

# Coverage Path Planning with Real-time Replanning and Surface Reconstruction for Inspection of Three-dimensional Underwater Structures using Autonomous Underwater Vehicles



## Enric Galceran

Perceptual Robotics Laboratory (PeRL), Department of Naval Architecture and Marine Engineering, University of Michigan, 2114-D Building 520, North Campus Research Complex (NCRC), 1600 Huron Parkway, Ann Arbor, Michigan 48105

e-mail: [egalcera@umich.edu](mailto:egalcera@umich.edu)

## Ricard Campos

Underwater Vision Laboratory, Computer Vision and Robotics Institute, University of Girona, Edifici P-IV, Campus de Montilivi 17071, Girona, Spain

e-mail: [rcampos@eia.udg.edu](mailto:rcampos@eia.udg.edu)

## Narcís Palomeras, David Ribas, Marc Carreras, and Pere Ridao

Underwater Robotics Research Center, Computer Vision and Robotics Institute, University of Girona, Pic de Peguera, 13 (La Creueta) 17003, Girona, Spain

e-mail: [npalomer@eia.udg.edu](mailto:npalomer@eia.udg.edu), [dribas@eia.udg.edu](mailto:dribas@eia.udg.edu), [marc.carreras@udg.edu](mailto:marc.carreras@udg.edu), [pere@eia.udg.edu](mailto:pere@eia.udg.edu)

Received 16 June 2014; accepted 24 September 2014

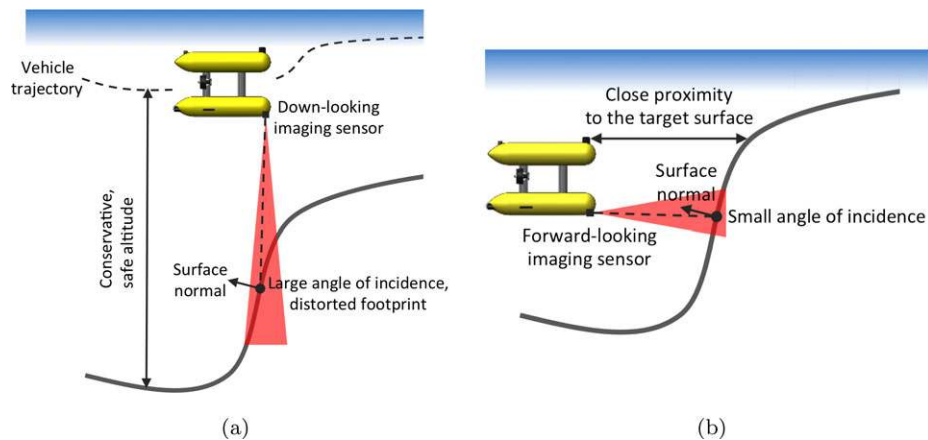
We present a novel method for planning coverage paths for inspecting complex structures on the ocean floor using an autonomous underwater vehicle (AUV). Our method initially uses a 2.5-dimensional (2.5D) prior bathymetric map to plan a nominal coverage path that allows the AUV to pass its sensors over all points on the target area. The nominal path uses a standard mowing-the-lawn pattern in effectively planar regions, while in regions with substantial 3D relief it follows horizontal contours of the terrain at a given offset distance. We then go beyond previous approaches in the literature by considering the vehicle's state uncertainty rather than relying on the unrealistic assumption of an idealized path execution. Toward that end, we present a replanning algorithm based on a stochastic trajectory optimization that reshapes the nominal path to cope with the actual target structure perceived *in situ*. The replanning algorithm runs onboard the AUV in real time during the inspection mission, adapting the path according to the measurements provided by the vehicle's range-sensing sonars. Furthermore, we propose a pipeline of state-of-the-art surface reconstruction techniques we apply to the data acquired by the AUV to obtain 3D models of the inspected structures that show the benefits of our planning method for 3D mapping. We demonstrate the efficacy of our method in experiments at sea using the GIRONA 500 AUV, where we cover part of a breakwater structure in a harbor and an underwater boulder rising from 40 m up to 27 m depth. © 2014 Wiley Periodicals, Inc.

## 1. INTRODUCTION

Thanks to technology breakthroughs in the past two decades, autonomous underwater vehicles (AUVs) have become a standard tool for surveying the ocean floor in a broad variety of applications, including marine geology (Escartin et al., 2008), underwater archeology (Bingham et al., 2010; Gracias et al., 2013), and fine-scale mapping of structures on the ocean floor (Johnson-Roberson et al., 2010; Tivey et al., 1997; Williams et al., 2009; Yoerger, Kelley, & Delaney, 2000), to name just a few. AUVs provide high-resolution maps thanks to near-bottom surveys, and they require little human supervision compared to their ship- or remotely

operated vehicle (ROV) -assisted counterparts, and hence at a lower cost.

Nonetheless, fully autonomous AUV surveys still have important limitations, particularly when targeting high-relief areas on the sea floor. At present, in most AUV survey missions, the vehicle passes a down-looking sensor over the sea floor by following a preplanned lawnmower-like survey path while keeping a safe altitude from the bottom. This is a valid approach for sea floor areas that are effectively planar at the survey scale. However, flying at a conservative altitude imposes serious limitations for a number of emerging applications demanding fine-scale sea



**Figure 1.** Askew sensor footprint due to a large angle of incidence (a) in contrast with a more desirable viewpoint along the surface normal (b).

floor surveys of rugged terrain. These applications require surveying the ocean floor in close proximity for acquisition of high-resolution imagery or even object grasping. Examples include monitoring of cold water coral reefs, oil and gas pipeline inspection, harbor and dam protection, and object recovery. Therefore, techniques that allow AUVs to maneuver in close proximity to the seabed without compromising vehicle safety are desired.

Additionally, following the elevation profile of the seabed does not provide satisfactory results when surveying rugged, high-relief terrain such as coral reefs or ship wrecks. These sites present very steep slopes that cannot be imaged with acceptable quality from an overhead point of view. It is rather desired that the AUV places its sensor so that a viewing angle close to the surface normal of the target structure is achieved, as illustrated in Figure 1. Therefore, in order to meet these requirements, flying at a conservative distance from the sea floor is no longer an option. The AUV must rather navigate amidst potential bulges sticking out of the bottom. Obviously, this increases the threat of collision.

The general task of passing a robot's sensor over all points in a target area is known as coverage path planning, and there is substantial research addressing this problem in the literature (see Section 2 below). However, typical coverage path-planning approaches assume perfect knowledge of the environment and no uncertainty in sensing or control, which are unrealistic assumptions in the vast majority of scenarios and especially in underwater environments, even when using techniques such as terrain-relative navigation (TRN) (Meduna, Rock, & McEwen, 2008) or simultaneous localization and mapping (SLAM) (Barkby et al., 2012) for enhanced localization. This limits real-world application of those approaches to very constrained, controlled environments.

In this paper, we present a new coverage path-planning method for inspection of 3D structures on the ocean floor

using AUVs that does not rely on these assumptions. Strictly speaking, our algorithms operate on a 2D manifold embedded in  $\mathbb{R}^3$ , i.e., the sea floor, which can be represented as a 2.5D model (an elevation map). However, covering such a geometrical structure requires the vehicle to move in a 3D workspace. In addition, the coverage paths we plan enable a full 3D perception of the target structures, as shown in our experiments. Bearing this in mind, we use the term “3D coverage” throughout this article.

Our method is a two-step process. First, a nominal coverage path seeking to provide sensor viewpoints close to the target surface's normal is planned using an *a priori* bathymetric map of the target structure (a bathymetric map is an elevation map of the ocean floor). To plan the path, our algorithm classifies the mapped area into effectively planar areas, which can be covered using a standard mowing-the-lawn pattern, and high-slope regions. For the latter, the algorithm uses a coverage pattern that follows the structure's horizontal contours at uniformly spaced depths while maintaining a fixed offset distance, accumulating data contour-by-contour along the vertical spatial dimension of the workspace. As a result, the path enables acquisition of a clear and continuous data product, simplifying the tasks of postprocessing and analysis for both humans and automated procedures.

Once a nominal path has been planned, we go beyond previous approaches by not relying on the unrealistic assumption of an idealized execution of the planned path. To handle the uncertainty in the vehicle's state and environment, we put forth a replanning algorithm based on stochastic trajectory optimization to adapt the initially planned coverage path in real time using range sensor measurements. The resulting path provides successful coverage under bounded position error. Additionally, we present a pipeline of state-of-the-art 3D surface reconstruction techniques, which we apply to the data collected in the

coverage tasks, obtaining full 3D surface models and optical maps. The traditional downward-looking configuration of multibeam sonars mounted on AUVs has typically promoted using 2.5D representations of the mapped surface of the sea floor. In contrast, our real-time coverage path-replanning algorithm provides viewpoints enabling recovery of point clouds describing full 3D surfaces, requiring more complex surface reconstruction methods. Our 3D mapping results show how the paths planned with our method are useful in mapping complex 3D structures, not amenable to traditional surveying of marine environments. The data products obtained demonstrate the high quality of the 3D perception enabled by our coverage path-planning method.

We show the feasibility of our approach in experiments at sea with the GIRONA 500, a reconfigurable AUV equipped with range and imaging sensors. Our experiments comprise coverage of a large concrete block on a breakwater structure in a harbor and coverage of a diving site featuring an underwater boulder rising from 40 to 27 m depth. Results show that our method successfully achieves coverage of the target structures, adapting the planned paths in agreement with real-time perception onsite and enabling full 3D mapping of the target structures.

The remainder of this paper is organized as follows. Section 2 briefly reviews related work on coverage path planning, while Section 3 introduces planning and reconstruction algorithms from the literature upon which this paper builds. The nominal coverage path-planning phase of our method is described in Section 4, and the replanning algorithm is presented in Section 5. Section 6 introduces the GIRONA 500 AUV and the sensor configuration we use, the scenarios in which we conduct the experiments to evaluate our method, and it describes the choice of planning parameters used in the coverage tasks according to experimental data. Section 7 reports on the results obtained in sea trials, including 3D reconstructions and a comparison with a traditional lawnmower-type survey. Finally, concluding remarks and directions for further research are given in Section 8.

## 2. RELATED WORK

As mentioned earlier, the task of the determining a path that passes over all points of a surface of interest while avoiding obstacles is known as coverage path planning. A large body of research has investigated coverage path planning [see Galceran & Carreras (2013b) for a recent survey] in 2D environments (Acar & Choset, 2002a; Acar et al., 2002, 2006; Butler, Rizzi, & Hollis, 1999; Choset, 2000; Gabrieli & Rimon, 2002; Luo & Yang, 2008; Wong, 2006) and applications have been reported in domains such as aerial robotics (Ahmadzadeh et al., 2006; Barrientos et al., 2011; Maza & Ollero, 2007; Xu, Virie, & Rekleitis, 2011), agricultural robotics (Oksanen & Visala, 2009), mine countermeasures (MCM) operations (Acar et al., 2003; Stack & Smith, 2003),

and marine robotics (Galceran & Carreras, 2012; Paull et al., 2012). Furthermore, 2D coverage algorithms for planning optimal paths (Huang, 2001; Jimenez et al., 2007; Manna-diar & Rekleitis, 2010) and for minimizing the robot's state uncertainty along the planned paths (Acar & Choset, 2002b; Bosse, Nourani-Vatani, & Roberts, 2007; Bretl & Hutchinson, 2013; Das, Becker, & Bretl, 2011; Tully, Kantor, & Choset, 2010) have been presented. In this latter category of 2D coverage under uncertainty, some approaches have particularly targeted underwater environments (Galceran et al., 2013; Hollinger, Mitra, & Sukhatme, 2012b; Paull, Seto, & Li, 2014), which, respectively, seek to maximize to quality of bathymetric data gathered during a survey, guarantee area coverage even in the case of severe AUV position estimate drift, and seek to maximize the quality of 3D reconstructions out of side-scan sonar data.

In contrast to 2D coverage, very few papers have addressed coverage path planning in 3D environments. Hert et al. presented a sensor-based coverage algorithm for AUVs 3D environments that are projectively planar (2.5D) (Hert, Tiwari, & Lumelsky, 1996). Their approach works by running a 2D coverage algorithm at uniformly spaced depths, and it entirely covers the volume free of obstacles. Lee et al. proposed an extension of the algorithm to cover only the environment's boundary, namely the seabed, which is the main focus in most applications (Lee, Choi, Lee, & Lee, 2009). However, details on how to detect the landmarks used by those algorithms to direct the vehicle are not provided, which makes its application to real-world environments difficult. Atkar, Choset, Rizzi, & Acar (2001) presented another sensor-based coverage algorithm for closed surfaces embedded in 3D space. This algorithm directs the robot to follow cross sections of the surface at an offset distance (similarly to our nominal coverage path-planning method), assuming the robot is equipped with an idealized omnidirectional range sensor ring. Those sensor-based, also known as online, 3D coverage algorithms are theoretically proven but lack experimental validation using real-world vehicles.

More recently, offline 3D coverage algorithms that use a model of the environment have been presented. A coverage algorithm specifically targeted for spray-painting of automotive parts was presented in Atkar et al. (2005), which provides uniform paint deposition. Cheng et al. presented a method for planning time-optimal trajectories for unmanned aerial vehicles (UAVs) covering the exterior of buildings in urban environments (Cheng, Keller, & Kumar, 2008). Their approach first simplifies the target surfaces into hemispheres and cylinders and then plans the trajectories on these simpler surfaces. Their proposal is validated in hardware-in-the-loop simulations using a fixed-wing aircraft. Recently, Englot and Hover introduced a sampling-based algorithm to achieve coverage of complex 3D structures for ship hull inspection (Englot & Hover, 2013). Also building upon the idea of sampling-based planning, the

algorithm in Papadopoulos, Kurniawati, & Patrikalakis (2013) provides coverage for inspection of complex structures using systems with differential constraints. While these latter algorithms based on sampling can handle 3D structures of unprecedented complexity, they do not account for uncertainty in the model of the environment nor in the robot's sensors. As a result, their application is constrained to idealized or highly controlled environments. Moreover, they require large amounts of computation time, and the paths they generate spread randomly in all dimensions of the workspace, making the vehicle's maneuvers hard to predict from an operator's standpoint, although Englot et al. presented a path-smoothing algorithm aiming to mitigate this latter issue (Englot & Hover, 2013).

This paper's account for an uncertain execution of the planned coverage paths connects with the general field of robotic planning under uncertainty. Algorithms for robotic planning under uncertainty use probabilistic models of the robot's state, actions, and environment to produce a feasible plan, which often seeks to minimize some cost function or maximize some objective function. Lambert & Le Fort-Piat (2000) presented probabilistic models for a robot's state, control and map uncertainty. Using the well-known A\* (Hart, Nilsson, & Raphael, 1968; Russell & Norvig, 2003) search algorithm with a cost function accounting for all three sources of uncertainty, they were able to plan safe trajectories for a mobile robot. Many researchers propose extensions to the sampling-based rapidly exploring random tree (RRT) and probabilistic roadmap (PRM) path-planning algorithms (Kavraki et al., 1996; LaValle & Kuffner, 2000) to handle uncertainty. The RRT extensions by Melchior & Simmons (2007) and Kewlani, Ishigami, & Iagnemma (2009) explicitly handle uncertainty associated with terrain parameters (e.g., friction). By taking this uncertainty into account, these planners try to avoid rough terrain. Huang & Gupta (2008) combined an extension to the RRT algorithm with a particle-based SLAM algorithm used to expand the tree. This integrated approach explicitly accounts for sensor, localization, and environment uncertainty in the planning stage.

Some planners seek to maximize the probability of success or rather to minimize an expected cost by taking into account the sensing uncertainty (Carrillo et al., 2012; Gonzalez & Stentz, 2009; Pepy & Lambert, 2006; Platt et al., 2010; Prentice & Roy, 2009; van den Berg, Abbeel, & Goldberg, 2011) while other path planners focus on generating paths with minimum probability of collision with obstacles (Blackmore, Ono, & Williams, 2011; Burns & Brock, 2006; Guibas et al., 2008; Missiuro & Roy, 2006; Nakhaei & Lamiroux, 2008).

Another class of approaches uses Markov decision processes (MDPs) with motion uncertainty to define a global control policy over the entire robot's workspace, providing a connection between planning and control (Alterovitz et al., 2007). To also include sensing uncertainty, partially observable Markov decision processes (POMDPs)

can be used (Bai et al., 2014; Candido & Hutchinson, 2010; Kurniawati, Hsu, & Lee, 2008; van den Berg, Patil, & Alterovitz, 2012). Although POMDPs are theoretically satisfactory, these approaches suffer from scalability problems. Alternatively, Du Toit & Burdick (2012) formulated the problem as a dynamic program (DP) and presented a receding horizon control algorithm that approximates the solution to said DP to navigate in cluttered uncertain environments with moving obstacles.

Active perception algorithms increase robot localization efficacy, or more generally, maximize the robot's information gain along the path by specifically considering the expected uncertainty associated with the next action to be executed by the robot (Burgard, Fox, & Thrun, 1997; Roy et al., 1999; Valencia et al., 2012). Several approaches in this category are particularly related to the underwater domain. Fairfield & Wettergreen (2008) proposed an active localization technique using multibeam sonar for enhanced localization of AUVs gathering bathymetric data. In the context of ship hull inspection, Kim & Eustice (2013) presented a next-best-view visual SLAM approach including planning of paths to revisit salient areas of the hull and reduce the accumulated uncertainty along a ship hull survey. Building upon this approach, Chaves, Kim, & Eustice (2014) combined a sampling-based algorithm with a Gaussian process predicting a measure of saliency in yet to be covered areas of the hull to minimize the robot's uncertainty along the inspection path. Also in a ship hull inspection context, Hollinger et al. (2012a) presented a view planning strategy seeking to improve the quality of the inspection. Finally, Hollinger & Sukhatme (2014) proposed sampling-based motion planning algorithms for planning informative robot trajectories. These algorithms seek to maximize an information metric along the planned path (such as information gain or variance reduction) while also satisfying a prespecified budget constraint (e.g., fuel, energy, or time). A proof-of-concept field implementation of the algorithms on an autonomous surface vehicle (ASV) is provided.

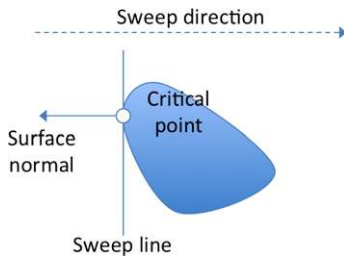
The unquestionable benefits of incorporating uncertainty into planning, however, come at the cost of dramatically higher computational complexity. Rather than directly planning in a probabilistic space, in this paper we use instead fast local replanning to account for discrepancies with the available prior information.

Earlier snapshots of the research reported in this paper have appeared in conference papers (Galceran & Carreras, 2013a; Galceran et al., 2014). This paper presents new experimental results and significantly extends the discussion of the methods employed to achieve them.

