Abstract—A framework for generating coverage paths for marine habitat mapping is proposed in this paper. The framework combines two existing coverage path planning algorithms with new ideas to provide automated, efficient survey paths that take into account the particularities of the application. On one hand, a recent algorithm especially targeted for marine environments is used to generate a survey path of a previously unmapped area. The method has the advantage of minimizing repeated coverage when using a surface vehicle or while surveying at constant depth with an underwater vehicle. On the other hand, only the regions where the marine habitat is present (which often come in the form of widespread “blobs”) need to be surveyed in future monitoring missions in the area. However, due to the changing nature of the marine habitats, determining the exact extent of those regions prior to mission is not possible. Rather than surveying the whole area anew, we propose to use a sensor-based planner that, given their approximate locations from a previous survey, covers the regions of interest (ROIs) on-line using acoustic or optical camera information. Additional procedures to generate a path that visits all the ROIs are provided. The approach is tested in simulation using a real-world bathymetric dataset and synthetic ROIs. Results show the feasibility of the proposed approach.

I. INTRODUCTION

In marine habitat monitoring, it is of interest to compare maps of a habitat site constructed at two distinct points in time [1]. This allows for assessing the proliferation or decline rate of the habitat. Typically, divers carry this task out in-situ nowadays. On the other hand, recent advances in seabed-mapping technology [2], especially in automated image mosaicing [3], [4], [5], opened the door to automate this task using an Autonomous Underwater Vehicle (AUV) or an Autonomous Surface Vehicle (ASV).

However, to fully automate the habitat mapping task, an automated method to plan a survey path that covers the target area is needed. Today, AUV and ASV survey paths are typically planned using vehicle vendor-provided software whereby a human user draws a sequence of points on a map, such as a bathymetry map or an electronic nautical chart [6]. Then, the vehicle follows the sequence by tracing straight-line segments from one waypoint to the next. This approach is clearly inconvenient, as it requires a considerable amount of human intervention to program the path. Therefore, the idea of a more automated, more flexible way of specifying those missions is attractive.

In this work, we propose a framework for generating coverage paths for marine habitat mapping taking these considerations into account. First, we use our previous work in CPP [25] to generate an efficient survey path for mapping

Fig. 1: Satellite imagery of a posidonia habitat in the coast of Majorca (Spain).

Completely determining the regions of interest (ROIs) to be covered is not possible using a previous map, as the habitat might have expanded in certain regions and declined in others. Indeed, identifying those changes is one of the main objectives of the habitat monitoring task. Nonetheless, it is possible to use an estimate of the ROI location based on a previous map of the area, and start to map the ROI on-line from that location.

In this work, we propose a framework for generating coverage paths for marine habitat mapping taking these considerations into account. First, we use our previous work in CPP [25] to generate an efficient survey path for mapping
A. Considerations and Assumptions

The algorithms in this work are intended to generate paths for survey AUVs and ASVs. This category of vehicles includes, for instance, the Bluefin and REMUS AUV families [7], the Sparus AUV (a survey vehicle with hovering capabilities) [8] or the Charlie ASV [9]. These vehicles are designed to trace routes at constant speed and constant depth or altitude without smooth maneuvers. With that in mind, we make the following considerations and assumptions.

1) Seabed Surface Characteristics: Taking into account the vehicle maneuvering limitations, the algorithms in this work focus on seabeds that are effectively planar. That is, surfaces without a strong 3-dimensional component.

2) Localization and Endurance: The vehicle is equipped with a navigation system that keeps track of the vehicle’s pose. Sufficient endurance to complete a survey mission without refueling is assumed.

3) Sensor Considerations: The vehicle is endowed with an imaging sensor, such as an optical or acoustic camera. Thus, practical constraints of the application, such as the desire to survey in parallel tracks for imaging purposes, are respected. It is assumed that the sensor allows for seabed image acquisition and detection of changes in seabed type (such as from sandy bottom to seagrass bottom). The details of such an algorithm are not in the scope of this work.

4) Obstacle Considerations: Obstacles are indeed considered in the off-line planning stage using an a priori map. However, the vehicles considered in this work have typically little or no capacity to sense and avoid obstacles. As a result, it is assumed no steep obstacles (such as protruding rocks) lay inside the ROIs covered by the sensor-based planner. This is a reasonable assumption considering the vehicle sensing limitations and the fact that obstacles are considered in the pre-mission planning. On the other hand, the sensor-based planner is able to dynamically detect and avoid “holes” within the ROIs.

