An Uncertainty-driven Sampling-based Online Coverage Path Planner for Seabed Mapping using Marine Robots

Mingxi Zhou[1] and Jianguang Shi[2]

***Abstract*— Seabed mapping is a common application for marine robots, and it is often framed as a coverage path planning problem in robotics. During a robot-based survey, the coverage of perceptual sensors (e.g., cameras, LIDARS and sonars) changes, especially in underwater environments. Therefore, online path planning is needed to accommodate the sensing changes in order to achieve the desired coverage ratio. In this paper, we present a sensing confidence model and a uncertainty-driven sampling-based online coverage path planner (SO-CPP) to assist in-situ robot planning for seabed mapping and other survey-type applications. Different from conventional lawnmower pattern, the SO-CPP will pick random points based on a probability map that is updated based on in-situ sonar measurements using a sensing confidence model. The SO-CPP then constructs a graph by connecting adjacent nodes with edge costs determined using a multi-variable cost function. Finally, the SO-CPP will select the best route and generate the desired waypoint list using a multi-variable objective function. The SO-CPP has been evaluated in a simulation environment with an actual bathymetric map, a 6-DOF AUV dynamic model and a ray-tracing sonar model. We have performed Monte Carlo simulations with a variety of environmental settings to validate that the SO-CPP is applicable to a convex workspace, a non-convex workspace, and unknown occupied workspace. So-CPP is found outperform regular lawnmower pattern survey by reducing the resulting traveling distance by upto 20%. Besides that, we observed that the prior knowledge about the obstacles in the environment has minor effects on the overall traveling distance. In the paper, limitation and real-world implementation are also discussed along with our plan in the future.**

I. INTRODUCTION

Seafloor mapping is an important practice as the bathymetric database is a key infrastructure [1] supporting various researches and applications, such as marine geological research [2], seabed habitat monitoring [3], naval mine-countermeasure [4]. With the rapid development in ocean technology and instrumentation, there is an increasing trend of using marine robots (i.e., autonomous underwater vehicles and autonomous surface vehicles) in seafloor mapping applications. Compared to manned ship surveys, marine robots provide an alternative approach that is more effective. For example, in deep sea, autonomous underwater vehicles (AUVs) could stay closer to the seabed to obtain high-resolution mapping results ([5][6][7]). On the other hand, in coastal regions, shore-launch autonomous surface vehicles (ASVs) are convenient and environmental friendly for mapping the coastal water, espcially in shallow areas [8].

In robotics, seafloor mapping could be referred to as a coverage path planning (CPP) problem or a viewpoint planning problem. The goal is to plan an optimal path such that the robot could fully explore a user-defined workspace while avoiding collisions [9]. CPP is a well-explored topic in robotics. The research focus started in known scenarios (known environment and known sensor performance, e.g., in [10]), then advanced into partially unknown conditions (unknown environment with known sensor performance, e.g., in [11] and [12]). In recent years, the emphasis started to expand into fully unknown condition (unknown environment and unknown sensor performance, e.g., in [13] and [14]) and multi-robot approach (e.g., in [15]).

In general, the CPP algorithms could be categorized into two classes, offline and online algorithms. The offline algorithms are normally applied to known environments where the mission is pre-programmed using provided data (e.g., seafloor topography) or modeled data (e.g., expected sonar coverage). In contrast, an online approach allows a robot to adapt its path actively using in-situ measurements. In the scenario of robot-based seafloor mapping, we could refer it to as a CPP problem in unknown conditions. In these missions, the seafloor topography is normally assumed unknown prior the mission and the sonar coverage is subject to change due to the seafloor topography. For example, the swath of a commonly used multibeam sonar changes with respect to seafloor depth and surficial properties [16]. As a result, offline approaches may not be an ideal solution. Using lawnmower pattern as an example, a large inter-distance between transects may result in uncovered voids while a small inter-distance may result in significant overlaps and extend the mission time. Therefore, an online solution is needed to allow a robot to perform path re-planning based on in-situ measurements for a better mission efficiency.