### 3. ALGORITHMIC BACKGROUND

In this section, we outline for the reader's convenience key algorithms upon which our paper builds: the boustrophedon decomposition algorithm, which we use to





**Figure 2.** Cell boundaries in Morse decomposition are placed at critical points, where the surface normal of the obstacle is perpendicular to the sweep slice, and parallel to the sweep direction.

plan coverage paths in effectively planar areas accounting for obstacles; the stochastic trajectory optimization motion planning (STOMP) algorithm, integral to our coverage path-replanning approach; and state-of-the-art 3D surface reconstruction techniques composing pipelines for optical and bathymetric surface reconstructions.

### 3.1. 2D Coverage Path Planning: The Boustrophedon Decomposition Algorithm

In this paper, we use the Morse-based boustrophedon<sup>1</sup> decomposition coverage path-planning algorithm introduced by Acar et al. (2002) to cover effectively planar areas of the target area on the sea floor. This algorithm generates a standard mowing-the-lawn pattern, but it is able to account for obstacles in a 2D workspace. As we detail later in Section 4.3, in our case the obstacles represent the high-slope regions of the terrain.

The boustrophedon decomposition is a 2D algorithm that breaks down the target workspace into obstacle-free regions, called cells. Given that they are obstacle-free, the cells can be easily covered using standard mowing-the-lawn patterns. The cell decomposition is encoded as an adjacency graph where nodes represent cells and edges represent adjacency relationships between cells. Thus, finding an exhaustive walk through the adjacency graph guarantees that all cells are visited and that there is full coverage of the workspace as a result.

To determine the cell decomposition, a vertical line is swept through the target area from left to right. Whenever the sweep line encounters a point on an obstacle boundary whose surface normal is perpendicular to the sweep line, as shown in Figure 2, a division between cells is placed along the line. The points where this occurs are called critical points.

At critical points, the connectivity of the sweep line changes, and therefore it is used to determine the cells in

the decomposition. Take for instance the example shown in Figure 3(a), where, at the critical point, the connectivity of the line changes from one to two, and hence the old cell is closed and two new cells are created. In Figure 3(b), at the critical point, the connectivity of the slice changes from two to one, and hence two old cells are closed and a new cell is created. Notice that the line connectivity remains constant within a cell, which guarantees that cells are obstacle-free.

Once the cell decomposition is constructed, it is represented as an adjacency graph, where adjacent cells (nodes) that share a common cell boundary have an edge connecting them. Next, an exhaustive walk through the adjacency graph is determined, and mowing-the-lawn paths are generated in each cell to obtain a full coverage path.

### 3.2. Path Reshaping via Stochastic Trajectory Optimization

We use the STOMP algorithm (Kalakrishnan et al., 2011) to reshape the nominal coverage path so it adapts to the actual target structure perceived onsite via onboard sensors. STOMP explores the space around an initial trajectory by generating noisy trajectories, which are then combined to produce an updated trajectory with lower cost in each iteration. Consider the example in Figure 4, where a cost designed to repel obstacles is used. In each iteration, the trajectory is updated to obtain a lower cost, achieving the effect of keeping it away from the obstacle (shaded in blue in Figure 4).

The algorithm takes as input the start and goal positions (which are kept constant during the optimization process), an initial trajectory from start to goal (which can be as simple as a straight line), and a cost function (which we detail below in Section 5.2.2 for our case). STOMP optimizes a cost function based on a combination of smoothness and application-specific costs, such as obstacles (as in the example above), constraints, or motor torques. An important characteristic of this algorithm is that it does not use gradient information, and so general costs for which derivatives are not available can be included in the cost function.

STOMP optimizes a trajectory,  $\theta$ , discretized as a sequence of  $N$  waypoints. Smoothness costs are represented as a positive semidefinite matrix  $\mathbf{R}$ , such that  $\theta^\top \mathbf{R} \theta$  is the sum of squared accelerations along the trajectory. The accelerations are obtained by means of a second-order finite-difference matrix  $\mathbf{A}$  that, when multiplied by the trajectory's position vector  $\theta$ , produces accelerations  $\ddot{\theta}$ :

$$\ddot{\theta} = \mathbf{A}\theta. \quad (1)$$

Thus, the sum of squared accelerations along the trajectory can be computed as

$$\ddot{\theta}^\top \ddot{\theta} = \theta^\top (\mathbf{A}^\top \mathbf{A}) \theta. \quad (2)$$

Defining the matrix  $\mathbf{R} = \mathbf{A}^\top \mathbf{A}$  allows us to minimize the sum of squared accelerations at each iteration by

<sup>1</sup>The term “boustrophedon” refers to the way an ox alternately drags a plow back and forth.

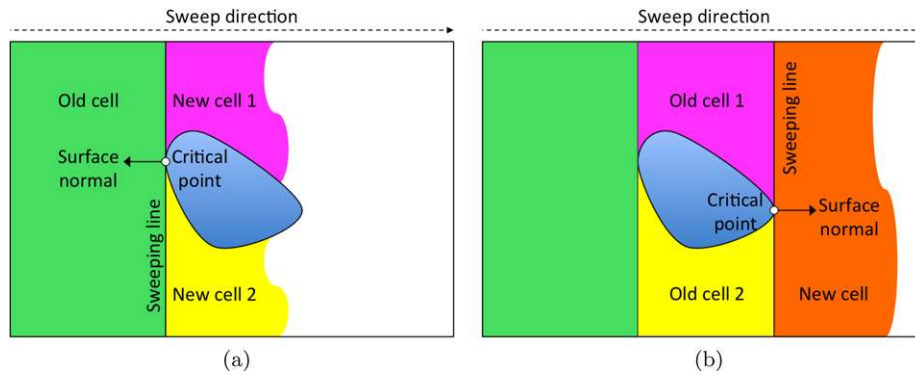


Figure 3. Cell determination with the Morse-based boustrophedon cell decomposition method.

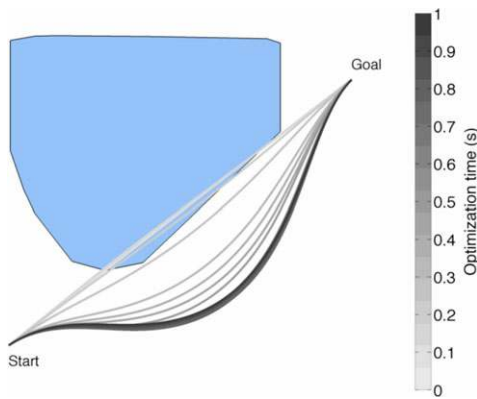


Figure 4. Example execution of the STOMP algorithm in a 2D workspace optimizing a cost function to repel obstacles. The blue-shaded polygon represents an obstacle. The evolution of the trajectory as it is optimized is shown in a time-colored manner.

incorporating the term  $\theta^T R \theta$  into the following cost function:

$$Q(\theta) = \sum_{i=1}^N q(\theta_i) + \frac{1}{2} \theta^T R \theta, \quad (3)$$

where  $\theta_i$  is the  $i$ th waypoint in the trajectory,  $N$  is the total number of waypoints, and  $q(\cdot)$  represents the user-defined cost function.

Further, the  $R$  matrix plays a second critical role in the STOMP algorithm. In each iteration of STOMP, first a set of noisy trajectories is generated by adding noise to the current trajectory, where the noise is sampled from a zero mean normal distribution with  $R^{-1}$  as its covariance matrix. This keeps the generated trajectories smooth and does not allow them to diverge from the start and goal. Figure 5 shows a representation of the noisy trajectories generated to explore the space around the initial trajectory.

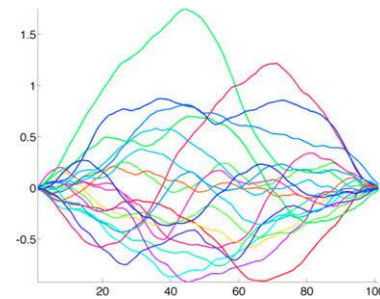


Figure 5. STOMP's trajectory exploration: 20 random samples of noise drawn from a zero mean normal distribution with covariance  $R^{-1}$ .

For each trajectory, its cost per time step is computed with the user-defined cost function  $q(\cdot)$  in Eq. (3). Based on this cost, a probability is assigned to each trajectory, per time step. The trajectory update for each time step is computed as the probability-weighted combination of the noisy trajectories for that time step. Finally, the trajectory parameter vector is updated and the cost for the updated trajectory is computed. The process repeats until convergence of the trajectory cost  $Q(\cdot)$ . For further details on the STOMP algorithm, we refer the reader to Kalakrishnan et al. (2011).

### 3.3. Surface Reconstruction

We show the benefits of our coverage path-planning method for 3D mapping using both bathymetric and optical surface reconstruction techniques. Regarding bathymetric data, we use the screened Poisson algorithm (Kazhdan & Hoppe, 2013) to recover a triangle mesh resembling the surface described by the unorganized range data collected by the multibeam sonar. In this method, points and their associated normals are seen as samples of the gradient of an indicator function volumetrically defining the object, such that the problem solved by the screened Poisson method is to find the indicator function  $\chi$  whose gradient best approximates

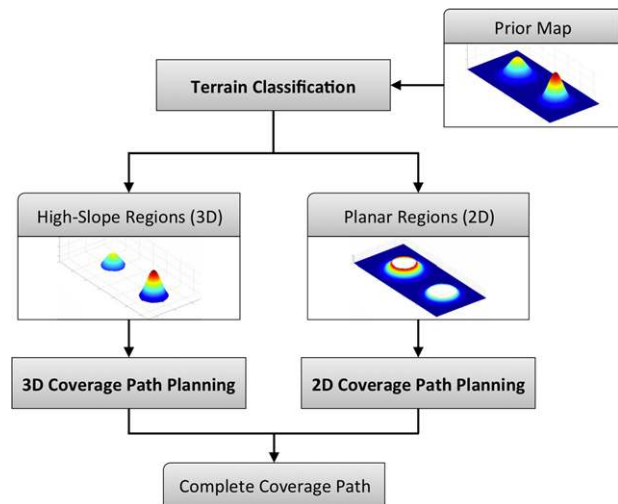
the vector field  $\vec{V}$  defined by the samples:  $\min \|\nabla\chi - \vec{V}\|$ . Once this indicator function is computed, the surface is extracted as its zero level set using a surface contouring technique (Boissonnat & Oudot, 2005).

As the reader may observe, and following the tendency of many surface reconstruction approaches in the state of the art, the screened Poisson method requires additional knowledge of per-point normals. Since multibeam readings provide pure range data, we estimate the normals at each point using the method of Hoppe (Hoppe et al., 1992). This method computes the normals at each point by fitting a plane using principal component analysis (PCA) in a local  $k$ -neighborhood around the point. Then, a globally coherent orientation of the normals is achieved by propagating the orientation of a given seed to its neighbors following the order defined by the minimum spanning tree of the points.

In addition, we apply another mapping technique based on optical data only. Using solely camera images, we follow a sequential pipeline composed by a camera trajectory estimation via structure from motion (Nicosevici et al., 2009), followed by a dense point set sampling through multiple-view stereo (Yang & Pollefeys, 2003). With the scene being also described as a point set, we can use again the previously mentioned surface reconstruction procedure (Kazhdan & Hoppe, 2013). In addition, and since we rely on optical data, we can enhance the reconstructed model using a final texture mapping step. In the present case, we use a direct per-vertex texturing, where each vertex in the mesh takes the color from the mean values of its reprojections on the original images.

#### 4. COVERAGE PATH PLANNING ON BATHYMETRIC MAPS

Our proposed nominal coverage path-planning method has the objective of generating a path for an AUV to cover a target region of the seabed potentially containing 3D structures, which cannot be covered satisfactorily with a traditional mowing-the-lawn pattern. As illustrated in Figure 6, the method is a two-level hierarchical procedure that takes a prior bathymetric map,  $\mathcal{B}(x, y)$ , as input. For every point  $(x, y)$  on the mapped area,  $\mathcal{B}(x, y)$  returns its depth. The method addresses coverage of high-slope (3D) regions and coverage of effectively planar (2D) regions as two separate, decoupled problems, generating a path specifically tailored for each type of region. This is done by first classifying the terrain (bathymetric map) into high-slope regions and effectively planar regions. Next, a coverage path that follows the horizontal cross sections of the surface is generated in the high-slope (3D) regions using a slicing algorithm that we put forth. Finally, a coverage path is planned for the remaining effectively planar (2D) regions of the target region of the sea floor. This latter coverage path is planned using a popular method from the coverage path-planning literature, where the high-slope regions are treated as obstacles.



**Figure 6.** Diagram of the proposed coverage path-planning algorithm for bathymetric maps.

##### 4.1. Terrain Classification

We classify the terrain into high-slope and effectively planar regions using a *slope map*,  $S(x, y)$ . The slope map is calculated for the mapped area as the norm of the gradient of  $\mathcal{B}$ , that is,

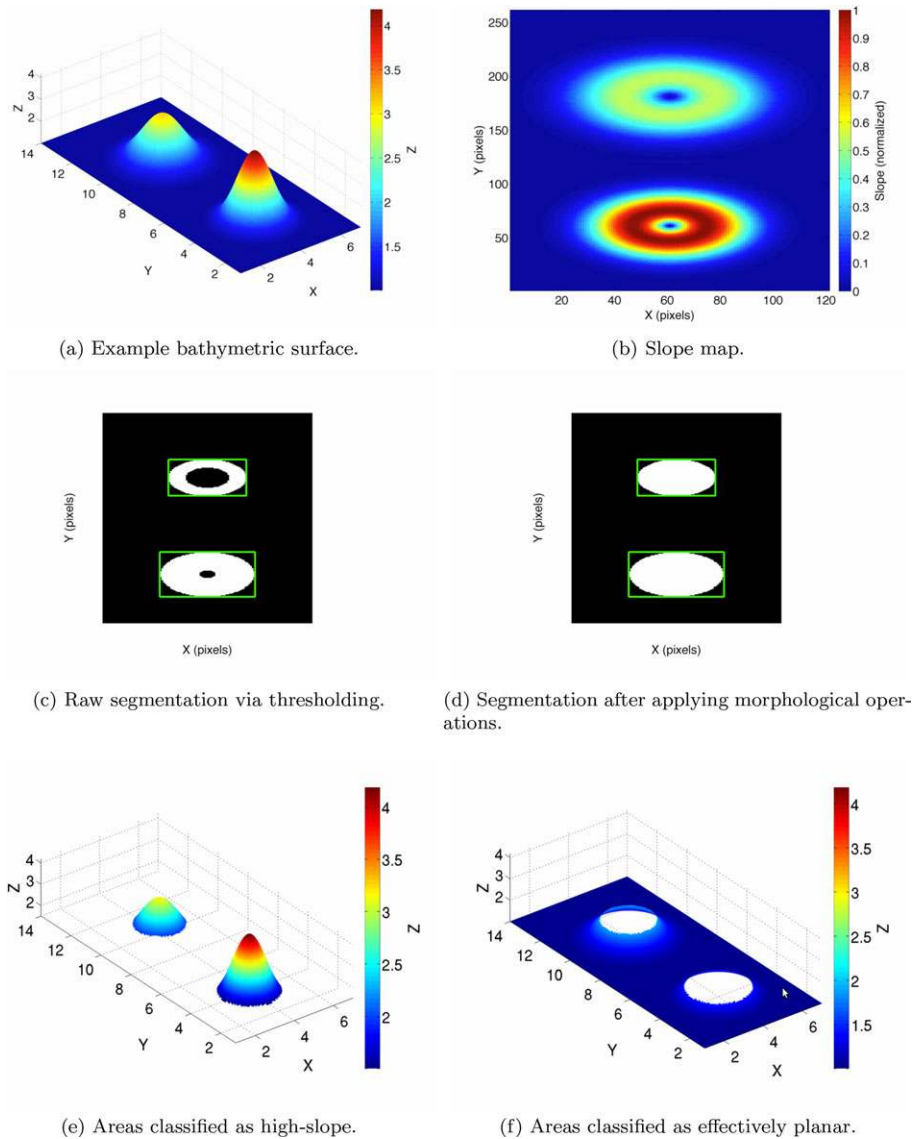
$$S(x, y) = \|\nabla\mathcal{B}\| = \left\| \frac{\partial\mathcal{B}}{\partial x}\vec{i} + \frac{\partial\mathcal{B}}{\partial y}\vec{j} \right\|,$$

where  $\vec{i}, \vec{j}$  are the standard unit vectors in the  $X$  and  $Y$  axis, respectively. Then, we apply a user-defined slope threshold,  $\delta_s$ , to  $S$  to obtain an initial binary classification:

$$T(x, y) = \begin{cases} 1 & \text{if } S(x, y) \geq \delta_s, \\ 0 & \text{if } S(x, y) < \delta_s. \end{cases}$$

The choice of  $\delta_s$  depends strongly on the scale of the mapped area. A valid option is to normalize  $S$  into the  $[0, \dots, 1]$  range and select  $\delta_s = 0.5$ . To filter out small regions and potential artifacts in the initial classification, we apply the dilate and erode morphological operations (Serra, 1982) to  $T$  using an appropriate structuring element.

Let us illustrate this terrain classification process with the simple example shown in Figure 7. First, according to the target surface [Figure 7(a)], the slope map is computed and normalized into the  $[0, \dots, 1]$  range [Figure 7(b)]. Then, a slope threshold  $\delta_s = 0.5$  is applied to the slope map to obtain an initial classification [Figure 7(c)]. Lastly, morphological operations are used to obtain the final classification [Figure 7(d)]. Note how the morphological operations eliminate the holes of the high-slope regions in the initial classification. However, a hole (a planar region “inside” a high-slope region) can also be covered by the coverage algorithms described below. The subsets of the original surface



**Figure 7.** Example of terrain classification into high-slope and effectively planar regions on a synthetic bathymetric surface. The white regions in (c) and (d), delimited by their bounding boxes (in green), correspond to high-slope areas; the black regions correspond to effectively planar areas.

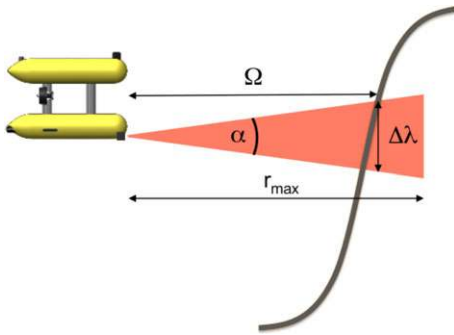
corresponding to each class (high-slope and effectively planar) are shown in Figures 7(e) and 7(f), respectively.

#### 4.2. Covering High-slope Regions Using a Slicing Algorithm

We propose a slicing algorithm to generate an in-detail coverage path for each identified high-slope region. The proposed algorithm draws inspiration from the algorithm of Atkar, Choset, Rizzi, & Acar (2001), which consists in a control law to guide a robot along cross sections of a 3D

structure using range sensors online. Our algorithm, however, plans using a full prior bathymetric map rather than reacting to sensor ranges at each time step. The main idea of the algorithm is to intersect a horizontal slice plane with the target surface at incremental depths, offset these intersections by a desired distance, and then link them together. The resulting intersections correspond to contour lines, or level curves, of the target bathymetric surface. As a result, the planned path acquires data contour-by-contour along the vertical spatial dimension of the workspace. Thus, our algorithm actually operates in a 2.5D workspace, as induced





**Figure 8.** Sensor FOV of a robot's sensor located at an offset distance  $\Omega$  from the target surface. The sensor footprint, given by the aperture angle  $\alpha$ , determines the distance between slice planes,  $\Delta\lambda$ . Note that  $\Omega$  must be big enough to accommodate the robot's volume and not bigger than the maximum sensor range  $r_{\max}$ .

by the prior bathymetric map. The viewpoints provided by the planned path, however, enable a full 3D perception of the target structure, as we show later in Section 7.

