2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Coverage Path Planning with Realtime Replanning for Inspection of 3D Underwater Structures

Enric Galceran, Ricard Campos, Narcís Palomeras, Marc Carreras and Pere Ridao

***Abstract*— We present a novel method for planning 3D coverage paths for inspection of complex structures on the ocean floor (such as seamounts or coral reefs) using an autonomous underwater vehicle (AUV). Our method initially uses an *a priori* map to plan a nominal coverage path that allows the AUV to pass its sensors over all points on the target structure. We then go beyond previous approaches in the literature by considering the vehicle's position uncertainty rather than relying on the unrealistic assumption of an idealized path execution. To this aim, we present a replanning algorithm based on 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 realtime during the inspection mission, adapting the path according to the measurements provided by the vehicle's range sensing sonars. We demonstrate the efficacy of our method in experiments at sea using the GIRONA 500 AUV where we cover a concrete block of a breakwater structure in a harbor and an underwater boulder rising from 40 m up to 27 m depth. Moreover, we apply state-of-the-art surface reconstruction techniques to the data acquired by the AUV and obtain 3D models of the inspected structures that show the benefits of our planning method for 3D mapping.**

I. INTRODUCTION

Many autonomous underwater vehicle (AUV) applications require covering a target area by passing a sensor (such as a camera or a sonar) over all the points in the area. At present, in most AUV survey missions the vehicle tackles this task by following a pre-planned lawnmower-like survey path while keeping a safe altitude from the sea bottom. This is a valid approach for seafloor areas that are effectively planar at the survey scale. However, flying at a conservative altitude, imposes serious limitations for a number of emerging applications requiring fine-scale seafloor surveys of rugged terrain in close proximity that allow 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 the AUV to maneuver in close proximity to the seabed without compromising vehicle safety are desired.

On the other hand, 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. Therefore, in order to meet these requirements, flying at a conservative distance from the seabed is no longer an option. The AUV must rather navigate amidst bulges sticking out of the bottom. Obviously, this increases the threat of collision.

Approaches to plan a coverage path accounting for obstacles have been proposed in the literature (see Sec. II). However, they typically assume perfect knowledge of the environment or perfect sensing, 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) or simultaneous localization and mapping (SLAM) for enhanced localization. This limits real-world application of those approaches to very constrained, controlled environments.

In this paper, we present a new 3D coverage path planning method for inspection of complex structures on the ocean floor using AUVs which does not rely on these assumptions. Our method initially plans a coverage path off-line using an *a priori* bathymetric map of the target structure (a bathymetric map is an elevation map of the ocean floor). 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. As a result, the path enables acquisition of a clear and continuous data product, simplifying the tasks of post-processing and analysis for both humans and automated procedures. Furthermore, unlike previous approaches, our method does not rely on an idealized execution of the path. To handle the vehicle's position uncertainty, we present a replanning algorithm based on stochastic trajectory optimization to adapt the initially planned coverage path in realtime using range sensor measurements. The resulting path is smooth and provides successful coverage under bounded position error. We validate our method in experiments at sea with the GIRONA 500, a reconfigurable AUV equipped with range and imaging sensors. Our experiments comprise coverage of a concrete block of a breakwater structure in a harbor and coverage of an underwater boulder rising from 40 m depth to 27 m depth. Results show that our method successfully achieves coverage of the target structures, adapting the planned paths in agreement with realtime perception on site. Additionally, we present 3D surface reconstructions generated from the data collected in the coverage tasks that

This research was sponsored by the Spanish government (COMAROB Project, DPI2011-27977-C03-02) and the MORPH EU FP7-Project under the Grant agreement FP7-ICT-2011-7-288704.

The authors are with the Underwater Robotics Research Center (CIRS), University of Girona, Girona, Spain. {enricgalceran,rcampos,marcc,pere}@eia.udg.edu

demonstrate the high quality of the data products enabled by our coverage path planning method.

