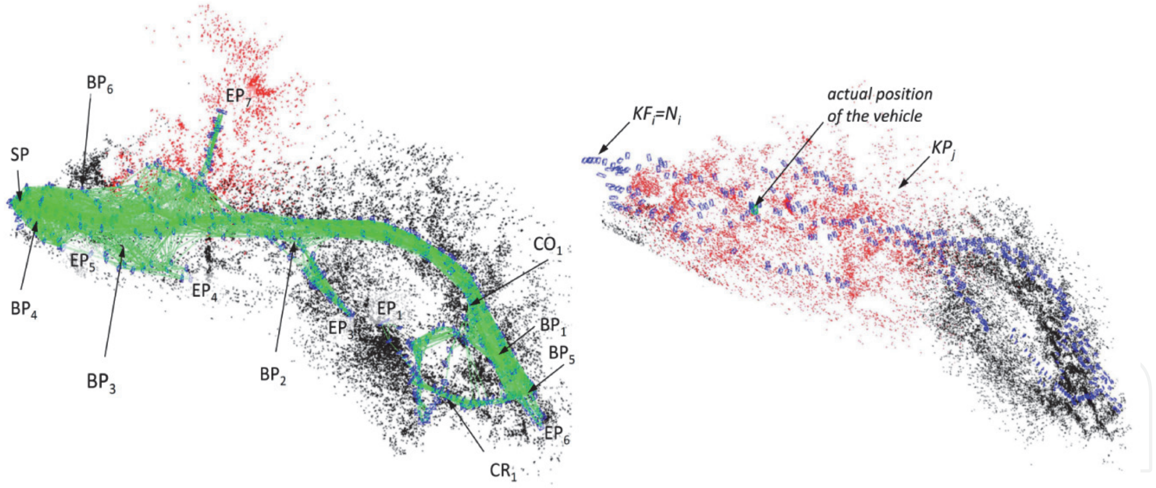

Figure 12.
Construction of a corridor network. Path graph with localization of BPs, EPs, COs, and CRs (left). Node sequence in a sparse map (right).

keypoints produced during the multiple round trips. It is observed that the density of keypoints is significantly higher in the right zone than in the left zone of the explored area. This is due to the marked self-similarity of the terrain that dispersed the keypoints over the terrain, while in the left zone, the bulky elements concentrate the keypoints around their peripheries.

The trunk corridor was long enough to encompass areas of different textures. Light disturbances were moderate; most of them transitions from dim to shining scenarios.

The sequence of the round trips can be followed by the order of occurrence of EPs and BPs. The strategy of expanding the network is the same as in the case before, i.e., create the trunk corridor, return to SP, annex a new branch, and begin a new round trip.

The most remarkable instance in this experiment occurred on a stretch in front of EP1, in the self-similar zone, where the C-SLAM briefly loses the vehicle location. Afterwards the vehicle was led in the direction of EP1 up to cross the trunk corridor. This CR was recognized by the system and aggregated to the network. From this point, the vehicle could return in reverse to a point that was also recognized by the system as a CO. Finally, after the sixth branch, C-SLAM completed the exploration by providing a global map of the corridor network.

8. Conclusions

This work deals with the autonomous navigation of underwater vehicles with the aim of achieving a broad exploration of the seabed. Unlike autonomous navigation systems in aerial, terrestrial, and space applications, for which the main source for localization is a GPS system, this approach basically uses only a monocular camera as sensor.

To establish the position of the vehicle, the proposed vision system takes advantage of the characteristics of texture of the seabed. Since underwater scenarios are generally very diverse and harsh due to water turbidity and light disturbances, vision-supported navigation poses a huge challenge for safe autonomous exploration. In particular, the lack of transparency in water forces the control system to lower the altitude to the seabed, which in turns demands a high degree of maneuverability to avoid collisions.