Consider a robot equipped with a limited field of view (FOV) sensor. The sensor FOV is determined by an aperture angle,  $\alpha$ , and a maximum range  $r_{\max}$ , as shown in Figure 8. To image the target surface with the sensor, the robot navigates at a user-defined fixed offset distance,  $\Omega < r_{\max}$ , from the target surface. Note that  $\Omega$  must be chosen to accommodate the robot's volume. The sensor footprint on the target surface determines the spacing between successive slice planes,  $\Delta\lambda$  (where  $\lambda$  is the current slice plane depth). Note that the footprint extent depends on the curvature of the target surface on the imaged area. We approximate the footprint extent as a circle of radius  $r = \Omega \tan \frac{\alpha}{2}$ , and therefore  $\Delta\lambda = 2r$ .

The slicing algorithm is detailed in Algorithm 1. The algorithm is applied to each identified high-slope region of the bathymetric map. For each high-slope region, the algorithm takes as input the corresponding subset of the bathymetric map  $\mathcal{B}_r$  [for the example described above, the subset depicted in Figure 7(e)], the offset distance,  $\Omega$ , and the slice plane spacing,  $\Delta\lambda$ .

An example run of the algorithm on the high-slope regions of Figure 7(e) is illustrated in Figure 9. First, the algorithm initializes the horizontal slice plane depth,  $\lambda$ , as the minimum depth (the shallowest) in  $\mathcal{B}_r$  (line 1). The set of intersection edges,  $E$ , is initialized as empty in line 2. The algorithm runs at incremental values of  $\lambda$  until  $\lambda$  surpasses the maximum depth in  $\mathcal{B}_r$  (line 3), that is, from the shallowest down to the deepest point in  $\mathcal{B}_r$  [see Figures 9(a) and 9(b)]. At each depth level, a horizontal plane is intersected with the bathymetric surface (line 4). The function INTERSECT() returns the set of edges composing the intersection, as illustrated in Figure 9(c). The intersection edges of the

current slice plane are added to  $E$ . Recall that these edges correspond to the level curves of  $\mathcal{B}_r$  at the current depth  $\lambda$ , and that these edges are not necessarily closed.

Notice that, when the while loop exits, the intersection edges lie exactly on the bathymetric surface. To obtain a coverage path at the desired offset distance, the edges are then projected onto the offset surface by OFFSETEDGES(), which projects all points in the edges outward from the target surface by an offset distance  $\Omega$ , as shown in Figure 9(d). To do so, each point in the original edge is offset along the projection of the bathymetric surface normal vector on the corresponding horizontal plane. This approach assumes there is enough clearance between protruding bulges in the terrain to accommodate the desired offset distance, which we have found to be a reasonable assumption in practice. A more general approach is to first construct an offset surface of the prior bathymetric map [using, e.g., Liu & Wang (2011)], and apply the slicing in such a surface without the need for edge offsetting.

Lastly, the final coverage path is generated by linking all edges in the set [function LINKEDGES(), line 7]. The function LINKEDGES() connects all edges in  $E$  in a greedy manner. Starting at the first edge in the set, belonging to the first depth level, it will link it to the closest edge in the set, and do so repeatedly for all remaining edges until all of them have been linked. To link a given pair of edges, LINKEDGES() simply traces a straight line path from the first edge to the second. If the straight line intersects the bathymetric surface, it traces the projection of the line on the bathymetric surface. The result of this linkage procedure is shown in Figure 9(e).

---

#### Algorithm 1: Slicing Algorithm

---

**Input:** High-slope region of a bathymetric map,  $\mathcal{B}_r(x, y)$   
**Parameters:** Offset distance,  $\Omega$ . Slice plane spacing,  $\Delta\lambda$ .

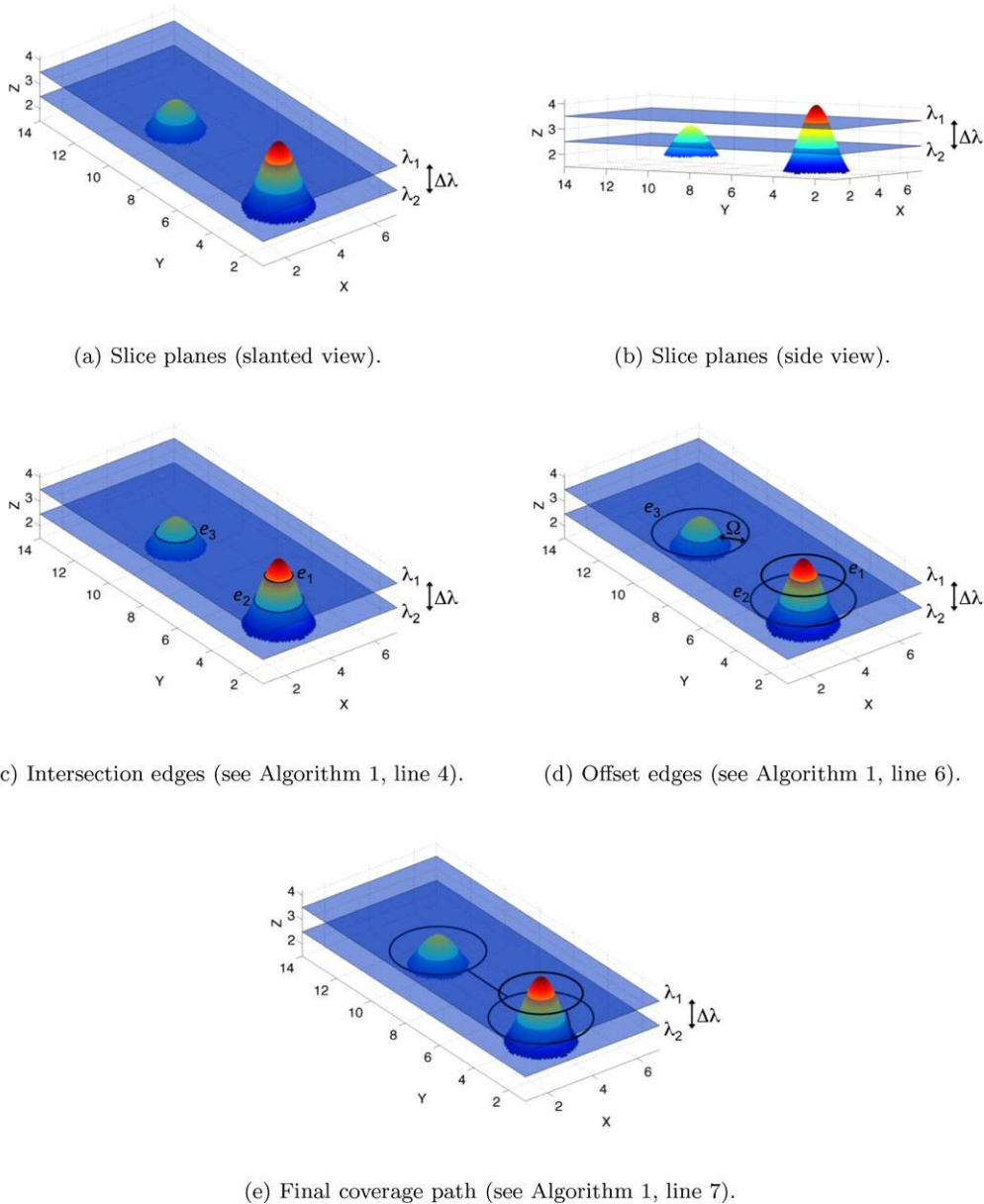
```

1  $\lambda \leftarrow \min \mathcal{B}_r(x, y) + \Delta\lambda$ 
2  $E \leftarrow \emptyset$ 
3 while  $\lambda < \max \mathcal{B}_r(x, y)$ 
4  $E \leftarrow E \cup \text{INTERSECT}(\lambda, \mathcal{B}_r)$  // see Figure 9(c)
5  $\lambda \leftarrow \lambda + \Delta\lambda$ 
6  $E \leftarrow \text{OFFSETEDGES}(E, \Omega)$  // see Figure 9(d)
7  $p \leftarrow \text{LINKEDGES}(E)$  // see Figure 9(e)
8 return p
```

---

### 4.3. Covering the Effectively Planar Regions using the Boustrophedon Decomposition Algorithm

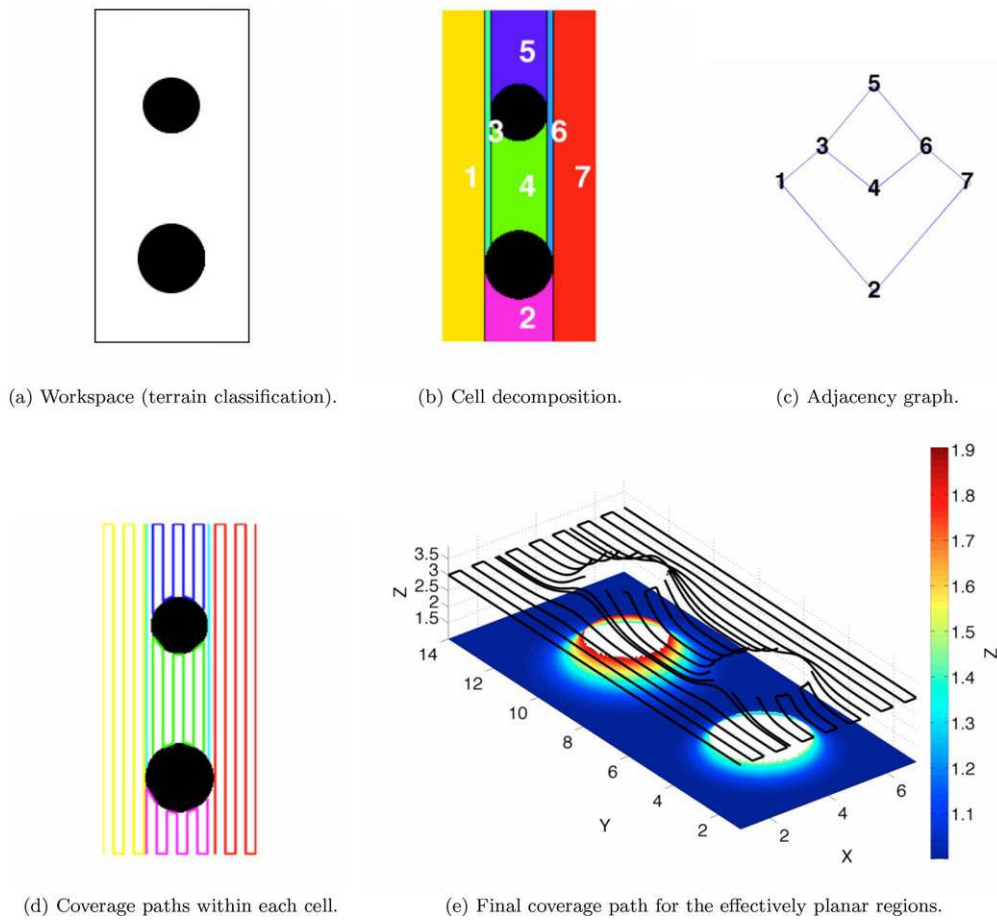
Once a coverage path is obtained for the high-slope regions using the slicing algorithm introduced above, we generate a coverage path for the remaining effectively planar regions treating the (already addressed) high-slope regions as obstacles. To do so, we use the Morse-based boustrophedon decomposition coverage path-planning algorithm introduced by Acar et al. (2002) and outlined in Section 3.



**Figure 9.** Application of the slicing algorithm on an example bathymetric surface. The target high-slope regions are intersected with the slice planes (a) and (b), producing a set of intersection edges (c). The intersection edges are then offset by the desired offset distance  $\Omega$  (d). Lastly, the final coverage path is obtained by linking the offset edges (e).

We apply the boustrophedon decomposition to the 2D workspace induced by the terrain classification procedure, where high-slope regions represent obstacles. Following up on the example introduced above, Figure 10 shows the execution of the algorithm. The workspace [Figure 10(a)] is decomposed into cells [Figure 10(b)], which are encoded in an adjacency graph [Figure 10(c)]. Then, an exhaustive

walk through the graph is determined to obtain the order in which to cover the cells, and finally individual coverage paths are generated within each cell [Figure 10(d)] and linked according to the exhaustive walk. Notice how the mowing-the-lawn path has been projected onto the surface to maintain a constant offset distance (altitude) from the bottom.



**Figure 10.** Application of the Morse-based boustrophedon decomposition algorithm for coverage of effectively planar areas on an example bathymetry.

#### 4.4. Obtaining a Complete Coverage Path

The union of both coverage paths (the coverage path for high-slope areas and the coverage path for effectively planar areas) provides full coverage of the entire target bathymetric surface, using different coverage patterns according to the terrain’s slope and providing the AUV with suitable viewpoints for inspection tasks. Figure 11 shows the coverage paths for both high-slope and effectively planar areas generated on the example target surface introduced above.

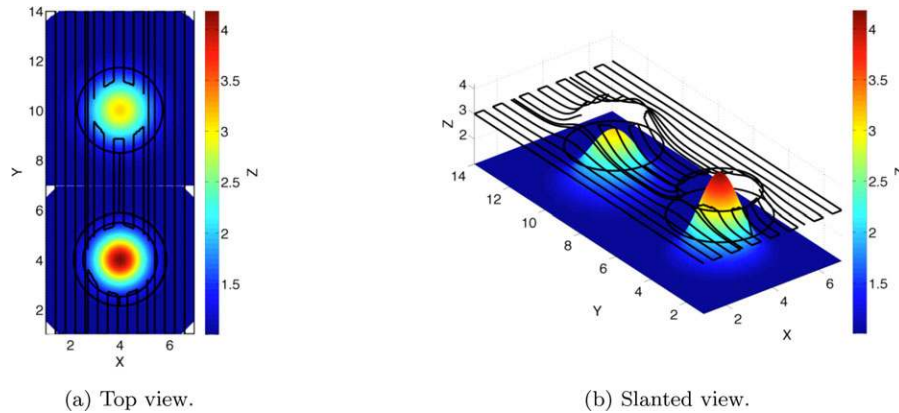
#### 4.5. Example Application to a Real-world Dataset

To further illustrate our coverage path-planning method, we next show its application to a real-world bathymetric dataset. Toward that end, we utilize a bathymetric map of a region of an underwater caldera near the island of Santorini, Greece, shown in Figure 12. The mapped area is approximately  $100 \times 250$  m with depths ranging from 310 down to 350 m. To plan the coverage path, we set an offset

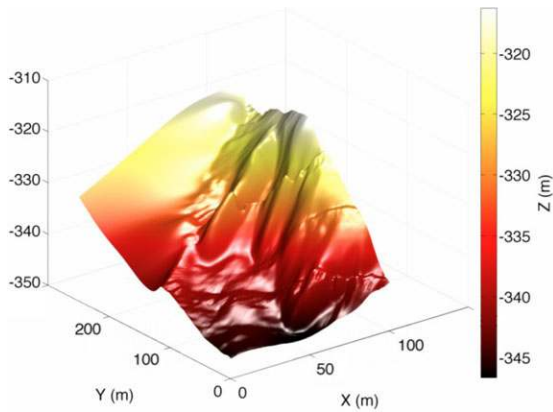
distance  $\Omega = 2$  m and assume a 60-degree FOV sensor to determine the interplane spacing. The slope map,  $S(x, y)$ , of the region of interest (ROI) is shown in Figure 13(a), with values ranging between 0.001 and 0.622. The slope map is classified using a threshold  $\delta_s = 0.5$ , which yields a single high-slope region after applying the described morphological operations as shown in Figure 13(b). The slicing and boustrophedon decomposition algorithms are then applied to the corresponding regions. Figure 14 shows the boustrophedon decomposition process on the effectively planar areas of the caldera scenario, while Figure 15 illustrates the slicing algorithm process. The resulting coverage path after combining the outcome from the boustrophedon decomposition algorithm with that of the slicing algorithm is shown in Figure 16.

### 5. REAL-TIME COVERAGE PATH REPLANNING

As mentioned earlier, most coverage path-planning algorithms described in the literature assume that the map used



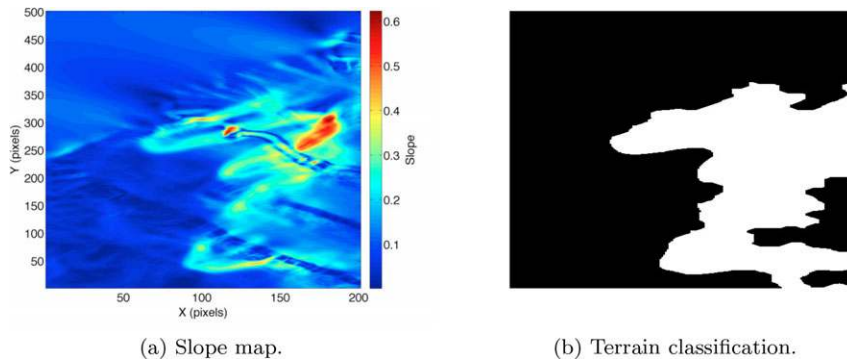
**Figure 11.** Coverage path for both high-slope and effectively planar areas on an example bathymetric surface.



