On-line 3D Path Planning for Close-proximity Surveying with AUVs

Juan D. Hernández* Guillem Vallicrosa* Eduard Vidal* Èric Pairet* Marc Carreras* Pere Ridao*

* *Underwater Vision and Robotics Research Centre Universitat de Girona, Catalonia, Spain (e-mail: juandhv@eia.udg.edu).*

Abstract:
We present an approach for planning collision-free paths on-line for an underwater multi-robot system, which is composed by a leading Autonomous Underwater Vehicle (AUV) endowed with a multibeam sonar and high processing capabilities and a second AUV. While the leading AUV follows a safe, pre-planned survey path, the second vehicle, herein referred to as Camera Vehicle (CV), must survey the bottom in close proximity while following the leader, complementing its survey capabilities. Due to their proximity to the bottom, the CV is exposed to a collision threat. We address this problem by incrementally building a 3D map of the environment onboard the leading vehicle by means of its multibeam sonar. Using this map, we plan on-line 3D paths that are transferred to the CV for close and safe surveying of the bottom. These paths are planned using the Transition-based RRT (T-RRT) algorithm, which is an RRT-variant that considers a cost function defined over the vehicle's configuration space, or costmap for short. By defining a costmap in terms of distance to the bottom and path distance, we are able to keep the paths at a desired offset distance from the bottom for constant-resolution surveying. We have integrated our path planning system with the software architecture of the SPARUS-II and GIRONA500 AUVs. We demonstrate the feasibility of our approach in simulation. The multi-robot system presented is based on the context of the MORPH FP7 EU project.

Keywords: Path Planning, Mapping, Surveys, On-line processing, Communication channels.

1. INTRODUCTION

Most underwater surveys are produced by AUVs that navigate at a safe and constant altitude from the seafloor. They are equipped with different sensors such as multibeam and imaging sonars that capture data used to build bathymetric maps (elevation maps of the seabed). These bathymetric maps are used for safe surface and sub-surface navigation, enabling a variety of applications, such as underwater archeology (Bingham et al., 2010) and underwater geology (Escartín et al., 2008).

However, there are scenarios where collecting data at a constant and conservative altitude may not be convenient, especially when imaging in close proximity is required (*i.e.,* visibility plays a critical role). In such situations, AUVs are required to stay closer to the seafloor. A standard approach is to keep reference values of altitude provided by profiling sonars (Caccia et al., 1999; Creuze et al., 2001) or altimeters (Caccia et al., 2003; Melo and Matos, 2012). However, these approaches lack a complete representation of the area, limiting their responsiveness to rugged, high relief terrain. Thus, moving through complex topography is still a challenge in the underwater domain. Extending technological bounds in this respect is key for widening the applicability of underwater robotics.

This paper presents the application of a sampling-based algorithm for planning collision-free paths on-line for the presented multi-vehicle system. Operating on a 3D map constructed incrementally onboard by the leading vehicle using its multibeam sensor, we plan 3D paths using the T-RRT algorithm (Jaillet et al., 2010), which considers a cost function defined over the robot's configuration space, or costmap for short. By defining a costmap in terms of distance to the bottom and path length, we are able to keep the vehicles at the desired, close offset distance from the bottom. The resulting paths are transferred via the communication links to the follower vehicle for safe surveying in close proximity. Our approach goes beyond the aforementioned reactive methods by using a complete representation of the environment, which allows a trade-off between keeping a desired altitude and path length. We tested our method using both a synthetic and a real-world dataset in simulation using a model of the GIRONA500 and SPARUS-II AUVs.

The remainder of this paper is organized as follows. Section 2 introduces previous research related to on-line path planning for AUVs and current state-of-the-art of sampling-based methods. Section 3 describes the complete setting of the problem, its requirements and specifications. Section 4 discusses our proposed method. In Section 5, a

⋆ This work was supported by the MORPH EU FP7-Project under the Grant agreement FP7-ICT-2011-7-288704 and partially supported by the Colombian Government through its Predoctoral Grant Program offered by Colciencias.

real-world dataset used to validate the capacity for path planning on-line is described, and the results are presented. Virtual datasets were used to evaluate the multi-vehicle performance. Finally, concluding remarks and directions for further research are given in Section 6.