B. Paper Organization

The remainder of this document is organized as follows. In Sec. II, we review related work on CPP. Sec. III introduces our previous work in CPP which allows for efficiently mapping an area using an ASV or an AUV. Sec. IV describes the sensor-based planner used to cover each of the ROI. Sec. V presents our proposed framework, which builds upon the previously introduced planners to generate a coverage path suitable for marine habitat mapping. The experimental setup used to demonstrate the proposed method is introduced in Sec. VI and results are presented in Sec. VII. Concluding remarks and directions for future work are given in Sec.VIII.

II. RELATED WORK

A considerable body of research has addressed the CPP problem in the last few years. A survey is presented in [10]. A class of CPP algorithms, termed exact cellular decomposition, break the free space (i.e., the space free of obstacles) down into simple, non-overlapping regions called cells. The union of all the cells exactly fills the free space. These regions, where no obstacles lay in, are “easy” to cover and can be swept by the robot using simple motions. For instance, each cell could be covered using a “mowing the lawn” pattern. The earliest algorithms to appear in this class were the trapezoidal decomposition [11] and the boustrophedon decomposition [12]. However, these algorithms apply only to polygonal environments. A decomposition that handles a more general class of obstacles is the Morse decomposition [13]. The algorithms cited so far are require the environment to be known a priori (i.e. a map is required). An on-line algorithm to iteratively construct the Morse decomposition using range sensor information was presented in [14]. However, this algorithm can only detect the cell boundaries when they are on the side of the robot. As a result, a cyclic path that includes retraction is used to detect all possible boundaries. This leads to potential repeated coverage. A new algorithm that overcomes this limitation was introduced in [15]. This algorithm allows for sensor-based coverage using simple “mowing the lawn” paths that minimize repeated coverage.

Another class of algorithms, termed approximate cell decomposition or grid-based, rely on a uniform discretization of the target environment. This discretization typically consists of a grid of uniform square cells. As a result, the union of the cells only approximates the target region [16]. In [17], a “wave front” propagated from a goal cell to a start cell is used to assign a specific number to each grid element. That is, the algorithm first assigns a 0 to the goal and then a 1 to all its surrounding cells. Then, all the unmarked cells neighboring the marked 1 are labeled with a 2. The process repeats incrementally until the start goal is reached. Then, a coverage path can be found by starting on the start cell and selecting the neighboring cell with the highest label that is unvisited. If two or more unvisited neighbors share the same label, one of them is selected randomly. This process to find a coverage path is equivalent to using pseudo-gradient descent from the start point on the numeric potential function constituted by the labeling, that is, following the equipotential curves from top to bottom. In another approximate cellular decomposition approach, [18] proposed an algorithm which
follows a spanning tree of a graph induced by the grid cells. The robot is able to cover every point precisely once, hence minimizing repeated coverage. An extension to cover not only unoccupied cells, but also the partially occupied ones was proposed in [19]. It is easy to create a grid map, as it can be represented as an array where each element contains occupancy information associated with a cell. On the other hand, grid maps suffer from exponential growth of memory usage because the resolution remains constant regardless of the complexity of the environment [20]. Also, they require accurate localization to maintain the map’s coherency [21], [22]. For these reasons, their application to vast areas such as the ones of interest in marine habitat monitoring is limited.

Finally, an on-line algorithm targeted to cover projectively planar ocean surfaces was introduced in [23]. An extension of this algorithm for covering only regions close to the ocean floor was presented in [24]. However, no procedure to detect the landmarks used by these algorithms based on sensor information is provided. This precludes the feasibility of a practical implementation.

III. MORSE-BASED EFFICIENT SEABED COVERAGE PATH PLANNING

To generate an optical or acoustic map of a target area that has not been previously mapped, we propose to use our previous CPP method especially targeted for marine environments [25]. This method is an off-line approach that requires a bathymetric map of the area to be covered. The method takes into account the varying sensor footprint when a marine vehicle navigates at a constant depth. This is the case of Autonomous Surface Vehicles (ASVs), which always navigate at the water surface level. Nonetheless, also several AUV tasks require covering the seabed while navigating at constant depth. Indeed, AUVs with motion constraints such as limited heave (motion along the vertical axis) capabilities can also benefit from a constant-depth seabed coverage method. In this situation, the sensor’s footprint over the seafloor changes depending on the height of the bottom surface. As a result, to ensure complete coverage, the interlap spacing between parallel tracks allowed by the shallowest depth must be used, producing an undesired, considerable amount of sensor coverage overlapping among the path laps. This situation is illustrated in Fig. 2.