**Figure 12.** Bathymetric map of the Santorini caldera where we apply our nominal coverage path-planning method.

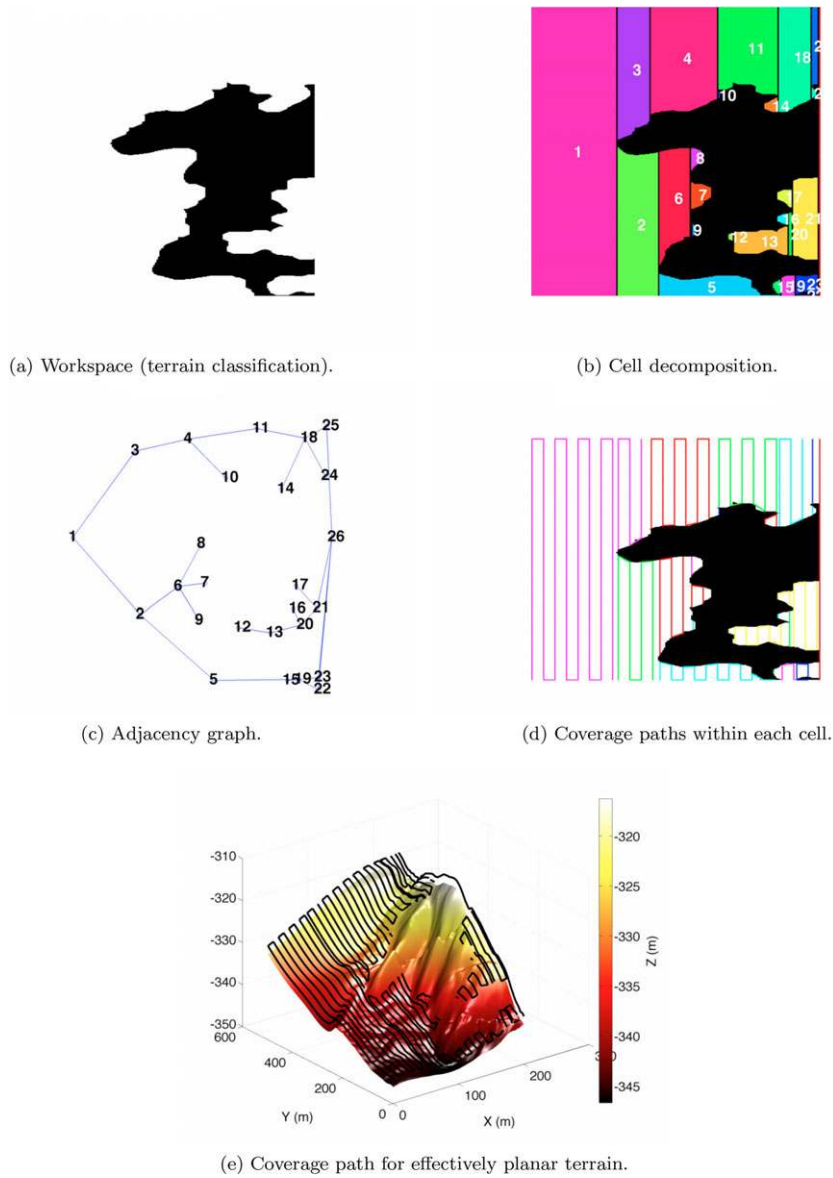
to plan the path accurately represents the environment, and that the vehicle will be able to precisely execute the path, ignoring any sensing or control uncertainties. Clearly, however, these assumptions do not hold when operating in the

challenging oceanic environment, where the robot’s position estimation is subject to error, disturbances such as currents affect the vehicle’s control actions, and the acoustic sensors used to map the environment might lead to inaccuracies and artifacts in the prior map. To be able to conduct AUV inspection tasks in such environments, we present a novel 3D coverage path-replanning method that does not rely on the unrealistic assumption of an idealized path execution. By contrast, our method reshapes a nominal coverage path according to the actual target structure perceived *in situ* using the vehicle’s onboard sensors during the mission. The method takes as input a nominal coverage path planned using the algorithm presented above in Algorithm 1. Recall that the planned coverage path follows the structure contours on the map at uniformly spaced depths maintaining a fixed offset distance from the target surface, accumulating data contour-by-contour along the vertical spatial dimension of the workspace. To handle the vehicle’s position, environmental, and control uncertainty during path execution, we use stochastic trajectory optimization to adapt the initially planned coverage path in real time according to



**Figure 13.** Slope map of the Santorini caldera bathymetric dataset (a) and its terrain classification showing a single high-slope region (b).





**Figure 14.** Application of the Morse-based boustrophedon decomposition algorithm for coverage of effectively planar areas on the Santorini caldera dataset.

the vehicle’s perception. The resulting path is smooth and provides successful coverage under bounded position error.

To design our replanning method, we focus on the estimation error in the  $X$  (surge) and  $Y$  (sway) degrees of freedom (DOFs), since the vehicle’s depth ( $Z$  or heave DOF) and attitude can typically be measured accurately using a pressure sensor and a motion reference unit (MRU), respectively. Thus we model the vehicle’s position error as follows. When the AUV executes the path, its position estimate in the  $X$  (surge) and  $Y$  (sway) DOFs at time  $t$ ,  $(x_t, y_t)$ , is given

by its ground truth position  $(\hat{x}_t, \hat{y}_t)$  subject to a random error  $\epsilon_t$ :

$$(x_t, y_t) = (\hat{x}_t, \hat{y}_t) + \epsilon_t, \tag{4}$$

where we assume  $\epsilon_t$  is bounded by

$$|\epsilon_t| \leq \epsilon_{\max} \forall t. \tag{5}$$

In a real-world AUV,  $\epsilon_t$  is typically brought about by GPS error while at the surface and by dead-reckoning drift while underwater. The upper bound  $\epsilon_{\max}$  can be estimated

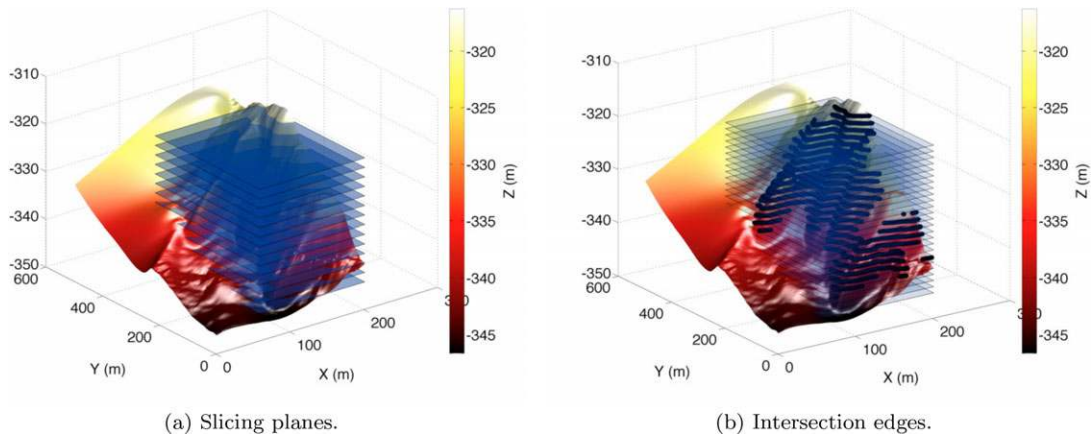


Figure 15. Application of the slicing algorithm for coverage of high-slope areas on the Santorini caldera dataset.

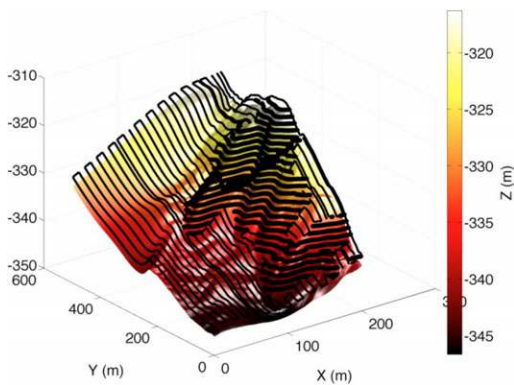


Figure 16. Nominal coverage path for the Santorini caldera dataset. The coverage path for the high-slope region is marked with thicker lines.

according to the accuracy of the AUV’s navigation sensors and the mission duration. Additionally, if a SLAM or map-based localization system is running on the vehicle during the mission, its expected precision can be factored into the error bound estimation. Note that  $\epsilon_r$  also factors in the error in the bathymetric map used to plan the nominal path. Thus, given a nominal path and  $\epsilon_{max}$ , the objective is for the AUV to provide full coverage of the exterior boundary surface of the 3D target structure. Note that, to address potential errors in the map used to plan the nominal path, we rely in the short-term, close-proximity range measurements provided by the vehicle’s sensors during task execution.

### 5.1. Choosing an Appropriate Offset Distance

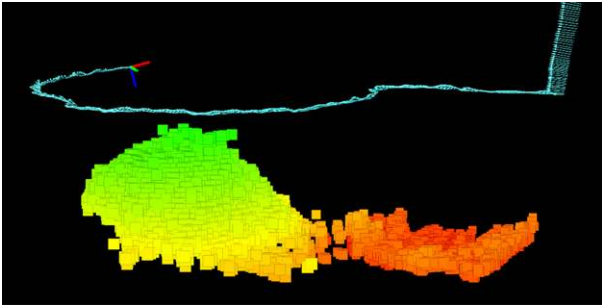
A key point of our method is the choice of the offset distance  $\Omega$  used to plan the nominal coverage path. Of course, as mentioned earlier,  $\Omega$  must be greater than the vehicle’s radius to avoid collisions, assuming a rigid vehicle modeled as a sphere.  $\Omega$  must also lie within the payload sensors range

limits. However, as we will discuss below, for our real-time replanning strategy to succeed,  $\Omega$  must be chosen such that for an error  $|\epsilon_r| \leq \epsilon_{max}$ , the coverage path does not intersect the target surface. This has two implications. First,  $\Omega$  must be greater than  $\epsilon_{max}$ . Second, sufficient clearance between our target structure and potential neighboring structures is required. More formally, for a given depth  $\lambda$  in the nominal path, each point in the nominal path must lie outside the Minkowski sum of the current slice edges and a disk of radius  $\epsilon_{max}$ . Since we assume the vehicle’s heading to be accurately estimated by a MRU, this requirement ensures that the nominal coverage plan is collision-free under  $|\epsilon_r| \leq \epsilon_{max}$ . Given the difficulty in precisely determining  $\epsilon_{max}$ , a safe, conservative value must be used (we describe the determination of  $\epsilon_{max}$  and  $\Omega$  for our system later in Sections 6 and 7).

We note that it is possible to adaptively select a more aggressive or more conservative value of  $\Omega$  for the replanning algorithm, for instance to achieve a closer proximity to the target structure as the uncertainty in the vehicle’s state is decreased by a SLAM or map-based localization system or, conversely, to increase the desired offset distance as the vehicle’s state uncertainty grows as a result of drift while dead reckoning. In any such case, a value of  $\Omega$  that accounts for the total error at the moment of generating the nominal plan must be used. Notice, however, that we have not evaluated this possibility in our experiments.

### 5.2. Real-time Replanning Algorithm

Starting from a nominal coverage path, we propose an iterative replanning method to adapt the path in real time according to a 3D occupancy grid map constructed from range measurements obtained during the inspection mission. In each iteration, our algorithm operates on the section of the nominal path yet to be processed within a given range from the robot. That piece of the nominal path is then reshaped using the STOMP algorithm (Section 3.2) that, given an appropriate cost function, produces a smooth trajectory



**Figure 17.** Example 3D occupancy grid map of an underwater environment obtained by means of Octomap on the GIRONA 500 AUV during one of our inspection trials. The map represents an underwater boulder spanning approximately 13 m from top to bottom. The map cells are color-coded by depth. The axes represent the robot’s pose and the cyan arrows represent the robot poses along the circular trajectory followed to acquire sensor data to construct the map.

that keeps the vehicle at the desired offset distance from the actual target structure. The vehicle then executes the optimized trajectory. The process repeats until the end of the nominal path is reached. Next, we first describe the mapping framework (Section 5.2.1) and the cost function we use in the optimization process (Section 5.2.2). Then, building upon those elements, we detail our coverage path-replanning algorithm (Section 5.2.3).

### 5.2.1. Online Mapping

To obtain a convenient representation of the environment for our replanning method, we incrementally construct and maintain a 3D map of the target structure onboard the vehicle in real time using range data. We use the Octomap (Hornung et al., 2013) probabilistic mapping framework for this purpose, which uses an octree map compression method to keep the 3D model compact and quickly accessible. Figure 17 shows an example 3D occupancy grid map constructed from multibeam sonar range measurements obtained during the sea trials we describe in the experimental outcomes section below. For planning purposes, we retrieve the maximum likelihood map, using the occupied cells therein to reshape the nominal path.

### 5.2.2. Cost Function

Our cost function seeks to keep all the points in the optimized trajectory at the desired offset distance  $\Omega$  from the target structure. The distance between a point  $x$  and the boundary surface of the target structure  $S$  is the shortest distance between  $x$  and all points  $s_i$  in  $S$ . Such a distance is given by the following function:

$$d(x, S) = \min_{s_i \in S} \|x - s_i\|. \quad (6)$$

We define our cost function so it penalizes the difference between the distance from the trajectory points to the target surface and the desired offset distance:

$$q(\theta_i) = |d(\theta_i, S) - \Omega|, \quad (7)$$

where  $d(\theta_i, S)$  is calculated according to the current online map. Recall that the additional smoothness cost  $\theta^T R \theta$  is already incorporated in Eq. (3).

### 5.2.3. Real-time Coverage Path-replanning Algorithm

We propose an iterative real-time replanning algorithm that uses range sensor data to reshape the nominal path to the actual target structure perceived *in situ*. The resulting path is smooth and keeps the desired offset distance  $\Omega$  from the target structure. Recall that our replanning algorithm assumes that for an error  $|\epsilon_i| \leq \epsilon_{\max}$ , the nominal coverage path does not intersect the actual target surface. The algorithm reshapes, in each iteration, the section of the nominal path yet to be processed within a range  $R$  from the vehicle’s position. The magnitude of  $R$  must be smaller than the maximum sensor range used to perceive the target structure since the environment is still unknown beyond that limit. Once optimized, the vehicle begins executing the path and the algorithm restarts from the current vehicle position. The process continues until the entire nominal coverage path has been processed.

Algorithm 2 details our real-time coverage path-replanning algorithm. In each iteration, the algorithm takes the section of the nominal path composed of all unprocessed waypoints within the given range  $R$  from the vehicle (lines 4–8). Next, an initial trajectory is built based on this path section (line 9). We do so by first building an initial geometric path. To construct this initial geometric path, the last waypoint (the most distant from the vehicle) in the current nominal path section is projected along the surface normal, so it lies at the desired distance  $\Omega$  from the target structure. This step is necessary because the goal of the initial trajectory remains constant during the optimization process. Then, the initial path is composed by (1) a straight line connecting the current vehicle position to the first waypoint of the current path section, (2) the current path section itself, and (3) a straight line connecting the last waypoint of the current path section to its projection along the surface normal. This initial path is then discretized into time steps to obtain an initial trajectory. This initial trajectory generation procedure is illustrated in Figure 18.

Next, the initial trajectory is optimized using the STOMP algorithm (line 10). The optimization takes place in the vehicle’s horizontal ( $X$ - $Y$ ) plane, leaving the vertical ( $Z$ ) coordinates of the nominal path unchanged. The current map  $\mathcal{M}$  is passed as an argument to compute the cost function given in Eq. (7). Finally, the vehicle begins executing the optimized trajectory (line 11) and the process repeats

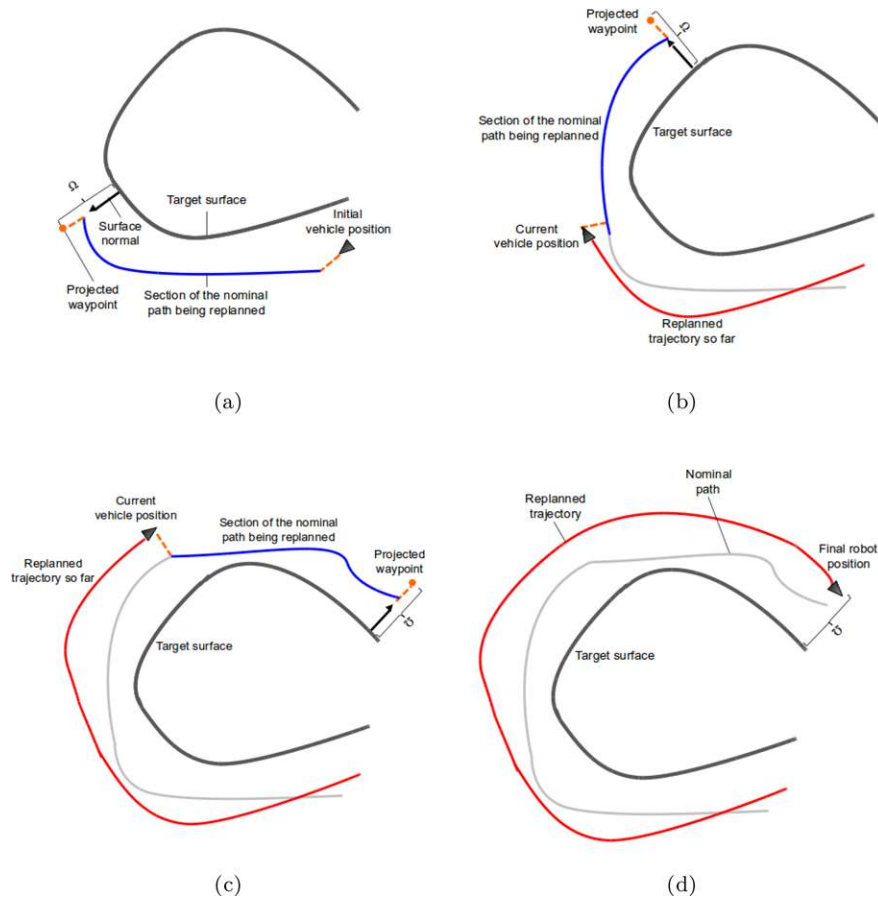


Figure 18. Illustration showing successive replanning stages of the proposed real-time coverage replanning algorithm.

until the end of the nominal path is reached, as illustrated in Figure 18.

---

**Algorithm 2:** Real-time Coverage Path Replanning

**Input:**

- Nominal coverage path as a list of  $K$  waypoints  $w_0, \dots, w_K$ .
- Current environment’s map,  $\mathcal{M}$ .
- Replanning step range,  $R$ .

```

1 Navigate to initial waypoint  $w_0$ 
2  $i \leftarrow 0$ 
3 while  $i < K$  do
4  $x \leftarrow \text{GETROBOTPOSITION}()$ 
5  $\text{pathSection} \leftarrow \emptyset$ 
6 while  $\text{DISTANCE}(x, w_i) < R$  and  $i < K$  do
7  $\text{pathSection}.\text{APPEND}(w_i)$ 
8  $i \leftarrow i + 1$ 
9  $\theta \leftarrow \text{INITIALTRAJECTORY}(\text{pathSection}, x)$ 
10  $\text{optimizedTrajectory} \leftarrow \text{STOMP}(\theta, \mathcal{M})$ 
11 EXECUTE( $\text{optimizedTrajectory}$ )

```

---

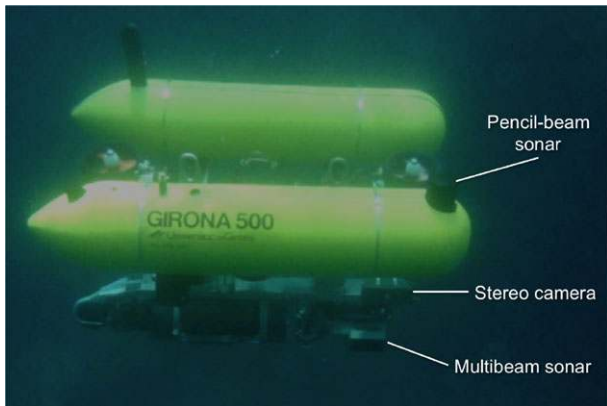
**6. EXPERIMENTAL SETUP**

We validated our strategy for autonomous inspection of underwater structures by conducting inspection tasks with the GIRONA 500 AUV (Ribas et al., 2012). In our experiments, we target a manmade structure in a harbor environment and an underwater boulder lying at 40 m depth in open waters. We next introduce GIRONA 500 and its sensor suite and outline the scenarios in which we conducted the experimental validation.