II. RELATED WORK

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 in 2D environments and applications have been reported in domains such as aerial robotics, agricultural robotics, mine countermeasures (MCM) operations and marine robotics (see [1] for a survey). Furthermore, 2D coverage algorithms for planning optimal paths [2] have been presented.

In contrast, very few papers have addressed coverage path planning in 3D environments. A coverage algorithm specifically targeted for spray-painting of automotive parts was presented in [3], which provides uniform paint deposition. Recently, Englot and Hover introduced a sampling-based algorithm to achieve coverage of complex 3D structures for ship hull inspection [4]. Also building upon the idea of sampling-based planning, the algorithm in [5] provides coverage for inspection of complex structures using systems with differential constraints. While these later 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, providing no control on the order in which the points of the target surface are covered.

In our previous work [6], we presented an off-line coverage path planning method for seafloor regions including high-relief terrain. The off-line planning phase presented below in this paper is based on an algorithm in [6].

III. OFF-LINE COVERAGE PATH PLANNING

The first phase of our coverage method consists in planning a nominal coverage path off-line using prior knowledge of the target environment. We build upon our previous work [6] to plan a coverage path for our target surface. While in our previous work we presented a method for covering both planar and high-relief regions, here we focus on planning coverage paths for high-relief regions only. Our off-line coverage path planning algorithm takes as input an *a priori* bathymetric map of the target region, $\mathcal{B} : \mathbb{R}^2 \rightarrow \mathbb{R}$ (given a point (x, y) on target region, $\mathcal{B}(x, y)$ returns its depth). Then, the algorithm plans a coverage path by intersecting a horizontal plane with $\mathcal{B}$ at uniformly spaced depths. The inter-plane spacing is determined by the robot's sensor footprint. These intersections are finally offset by a desired distance Ω from the target surface and linked together to form a coverage path. We refer the reader to [6] for further details on the algorithm. Our off-line algorithm works in a 2.5D environment (a bathymetric map), but the replanning procedure discussed below permits to adapt the path to the shape of an actual 3D environment, providing a full 3D perception of the target structure.

IV. REALTIME REPLANNING

Once a nominal coverage path has been planned off-line, we propose an iterative replanning method to adapt the path in realtime according to range sensor information. 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 realtime using range data. We use the Octomap [7] probabilistic mapping framework, which is based on an octree map compression method that keeps the 3D model compact and quickly accessible. At 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 a trajectory optimization algorithm that, given an appropriate cost function, produces a smooth trajectory 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 trajectory optimization algorithm (IV-A) and the cost function we use in the optimization process (IV-B). Then, building upon those two elements, we detail our coverage path replanning algorithm (IV-C).

A. Stochastic Trajectory Optimization

We use the Stochastic Trajectory Optimization for Motion Planning (STOMP) algorithm [8] to reshape the nominal coverage path so it adapts to the actual target structure perceived on site 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. STOMP optimizes a cost function based on a combination of smoothness and application-specific costs, such as constraints, obstacles or motor torques.

STOMP considers trajectories of a fixed duration, T, discretized into N waypoints equally spaced in time. For simplicity in the notation, we describe the algorithm for a 1-dimensional trajectory in Algorithm 1; this extends naturally to multiple dimensions. This 1-dimensional trajectory is represented as a vector $\boldsymbol{\theta} \in \mathbb{R}^N$. 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 IV-B for our case).

STOMP represents smoothness costs as a positive semi-definite matrix such that $\boldsymbol{\theta}^\top \boldsymbol{R}\boldsymbol{\theta}$ is the sum of squared accelerations along the trajectory. The accelerations are obtained by means of a finite difference matrix that when multiplied by the position vector $\boldsymbol{\theta}$, produces accelerations $\ddot{\boldsymbol{\theta}}$:

$$\ddot{\boldsymbol{\theta}} = \boldsymbol{A}\boldsymbol{\theta}, \tag{1}$$

$$\ddot{\boldsymbol{\theta}}^\top \ddot{\boldsymbol{\theta}} = \boldsymbol{\theta}^\top (\boldsymbol{A}^\top \boldsymbol{A})\boldsymbol{\theta}. \tag{2}$$