2. RELATED WORK

Path planning is the problem of finding valid (collision-free) paths from a start to a goal configuration in the configuration space (C-Space). Classical approaches include grid-based methods such as A* or D* (Stentz, 1994). These methods have been applied to path planning problems in 3-dimensional workspaces (Carsten et al., 2006). However, since they require the discretization of the state space, these methods suffer from scalability problems, thus restricting them to problems in low-dimensional C-Spaces.

2.1 Sampling-based Algorithms

Sampling-based algorithms have been developed to overcome these restrictions. These methods exploit the fact that, while explicitly building the space free of obstacles (the free space) is expensive, checking a given state for collisions can be done quickly. Thus, sampling based methods build a discrete representation of the free space, typically in the form of a graph or tree, by checking its nodes and edges for collisions. This category of algorithms includes Probabilistic Roadmap (PRM) (Kavraki et al., 1996), Expansive-Spaces Tree (EST) (Hsu, 2000), and Rapidly-exploring Random Tree (RRT) (LaValle and Kuffner, 1999). Of particular interest are the many extensions developed on the basis of this concept. One of them, termed RRT-Connect, proposes to build two trees rooted at the start and goal states, respectively, to speed up convergence (Kuffner and LaValle, 2000). Another extension of the original algorithm is the asymptotic Optimal RRT (RRT*) (Karaman and Frazzoli, 2011), which is guaranteed to asymptotically converge to an optimal solution. Finally, T-RRT takes into consideration a costmap defined over the configuration space and finds low-cost paths that follow valleys and saddle points on such costmap (Jaillet et al., 2010). The T-RRT is of special interest in this research. Its utility and applicability for our solution approach is explained in Section 4.

2.2 3D Path Planning

In contrast to some terrestrial applications where motion is restricted to a plane, in domains like aerial or underwater robotics, path planning in three dimensions (3D) is required. Thus, several approaches have been proposed in this regard.

Carsten et al. (2006) presented a method that extends a 2D grid-based technique to solve 3-dimensional path planning problems, in which considers the costs for different types of movements. In the field of underwater robotics, most contributions are limited to simulations results. Guo and Gao (2009) presented path planning simulations for a multi-robot system, where obstacles, placed randomly, are represented as spheres and robots as particles. Zhu et al. (2012) proposed the usage of neural networks to control the task assignment and the path planning of a multi-AUV system, in which workload and energy balance must be guaranteed.

Other works that must be highlighted include real-world results. Galceran et al. (2014) proposed a coverage path planning for inspection in close proximity of 3D natural structures on the ocean floor charted as 2.5D bathymetric maps. However, this approach is not applicable in our context, since it has not been developed to solve start-to-goal queries, but to perform exhaustive coverage.

3. PROBLEM DESCRIPTION

To what our path planning problem concerns, an AUV must attempt to follow the leading vehicle trajectory but in close-proximity to the seafloor. To accomplish this, while the leading vehicle performs its trajectory, it also builds a navigation map and plans a collision-free path for the low-altitude follower vehicle. Each period of time T, the leading vehicle processes its position and defines a desired start/goal configuration. A previous captured position defines the X-Y coordinates of the start configuration, while the Z coordinate is calculated as the desired altitude from the seafloor. Likewise, the goal configuration is specified but using the latest captured position. The complete task can be considered as a repetitive and incremental planning of consecutive segments.

For each path planning segment, a bounding box is defined to limit acceptable paths. X-Y limits are specified according to the start and goal desired configurations. Limits in depth, on the other hand, are defined depending on leading navigation altitude. A final relevant aspect is given by the periodicity of the path planning process, which implies that an updated version of the map is required to solve each planning problem.

Summing up, the main concern of this work is to define, select and implement the necessary algorithms or methods to plan the paths for an AUV vehicle that navigates at a desired altitude from the seafloor, in such way that follows the leading vehicle movements. It is assumed that the leading vehicle can reliably communicate the waypoints of the path solution to the follower vehicle. Details of underwater communication are given in Section 5.1.

4. 3D PATH PLANNING FOR CLOSE-PROXIMITY SEA BOTTOM SURVEYING

This section details our proposed approach for planning collision-free paths for a follower AUV surveying the bottom in close proximity, which is based in three key modules. First, an on-line 3D mapping module constructs a convenient representation of the environment for path planning (4.1). Second, we define a costmap to keep the vehicle at a desired altitude from the bottom (4.2). Third, the path planning algorithm itself, based on T-RRT to account for the costmap (4.3).