**6.1. Experimental Platform: The GIRONA 500 AUV**

GIRONA 500, shown in Figure 19, is a reconfigurable AUV designed to operate at depths up to 500 m. The vehicle is composed of an aluminum chassis supporting three torpedo-shaped hulls (0.3 m in diameter and 1.5 m in length) and a variable number of thrusters. The typical thruster configuration, which we use for our experiments, consists in four thrusters providing controllability in the surge ( $X$ ), sway ( $Y$ ), heave ( $Z$ ), and heading (yaw) DOFs. The design of the vehicle offers good hydrodynamic performance and



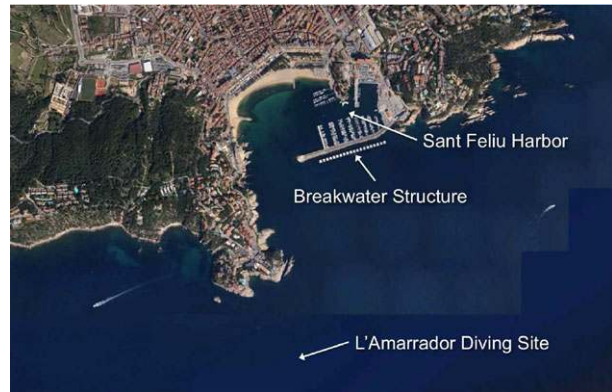


**Figure 19.** The GIRONA 500 AUV during the 3D coverage with replanning sea trials. The vehicle is equipped with a pencil-beam sonar and side-looking multibeam sonar and stereo camera.

room for equipment while keeping the vehicle compact, allowing deployment from small vessels. The overall dimensions of the AUV are 1 m height, 1 m width, 1.5 m length, and it weighs under 200 kg, the actual weight depending on the particular vehicle configuration and payload.

### 6.1.1. Sensor Suite

GIRONA 500's standard navigation sensor suite includes a pressure sensor, Doppler velocity log (DVL), inertial measurement unit (IMU), and GPS to receive fixes while at the surface. The measurements from these sensors are integrated via an extended Kalman filter (Kalman, 1960) to perform dead-reckoning navigation and estimate the vehicle's pose. In addition, an ultrashort baseline (USBL) system allows us to localize and track the vehicle from a support vessel at the surface. We note that, in the experiments presented below, there was no SLAM or map-based localization module running aboard the vehicle. As a result, the pose uncertainty along the vehicle's trajectory can grow without bound. To perceive the environment and collect valuable sensory data for inspection tasks, we equip the vehicle with a SeaKing pencil-beam sonar by Trittech, a Delta T multibeam bathymetry sonar by Imagenex, and a stereo camera system by Point Grey. The pencil-beam sonar is mounted on the front end of the upper-right hull of the vehicle to scan on a horizontal plane, being able to detect forthcoming objects along the trajectory of the robot. The multibeam sonar is mounted looking sideways, with its beams spanning a 120 degree fan in a vertical plane, being able to perceive structures to the side of the vehicle as it advances. These two sensors provide range measurements that are used to update the 3D occupancy grid map online during the mission. Finally, the stereo camera system is mounted side-looking



**Figure 20.** Location of the target structures on which we conducted our experiments. Satellite imagery: Google Earth, TerraMetrics, Institut Cartogràfic de Catalunya.

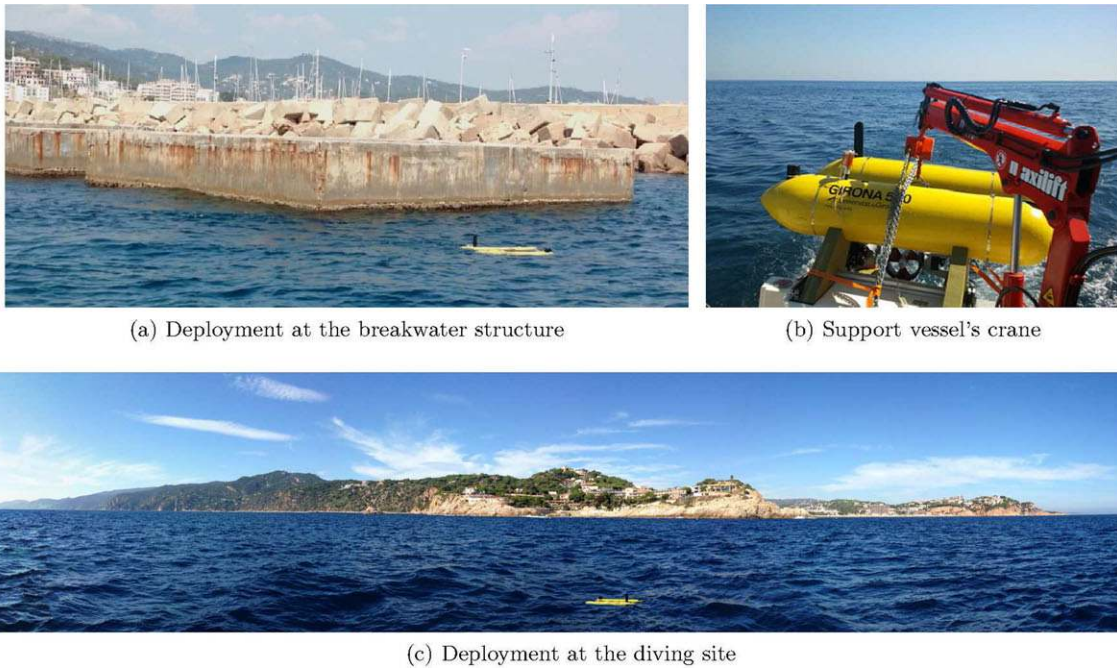
as the multibeam and is able to gather optical imagery of the target structures with a 65 degree FOV.

## 6.2. Scenarios

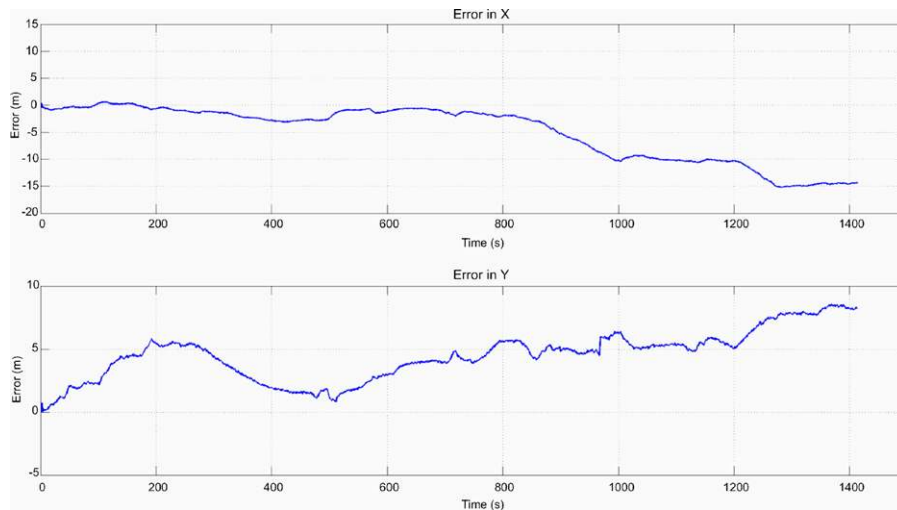
We planned and executed autonomous inspection tasks with the GIRONA 500 AUV in two different scenarios, both nearby the harbor of Sant Feliu de Guíxols in the Costa Brava of Catalonia, northeastern Spain. First, we conducted the inspection of a large concrete block part of a breakwater structure that protects the harbor from the effects of weather and long-shore drift. This breakwater structure is composed of 20 such blocks, each block's footprint being approximately  $5 \times 5$  m, spanning from 2 m above the surface down to the bottom at 10 m depth. Second, we planned and executed the inspection of a popular diving site featuring rich marine biodiversity known as "l'Amarrador." This diving site is located approximately 1 Km into the sea from the harbor of Sant Feliu and features a natural underwater boulder based at 40 m depth and rising up to 27 m depth. Figure 20 shows the location of the Sant Feliu harbor and both scenarios on satellite imagery. Deployment of GIRONA 500 at the target sites is shown in Figure 21. We deploy and recover the vehicle from a 7-m-long boat equipped with a crane for lifting the vehicle, shown in Figure 21(b).

## 6.3. State Error Estimation

As mentioned above, a key point of our proposed coverage scheme is estimating the total maximum position error  $\epsilon_{\max}$  that will be incurred during the task in order to determine the desired offset distance  $\Omega$ . We know from prior field trials that the error induced by GIRONA 500's dead-reckoning system is approximately 0.01% of the trajectory length [see, e.g., Zandara et al. (2011)]. Figure 22 shows the position error in the  $X$  and  $Y$  coordinates along a typical lawnmower-type survey performed with GIRONA 500's sensor suite, where



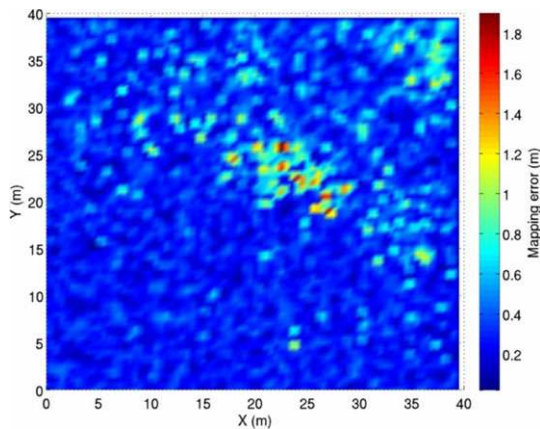
**Figure 21.** GIRONA 500 initiating the planned inspection tasks at the target sites. A small surface vessel equipped with a crane allows us to deploy and recover the vehicle.



**Figure 22.** Position error in the  $X$  (top) and  $Y$  (bottom) coordinates along a typical lawnmower-type survey performed with GIRONA 500's sensor suite.

ground truth is provided by GPS measurements. The survey took 24 min to complete, the trajectory being 1,500 m long. However, in this paper we execute much shorter missions of under 300 m total length, depending on the experiment and deployment point. Additionally, the testing sites in this paper provide good bottom lock for the DVL, resulting in accurate vehicle velocity estimates.

In addition, typical error in prior bathymetric maps of the scale we use to plan the coverage tasks in this paper is typically well below 1 m, although there are some sporadic areas where the error can approach 2 m. As a typical example, Figure 23 shows the mapping error of “l’Amarrador” prior map (introduced below in Section 7.2). High error peaks are often located in salient regions of the terrain, such



**Figure 23.** Map error of “l’Amarrador” site prior map we use to plan a coverage task. The map error is computed as the standard deviation of the bathymetry points falling in each cell of the 2.5D grid model composing the map.

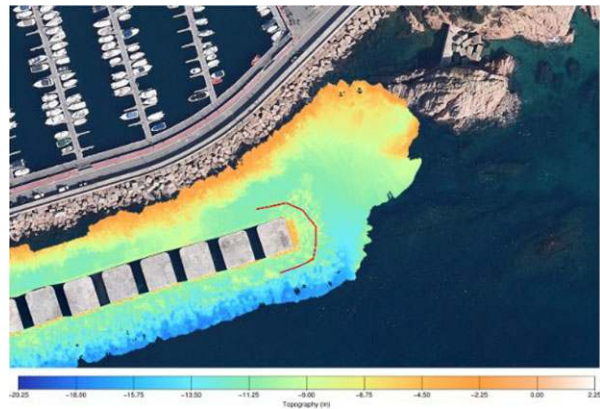
as an underwater boulder summit in this case. The map error is computed as a measure of self-consistency as the standard deviation of the bathymetry points falling in each cell of the 2.5D grid model composing the map, each cell being  $0.3 \times 0.3$  m in this case. For further reference, see also the mapping results in Galceran et al. (2013).

Finally, another important factor contributing to position error is the GPS initialization provided by the commodity GPS device mounted on GIRONA 500 when starting a mission at the surface.

To obtain an estimate of the typical position error as a blend of the aforementioned factors, we performed a total of six vertical dives at a known location nearby an underwater boulder. The expected range to the boulder is known according to a prior bathymetric map of the area, and the actual range was measured by GIRONA 500’s horizontally scanning pencil-beam sonar. The average error measured was 3.7 m, with the maximum error never surpassing 5 m (4.6 m was the actual maximum error we measured). Therefore, in both experiments, we estimate that we are dealing with a maximum position error of 5 m. Hereby, we set the estimation of the total maximum error  $\epsilon_{\max}$  in our replanning strategy to a different conservative value depending on each particular setup, as we detail in the experimental outcomes section below.

## 7. EXPERIMENTAL OUTCOMES

We next present experimental outcomes that show the potential of the 3D coverage path-planning and surface reconstruction techniques introduced earlier in this article. Both our 3D nominal coverage path-planning and -replanning algorithms have been implemented in Python and integrated with GIRONA 500’s software architecture (Palomeras et al., 2012) using the robot operating system (ROS) frame-



**Figure 24.** Bathymetric map of the area surrounding Sant Feliu harbor’s breakwater structure overlapped on satellite imagery. The nominal coverage path, targeting the rightmost block of the breakwater structure, is shown in red. Satellite imagery: Google Earth, TerraMetrics, Institut Cartogràfic de Catalunya.

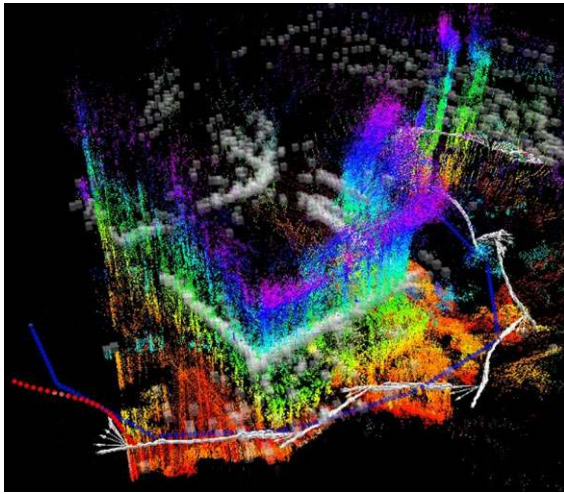
work (ROS, 2014) to run onboard the AUV. The implementation produces a nominal 3D coverage path in less than a second on the prior bathymetric maps used in our experiments, while a typical replanning step completes in 0.5 s, which is enough to reliably execute the inspection tasks at the slow speeds ( $< 0.5$  m/s) at which GIRONA 500 operates in these experiments. To present our results, we replay the mission logs and visualize the data using Rviz, the visualization package provided by the ROS framework. As previously mentioned, we have tested our method by performing two coverage tasks with the GIRONA 500 AUV at sea, inspecting a large concrete block in a breakwater structure and “l’Amarrador” site featuring a 13-m-high underwater boulder.<sup>2</sup> The concrete block scenario served as a minimal validation of our real-time replanning technique, where the nominal path is a simple offset edge around the block. This initial validation allowed us to confidently move on to the more challenging diving site at 40 m depth, where we demonstrate the full potential of our nominal coverage path-planning and -replanning techniques. The *a priori* bathymetric charts used to plan the nominal coverage paths were created by members of our lab using a vessel equipped with GIRONA 500’s Delta T multibeam sonar.

### 7.1. Inspection on a Breakwater Structure

As stated earlier, the first coverage task in which we test our method serves as a minimal test of our implementation. Figure 24 shows the *a priori* bathymetric chart (overlapped on satellite imagery) we use to plan a nominal coverage path (also shown in Figure 24) for this task using our nominal

<sup>2</sup>A video showcasing these experiments can be found at <http://www.youtube.com/watch?v=2REWf6jbdZ0>



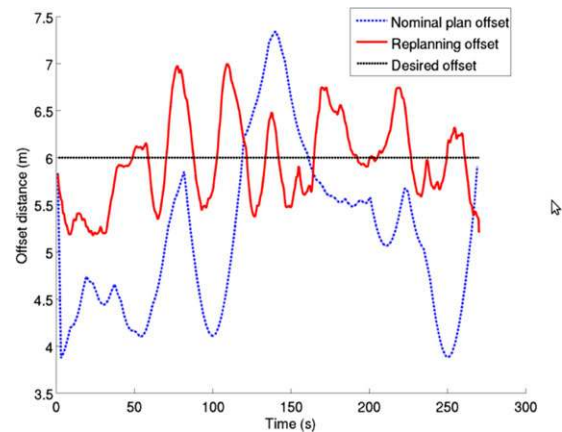


**Figure 25.** Real-time replanning on the concrete block coverage experiment at the last replanning step of the task: nominal coverage path (blue-dotted line); optimized trajectory that the robot is executing at that particular instant (red-dotted line); overall trajectory (white arrows); occupied cells in the online map (white cubes). The depth-colored range data acquired by the multibeam sonar is also displayed.

coverage planning method. In this minimal validation experiment, we target the rightmost block of the structure and we plan a coverage path of a single contour at 5 m depth, which will allow the multibeam sonar to image most of the in-water part of the block. Aiming to capture optical data of the structure, and since we deal with a somewhat controlled environment in this experiment, we use a relatively short offset distance  $\Omega = 6.0$  m to plan the nominal path (hence assuming  $\epsilon_{\max} < 6.0$  m). The nominal path resulting from the nominal path-planning phase is also shown in Figure 25. Note that the path is not closed (it resembles a semicircle) and therefore provides coverage of only three of the four vertical faces of the block.

The trajectory followed by the robot during the real-time replanning phase is shown in Figure 25 with the online map and the depth-colored raw range data acquired by the side-looking multibeam sonar. It can be observed that the map includes many outliers, mainly due to surface reflections of the pencil-beam and multibeam sonar beams. Nonetheless, the resulting trajectory provides full sensor coverage of the targeted in-water part of the structure.

Figure 26 shows the desired offset distance, the offset distance achieved by our replanning scheme along the executed trajectory, and the offset distance associated with the nominal plan. All distances are computed as per Eq. (6) using the online map incrementally constructed for the purpose of replanning. Note that we did not execute the nominal path. In fact, directly executing the nominal path without any reshaping strategy can drive the AUV dangerously



**Figure 26.** Offset distance achieved along the replanned trajectory compared to the nominal plan in the breakwater structure scenario.

close to the target structure, as shown in the plot. Nonetheless, as evidenced by the oscillations about the desired offset distance incurred by the replanned trajectory, the aforementioned outliers in the online map pose a difficulty to the optimization procedure as new data are added to the map. Overall, however, the proposed replanning strategy achieves a safer trajectory by staying closer to the prescribed offset distance in the nominal plan.

## 7.2. Inspection of “l’Amarrador” Diving Site

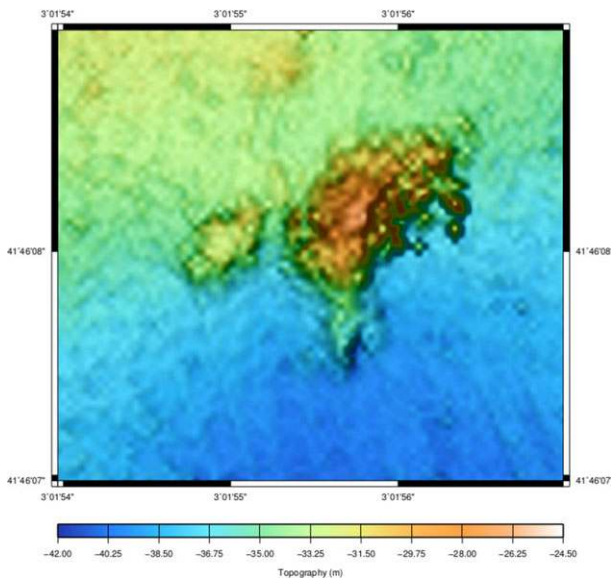
We now show results obtained at “l’Amarrador” diving site. Recall that the underwater boulder in this site rises from 40 m depth up to 27 m, being approximately 13 m high. We will apply to this scenario our nominal coverage path-planning algorithm and our real-time replanning algorithm. In addition, we will also show how our nominal coverage path-planning algorithm can be applied to the 3D occupancy grid map constructed online during the mission.