This work describes a vision-based system named C-SLAM that provides a nested loop structure to integrate control, guidance, navigation, and route planning systems into one. To explore the seabed, C-SLAM implements a strategy based on active SLAM. In contrast to many methods in the literature that employs topological models of the environment like feature graphs [13], Bayes tree [14], or grid maps [15], C-SLAM presents the environment as a network of interconnected corridors made up nodes and bifurcation points. It is claimed in this work that this simple topology is highly adequate and robust for harsh underwater environments with self-similarities, strong spatiotemporal light perturbations, and lack of water transparency [23].

Compared to other optimization techniques proposed for underwater applications such as random trees [38], particle swarming [36], octo-trees [39], level setting [40], and genetic background [41], among others, this proposal has a significant level of novelty. It radially searches in the field of vision for the optimal direction to explore. This direction is subject to satisfy quantifiable requirements for the navigation. The requirements are integrated in a weighted linear combination that is maximized step by step. This produces an active scoring list of promising points for selection of nodes and future bifurcation points providing adaptation to the environment and robustness of the node sequence.

The heuristic of the proposal have similitudes with the visual teach and repeat method for long-range autonomy [16]. Therein a pathway is created in an unfamiliar outdoor environment, at first by teleoperation and then followed by a rover. However, C-SLAM is essentially designed for autonomous exploration underwater and deals with a dense exploration in 2D. Besides, the submarine navigates basically on a plane over the seabed contemplating topographic features from above, not over the terrain. So, comparatively, C-SLAM has the advantage to choose the optimal route in order to be able to safely return to the start point. Another difference is that the C-SLAM can cope with the lack of map scale and odometric errors and even so ensure the vehicle return.

In order to expand the explored region as far as possible, a suitable metric in the non-scaled topological space is defined in relation to the vehicle energy autonomy. The key idea is to lead the vehicle by a corridor just up to the no-return point in order to make the return trip safe. In contrast to other approaches that solve the problem in a rather complex form, for instance, employing dynamic programming onto bathymetric and sea current maps [26, 27], C-SLAM proposes a statistic-based connection of odometric information and energy consumption.

The approach was experimentally tested in reduced-scale in a basin, wherein subaquatic environments with a good resemblance to a real seabed were staged. Experimental trials have demonstrated the feasibility of the approach in future applications where an autonomous underwater vehicle can host the C-SLAM vision system for large-scale underwater exploration. With the exception of extremely turbid environments, practically in all other cases, C-SLAM was able to make correct decisions to create and expand the underwater corridor network in a stable manner.

Acknowledgements

I thank E. Trabes for the valuable assistance in implementing the concepts and ideas elaborated in this work. I also thank J.L. Bustamante, J.L. Bonitatibus, G.D. Van Waarde, and L. Nuciari for their support in the experiments carried out at IADO. This work was financially supported by CONICET, CCT-Bahía Blanca, Argentine.

Author details

Mario Alberto Jordan
Argentinean Institute of Oceanography (IADO-CONICET), Electrical Engineering and Computer Department (DIEC), University of the South (UNS), Bahia Blanca, Argentine

*Address all correspondence to: mjordan@criba.edu.ar

References

[1] Lowry S, Snderhauf N, Newman P, Leonard JJ, Cox D, Corke P, et al. Visual place recognition: A survey. IEEE Transactions on Robotics. 2016;**32**(1):1-19

[2] Cadena C, Carlone L, Carrillo H, Latif Y, Scaramuzza D, Neira N, et al. Past, present, and future of simultaneous localization and mapping. Towards the robust-perception age. Open access arXiv.1606.05830v2. 2016

[3] Horgan H, Toal D. Computer vision applications in the navigation of unmanned underwater vehicles, Chap. 11. In: Inzartsev AV, editor. Robotics, Mobile Robotics, Underwater Vehicles. 2009. DOI: 10.5772/6703. ISBN 978-953-7619-49-7

[4] Sonka M, Hlavac V, Boyle R. Image Processing, Analysis, and Machine Vision. Stamford (Connecticut): Thomson; 2008. ISBN 0-495-08252-X

[5] Agarwal S, Lazarus SB, Savvaris A. Monocular vision based navigation and localization in indoor environments. In: Proceedings of IFAC Workshop on Embedded Guidance, Navigation and Control in Aerospace, Bangalore, India. Vol. 45(1). 2012