4.1 3-dimensional Workspace Representation

The leading vehicle captures seafloor relief data through a multibeam sonar. That information is used to create a 3D model employed to plan the paths. There are different alternatives to represent 3D workspaces such as point clouds, elevation maps, and multi-level surface maps. Such approaches have features that do not fulfill some relevant aspects of our research. For instance, point clouds store large amounts of information, making it a memory-inefficient option for an on-line application. On the other hand, together with point clouds, elevation and multi-level maps do not permit to differentiate between obstacle-free areas and unexplored areas, which can be critical when performing missions in environments where no previous information is available, as is our case.

Considering these facts, we decided to use Octomap (Hornung et al., 2013), an octree-based framework for modeling volumetric information. Octomaps have three main characteristics that contribute directly to our online mapping and path planning application. The first of them is the probabilistic state representation that considers previous information when updating states, which calculates new state values according to probabilistic functions, thus not only updating map information, but also protecting it from noisy measurements. The second is the capacity of representing unexplored areas, which is the particular interest for exploration tasks. Finally, Octomap offers an efficient method for modeling volumetric information, since its computational costs in terms of time of access and memory consumption are less that other alternatives, for instance, it does not have to be initialized with a predefined size, but can be enlarged or extended as demanded.

4.2 Costmap for Surveying at a Desired Altitude

To ensure that the resulting path for the low-altitude AUV attempts to meet the altitude constraint, a costmap was defined over the configuration space. The respective cost, $0 \leq Cost \leq 100$, where 0 and 100 are the minimum and maximum cost respectively, is presented in Eq. 1. The Cost depends on the depth (d) and includes adjustable parameters. One of those parameters is the expected depth d_e, which is calculated as the difference of seafloor depth and the desired altitude. Another is an admissible range of altitude Δd_a, which permits to define an interval of altitudes in which the cost has its minimal cost (clearly observed in Fig. 1).

$$Cost(d) = \begin{cases} \left(1 - \dfrac{d}{d_e}\right) 100, & d < d_e - \dfrac{\Delta d_a}{2} \\ 0, & d_e - \dfrac{\Delta d_a}{2} \leq d \leq d_e + \dfrac{\Delta d_a}{2} \\ \left(\dfrac{d}{d_e} - 1\right) 100, & d > d_e + \dfrac{\Delta d_a}{2} \end{cases} \quad (1)$$

Nonetheless, it is important to remark that defining high-cost values to certain zones does not restrict or limit them as possible paths, but it will attempt to avoid them as much as possible. In other words, this approach is different than defining restricted areas, where the vehicle would not be allowed to move through. An example of these situations is when the only possible path coincides with the restricted zone (highest cost), in which our approach will permit that path as a valid solution.

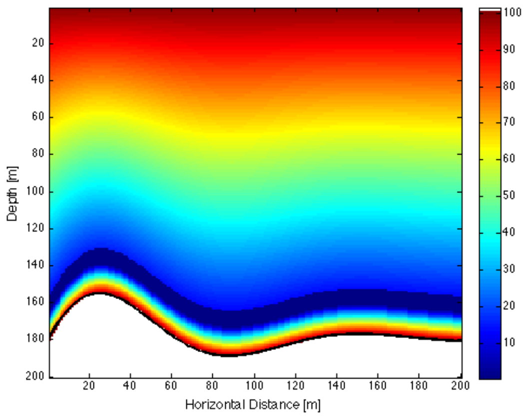


Fig. 1. Costmap projected in vehicle's X(surge)-Z(heave) plane.

4.3 Path Planning

In what path planning regards, detailed inspection tasks in underwater environments, as proposed in this research, require planning motion/paths in 3-dimensional spaces. Furthermore, when considering some requirements such as a complete 6-DOF model, on-line and incremental path searching, or even the extension of the workspace, the computational cost of representing and evaluating completely the C-space, as required by several combinatorial methods, can be prohibitive. Sampling-based algorithms, on the other hand, have as a basic principle a probabilistic exploration of the C-space.