### 7.2.1. Nominal Coverage Path Planning using a Prior Map

We start by generating a nominal coverage path for the entire diving site using a prior bathymetric chart of the site, shown in Figure 27. This bathymetric chart was generated out of multibeam range data using the MB-System mapping software (MBARI, 2013), and was filtered to remove outliers and adjusted to maximize the map’s self-consistency. Each cell in the uniform grid composing the bathymetric model is  $40 \times 40$  cm.

The terrain classification for “l’Amarrador” site is shown in Figure 28. The application of the boustrophedon algorithm for coverage of the effectively planar region is illustrated in Figure 29, while Figure 30 shows the application of the slicing algorithm for coverage of the high-slope region. The final nominal coverage path for the site





**Figure 27.** Prior bathymetric map of “l’Amarrador” site.

is shown in Figure 31. The coverage path for the effectively planar region is basically a standard mowing-the-lawn path like those used by most AUVs in survey missions. Today, most AUVs are able to track such mowing-the-lawn paths. Therefore, we will focus our discussion on the execution of the more challenging coverage paths for high-slope regions.

The plan for the high-slope region (i.e., the boulder) consists of two contours spaced 2 m apart in the vertical axis. This spacing provides some redundant coverage, which is of interest for testing SLAM and 3D reconstruction algorithms since the overlap allows these algorithms to match sensory data to previously seen features on the environment. There are two important factors to take into account when choosing an offset distance to plan this task. First, this site is in an open sea environment, and there exists a threat of strong currents. Second, the mission is significantly longer,

incurring a potentially larger error due to dead-reckoning drift. For these reasons, we use a more conservative offset distance than in the previous task:  $\Omega = 10$  m. Unfortunately, at this offset distance, the water turbidity conditions did not allow for optical imaging of the underwater boulder. Therefore, only the sonar range data are of interest in this experiment.

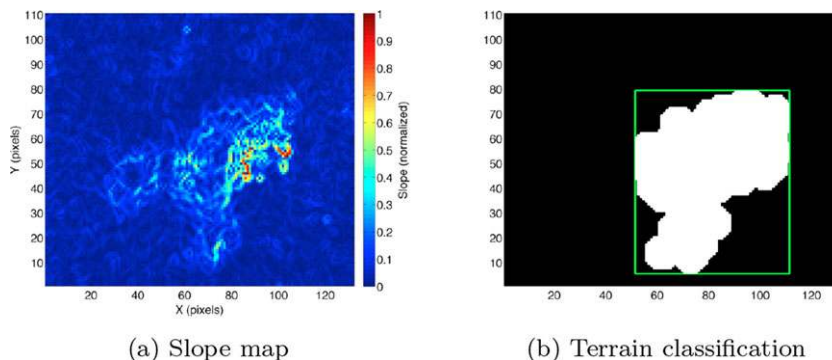
### 7.2.2. Nominal Coverage Path Planning using an Online Map

Next, we show that our nominal coverage path-planning method can be applied also to a 3D occupancy grid map constructed online onboard an AUV using Octomap, as described in Section 5.2.1. This capability might be of interest when a prior bathymetric chart of the target site is thought to be inaccurate or outdated, or when a prior chart is not available at all.

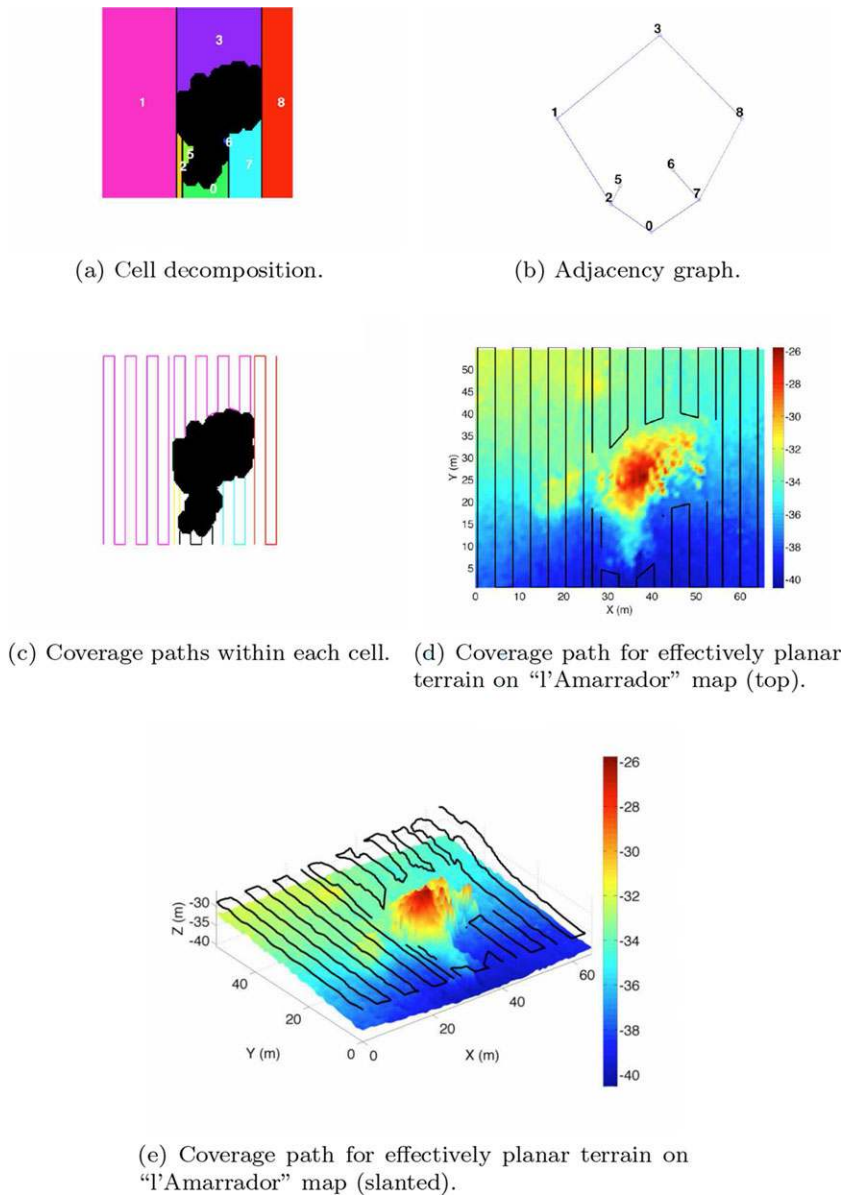
In one of the sea trials at l’Amarrador,” we commanded GIRONA 500 to follow a preplanned, constant-depth circular trajectory feeding range data to the Octomap mapping system. The trajectory was centered at the boulder’s peak. To prevent a collision, it kept a constant depth of 22 m (well above the boulder’s peak at 27 m) and a radius of 25 m, with the vehicle turning clockwise with its sensors pointing inward toward the boulder. After completing the circular trajectory, the slicing algorithm for 3D coverage path planning was applied to the 3D map. The result is shown in Figure 32. Although the map contains a substantial number of outlier cells due to spurious multibeam measurements, the path with offset  $\Omega = 10$  m produced by the algorithm is feasible. Nonetheless, since we had a post-processed, more refined prior chart available, we used the nominal path planned on this latter chart for inspecting the site.

### 7.2.3. Real-time Coverage Path Replanning

Figure 33 shows two instants of the real-time replanning phase together with the nominal coverage path, which



**Figure 28.** Slope map and terrain classification for “l’Amarrador” site.



**Figure 29.** Application of the Morse-based boustrophedon decomposition algorithm for coverage of effectively planar areas on the “l’Amarrador” scenario.

GIRONA 500 is reshaping so it agrees with the perceived sonar range data of the underwater boulder. To minimize the effect of potential artifacts in the online 3D map, and taking into account that we deal with an unstructured environment in this experiment rather than a manmade one, we use only a 3-m-thick horizontal slice of it, spanning 1.5 m above and 1.5 m below the AUV, as represented by the white cubes in Figure 33. As can be observed in the figure, the vehicle starts at the surface, dives down to the depth of the first coverage edge of the plan in a safe area, and starts

the coverage task. Along the overall coverage trajectory in this experiment, and according to the online map, the AUV kept a mean distance to the target structure of 9.41 m, with a standard deviation of 0.93 m. The trajectory executed to cover the deepest coverage edge of the plan is shown in comparison with the nominal path in Figure 34. Note how the coverage trajectory, in contrast with the nominal path, adapts to the actual shape of the boulder perceived onsite. The trajectory provides successful coverage of the underwater boulder, allowing a full 3D perception of the target

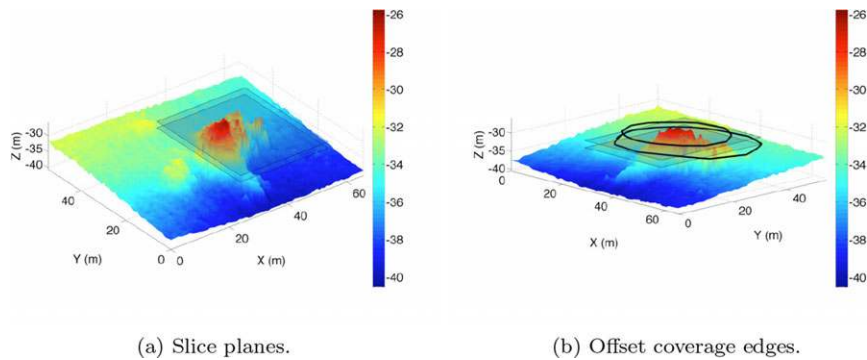


Figure 30. Application of the slicing algorithm for coverage of high-slope areas on the “l’Amarrador” scenario.

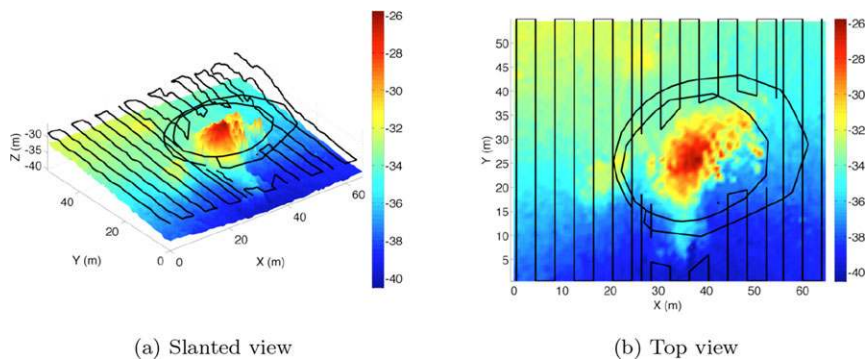


Figure 31. Full nominal coverage plan for “l’Amarrador” site.

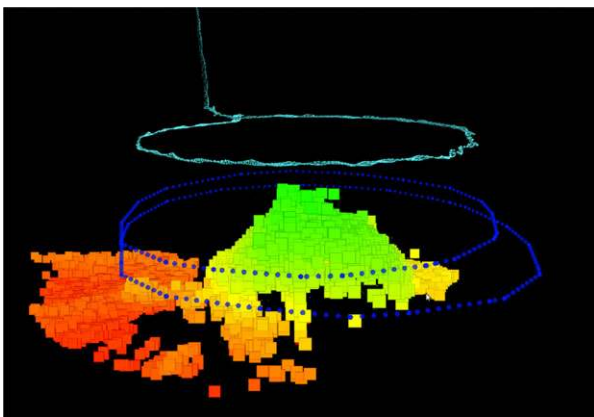


Figure 32. Nominal 3D coverage path planning at l’Amarrador’ site using a 3D occupancy grid map constructed online. The cells of the map are color-coded by depth. The blue-dotted line shows the planned coverage path. The cyan arrows represent the poses along the circular trajectory followed by the AUV to acquire the map.

structure as demonstrated in the resulting 3D maps presented in the following section.

As in the breakwater structure results above, Figure 35 shows, for the underwater boulder scenario, the desired

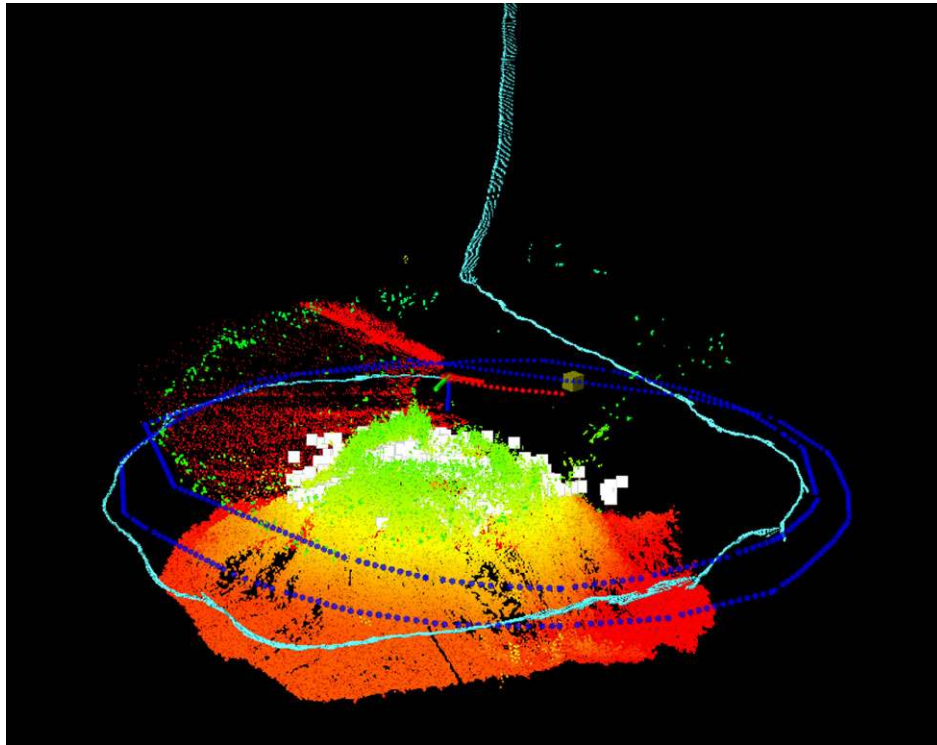
offset distance, the offset distance achieved by our replanning scheme along the executed trajectory, and the offset distance associated with the nominal plan (which was not executed). As evidenced already in Figure 34, the nominal plan deviates significantly from the desired offset distance, leading to an increased threat of collision. Conversely, the replanned coverage trajectory stays closer to the desired offset, and as a result provides sensor viewpoints closer to those mandated by the nominal coverage plan.

### 7.3. Surface Reconstruction Outcomes

This section shows the direct application of the methods in this article to the problem of surface reconstruction on complex scenarios. We show in the following the results of the surface reconstruction pipeline on the two previously presented datasets. All the data products we show are the direct result of the automatic 3D mapping techniques we use, without any manual tuning or refinement.

#### 7.3.1. Breakwater Structure

Figure 36 shows the surface reconstructed from the multi-beam raw range data, globally positioned with respect to the navigation information of the AUV. Note how the point cloud depicted in Figure 25 is far from ideal, as it contains



**Figure 33.** Real-time replanning on “L’Amarrador” underwater boulder at the end of the deepest coverage edge: nominal coverage path (blue-dotted line); optimized trajectory that the robot is executing at that particular instant (red-dotted line); overall trajectory (cyan arrows); occupied cells in the current slice of the online map (white cubes); and last processed waypoint of the nominal plan (yellow cube). The current pose of the vehicle is represented by the red-green-blue 3D axis. The depth-colored range data acquired by the multibeam sonar are also displayed.

high levels of noise and outliers coupled with registration errors. As previously commented, since the surface reconstruction method used (Kazhdan & Hoppe, 2013) requires oriented point sets, we computed per-point normals with the method of Hoppe (Hoppe et al., 1992), using a neighborhood of  $k = 200$  points. We found this value for the  $k$  parameter to be a good tradeoff between the density on the data and the noise level to take into account during PCA computations, and varying its value by small amounts did not change the results noticeably. Despite the defect-laden nature of the input data, the screened Poisson method is able to recover the surface with reasonable accuracy. However, data defects cause some nonexistent artifacts to show up in the top- and bottommost parts of the model and some undesirable roughness in its front wall.

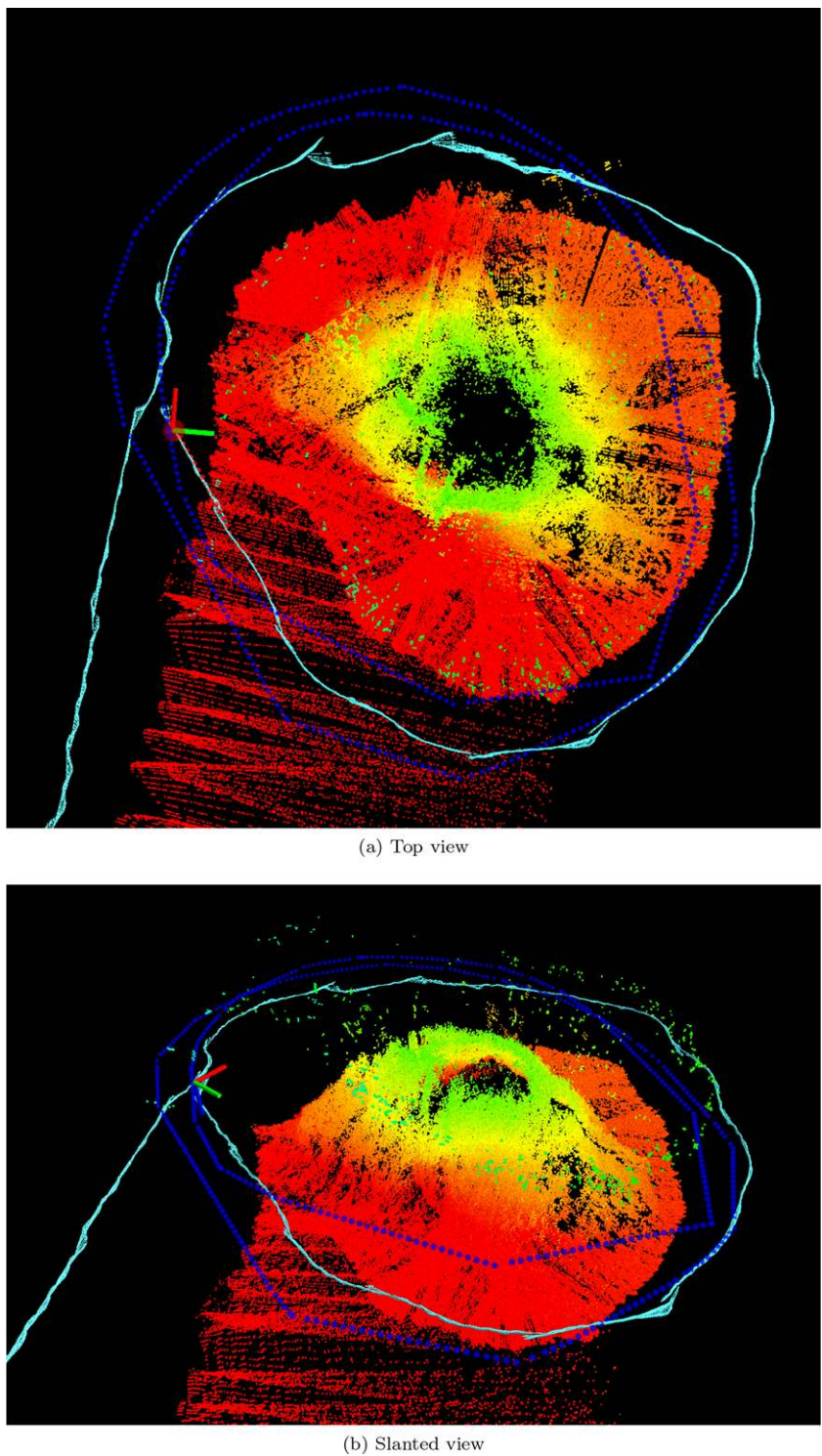
Finally, we used the onboard stereo camera to reconstruct a 3D model of the site using just optical data. As visible in Figure 37(a), due to low visibility conditions brought about by water turbidity, we are only able to obtain a partial reconstruction of the structure. This produces a less complete point set model than its range-based counterpart. Additionally, for the same reason, the point cloud suffers from

large defects in the form of noise and outliers caused by accumulating cascading errors in both the structure from motion and the dense point set sampling stages. Nevertheless, as shown in Figures 37(b) and 37(c), the surface reconstruction pipeline retrieved an acceptable final model (again, using  $k = 200$  when computing the normals). Note that, regardless of the high levels of corruption for the input point set, the front face of the block is reconstructed with more detail than in the model obtained using range data, due to the larger density of samples of the optical point cloud.