Selecting $\boldsymbol{R} = \boldsymbol{A}^\top \boldsymbol{A}$ ensures that $\boldsymbol{\theta}^\top \boldsymbol{R}\boldsymbol{\theta}$ respresents the sum of squared accelerations along the trajectory. In each iteration of Algorithm 1, first a set of K noisy trajectories is generated by sampling the noise from a zero mean normal distribution with $\boldsymbol{R}^{-1}$ as its covariance matrix (line 2). This keeps the generated trajectories smooth and does not allow them to diverge from the start or goal. For each trajectory, its cost per time-step $S(\tilde{\boldsymbol{\theta}}_{k,i})$ is computed (line 4). Based on this cost, a probability $P(\tilde{\boldsymbol{\theta}}_{k,i})$ is assigned to each trajectory, per time-step (line 5). The parameter λ regulates the sensitivity of the exponentiated cost, and is optimized to maximally discriminate between the experienced costs by computing the exponential term in line 5 as:

$$e^{-\frac{1}{\lambda} S(\tilde{\boldsymbol{\theta}}_{k,i})} = e^{-h \frac{S(\tilde{\boldsymbol{\theta}}_{k,i}) - \min S(\tilde{\boldsymbol{\theta}}_{k,i})}{\max S(\tilde{\boldsymbol{\theta}}_{k,i}) - \min S(\tilde{\boldsymbol{\theta}}_{k,i})}}, \tag{3}$$

with $h = 10$ as suggested in [8]. The update for each time-step is computed in line 7 as the probability-weighted combination of the noisy trajectories for that time-step. In line 8 the update is smoothed using the $\boldsymbol{M}$ matrix, which is formed by scaling each column of $\boldsymbol{R}^{-1}$ such that the largest element in the column has magnitude $1/N$. Multiplication by $\boldsymbol{M}$ ensures that the updated trajectory remains smooth. Finally, the trajectory parameter vector is updated in line 9 and the cost for the updated trajectory is computed in line 10.

Algorithm 1: STOMP

Input:
- Start and goal positions x_0 and x_N.
- An initial 1-D discretized trajectory vector $\boldsymbol{\theta}$.
- A state-dependent cost function $q(\boldsymbol{\theta}_i)$.

Precompute:
- $\boldsymbol{A}$: second-order finite difference matrix (Eq. **??**).
- $\boldsymbol{R}^{-1} = (A^\top A)^{-1}$.
- $\boldsymbol{M} = \boldsymbol{R}^{-1}$, with each column scaled such that the maximum element is $1/N$.

1 **while not** *convergence of trajectory cost* $Q(\boldsymbol{\theta})$ **do**
2 Create K noisy trajectories, $\tilde{\boldsymbol{\theta}}_1, \ldots, \tilde{\boldsymbol{\theta}}_K$ with parameters $\boldsymbol{\theta} + \boldsymbol{\epsilon}_k$, where $\boldsymbol{\epsilon}_k \sim \mathcal{N}(0, \boldsymbol{R}^{-1})$
3 **for** $k = 1 \ldots K$ **do**
4 $S(\tilde{\boldsymbol{\theta}}_{k,i}) \leftarrow q(\tilde{\boldsymbol{\theta}}_{k,i})$
5 $P(\tilde{\boldsymbol{\theta}}_{k,i}) \leftarrow \frac{e^{-\frac{1}{\lambda} S(\tilde{\boldsymbol{\theta}}_{k,i})}}{\sum_{l=1}^{K} [e^{-\frac{1}{\lambda} S(\tilde{\boldsymbol{\theta}}_{l,i})}]}$
6 **for** $i = 1 \ldots (N-1)$ **do**
7 $[\delta\tilde{\boldsymbol{\theta}}]_i \leftarrow \sum_{k=1}^{K} P(\tilde{\boldsymbol{\theta}}_{k,i}) [\boldsymbol{\epsilon}_k]_i$
8 $\delta\boldsymbol{\theta} \leftarrow \boldsymbol{M}\delta\tilde{\boldsymbol{\theta}}$
9 $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \delta\boldsymbol{\theta}$
10 Update trajectory cost $Q(\boldsymbol{\theta}) = \sum_{i=1}^{N} q(\boldsymbol{\theta}_i) + \frac{1}{2}\boldsymbol{\theta}^\top \boldsymbol{R}\boldsymbol{\theta}$
11 **return** $\boldsymbol{\theta}$