[6] Li J, Arbor A, Eustice RM, Johnson-Roberson M. Underwater robot visual place recognition in the presence of dramatic appearance change. In: Proceedings of the IEEE OCEANS 2015—MTS/IEEE Washington DC; IEEE. 2015. pp. 1-6

[7] Smith RN, Py F, Rajan K, Sukhatme GS. Adaptive path planning for tracking ocean fronts with an autonomous underwater vehicle. In: The 14th International Symposium on Experimental Robotics. 2014. pp. 761-775

[8] Oleari F, Kallasi F, Rizzini DL, Aleotti J, Caselli S. An underwater stereo vision system: From design to deployment and dataset acquisition. In: Proceedings of the OCEANS 2015; 18-21 May; Genova. IEEE; 2015, p. 1-6

[9] Lee D, Kim D, Lee S, Myung H, Choi HT. Experiments on localization of an AUV using graph-based SLAM. In: Proceedings of the 10th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI). IEEE; 2013. pp. 526-527

[10] Folkesson J, Lonard J. Autonomy through SLAM for an underwater robot. In: Robotics Research. Springer Tracts in Advance Robotics; 2011

[11] Hessner K, Reichert K, Rosenthal W. Mapping of sea bottom topography in shallow seas by using a nautical radar. In: 2nd International Symposium on Operationalization of Remote Sensing; 16-20 August; Enschede, The Netherlands. 1999

[12] Trabes E, Jordan MA. Node-based method for SLAM navigation in self-similar underwater environments: A case study. MDPI Robotics. 2017; **6**(4):29

[13] Chaves SM, Eustice RE. Efficient planning with the Bayes tree for active SLAM. In: International Conference on Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ; Daejeon, South Korea. IEEE; 2016

[14] Maurovi I, Seder M, Lenac K, Petrovi I. Path planning for active SLAM based on the D* algorithm with negative edge weights. In: IEEE Transactions on Systems, Man, and Cybernetics: Systems. Piscataway, New Jersey: IEEE; 2017. p. 99

[15] Mu B, Giamou M, Paull L, Agha-Mohammadi AA, Leonard J, How J. Information-based active SLAM via

topological feature graphs. In: IEEE 55th Conference on Decision and Control (CDC). IEEE; 2016

[16] Furgale P, Barfoot TD. Visual teach and repeat for long-range rover autonomy. Journal of Field Robotics. 2010;**27**(5):534-560

[17] Jordán MA, Bustamante JL. Adaptive control for guidance of underwater vehicles. In: Inzarsev A, editor. Underwater Vehicles. Vienna, Austria: In-Tech; 2009. pp. 251-278. ISBN: 978-953-7619-49-7

[18] Conor M, Py F, Rajan K, Ryan J. Adaptive control for autonomous underwater vehicles. In: Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008; Chicago, Illinois, USA; July 13-17. p. 2008

[19] Swirski Y, Schechner YY, Herzberg B, Negahdaripour S. Underwater stereo using natural flickering illumination. In: IEEE OCEANS'10, Sydney. 2010

[20] Trabes E, Jordan MA. On-line filtering of sunlight caustic waves in underwater scenes in motion. In: 7th International Scientific Conference on Physics and Control; 19-22 August 2015; Istanbul, Turkey. p. 2015

[21] Roosmalen PMBV, Lagendijk RL, Biemond J. Flicker reduction in old film sequences. In: Time-varying Image Processing and Moving Object Recognition 4. Amsterdam, Netherlands: Elsevier Science; 1997. pp. 9-17

[22] Gracias N, Negahdaripour N, Neumann L, Prados R, García R. A motion compensated filtering approach to remove sunlight flicker in shallow water images. In: Proc. MTS/IEEE Oceans. 2008

[23] Jordan M, Trabes E, Delrieux C. A robust approach for monocular visual odometry in underwater environments. In: Proceedings of the ASAI, 48 JAIIO. 2019