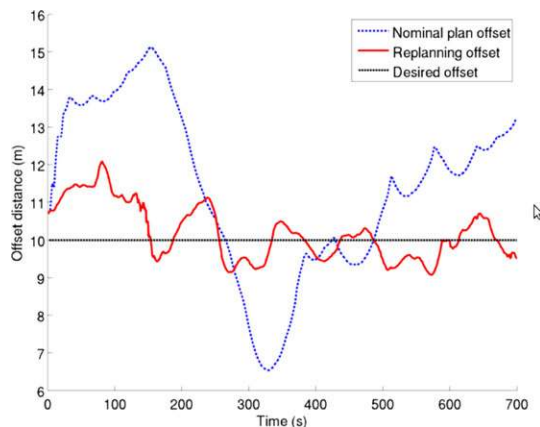
### 7.3.2. “L’Amarrador” Diving Site

Figure 38 shows the reconstructed surface from the raw range data in Figure 34, with normals computed with a neighborhood of  $k = 100$  points (note that  $k$  is smaller in this case due to a smaller sampling density when compared to the previously presented point sets). It is obvious that the point set shown in Figure 34 presents fewer outliers than the datasets discussed above, but on the contrary attains a larger amount of noise. Additionally, in this dataset





**Figure 34.** Coverage trajectory on “l’Amarrador” underwater boulder: nominal coverage path (blue-dotted line) and overall trajectory (cyan arrows). The current pose of the vehicle is represented by the red-green-blue 3D axis. The depth-colored range data acquired by the multibeam sonar are also displayed.



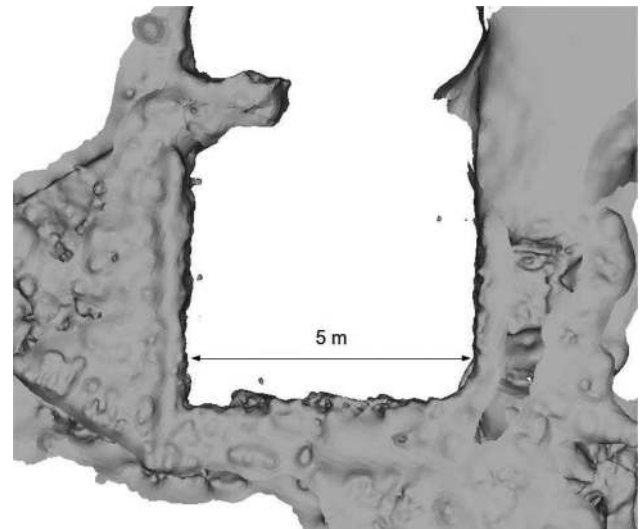
**Figure 35.** Offset distance achieved along the replanned trajectory compared to the nominal plan in the underwater boulder scenario.

the effect of drifting in the global navigation results in evident registration errors in the final point set, which greatly complicates the extraction of a global coherent surface in loop-closing areas.

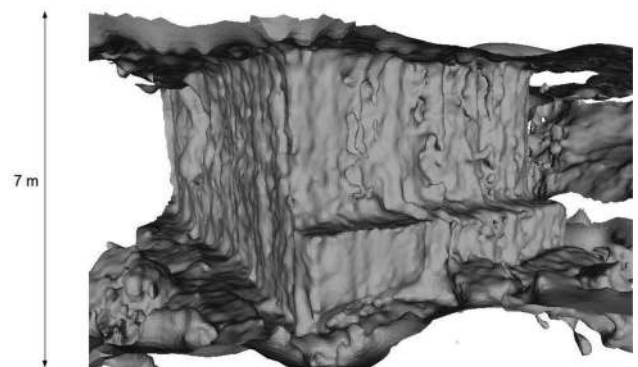
Nonetheless, the overall reconstructed surface provides a detailed representation of the shape of the surveyed underwater boulder, increasing by far the resolution from the prior bathymetry in Figure 27. We note, however, that undersampled parts are overly extrapolated (in particular at the peak of the boulder), and noise and registration errors create some small artifacts, depicted as unrealistic off-surface blobs. In the top part of Figure 38, we can see that this area of the object, presenting a larger amount of registration error, is also presenting the worst results, in the form of a spiky surface. In this regard, applying a global postprocessing optimization method to the dataset, merging registration information from a larger set of sensors would result in a better estimation of the global trajectory followed by the vehicle. Consequently, a better positioning of the individual range scans in the global frame would correspond to the generation of cleaner and more precise point sets, posing fewer difficulties to the surface reconstruction process.

#### 7.4. Comparison with a Standard Boustrophedon Survey

Next, we provide a qualitative comparison of the coverage method proposed in this paper with a traditional boustrophedon survey. The boustrophedon survey was conducted autonomously by the GIRONA 500 AUV navigating at 18 m depth, keeping a safe distance from the boulder's summit at 27 m. Here we used a down-looking multibeam sonar configuration, with a sufficiently short interlap spacing to provide overlap among the sonar swaths. We note that, if operating in close proximity as required for inspection



(a) Top view



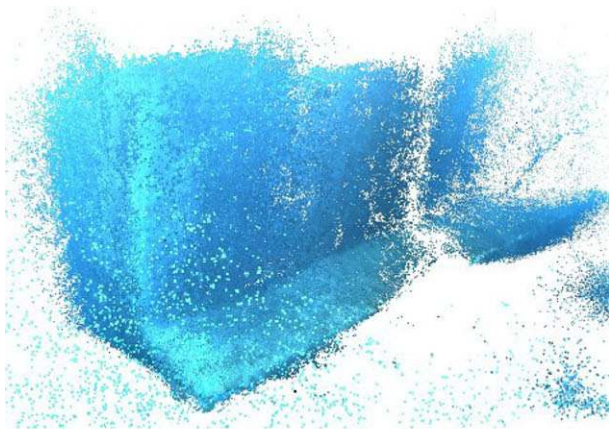
(b) Slanted view

**Figure 36.** Surface reconstruction of the concrete block from range data (scales are approximate).

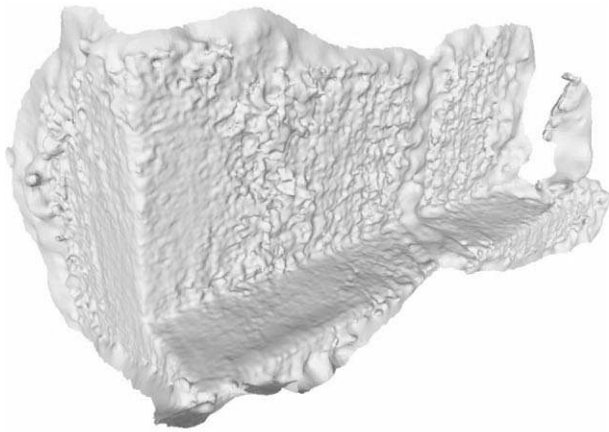
tasks, the boulder present in this site poses a threat to traditional bottom-keeping approaches based on single-range echosounders. Figure 39 shows the vehicle trajectory and the point cloud gathered during the standard survey. Remarkably, due to the askew angle of incidence of the sonar beams, the most vertical face of the boulder presents a significant hole with no bathymetric points. Likewise, vertical protrusions on the surface of the boulder present less point density than in the slicing-based coverage survey presented above.

As a result, when applying the 3D surface reconstruction pipeline to this dataset (Figure 40), we obtain a considerably less detailed reconstruction. This is a direct result of the sparsity of the data in the vertical faces of the boulder provided by the boustrophedon survey, which leads to a rougher estimation of the surface normals (a key step in the

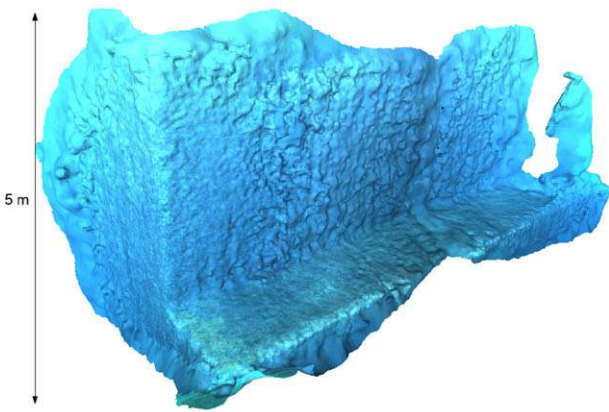




(a)

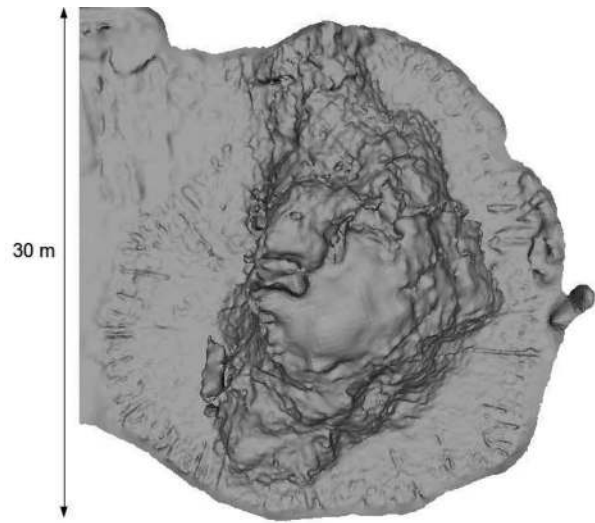


(b)

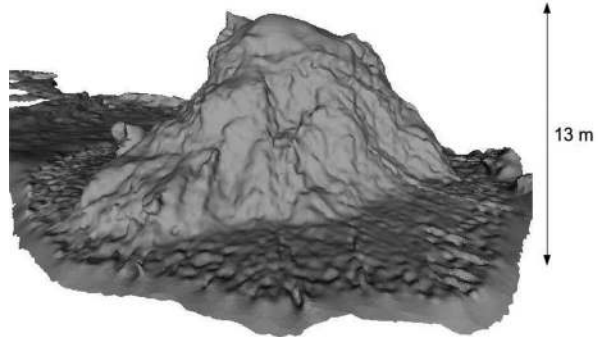


(c)

**Figure 37.** Slanted views of the optical reconstruction on the concrete block dataset. (a) Dense point set with per-vertex texture mapping, (b) surface, (c) textured surface (scale is approximate).

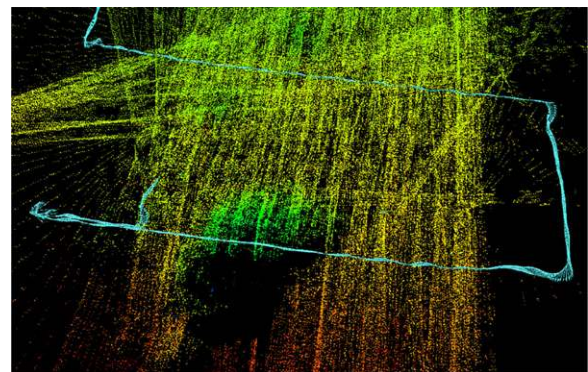


(a) Top view

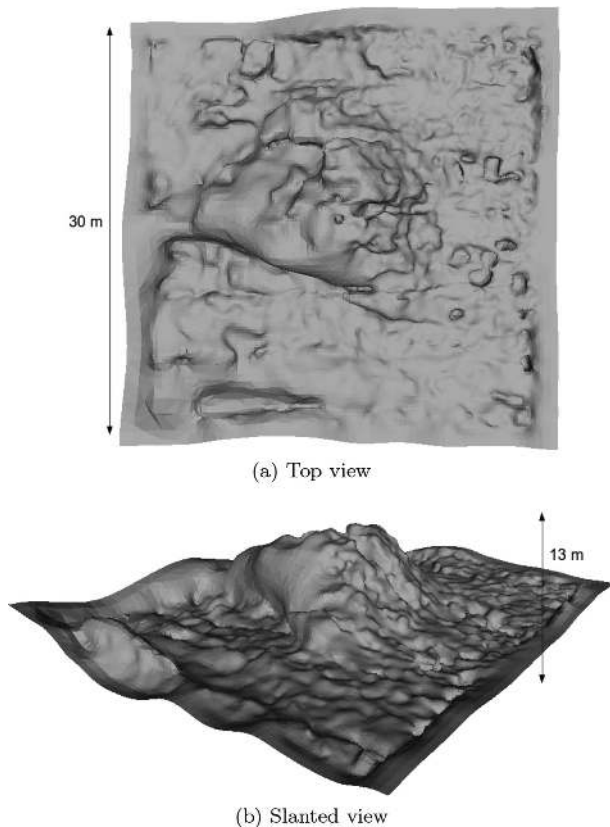


(b) Slanted view

**Figure 38.** Surface of “l’Amarrador” underwater boulder, recovered from the raw range data (scales are approximate).



**Figure 39.** Section of the vehicle trajectory and bathymetric point cloud gathered in a standard boustrophedon survey at “l’Amarrador” site. Note the lack of data in the boulder’s most vertical face, in the bottom-center part of the figure.



**Figure 40.** Surface of “l’Amarrador” underwater boulder, recovered from the standard boustrophedon survey raw range data (scales are approximate).

reconstruction process). For this particular case, we empirically found a neighborhood of  $k = 50$  points in computing the normals to produce the best results for this dataset. Nonetheless, the result resembles a “smoothed” version of that in Figure 38, where the finer-detail features of the vertical faces of the boulder are not captured. Conversely, due to the down-looking sensor configuration, the boustrophedon survey images with more density the summit of the boulder, leading to a less extrapolated reconstruction in that part of the surface. However, this issue could be mitigated by adjusting the slope threshold in the terrain segmentation phase to consider the boulder’s summit as an effectively planar region.

## 8. CONCLUSION

In this paper, we have presented a 3D coverage path-planning method for inspection of complex underwater structures. The method first plans a nominal coverage path on a prior map of the target structure, covering both 2D and 3D regions. By classifying the target region into effectively planar and high-slope regions, coverage paths suited

for each type of regions are generated. In effectively planar regions, our method favors a traditional mowing-the-lawn pattern, while in high-slope regions the planned coverage path follows contours of the target surface at a given offset distance. As a result, our method provides a clear and continuous data product by sequentially covering horizontal slices of the target surface along the vertical dimension of the workspace. This contrasts with state-of-the-art sampling-based 3D coverage path-planning algorithms, whose paths spread randomly in all dimensions of the workspace, making the vehicle’s maneuvers hard to predict from an operator’s standpoint and resulting in a less structured data flow for postprocessing tasks.

Rather than assuming a perfect execution of the nominal path, we use a replanning algorithm based on stochastic trajectory optimization to reshape the nominal path in real time during the mission. Provided a bound for the vehicle’s position error, the replanning algorithm is able to adapt the nominal path according to range sensor measurements onboard the vehicle. Our method has proven successful in inspection tasks in two different scenarios, involving coverage of a part of a breakwater structure and of an underwater boulder rising from 40 m up to 27 m depth. Moreover, we have presented surface reconstruction pipelines to obtain 3D models of the inspected sites that show the benefits of our coverage path-planning method for 3D mapping of complex structures, not amenable to standard mowing-the-lawn surveys.

Nonetheless, one aspect of the proposed approach that could be improved is the binary threshold used for terrain classification. In certain areas, it might be hard to choose an appropriate threshold, since regions with a slope close to the threshold could potentially be addressed by a boustrophedon survey even if they have been classified as high-slope regions, and vice versa. Perhaps using a segmentation approach like that proposed in Atkar et al. (2009) to segment the target terrain into regions bounded by high-curvature edges could result in more meaningful regions for which it is more clear what coverage strategy to use. However, the method proposed by Atkar et al. (2009) exploited high curvature features present in automotive surfaces, such as weld lines, which have no clear counterpart in underwater environments. Exploring possible algorithmic combinations of the two proposed coverage patterns is an interesting subject for further research. In addition, the nominal coverage path-planning algorithm we propose could benefit from advances in the area of motion planning under uncertainty (such as those surveyed in Section 2), leading to paths that result in less localization error or that maximize the information gain during the mission, for example.

We are currently working on further testing of the proposed method in other challenging sites of interest and using different sensor configurations such as side-scan sonar. In this paper, we have provided qualitative evaluations of the 3D reconstruction performance, since due to



uncertainty in the robot's position and noise and troublesome second returns from the multibeam sonar it is difficult to obtain a ground truth model of the surface. Thus, finding a method to obtain such ground truth would help quantify the performance of 3D mapping algorithms in underwater environments.

In the future, we plan to incorporate map uncertainty in the cost function to obtain safer trajectories with smaller probability of collision. Finally, splitting the workload of the coverage tasks among multiple AUVs would provide shorter mission times, for instance by assigning a piece of the path to each robot. However, multiple robots could do more, such as use each other as beacons for more robust localization and mapping. This multirobot extension is an interesting open problem for further research.

## ACKNOWLEDGMENTS

This research has been sponsored by the Government of Spain (COMAROB Project, DPI2011-27977-C03-02), the MORPH EU FP7-Project (grant agreement FP7-ICT-2011-7-288704), and the Eurofleets2 EU FP7-Project (grant agreement FP7-INF-2012-312762). The authors are grateful to Lluís Magí, Carles Candela, and Arnau Carrera for helping with the GIRONA 500 operations.