But deepening on those specifications and constraints, even an exhaustive and probabilistic exploration of the C-space is not only unnecessary, but also time-consuming. In fact, in our context, the problem can be summed up as a repetitive process of path planning over a new map, even when it is actually an extension of a previous one. For this reason, our proposed approach must use a single-query planner (*e.g.,* RPP, EST, RRT) instead of a multi-query one (Choset et al., 2005).

The current reference and state-of-the-art of sampling-based single-query planners is the widely known RRT*, as mentioned in Section 2. In its early development phase, the RRT was geared towards solving kinodynamic motion planning problems, *i.e.,* to move from path planning to motion planning where time and dynamics are included. Finally, to compute a path that meets altitude requirements, the T-RRT algorithm was used and results in a convenient method for our proposed approach.

T-RRT calculates a low-cost path that follows valleys and saddle points of the configuration-space costmap. To achieve this, it verifies the quality of the path by a Minimal Work (mechanical work) criterion. The key principle is that positive variations of the cost function can be seen as forces acting against motion, and thus, producing mechanical work. Contrasted to a classical RRT implementation, an additional transition validation is performed, which accepts or rejects new potential states before including them into the solution tree, as can be appreciated in Algorithm 1. In our approach, T-RRT uses the costmap explained in Section 4.2, where moving away from desired altitude will result in an increment of mechanical work.

Algorithm 1: T-RRT (Jaillet et al., 2010)

Input:
q_{init} and q_{goal} states.
Input:
Cost function, $c(q) = Cost(d(q))$
Output:
The tree (T). Contains the collision-free path.
begin
  T ←InitTree(q_{init})
  while not *StopCondition*(T, *goal*) **do**
    q_{rand} ←SampleConf(C-Space)
    q_{near} ←NearestNeighbor(q_{rand}, T)
    q_{new} ←Extend(T, q_{rand}, q_{near})
    if $q_{new} \neq NULL$
    and *TransitionTest*($c(q_{near})$,$c(q_{new})$,$d_{near-new}$)
    and *MinExpandControl*(T, q_{near}, q_{rand}) **then**
      AddNewNode(T, q_{new})
      AddNewEdge(T, q_{near}, q_{new})
end

5. RESULTS

In our simulation tests, the leading vehicle role is performed by the SPARUS-II AUV (Carreras et al., 2013) (see Fig. 2), a torpedo-shaped vehicle with hovering capabilities, rated for depths up to $200m$. The robot has three thrusters (two horizontal and one vertical) and can be actuated in surge, heave and yaw degrees of freedom (DOF). As a follower vehicle, we used the GIRONA500 AUV (Ribas et al., 2012) (see Fig. 3), a reconfigurable vehicle with hovering capabilities, designed for depths up to $500m$. The robot is configured to be actuated in surge, sway, heave and yaw, *i.e.,* a 4-DOF system. Both vehicles are equipped with a navigation sensor suite including a pressure sensor, a Doppler Velocity log (DVL), an Inertial Measurement Unit (IMU) and a Global Positioning System (GPS) to receive position fixes while at surface.

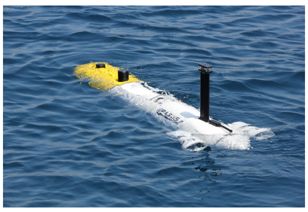


Fig. 2. SPARUS-II, a torpedo-shaped AUV.

We tested our path planning approach in a synthetic environment, which has been created with a 3D graphics tool and represents a challenging seabed region for an AUV to navigate through, presenting several steep protrusions. Furthermore, we have used a real-world dataset to evaluate the capacity for planning 3D collision-free paths in a real-world scenario. In such dataset, the AUV covered a 135-by-117 m area by following a "mowing the lawn" trajectory at a constant depth of 21 m with a total length of 1173 m. The target area is located at "l'Amarrador" diving site, approximately 1 Km off the harbor of Sant

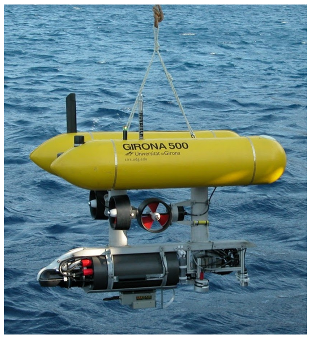


Fig. 3. GIRONA500 AUV

Feliu de Guixols in Catalonia, Spain. The site features a seamount rising from 40 m to 27 m depth, arising a collision threat for an AUV surveying the bottom in close proximity. The AUV was equipped with a Delta T multibeam bathymetry sonar from Imagenex in this mission, providing range measurements to the seabed. Thanks to the mission replaying tools provided by the ROS framework, we are able to incrementally construct a 3D map of the environment using the bathymetric data provided in this dataset.