To address this issue, the method proceeds as follows. First, the target bathymetric map is segmented in regions of similar depth characteristics using the k-means algorithm [26]. Then, Morse decomposition is applied in each region. The Morse decomposition divides the space free of obstacles into cells. To determine the cell boundaries, a vertical line is swept horizontally through the target space. Wherever the line encounters a point in an obstacle boundary where the gradient of the obstacle boundary is parallel to the line’s sweep direction, a vertical cell division is placed. The cell decomposition is encoded as a graph that captures the adjacency relationship (two cells sharing a boundary) between cells. Then, an exhaustive walk that visits all the cells (nodes) in the graph is calculated. Finally, to achieve complete coverage, all the cells are visited in the order given by the exhaustive walk. However, specific coverage paths still need to be determined inside each cell. Specific parallel track paths are then generated in the orientation orthogonal to the main gradient of the surface embraced by the cell, seeking to maximize the interlap spacing. We refer the reader to [25] for more details on this coverage method.

IV. ON-LINE COVERAGE OF ROI

We propose to use the on-line coverage algorithm presented in [15] to achieve coverage of the ROIs. The motivation behind this choice is two-fold. First, the algorithm is simple to implement. Second, in contrast to the on-line version of Morse decomposition [14], it allows for using simple parallel track paths that minimize repeated coverage.

Recall that, in the context of this work, obstacles found by this algorithm correspond to “holes” in the ROIs being covered.

The algorithm iteratively constructs a cell decomposition by detecting five different types of event where cell boundaries are placed:

1) **Split event:** A free space segment in the previous strip is split into two by the emergence of an obstacle, as in Fig. 3(a).
2) **Merge:** Two free space segments in the previous strip are merged into one by the disappearance of an obstacle, as in Fig. 3(b).
3) **Lengthen:** The current strip is much longer than the previous strip, as in Fig. 3(c).
4) **Shorten:** The current strip is much shorter than the previous strip, as in Fig. 3(d).
In all the events shown, \( c_i \) is the current cell (shaded) and \( s_i \) is the current strip. The dashed arrow indicates the sweep direction.

5) **End**: The previous free space segment is the final one in the current cell, as in Fig. 3(e).

In the context of this work, the events are detected using imagery data provided by the robot’s acoustic or optical camera.

The cell decomposition is encoded as a topological map. A topological map is represented as a planar graph, where the nodes represent landmarks (i.e., split, merge, end, lengthen or shorten events) and edges indicate the types of motion required to travel between nodes they are incident upon. For example, whether the edge is next to a wall and which side the wall is on. They also store estimated distances separating the two nodes they connect.

The on-line algorithm iteratively constructs the topological map associated to the slice decomposition of the environment using a finite state machine with three states –boundary, normal, and travel. Fig. 4 shows its state transition diagram. The algorithm starts in the boundary state, as it is assumed that the robot is initially located in a corner of the environment.

In the boundary state, the robot explores the current cell boundary. The aim of the boundary exploration is to expose all cells neighboring the current border. Whenever the robot arrives at an event or at an end of the cell boundary, the topological map is updated. When the boundary exploration has finished, the algorithm switches to the travel state. In the travel state, the robot searches the topological map for an uncovered cell and it is directed to that cell. Then, the robot enters the normal state, where it follows a “mowing the lawn” pattern to cover the current cell. Again, whenever an event is found, the topological map is updated and the algorithm switches to the boundary state. The algorithm finishes when there are no more uncovered cells in the topological map.

![Fig. 4: State transition diagram of the sensor-based coverage algorithm.](image_url)

V. CPP FRAMEWORK FOR MARINE HABITAT MAPPING

We propose a framework based on the two algorithms described above to generate efficient coverage paths for marine habitat mapping. This framework consists of the following stages.

A. **First-time Mapping**

On previously unmonitored areas, our Morse-decomposition-based coverage method is used to generate a coverage path. As described, the method requires an \textit{a priori} bathymetry map or nautical chart of the target area. Taking into account the bathymetric data and the sensor footprint, the method minimizes repeated coverage. During execution of the planned path, and depending on the robot’s on-board sensors, optical, acoustic and/or other kinds of oceanographic data can be blended together to obtain a rich survey map of the area.