## REFERENCES

- Acar, E., & Choset, H. (2002a). Sensor-based coverage of unknown environments: Incremental construction of morse decompositions. *International Journal of Robotics Research*, 21(4), 345–366.
- Acar, E. U., & Choset, H. (2002b). Exploiting critical points to reduce positioning error for sensor-based navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation* (vol. 4, pp. 3831–3837).
- Acar, E. U., Choset, H., & Lee, J. Y. (2006). Sensor-based coverage with extended range detectors. *IEEE Transactions on Robotics*, 22(1), 189–198.
- Acar, E. U., Choset, H., Rizzi, A. A., Atkar, P. N., & Hull, D. (2002). Morse decompositions for coverage tasks. *International Journal of Robotics Research*, 21(4), 331–344.
- Acar, E. U., Choset, H., Zhang, Y., & Schervish, M. (2003). Path planning for robotic demining: Robust sensor-based coverage of unstructured environments and probabilistic methods. *International Journal of Robotics Research*, 22(7-8), 441–466.
- Ahmadzadeh, A., Keller, J., Jadbabaie, A., & Kumar, V. (2006). An optimization-based approach to time critical cooperative surveillance and coverage with unmanned aerial vehicles. In *International Symposium on Experimental Robotics*.
- Alterovitz, R., Simeon, T., & Goldberg, K. (2007). The stochastic motion roadmap: A sampling framework for planning with Markov motion uncertainty. In *Robotics: Science and Systems III* (Proceedings RSS 2007).
- Atkar, P., Conner, D., Greenfield, A., Choset, H., & Rizzi, A. (2009). Hierarchical segmentation of piecewise pseudoextruded surfaces for uniform coverage. *IEEE Transactions on Automation Science and Engineering*, 6(1), 107–120.
- Atkar, P., Greenfield, A. L., Conner, D. C., Choset, H., & Rizzi, A. (2005). Uniform coverage of automotive surface patches. *The International Journal of Robotics Research*, 24(11), 883–898.
- Atkar, P. N., Choset, H., Rizzi, A. A., & Acar, E. U. (2001). Exact cellular decomposition of closed orientable surfaces embedded in  $\mathbb{R}^3$ . In *Proceedings of the International Conference on Robotics and Automation* (vol. 1, pp. 699–704).
- Bai, H., Hsu, D., & Lee, W. S. (2014). Integrated perception and planning in the continuous space: A pomdp approach. *The International Journal of Robotics Research*, 33(9), 1288–1302.
- Barkby, S., Williams, S., Pizarro, O., & Jakuba, M. (2012). Bathymetric particle filter slam using trajectory maps. *The International Journal of Robotics Research*.
- Barrientos, A., Colorado, J., del Cerro, J., Martinez, A., Rossi, C., Sanz, D., & Valente, J. (2011). Aerial remote sensing in agriculture: A practical approach to area coverage and path planning for fleets of mini aerial robots. *Journal of Field Robotics*, 28(5), 667–689.
- Bingham, B., Foley, B., Singh, H., Camilli, R., Delaporta, K., Eustice, R., Mallios, A., Mindell, D., Roman, C. N., & 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.
- Blackmore, L., Ono, M., & Williams, B. (2011). Chance-constrained optimal path planning with obstacles. *IEEE Transactions on Robotics*, 27(6), 1080–1094.
- Boissonnat, J.-D. & Oudot, S. (2005). Provably good sampling and meshing of surfaces. *Graphical Models*, 67, 405–451.
- Bosse, M., Nourani-Vatani, N., & Roberts, J. (2007). Coverage algorithms for an under-actuated car-like vehicle in an uncertain environment. In *Proceedings of the IEEE International Robotics and Automation Conference* (pp. 698–703).
- Bretl, T., & Hutchinson, S. (2013). Robust coverage by a mobile robot of a planar workspace. In *Proceedings of the International Conference on Robotics and Automation*.
- Burgard, W., Fox, D., & Thrun, S. (1997). Active mobile robot localization by entropy minimization. In *Advanced Mobile Robots, 1997. Proceedings, Second EUROMICRO workshop* (pp. 155–162).
- Burns, B., & Brock, O. (2006). Sampling-based motion planning using uncertain knowledge. Technical report, University of Massachusetts at Amherst.
- Butler, Z. J., Rizzi, A. A., & Hollis, R. L. (1999). Contact sensor-based coverage of rectilinear environments. In *Proceedings of the IEEE International Intelligent Control/Intelligent Systems and Semiotics Symposium* (pp. 266–271).
- Candido, S., & Hutchinson, S. (2010). Minimum uncertainty robot path planning using a pomdp approach. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference* (pp. 1408–1413).

- Carrillo, H., Latif, Y., Neira, J., & Castellanos, J. (2012). Fast minimum uncertainty search on a graph map representation. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference* (pp. 2504–2511).
- Chaves, S. M., Kim, A., & Eustice, R. M. (2014). Opportunistic sampling-based planning for active visual slam. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*.
- Cheng, P., Keller, J., & Kumar, V. (2008). Time-optimal UAV trajectory planning for 3d urban structure coverage. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference* (pp. 2750–2757).
- Choset, H. (2000). Coverage of known spaces: The boustrophedon cellular decomposition. *Autonomous Robots*, 9(3), 247–253.
- Das, C., Becker, A., & Bretl, T. (2011). Probably approximately correct coverage for robots with uncertainty. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference* (pp. 1160–1166).
- Du Toit, N., & Burdick, J. (2012). Robot motion planning in dynamic, uncertain environments. *IEEE Transactions on Robotics*, 28(1), 101–115.
- Englot, B., & Hover, F. S. (2013). Three-dimensional coverage planning for an underwater inspection robot. *The International Journal of Robotics Research*, 32(9-10), 1048–1073.
- Escartin, J., Garcia, R., Delaunoy, O., Ferrer, J., Gracias, N., Elibol, A., Cufi, X., Neumann, L., Fornari, D. J., Humphris, S. E., & Renard, J. (2008). Globally aligned photomosaic of the lucky strike hydrothermal vent field (mid-atlantic ridge, 37 deg 18.5 min n): Release of georeferenced data, mosaic construction, and viewing software. *Geochemistry, Geophysics, Geosystems*, 9(12), n/a–n/a.
- Fairfield, N., & Wettergreen, D. (2008). Active localization on the ocean floor with multibeam sonar. In *OCEANS 2008* (pp. 1–10).
- Gabriely, Y., & Rimón, E. (2002). Spiral-stc: An on-line coverage algorithm of grid environments by a mobile robot. In *Proceedings of the IEEE International Conference on Robotics and Automation* (vol. 1, pp. 954–960).
- Galceran, E., Campos, R., Palomeras, N., Carreras, M., & Ridao, P. (2014). Coverage path planning with realtime replanning for inspection of 3d underwater structures. In *Proceedings of the International Conference on Robotics and Automation*.
- Galceran, E., & Carreras, M. (2012). Efficient seabed coverage path planning for ASVs and AUVs. In *Proceedings of the IEEE/RSJ International Intelligent Robots and Systems Conference*.
- Galceran, E., & Carreras, M. (2013a). Planning coverage paths on bathymetric maps for in-detail inspection of the ocean floor. In *Proceedings of the International Conference on Robotics and Automation*.
- Galceran, E., & Carreras, M. (2013b). A survey on coverage path planning for robotics. *Robotics and Autonomous Systems*, 62(12), 1258–1276.
- Galceran, E., Nagappa, S., Carreras, M., Ridao, P., & Palomer, A. (2013). Uncertainty-driven survey path planning for bathymetric mapping. In *Proceedings of the IEEE/RSJ International Intelligent Robots and Systems*.
- Gonzalez, J., & Stentz, A. (2009). Using linear landmarks for path planning with uncertainty in outdoor environments. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference* (pp. 1203–1210).
- Gracias, N., Rafael Garcia, P. R., Escartin, J., L Hour, M., Cibecchini, F., Campos, R., Carreras, M., Ribas, D., Palomeras, N., Magi, L., Palomer, A., Nicosevici, T., Prados, R., Hegedusz, R., Neumann, L., de Filippo, F., & Mallios, A. (2013). Mapping the moon: Using a lightweight auv to survey the site of the 17th century ship la lune. In *Proceedings of the IEEE OCEANS*.
- Guibas, L., Hsu, D., Kurniawati, H., & Rehman, E. (2008). Bounded uncertainty roadmaps for path planning. In *Proceedings of the Workshop on Algorithmic Foundations of Robotics*.
- Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems, Science, and Cybernetics*, SSC-4(2), 100–107.
- Hert, S., Tiwari, S., & Lumelsky, V. (1996). A terrain-covering algorithm for an auv. *Autonomous Robots*, 3, 91–119.
- Hollinger, G. A., Englot, B., Hover, F. S., Mitra, U., & Sukhatme, G. S. (2012a). Active planning for underwater inspection and the benefit of adaptivity. *The International Journal of Robotics Research*, 32(1), 3–18.
- Hollinger, G. A., Mitra, U., & Sukhatme, G. S. (2012b). Active and adaptive dive planning for dense bathymetric mapping. In *Proceedings of the International Symposium on Experimental Robotics* (pp. 803–817).
- Hollinger, G. A., & Sukhatme, G. S. (2014). Sampling-based robotic information gathering algorithms. *The International Journal of Robotics Research* 33(9), 1271–1287.
- Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., & Stuetzle, W. (1992). Surface reconstruction from unorganized points. *SIGGRAPH Computer Graphics*, 26(2), 71–78.
- Hornung, A., Wurm, K., Bennewitz, M., Stachniss, C., & Burgard, W. (2013). Octomap: An efficient probabilistic 3d mapping framework based on octrees. *Autonomous Robots*, 34(3), 189–206.
- Huang, W. H. (2001). Optimal line-sweep-based decompositions for coverage algorithms. In *Proceedings of the ICRA Robotics and Automation, IEEE International Conference* (vol. 1, pp. 27–32).
- Huang, Y., & Gupta, K. (2008). Rrt-slam for motion planning with motion and map uncertainty for robot exploration. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference* (pp. 1077–1082).
- Jimenez, P. A., Shirinzadeh, B., Nicholson, A., & Alici, G. (2007). Optimal area covering using genetic algorithms. In *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics* (pp. 1–5).
- Johnson-Roberson, M., Pizarro, O., Williams, S. B., & Mahon, I. (2010). Generation and visualization of large-scale three-dimensional reconstructions from underwater robotic surveys. *Journal of Field Robotics*, 27(1), 21–51.

- Kalakrishnan, M., Chitta, S., Theodorou, E., Pastor, P., & Schaal, S. (2011). Stomp: Stochastic trajectory optimization for motion planning. In *Robotics and Automation (ICRA)*, 2011 IEEE International Conference (pp. 4569–4574).
- Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME—Journal of Basic Engineering*, 82(Series D), 35–45.
- Kavraki, L., Svestka, P., Latombe, J.-C., & Overmars, M. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4), 566–580.
- Kazhdan, M., & Hoppe, H. (2013). Screened poisson surface reconstruction. *ACM Transactions Graphics*, 32(3), 29:1–29:13.
- Kewlani, G., Ishigami, G., & Iagnemma, K. (2009). Stochastic mobility-based path planning in uncertain environments. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference* (pp. 1183–1189).
- Kim, A., & Eustice, R. (2013). Perception-driven navigation: Active visual slam for robotic area coverage. In *Robotics and Automation (ICRA)*, 2013 IEEE International Conference (pp. 3196–3203).
- Kurniawati, H., Hsu, D., & Lee, W. (2008). Sarsop: Efficient point-based pomdp planning by approximating optimally reachable belief spaces. In *Proceedings of the Robotics: Science and Systems*.
- Lambert, A., & Le Fort-Piat, N. (2000). Safe task planning integrating uncertainties and local maps federations. *The International Journal of Robotics Research*, 19(6), 597–611.
- LaValle, S. M., & Kuffner, J. J. (2000). Rapidly-exploring random trees: Progress and prospects. In *Proceedings Workshop on the Algorithmic Foundations of Robotics*.
- Lee, T.-S., Choi, J.-S., Lee, J.-H., & Lee, B.-H. (2009). 3-d terrain covering and map building algorithm for an auv. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 4420–4425).
- Liu, S., & Wang, C. (2011). Fast intersection-free offset surface generation from freeform models with triangular meshes. *IEEE Transactions on Automation Science and Engineering*, 8(2), 347–360.
- Luo, C., & Yang, S. (2008). A bioinspired neural network for real-time concurrent map building and complete coverage robot navigation in unknown environments. *IEEE Transactions on Neural Networks*, 19(7), 1279–1298.
- Mannadiar, R., & Rekleitis, I. (2010). Optimal coverage of a known arbitrary environment. In *Proceedings of the IEEE International Robotics and Automation (ICRA) Conference* (pp. 5525–5530).
- Maza, I., & Ollero, A. (2007). Distributed autonomous robotic systems 6, Multiple UAV cooperative searching operation using polygon area decomposition and efficient coverage algorithms (pp. 221–230). Springer.
- MBARI (2013). Mb-system: Software for the processing and display of swath sonar data. <http://www.mbari.org/data/mbsystem/> [online; accessed 08-March-2013].
- Meduna, D., Rock, S., & McEwen, R. (2008). Low-cost terrain relative navigation for long-range auvs. In *OCEANS 2008* (pp. 1–7).
- Melchior, N., & Simmons, R. (2007). Particle rrt for path planning with uncertainty. In *Robotics and Automation, 2007 IEEE International Conference* (pp. 1617–1624).
- Missiuro, P., & Roy, N. (2006). Adapting probabilistic roadmaps to handle uncertain maps. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference* (pp. 1261–1267).
- Nakhai, A., & Lamiroux, F. (2008). A framework for planning motions in stochastic maps. In *Control, Automation, Robotics and Vision, 2008. ICARCV 2008. 10th International Conference* (pp. 1959–1964).
- Nicosevici, T., Gracias, N., Negahdaripour, S., & Garcia, R. (2009). Efficient three-dimensional scene modeling and mosaicing. *Journal of Field Robotics*, 26, 759–788.
- Oksanen, T., & Visala, A. (2009). Coverage path planning algorithms for agricultural field machines. *Journal of Field Robotics*, 26(8), 651–668.
- Palomeras, N., El-Fakdi, A., Carreras, M., & Ridao, P. (2012). Cola2: A control architecture for auvs. *IEEE Journal of Oceanic Engineering*, 37(4), 695–716.
- Papadopoulos, G., Kurniawati, H., & Patrikalakis, N. M. (2013). Asymptotically optimal inspection planning using systems with differential constraints. In *Proceedings of the International Conference on Robotics and Automation*.
- Paull, L., SaeediGharahbolagh, S., Seto, M., & Li, H. (2012). Sensor driven online coverage planning for autonomous underwater vehicles. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference* (pp. 2875–2880).
- Paull, L., Seto, M., & Li, H. (2014). Area coverage planning that accounts for pose uncertainty with an auv seabed surveying application. In *Proceedings of the IEEE International Conference on Robotics and Automation*.
- Pepy, R., & Lambert, A. (2006). Safe path planning in an uncertain-configuration space using rrt. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference* (pp. 5376–5381).
- Platt, R., Tedrake, R., Kaelbling, L., & Lozano-perez, T. (2010). Belief space planning assuming maximum likelihood observations. In *Proceedings of the Robotics: Science and Systems*.
- Prentice, S., & Roy, N. (November/December 2009). The belief roadmap: Efficient planning in belief space by factoring the covariance. *The International Journal of Robotics Research*, 28(11-12), 1448–1465.
- Ribas, D., Palomeras, N., Ridao, P., Carreras, M., & Mallios, A. (2012). Girona 500 auv, from survey to intervention. *IEEE/ASME Transactions on Mechatronics*, 17(1), 46–53.
- ROS (2014). Robot operating system. <http://www.ros.org/> [online; accessed 15-June-2013].
- Roy, N., Burgard, W., Fox, D., & Thrun, S. (1999). Coastal navigation-mobile robot navigation with uncertainty in dynamic environments. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference* (vol. 1, pp. 35–40).
- Russell, S. J., & Norvig, P. (2003). *Artificial intelligence: A modern approach*. Pearson Education.

- Serra, J. (1982). Image analysis and mathematical morphology. Academic Press.
- Stack, J. R., & Smith, C. M. (2003). Combining random and data-driven coverage planning for underwater mine detection. In Proceedings OCEANS 2003 (vol. 5, pp. 2463–2468).
- Tivey, M. A., Bradley, A., Yoerger, D., Catanach, R., Duester, A., Liberatore, S., & Singh, H. (1997). Autonomous underwater vehicle maps seafloor. *Eos, Transactions American Geophysical Union*, 78(22), 229–230.
- Tully, S., Kantor, G., & Choset, H. (2010). Leap-frog path design for multi-robot cooperative localization. In Howard, A., Iagnemma, K., & Kelly, A. (eds.), *Field and service robotics*, volume 62 of Springer Tracts in Advanced Robotics (pp. 307–317). Berlin/Heidelberg: Springer.
- Valencia, R., Miro, J., Dissanayake, G., & Andrade-Cetto, J. (2012). Active pose slam. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference* (pp. 1885–1891).
- van den Berg, J., Abbeel, P., & Goldberg, K. (2011). Lqg-mp: Optimized path planning for robots with motion uncertainty and imperfect state information. *The International Journal of Robotics Research*, 30(7), 895–913.
- van den Berg, J., Patil, S., & Alterovitz, R. (2012). Motion planning under uncertainty using iterative local optimization in belief space. *The International Journal of Robotics Research*, 31(11), 1263–1278.
- Williams, S. B., Pizarro, O., Mahon, I., & Johnson-Roberson, M. (2009). Simultaneous localisation and mapping and dense stereoscopic seafloor reconstruction using an auv. In Khatib, O., Kumar, V., & Pappas, G. J. (eds.), *Experimental robotics*, volume 54 of Springer Tracts in Advanced Robotics (pp. 407–416). Berlin/Heidelberg: Springer.
- Wong, S. C. (2006). Qualitative topological coverage of unknown environments by mobile robots. Ph.D. thesis, The University of Auckland.
- Xu, A., Virie, P., & Rekleitis, I. (2011). Optimal complete terrain coverage using an unmanned aerial vehicle. In *Proceedings of the 2011 IEEE International Conference on Robotics & Automation*.
- Yang, R., & Pollefeys, M. (2003). Multi-resolution real-time stereo on commodity graphics hardware. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference* (vol. 1, pp. I-211–I-217).
- Yoerger, D. R., Kelley, D. S., & Delaney, J. R. (2000). Fine-scale three-dimensional mapping of a deep-sea hydrothermal vent site using the jason rovs system. *The International Journal of Robotics Research*, 19(11), 1000–1014.
- Zandara, S., Ridaou, P., Ribas, D., Campos, R., & Mallios, A. (2011). Kornati bathymetry survey data-set for navigation and mapping. In *Control automation (MED), 2011 19th Mediterranean Conference* (pp. 443–448).