This section explains in some detail the communication protocol between the vehicles and the results obtained in both cases, synthetic and real-world environments.

5.1 Underwater communications

Communication between vehicles is handled over simulated acoustic modems, provided by Evologics, which behave exactly as the real ones installed on the AUVs. The extra step differentiating them from a real one is that position of the modem in space must be provided.

The protocol used for communication is kept as simple as possible to leave space for other communications such as safety commands. The path planning vehicle must know the position of the follower and must send the latter the waypoints to follow. A simple message, which contains the time, the vehicle id (a letter) and the (x, y, z, ψ) position, is sent from the follower. The waypoints are also sent as time, id and (x, y, z, ψ) position.

Vehicles send messages using a Time division multiple access (TDMA) method. This allows different transmitters to work in the same channel by assigning them different time slots. Keeping this communication protocol allows for further expanding such as including more follower vehicles.

5.2 Simulation

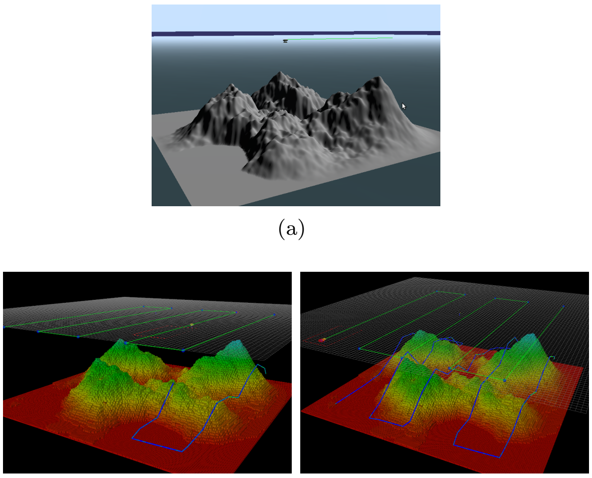
SPARUS-II and GIRONA500 [1] are controlled through the Component Oriented Layer-based Architecture for Autonomy (COLA2) (Palomeras et al., 2012), a control architecture that is in constant development and is completely integrated into the Robot Operating System (ROS). Besides operating in real robots, COLA2 can interact with the UnderWater Simulator (UWSim) (Prats et al., 2012), where synthetic scenarios can be included and can provide information to virtual sensors to simulate completely

[1] http://cirs.udg.edu/

the robot's behavior. This information is used as a first validation stage before testing on real robots.

On the other hand, as occurred with Octomaps, the Open Motion Planning Library (OMPL) offers a useful framework that can be extended and adapted to specific path planning problems (Şucan et al., 2012). Most relevant sampling-based methods, including the T-RRT, are available in OMPL. It was used for testing our approach, and the necessary adjustments were added, such as receiving Octomaps as C-space, including its respective collision-check state verification. Validation of our approach was done in 3 different stages, which were performed as follows.

Offline Mapping and On-line Path Planning. The software architecture was adapted for performing path planning tasks incrementally and on-line. For this, a synthetic scenario was created and the workspace was transformed into an equivalent Octomap (see Fig. 4a). On this set of tests, a virtual model of SPARUS-II performs a survey task at a safe altitude (4 meters deep). The representation of the workspace, the Octomap, is assumed to be static and pre-calculated. SPARUS-II extends a collision-free path at a close-to-bottom altitude (4 meters) each T seconds, which is defined as the path planning period of time. Results can be observed in Fig. 4.