B. Cost Function

Our cost function seeks to keep all the points in the optimized trajectory at the desired offset distance Ω 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 distance is given by the following function:

$$d(x, S) = \min_{s_i \in S} ||x - s_i|| \,. \tag{4}$$

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

$$q(\boldsymbol{\theta}) = \sum_{t=0}^{T} |d(\boldsymbol{\theta}_t, S) - \Omega|, \tag{5}$$

where $d(\boldsymbol{\theta}_t, S)$ is calculated according to the current map. Recall that the additional smoothness cost $\boldsymbol{\theta}^\top \boldsymbol{R}\boldsymbol{\theta}$ is already incorporated in Algorithm 1.

C. Realtime Coverage Path Replanning Algorithm

We propose an iterative realtime 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 Ω from the target structure. As previously mentioned, our replanning algorithm assumes that the nominal coverage path does not intersect the actual target surface under the robot's position error. 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 executes the path assuming the environment's map representation does not change for the duration of the path execution. Therefore, R can be chosen to regulate the length of the replanning "steps" so newly incorporated range measurements providing a more up-to-date map can be taken into account early in the next iteration.

Our realtime coverage path replanning algorithm is detailed in Algorithm 2. 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 Ω 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 procedure is illustrated in Fig. 1.

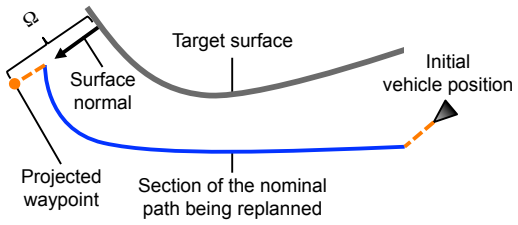


Fig. 1: Illustration of the initial trajectory construction in a replanning step (top view). The initial path will consist of both the slice of the nominal path (blue) and the added segments (orange).

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. 5. Finally, the optimized trajectory is executed (line 11) and the process repeats until the end of the nominal path is reached.

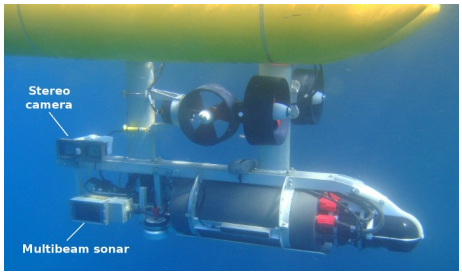
Algorithm 2: Realtime Coverage Path Re-planning