In this paper, we present a new online coverage path planning algorithm to address the seafloor mapping problem with marine robots. The algorithm is developed based on probabilistic roadmap (PRM) which was originally designed and widely used to plan a safe path to a defined location. There are only few sampling-based approach coverage path planning in an unknown environment. Therefore, this paper aims to fill this gap. The proposed method has been evaluated under extensive simulation runs. From the results presented later in this paper, the designed algorithm guarantees the coverage in non-convex workspace occupied by priorly unknown obstacles.

*This project is supported by the Department of the Navy, Naval Undersea Warfare Center, Division Keyport, under the Grant Number N00253-19-1-0005, and National Science Foundation under Grant No. #1945924

[1]Mingxi Zhou, Assistant Professor, Graduate School of Oceanography, University of Rhode Island, 215 South Ferry Road, Narragansett, RI, USA mzhou@uri.edu

[2]Jianguang Shi, Assistant Professor, Hangzhou Dianzi University sjg@hdu.edu.cn

The main novelty of the proposed algorithm can be summarized as follows.

- A uncertainty model is introduced to quantify the sensing confidence in each ensonified cell in the workspace based on in-situ sonar measurements. The dynamic sensing confidence map of the workspace is then used for sampling possible waypoint candidates. Compared to a typical PRM with an uniform probability in the workspace, the sensing confidence will allow more waypoints candidates to be drawn in the uncertain or uncovered cells; ultimately shorten the coverage path and guarantee the desired coverage ratio.
- The algorithm uses two new multi-variable functions to select the optimal path among all viable paths from the sampled waypoints.
- The algorithm requires minimum user-defined variables as most of the parameters, e.g., sample size and swath width, are determined automatically using in-situ data. The adaptive approach aims to reduce unnecessary computational resources and to obtain a better prediction on the vehicle coverage during the planning stage.

The remaining paper is organized as follows. In Section II, related work on online CPP is reviewed. The SO-CPP algorithm is introduced with details in Section III. Extensive simulations have been performed with results and discussion presented in Section IV. In Section V, we summarize the paper and outline the directions of our future development for the SO-CPP.

II. RELATED WORK

Regardless of extensive CPP work surveyed in [17], [18] and [19], there are only few online algorithms have been designed, especially for underwater environments. The simplest online coverage algorithm could be designed as a greedy approach where the algorithm will guide the AUV moving towards the direction with the highest reward (minimum time or maximum coverage gain). With this myopic planning strategy, vehicle efficiency may not be optimized [20]. Therefore, more then one variables are normally included in the cost function [20] or planned in a recursive way [21]. In [22], an online CPP algorithm is designed based on multi-objective function which computes the desired vehicle heading on-the-fly. The algorithm was demonstrated on an AUV equipped with a sidescan sonar in seafloor mapping missions. The advantage of the algorithm is highlighted by comparing to offline lawnmower missions and random walk planning methods. An online 3D CPP method for inspection of complex underwater structures is presented in [14]. They proposed a replanning algorithm based on stochastic trajectory optimization that reshapes the nominal path to cope with the actual target structure. In [7], an adaptive seabed coverage algorithm is introduced where the transects of an AUV is determined actively during the mission based on the uncertainty derived from a Gaussian process model. In [23], an online CPP approach based on an optimized backtracking mechanism is presented for mobile robots in an unknown workspace with static obstacles. A dynamic path planning approach for multirobot coverage considering energy constraints is proposed in [24]. The algorithm constructs the sensor-based coverage paths using Generalized Voronoi Diagram (GVD) that accounts for robot energy capacities. Moreover, Biological neural network is also an effective approach in avoiding obstacle and exploring uncovered area for CPP problems ([25] and [26]). However, all the above online approaches has less consideration on the sensor coverage, and the algorithm normally segments the workspace into sub-domains then use lawnmower or spiral pattern to generate local trajectory.

Over the years, sampling based path-planning algorithms, such as rapidly-exploring random tree [27], probabilistic roadmap [28] and their iterations, have drawn increased attentions in robot planning in obstacle-occupied environments. Sampling-based approach has advantages in saving computational time by explicit construction of obstacles in the state space, especially, when dealing with high-dimensional motion planning [27]. Recently, sampling-based methods have been also adopted to solve the CPP problem. A well-studied approach is to transform the CPP problem to a traveling salesman problem or a Art Gallery Problem where utilities are assigned to each sampled point. In [11] and [29], the authors have proposed and demonstrated an offline sampling-based CPP algorithms for 3D underwater inspection applications. In [30], the sampling approach were adopted to plan the next best view point for underwater exploration applications. Both works consider the utility of a sampled point that is independent of others.