B. **Determining ROIs and Selecting Starting Locations**

Once a previous survey of the area is available, ROIs (namely, regions with habitat presence) can be monitored using the described on-line algorithm, discarding regions with no habitat presence. As described, the availability of
an algorithm able to segment the ROIs is assumed. Once the ROIs are segmented, a starting point needs to be selected to cover their current extent.

Because of the fact that the boundary of marine habitats is typically the most affected area by external harshnesses, there is where more variability is expected from survey to survey [1]. Therefore, we choose the centroid of the ROI in the previous survey map as the starting point for covering the current extent of the ROI, given that the centroid is the most far away point from the ROI boundary. The ROI centroids can be easily determined using standard “blob processing” techniques from the Computer Vision community. However, the exact centroid of a given ROI in the previous map might have vanished in the current extent of the ROI. This issue is addressed in V-D.

We note that expert advice can also be injected in this stage to aid in the starting location selection process.

C. Visiting All the ROIs

Once the starting points for each ROI have been selected, the order in which the ROIs will be covered needs to be determined. Once the order is determined, paths from one ROI to the next need to be planned.

1) ROI Covering Order: Naïvely, an arbitrary or random order can be used to cover the ROIs. However, this can lead to a very costly solution in terms of path length. We instead propose to construct a graph where each of its nodes is assigned to each ROI centroid in the previous survey map and edges represent Euclidean distances between them. Then, we want to find a path that visits all the nodes while minimizing the travel distance. Such a problem is known as the Traveling Salesman Problem (TSP) and it is known to be NP-hard [27]. This means that the computation time required to solve the problem increases drastically when the size of the problem increases. As a result, heuristic approximations are often favored to compute a solution in reasonable time. We use the Lin-Kernighan heuristic citeLin1973 to compute a solution to the TSP posed by the ROIs.

It is worth noticing that the solution to the TSP finds a tour starting and ending at the same node (ROI). This can also ease vehicle’s recovery after the mission. However, a solution that visits all the ROIs without returning to the initial ROI might be preferred to minimize path length and hence fuel consumption. This is easily incorporated in the TSP solution [27].

2) ROI-to-ROI Paths: Once the order in which the ROIs will be covered has been determined, paths connecting the starting points associated to each ROI in the order given by the TSP solution are generated. These paths are computed in the bathymetric map or nautical chart used in V-A, hence accounting for obstacles. We compute these “link” paths using a simple Bug1 planner [28] off-line.

D. On-line Coverage of ROIs

To cover the current extent of each ROI on-line using imagery data, we execute the sensor-based planner described in Sec IV. Here, the sensor-based planner uses imagery data to detect the external and “hole” boundaries in the ROI online. The selected starting points in V-B are used to initiate the planner. However, as described, the exact centroid of a given ROI in the previous map might have been vanished in the current extent of the ROI. This issue is addressed in V-D. In such a situation, we propose to trace a spiral path (determined by the vehicle’s minimum turning radius) from the centroid. This spiral path is defined as an Archimedean spiral

\[(x, y) = (rt \cos(t), rt \sin(t)),\]

where \(r\) is the vehicle’s minimum turning radius and \(t\) determines the length of the path. The vehicle follows the spiral path until either an ROI is detected or a certain spiral path length threshold is surpassed. In the later case, the vehicle assesses that no such ROI exists in the current habitat extent.

E. Moving to the Next ROI

Once the coverage of a given ROI is completed, the vehicle moves to the next ROI in the determined sequence by following the corresponding ROI-to-ROI path generated in V-C.2. Then, coverage of a new ROI is executed. The process repeats until all the ROIs have been covered.

VI. EXPERIMENTAL SETUP

A. Real-World Dataset

We demonstrate our proposed approach using a real-world bathymetric dataset to demonstrate the successive steps of our method. Members of the Underwater Robotics Research Center (CIRS) from the University of Girona recorded the dataset near the Formigues islands in July 2009. The Formigues islands are an archipelago of sixteen little islets located about 1300 m off the Canet cape in the Costa Brava in Girona, Spain.

A 253 m width, 148 m height rectangular area was mapped, with its lower-left corner located at 41° 51' 34.35" N, 3° 10' 38.30" E. The map’s abscissae and ordinates axis are aligned with the Earth’s geographical latitude and longitude coordinate system. Depth in the mapped area ranges from about 5.5 m down to 16 m. The bathymetric data were obtained by means of a system composed of a down-looking multi-beam sonar, a motion reference unit and a GPS receiver. The whole system was attached to a boat’s hull. For the demonstration purposes of this paper, basic image processing techniques were used to filter out noise originally present in the map and missing data points were filled using neighboring pixels information. Fig. 5 shows the obtained bathymetric map.