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 ← 0
3  while i < K do
4      x ← GetRobotPosition()
5      pathSection ← ∅
6      while Distance(x, w_i) < R and i < K do
7          pathSection.append(w_i)
8          i ← i + 1
9      θ ← InitialTrajectory(pathSection, x)
10     optimizedTrajectory ← STOMP(θ, M)
11     Execute(optimizedTrajectory)
```

V. EXPERIMENTAL PLATFORM: THE GIRONA 500 AUV

We tested our method using the GIRONA 500 AUV, a reconfigurable vehicle rated for depths up to 500 m shown in Fig. 2. GIRONA 500 is a hovering-capable AUV actuated in 4 degrees of freedom (DOFs): surge, sway, heave and yaw. The vehicle is approximately 1 m by 1 m by 1.5 m length, weighing less than 200 Kg. The vehicle is equipped with a navigation sensor suite including a pressure sensor, a Doppler velocity log (DVL), an inertial measurement unit (IMU) and a GPS to receive fixes while at the surface.A horizontally-scanning SeaKing pencil-beam sonar by Tritech is used to perceive the target structure and construct the map on-line. For our coverage experiments, GIRONA 500 additionally mounts two side-looking sensors in its payload volume: a Bumblebee 2 stereo camera by Point Grey and a Delta T multibeam bathymetry sonar by Imagenex. The Bumblebee 2 stereo camera features a 65-degree field of view and the multibeam sonar features a 120-degree swath aperture.



(a)

Fig. 2: The GIRONA 500 AUV equipped with side-looking multibeam sonar and stereo camera.

VI. RESULTS

Our replanning algorithm has been implemented in Python and integrated with the rest of GIRONA 500's software architecture.The implementation produces an optimized path at each replanning step in less than one second. We have tested our method by performing two coverage tasks with the GIRONA 500 AUV*. In the first task we cover a concrete block of a breakwater structure in a harbor. In the second task, we cover a 13 m high underwater boulder. In both experiments we set the maximum range of the pencil-beam sonar at 20 m and we set the replanning range at $R = 5$ m. Likewise, in both experiments, we estimate to be dealing with a maximum The *a priori* bathymetric charts we use to plan the nominal coverage paths were created by members of our team using a Delta T multibeam sonar.

We validate the benefits of our method for 3D mapping using both surface reconstruction techniques. Our 3D mapping results show how the paths planned with our method are useful in mapping complex 3D structures, not amenable for traditional 2.5D mapping of marine environments. The unorganized range data collected by the multibeam sonar promotes the use of 3D surface reconstruction techniques. More precisely, we apply the screened Poisson method [9] to recover a triangle mesh resembling the surface described by the range data. All the data products we show are the direct result of the 3D mapping techniques we use, without any manual tuning or refinement.

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

A. Coverage of a Concrete Block of a Breakwater Structure

The first coverage task in which we test our method serves as a minimal test of our implementation. The target structure is a concrete block of a breakwater structure composed of twenty of such blocks. Each block's footprint is approximately 5 m $\times$ 5 m, spanning from 2 m above the surface down to the sea bottom at 10 m depth. This structure is located in the harbor of Sant Feliu de Guixols, in the Costa Brava in Catalonia, Spain. It is located next to its main pier and provides protection from the effects of bad weather and longshore drift. Fig. 3 shows the *a priori* bathymetric chart (overlapped on satellite imaging) we use to plan a nominal coverage path (also shown in Fig. 3) for this task using our off-line planning method. In this minimal validation experiment, we target the right-most 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. The nominal path resulting from the off-line planning phase is also shown in Fig. 4. Note that the path is not closed (it resembles a semi-circle) since there is not enough clearance between the concrete blocks for the vehicle to go through. Therefore, the path provides coverage on only three of the four vertical faces of the block.