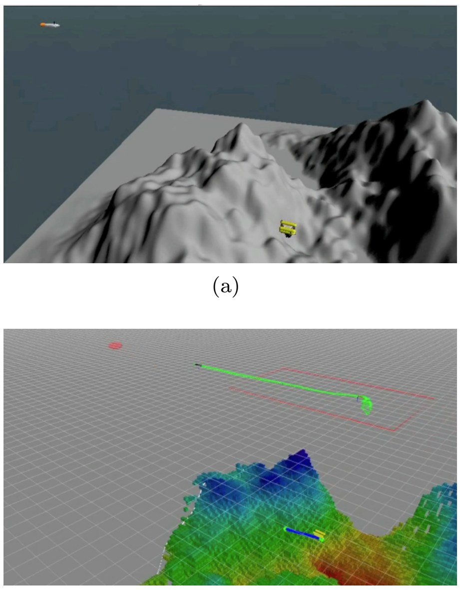
(a)

(b) (c)

Fig. 4. Offline mapping and on-line path planning. Leading vehicle performs a survey mission at a safe altitude (green) while computes a collision-free path close to the seabed (blue). Map is assumed to known (pre-calculated).

On-line Mapping and Path Planning. To fulfill the requirements and specifications presented throughout this paper, workspace must be assumed to be unexplored. For that reason, the next expected step was to include a module in charged to build incrementally and on-line a map in which the planning module has to find the collision-free path. Once again, each T seconds, SPARUS-II extends the resulting path for the GIRONA500. Figure 5 shows both vehicles the leader and follower conducting a mission. Incremental mapping and path planning can be appreciated in Fig. 6.

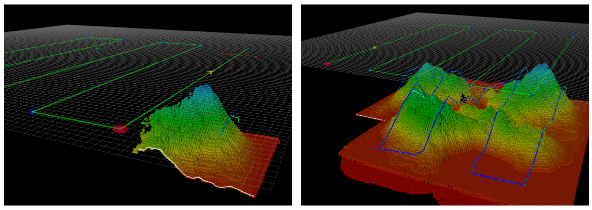
On-line Mapping and Path Planning on a Real-world Dataset. The final test used to validate our method was to perform mapping and path planning on-line using a real-world dataset, as explained above. Figure 7a presents the results of the complete mission. This dataset is particularly interesting, since it performs the survey in an area where a 10-meter high seamount (clearly observed) is located. Figure 7b details in that area of interest and shows how a collision-free path has been calculated.



(a)

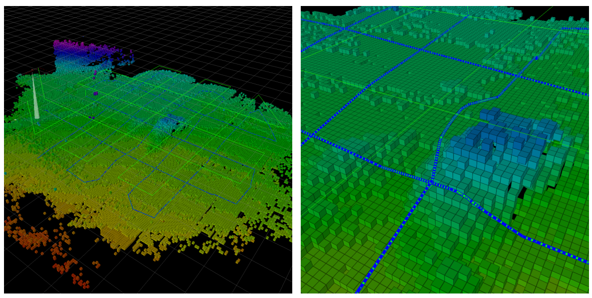
(b)

Fig. 5. On-line mapping and path planning. Leading vehicle performs a survey mission at a safe altitude (green) while builds a maps and computes a collision-free path close to the seabed (blue) for the follower vehicle.



(a) (b)

Fig. 6. Complete survey. Incremental on-line mapping and path planning.



(a) (b)

Fig. 7. On-line mapping and path planning on a real-world dataset. In (b) can be observed how the resulting path avoids a seamount.

6. CONCLUSIONS AND FURTHER WORK

We presented a new approach for planning collision-free paths in on-line for an AUV. The resulting path meets a specified altitude requirement, which permits close proximity exploration of the seafloor. The proposed method finds the solution by using a costmap defined through the vehicle's configuration space. The costmap is expressed in terms of distance to the bottom and path distance, which permits to keep the paths at a desired offset distance from the bottom. Our approach has been tested in simulation using both synthetic and real-world datasets, which has demonstrated its correctness and effectiveness.

We will focus our immediate efforts on considering and including the kinodynamic constraints when calculating the path. Furthermore, we expect to perform tests in a complete setting of the problem, including the second vehicle performing the obtained path on-line.

REFERENCES

Bingham, B., Foley, B., Singh, H., Camilli, R., Delaporta, K., Eustice, R., Mallios, A., Mindell, D., Roman, C., and Sakellariou, D. (2010). Robotic tools for deep water archaeology: Surveying an ancient shipwreck with an autonomous underwater vehicle. *Journal of Field Robotics*, 27(6), 702–717.