[24] Chellappa R, Qian G, Srinivasan S. Structure from motion: Sparse versus dense correspondence methods. In: Proc. Int. Conf. on Image Processing, 1999. ICIP 99. 1999

[25] Jordan M, Bustamante JL. On the autonomy in unmanned underwater vehicles with the most distant point of no return. In: VI Jornadas Argentinas de Robótica; 3 al 5 November; Buenos Aires, Argentina. 2010

[26] Chyba M, Haberkorn T, Singh SB, Smith RN, Choi SK. Increasing underwater vehicle autonomy by reducing energy consumption. Ocean Engineering. 2009;**36**:62-73

[27] Zeng Z, Lian L, Sammut K, He F, Tang Y, Lammas A. A survey on path planning for persistent autonomy of autonomous underwater vehicles. Ocean Engineering. 29 June 2019;**9** (2654):1-22

[28] Low KH, Dolan JM, Khosla P. Adaptive multi-robot wide-area exploration and mapping. In: Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems. Vol. 1. 2008. pp. 23-30

[29] Zhu D, Huang H, Yang SX. Dynamic task assignment and path planning of multi-AUV system based on an improved self-organizing map and velocity synthesis method in three-dimensional underwater workspace. IEEE Transactions on Cybernetics. 2013; **43**(2):504-514

[30] Alvarez A, Caiti A, Onken R. Evolutionary path planning for autonomous underwater vehicles in a variable ocean. IEEE Journal of Oceanic Engineering. 2004;**29**(2):418-429

[31] Guruji AK, Agarwal A, Parsediya DK. Time-efficient A*

algorithm for robot path planning. Procedia Technolgy. 2016;**23**: 144-149. DOI: 10.1016/j.protcy.2016.03.010

[32] Mur-Artal R, Montiel JMM, Tardós JD. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. Piscataway, New Jersey: IEEE; 2015. DOI: 10.1109/TRO.2015.2463671

[33] Engel J, Koltun V, Cremers D. Direct sparse odometry. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2018;**40**(3):611-625

[34] Engel J, Schöps J, Cremers D. LSD-SLAM: Large-scale direct monocular SLAM. In: Proc. of the European Conf. on Computer Vision (ECCV). Midtown Manhattan, Nueva York: Springer; 2014. pp. 834-849

[35] Newcombe RA, Lovegrove SJ, Davison AJ. DTAM: Dense tracking and mapping in real-time. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). Barcelona, Spain: IEEE; 2011. pp. 2320-2327

[36] Farnebäck G. Very high accuracy velocity estimation using orientation tensors, parametric motion, and simultaneous segmentation of the motion field. In: Proc. Eighth International Conference on Computer Vision. Vol. 1. Vancouver, Canada: IEEE Computer Society Press; 2001. p. 171-177

[37] Jordan M, Bustamante JL. Numerical stability analysis and control of umbilical-rov systems in taut-slack condition. Nonlinear Dynamics. 2006; **49**:163-191

[38] Rao D, Williams SB. Large-scale path planning for underwater gliders in ocean currents. In: Australasian Conference on Robotics and Automation (ACRA); December 2-4, 2009; Sydney, Australia. Vol. 110, Part A. Amsterdam, Netherlands: Elsevier; 2015. pp. 303-313

[39] Yan ZA, Li J, Wu Y, Zhang G. Real-time path planning algorithm for AUV in unknown underwater environment based on combining PSO and waypoint guidance. MDPI. 2018;**19**(1). Article number: 20

[40] Hernández JD, Vidal E, Vallicrosa G, Galceran E, Carreras M. Online path planning for autonomous underwater vehicles in unknown environments. In: Proc. IEEE Int. Conf. on Robotics and Automation. 2015. DOI: 10.1109/ICRA.2015.7139336

[41] Liang S, Zhi-Ming Q, Heng L. A survey on route planning methods of AUV considering influence of ocean currents. In: Conference: 2018 IEEE 4th Int. Conf. on Control Science and Systems Engineering (ICCSSE); 21-23 Aug. 2018; Wuhan, China