Fig. 3: Bathymetric map of the harbor area overlapped on Google Earth and nominal path (in red).

The trajectory followed by the robot during the realtime replanning phase is shown in Fig. 4 with the on-line map and the depth-colored raw range data acquired by the side-looking multibeam sonar. Although mitigated by the Octomap probabilistic mapping, it can be observed that the map includes many outliers, mainly due to surface reflections of the pencil-beam sonar beams. Nonetheless, the resulting trajectory provides full sensor coverage of the targeted in-water part of the structure. Along this coverage trajectory, and according to the on-line map, the AUV kept a mean distance to the target structure of 7.71 m, with a standard deviation of 2.03 m. The trajectory was executed in 498 s.

Fig. 5 shows the surface reconstructed from the raw range data. Note how the point cloud depicted in Fig. 4 is far from ideal, as it contains high levels of noise and outliers coupled with registration errors. Prior to reconstruction, and since the

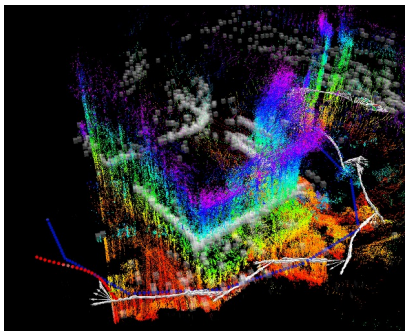
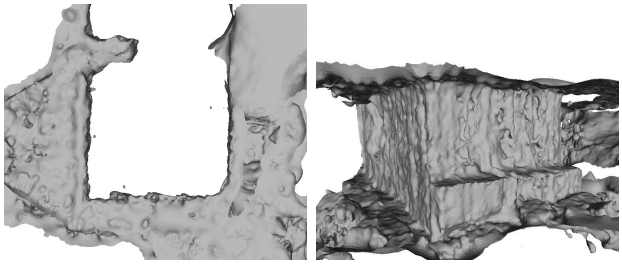


Fig. 4: Realtime replanning on the concrete block coverage experiment: nominal coverage path (blue-dotted line); optimized trajectory which the robot is executing at that particular instant (red-dotted line); overall trajectory (white arrows); occupied cells in the on-line map (white cubes). The depth-colored range data acquired by the multibeam sonar is also displayed.

method used [9] requires oriented point sets, we computed per-point normals with the method of Hoppe [10], using a neighborhood of $k = 200$ points. 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 non-existent artifacts to show up in the top- and bottom-most parts of the model and some undesirable roughness in its front wall.



(a) Top view (b) Slanted view

Fig. 5: Surface reconstruction of the concrete block.

B. Coverage of "l'Amarrador" Underwater Boulder

Our second coverage task targets "l'Amarrador" underwater boulder, which rises from 40 m up to 27 m depth. It is located approximately 1 Km off the harbor of Sant Feliu de Guixols. The nominal coverage path for this site is shown in Fig. 6. The plan consists of 2 contours spaced 2 m apart in the vertical axis. This spacing provides redundant coverage, which is of interest to us to obtain a dataset for testing SLAM and 3D reconstruction algorithms. There are two important factors to take into account when choosing an offset distance to plan this task. On one hand, this site is in an open sea environment and there exist a threat of strong currents. On the other hand, the mission is significantly longer, incurring a potentially bigger 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 boulder. Therefore, only the sonar range data is of interest in this experiment.

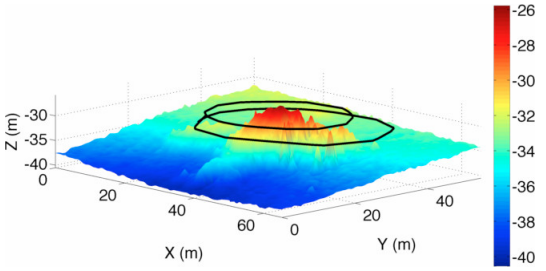
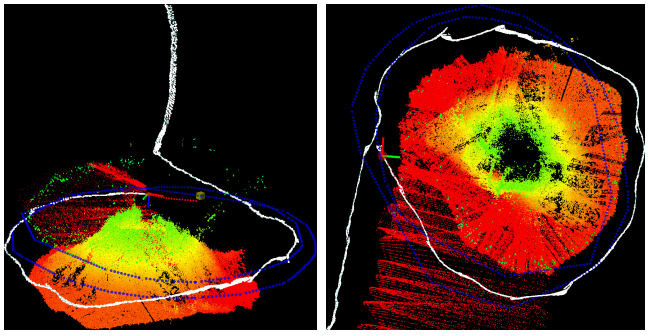


Fig. 6: Nominal coverage path for "l'Amarrador" underwater boulder.

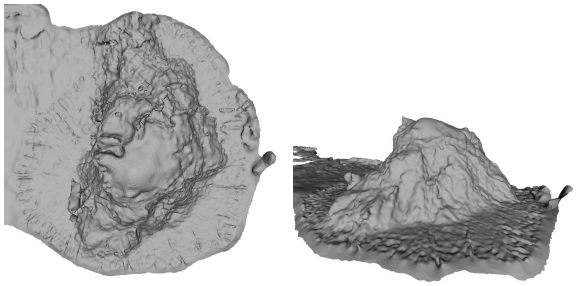