To address CPP problems with sample-based approach, a necessary stage is to sample the workspace in such a way that will drive the robot to cover the defined workspoace. In [11] and [29] , this process is done under the assumption that the robot only collect sensor information the the nodes of a graph. For a seafloor mapping mission, it is more realistic to quantify the coverage along a transect between two sampled waypoints and the quantification has to use in-situ sonar performance, e.g., the swath width. Moreover, sampling-based path planning normally uses a uniform probability distributed in the workspace. The SO-CPP that will be introduced in the following section uses a dynamic probability map that is updated based on sonar measurements, and it quantifies the ”rewards” of possible paths based on the observed swath width.

III. THE ALGORITHM

A. Gridded workspace presentation

The goal of the SO-CPP algorithm is to guide a robot to map a user-defined area to the desired coverage ratio, e.g., 99.9%. Often time, marine robots are operated in a 2D horizontal plane, e.g., at the water surface for ASVs or be kept at a constant depth for AUVs. Hence, our workspace, $\mathcal{W}$, is defined as a 2D map that is bounded by the lines connecting a set of user-defined vertexes. It could be non-convex and may contain obstacles.

In our algorithm, the workspace is divided into small cells (i.e., 1 m by 1 m grids). Then, the mission goal is to collect sonar samples in each cell until the total number of observed cells has reached the user-defined level. In each cell, we define two values, the observation condition ($O_{x,y}$) and sensing confidence ($S_{x,y}$), where the subscript indicates the cell's coordinate. The observation condition is initially set to 1 for all grids, meaning all the cells are unexploited.

The observation condition will be updated based on sonar measurements (see Eq. 1). Figure 1 depicts that the AUV is configured with a downward-looking multibeam sonar (MBS) and a forward scanning sonar (FSS). As indicated by Eq. 1, different sonar will result in different observation conditions for computing the updated coverage and obstacle avoidance purposes.

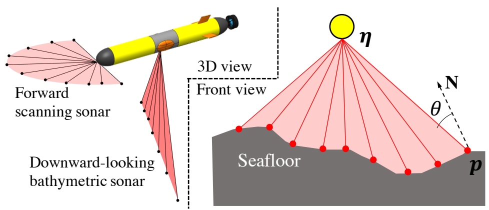

Fig. 1. An AUV equipped with sonars for seafloor mapping

$$O_{x,y} = \begin{cases} 0 & \text{a MBS point obtained in the cell} \\ 1 & \text{no sonar point obtained in the cell} \\ 2 & \text{a FSS point obtained in the cell} \end{cases} \quad (1)$$

The sensing confidence, $S_{x,y}$, is another property which indicates the resulting mapping confidence from the sonar measurements collected in a cell. Initially, $S_{x,y}$ is set to 0 for all cells, and it will be updated based on the MBS measurements. By assuming each sonar measurement as an independent event, the resulting sensing confidence could be computed using Eq. 2 where $C_{x,y}(k)$ indicates the sensing confidence of the k-th measurement in the cell, (x, y).

$$S_{x,y} = 1 - \prod_{i=1}^{k}(1 - C_{x,y}(k)) \quad (2)$$

$$EI = SL + 2AL + 2TL + NL + TS \quad (3)$$

$$C_{x,y}(k) = \cos^2\theta = \left(\frac{\mathbf{p} \bullet \mathbf{N}}{|\mathbf{p}|\,|\mathbf{N}|}\right)^2 \quad (4)$$

In this paper, the sensing confidence of a sonar measurement, $C_{x,y}(k)$, is derived using the sonar equation shown in Eq. 3, where the terms are in decibels. The target strength (TS) is a major factor affecting the echo intensity (EI) if we assume the source level (SL) and nose-level (NL) are identical for each sonar beam, and the attenuation loss (AL) and transmission loss (TL) can be compensated using the time-varying gain. A higher EI means a more distinct acoustic return that could be separated from background noise. If we reformat Eq. 3 by converting decibels into power unit, the power of echo intensity will be proportional to the power of TS. For seafloor, TS is proportional to the square of the cosine of the incident angle [31] which is denoted as θ in Fig. 1 where the 2D normal vector, $\mathbf{N}$, could be computed from the MBS swath. Herein, we define the sensing confidence of a data point in the k-th MBS swath, $C_{x,y}(k)$, is computed using Eq. 4 where $\mathbf{p}$ is a MBS sonar point relative to the vehicle body, and $\mathbf{N}$ is the normal vector of the swath at $\mathbf{p}$. Using the vehicle's pose information, the corresponding cell, (x,y), of $\mathbf{p}$ could be found. For different mapping sensors, different sensing confidence functions could be developed, e.g., the one defined in [22] for sidescan sonar.