B. Environment Modeling

To demonstrate the ability of our method to minimize repeated coverage in such situations, we run simulations where the vehicle navigates at constant depth. Thus, the environment modeling is composed of three elements: the bathymetric map, the planar surface in which the vehicle
navigates (the vehicle’s workspace) and the ROIs. We model the bathymetry map as a heightmap, $H$, where the elevation values are negative downwards, as shown in Fig. 5. The planar navigation surface, i.e., the vehicle’s workspace, $W$, is modeled as an occupancy grid corresponding to a horizontal slice of the heightmap laying at the constant vehicle depth, $D$, such that for a location $(x, y)$ in the heightmap

$$W(x, y) = \begin{cases} 
1 & \text{if } H(x, y) \geq D \\
0 & \text{if } H(x, y) < D 
\end{cases}$$

In other words, a point in the workspace is an obstacle (i.e., it is occupied) if the elevation of the target surface is above the vehicle’s constant depth. Due to lack of real data, we overlay synthetic ROIs on top of the bathymetric map to demonstrate our method. Fig. 6 shows the obtained workspace by slicing the target surface at $D = -8.5$ m and the added synthetic ROIs, overlaid on the bathymetric map.

C. Vehicle Modeling

In this work, we modeled the vehicle as a point, i.e., having no volume and weight, so it is considered as a fully actuated vehicle. Nonetheless, a vehicle with only the surge (motion along the $X$ axis) and heading (rotation about the $Z$ axis) degrees of freedom could theoretically follow the generated path.

The vehicle has a down-looking optical camera that images a field-of-view beneath the vehicle determined by an angle of view, $\alpha$. It is assumed that, from a given point in the planar navigation surface, the sensor range can reach the underlying target surface.

A simple thresholding detector is used to determine the ROI boundaries.

VII. RESULTS

A. First-time Mapping

Fig. 7 shows a comparison of the coverage density maps obtained using the a standard Morse decomposition and our CPP method. In the standard Morse decomposition path, the minimum field-of-view width along a lap determines the spacing to the next lap among the parallel tracks. Ideally, each point in the target space should be covered once and hence be represented in a dark blue color. The coverage density map corresponding to the “naïve” Morse decomposition method in Fig. 7(a) shows a high amount of overlapping, denoted by yellow and red colors. On the other hand, the coverage overlapping is highly reduced with our CPP method, as denoted by the blue colors on Fig. 7(b).

On the other hand, as shown in our previous work [25], our CPP method yields a path approximately 34% shorter than the “naïve” Morse decomposition approach on the real-world environment presented in this paper.
B. ROIs Coverage

The coverage path generated to cover only the ROIs is shown in Fig. 8.

![Fig. 8: Coverage path covering only ROIs](image)

The vehicle starts covering the upper-left ROI and ends at the bottom-right ROI. It is shown that the generated path efficiently covers only ROIs, a part from the “link” paths to navigate from one ROI to another. We selected a starting point for the bottom-left ROI in its “hole”. As a result, the planner traces a spiral path until the ROI is detected. Then, on-line coverage proceeds as in the other regions. The vehicle was not required to return to the initial ROI in this simulation (recall V-C.1).

VIII. CONCLUSION AND FURTHER WORK

A CPP framework for automated habitat mapping has been presented in this paper. The framework uses an extension of Morse decomposition to plan an efficient pre-mission coverage path for previously unmonitored areas. This method takes into account previously available bathymetric data and sensor characteristics to minimize repeated coverage for vehicles that navigate at constant depth. ROIs are determined in a previous survey map, and an optimal tour that visits every ROI is computed by solving the underlying TSP. Paths connecting the ROIs in the order determined by the TSP solution are then planned. A sensor-based CPP algorithm is used to cover each ROI in the corresponding order.

Results obtained in simulations using a real-world bathymetric map and overlaid synthetic ROIs show the feasibility of the proposed approach.

Further work will include the implementation of the proposed method on the Sparus AUV, a prototype torpedo-shaped AUV constructed at our lab. Application of the proposed method in sea trials to map real habitats is our next aim.

ACKNOWLEDGMENT

This research was sponsored by the Spanish government (DPI2011-27977-C03-02) and the TRIDENT EU FP7-Project under the Grant agreement No. ICT-248497.

REFERENCES