Fig. 7 shows an instant of the realtime replanning phase together with the nominal path, which GIRONA 500 is reshaping so it agrees with the perceived sonar range data of the boulder. Along the overall coverage trajectory in this experiment, and according to the on-line map, the AUV kept a mean distance to the target structure of 9.41 m, with a standard deviation of 0.93 m. The vehicle needed 150 s to approach the initial waypoint and dive to the first contour depth, 36 s to approach the structure at the desired offset distance, and approximately 600 s to cover each contour, leading to a total mission time of 1386 s (about 23 minutes). The trajectory provides successful coverage of the boulder, as demonstrated in the resulting 3D maps below.



(a) Slanted view (b) Top view

Fig. 7: Realtime coverage trajectory at "l'Amarrador" boulder (legend as in Fig. 4).

Indeed, Fig. 8 shows the reconstructed surface from the raw range data in Fig. 7, with normals computed with a neighborhood of $k = 100$ points. The overall surface faithfully represents the shape of the surveyed underwater boulder, increasing by far the resolution from the off-line model in Fig. 6. However, we note that undersampled parts are overly extrapolated (in particular at the top of the mount) and registration errors create some small artifacts.



(a) Top view (b) Slanted view

Fig. 8: Surface of "l'Amarrador" boulder, recovered from the raw range data.

VII. CONCLUSION AND FURTHER WORK

We have proposed a 3D coverage path planning method for inspection of complex underwater structures. The method first plans a nominal coverage path off-line on a bathymetric map of the target structure. The off-line follows contours of the target structure at a given offset distance. A replanning algorithm is used to cope with state uncertainty by reshaping the nominal path in realtime during the mission according to range sensor measurements. Our method has proven successful in two coverage experiments, involving coverage of a part of a breakwater structure and of an underwater boulder. Moreover, we have presented 3D models of the inspected sites that show the benefits of our method for 3D mapping.

We are currently working in further testing of the proposed method in other challenging sites and using different sensor configurations. Finally, splitting the workload of the coverage tasks among multiple AUVs is also an interesting avenue for future work.

REFERENCES

[1] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robotics and Autonomous Systems*, vol. 62, no. 12, pp. 1258–1276, December 2013.

[2] R. Mannadiar and I. Rekleitis, "Optimal coverage of a known arbitrary environment," in *Proc. ICRA*, 2010, pp. 5525–5530.

[3] P. Atkar, A. L. Greenfield, D. C. Conner, H. Choset, and A. Rizzi, "Uniform coverage of automotive surface patches," *The Int. Journal of Robotics Research*, vol. 24, no. 11, pp. 883 – 898, 2005.

[4] B. Englot and F. Hover, "Sampling-based coverage path planning for inspection of complex structures," in *Proc. ICAPS*, 2012.

[5] G. Papadopoulos, H. Kurniawati, and N. M. Patrikalakis, "Asymptotically optimal inspection planning using systems with differential constraints," in *Proc. International Conference on Robotics and Automation*, 2013.

[6] E. Galceran and M. Carreras, "Planning coverage paths on bathymetric maps for in-detail inspection of the ocean floor," in *Proc. ICRA*, 2013.

[7] A. Hornung, K. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: an efficient probabilistic 3d mapping framework based on octrees," *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, 2013.

[8] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal, "Stomp: Stochastic trajectory optimization for motion planning," in *Proc. ICRA*, 2011, pp. 4569–4574.

[9] M. Kazhdan and H. Hoppe, "Screened poisson surface reconstruction," *ACM Trans. Graph.*, vol. 32, no. 3, pp. 29:1–29:13, July 2013.

[10] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points," *SIGGRAPH Comput. Graph.*, vol. 26, no. 2, pp. 71–78, July 1992.