Caccia, M., Bono, R., Bruzzone, G., and Veruggio, G. (2003). Bottom-following for remotely operated vehicles. *Control Engineering Practice*, 11(4), 461–470.

Caccia, M., Bruzzone, G., and Veruggio, G. (1999). Active sonar-based bottom-following for unmanned underwater vehicles. *Control Engineering Practice*, 7(4), 459–468.

Carreras, M., Candela, C., Ribas, D., Mallios, A., Magí, L., Vidal, E., Palomeras, N., and Ridao, P. (2013). Sparus II, design of a lightweight hovering AUV. In *5th International Workshop on Marine Technology (MARTECH)*. Girona.

Carsten, J., Ferguson, D., and Stentz, A. (2006). 3D Field D*: Improved Path Planning and Replanning in Three Dimensions. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 3381–3386.

Choset, H., Lynch, K.M., Hutchinson, S., Kantor, G.A., Burgard, W., Kavraki, L.E., and Thrun, S. (2005). *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press, Boston.

Creuze, V., Jouvencel, B., and Baccou, P. (2001). Seabed following for small autonomous underwater vehicles. In *MTS/IEEE OCEANS*, 369–374.

Escartín, J., García, R., Delaunoy, O., Ferrer, J., Gracias, N., Elibol, A., Cufi, X., Neumann, L., Fornari, D.J., Humphris, S.E., and Renard, J. (2008). Globally aligned photomosaic of the Lucky Strike hydrothermal vent field (Mid-Atlantic Ridge, 37o18.5'N): Release of georeferenced data, mosaic construction, and viewing software. *Geochemistry, Geophysics, Geosystems*, 9(12).

Galceran, E., Campos, R., Palomeras, N., Carreras, M., and Ridao, P. (2014). Coverage path planning with realtime replanning for inspection of 3D underwater structures. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, 6586–6591.

Guo, S. and Gao, B. (2009). Path-planning optimization of underwater microrobots in 3-D space by PSO Approach. In *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 1655–1620. IEEE Press.

Hornung, A., Wurm, K.M., Bennewitz, M., Stachniss, C., and Burgard, W. (2013). OctoMap: an efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 34(3), 189–206.

Hsu, D. (2000). *Randomized single-query motion planning in expansive spaces*. Ph.D. thesis, Stanford University.

Jaillet, L., Cortés, J., and Simeon, T. (2010). Sampling-Based Path Planning on Configuration-Space Costmaps. *IEEE Transactions on Robotics*, 26(4), 635–646.

Karaman, S. and Frazzoli, E. (2011). Sampling-based Algorithms for Optimal Path Planning. *The International Journal of Robotics Research*, 30(7), 846–894.

Kavraki, L.E., Švestka, P., Latombe, J.C., and Overmars, M.H. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4), 566–580.

Kuffner, J.J. and LaValle, S.M. (2000). RRT-connect: An efficient approach to single-query path planning. In *IEEE International Conference on Robotics and Automation (ICRA)*, 995–1001.

LaValle, S.M. and Kuffner, J.J. (1999). Randomized kinodynamic planning. In *IEEE International Conference on Robotics and Automation (ICRA)*, 473–479.

Melo, J. and Matos, A. (2012). Bottom estimation and following with the MARES AUV. *Oceans, 2012*, 1–8.

Palomeras, N., El-Fakdi, A., Carreras, M., and Ridao, P. (2012). COLA2: A Control Architecture for AUVs. *IEEE Journal of Oceanic Engineering*, 37(4), 695–716.

Prats, M., Perez, J., Fernandez, J.J., and Sanz, P.J. (2012). An open source tool for simulation and supervision of underwater intervention missions. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2577–2582.

Ribas, D., Palomeras, N., Ridao, P., Carreras, M., and Mallios, A. (2012). Girona 500 AUV: From Survey to Intervention. *IEEE/ASME Transactions on Mechatronics*, 17(1), 46–53.

Stentz, A. (1994). Optimal and efficient path planning for partially-known environments. In *IEEE International Conference on Robotics and Automation (ICRA)*, 3310–3317.

Şucan, I.A., Moll, M., and Kavraki, L.E. (2012). The Open Motion Planning Library. *IEEE Robotics & Automation Magazine*, 19(4), 72–82.

Zhu, D., Huang, H., and Yang, S.X. (2012). 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*, 43(2), 504–514.