B. Algorithm workflow

A detailed flow chart of the algorithm is presented in Fig. 2, and Fig. 3 exemplifies a coverage mission in an unknown obstacle-occupied workspace. The SO-CPP will replan a robot's path if the vehicle has reached the last programmed waypoint or its planned path to the next waypoint intersects with an newly observed obstacle. For example, Fig.3(a) shows a planned path that intersects with the obstacles because that the robot doesn't know their existence at the beginning. When the robot has traveled to the location shown in Fig. 3(b), robot has gained knowledge about the obstacle from its FSS. A replanning is activated as vehicle has detected an intersection between its path to the next waypoint and the newly detected obstacle.

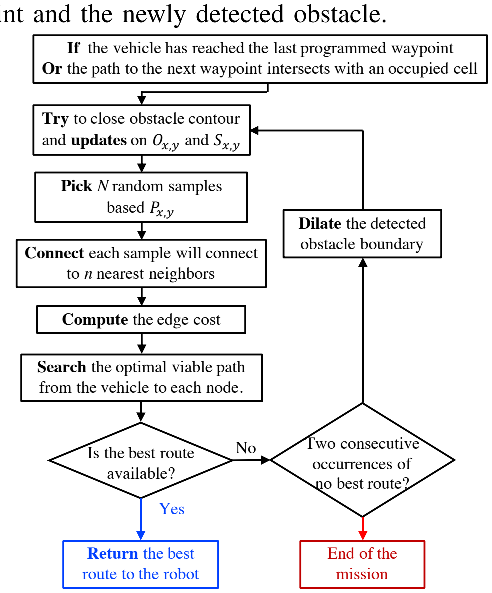

Fig. 2. The workflow the SO-CPP

In the first step during the planning, the SO-CPP will try to close the contour of the detected obstacles using morphological closing algorithm which is commonly available in MATLAB and openCV. In all of our simulations presented in Section 4, we used an identical morphological structure of a 10 m disk where 10 m is the turning diameter of the simulated robot. During this step, the algorithm will update $O_{x,y}$ and $S_{x,y}$ of the flooded cells to 2 and 1, indicating the cells are explored and are occupied by obstacles. For example, in Fig.3(d), the square obstacle is flooded since the robot has profiled the overall contour of the obstacle from the FSS.

$$P_{x,y} = \frac{(1 - S_{x,y})}{\sum(1 - S_{x,y})} \quad \forall x, y \in \mathcal{W} \quad (5)$$

Next, the algorithm will draw N numbers of samples in the workspace based on the probability map, $P_{x,y}$, which is derived using Eq. 5 from the sensing confidence distribution in the workspace. As a result, more samples (candidate waypoints) will be made in the regions with low sensing confidence or uncovered, cells i.e., the darker regions shown in Fig.3. One constrain we applied here is to avoid any samples within the 10 m which is also the acceptance radius for waypoint tracking.

After the candidate waypoints are obtained, SO-CPP will construct a roadmap by connecting each sample, including the current vehicle position, to n nearest samples. For each pair of connected samples (or called nodes), the SO-CPP computes the cost along each connected line (or called the edge). The cost consists of two components, traveling distance and the redundant-to-new information (RTNI) ratio. The traveling distance, $D_{i,j}$, is the Euclidean distance between two nodes, i and j, while RTNI is determined using in-situ sonar measurements. During the mission, the system will record the minimum width of the sonar swath, based on which SO-CPP will predict the potential coverage area, $A_{i,j}$ along an edge. In $A_{i,j}$, the algorithm will count the number of unobserved cells and observed cells. In a mathematical format, the RTNI ratio is computed using Eq. 6 where the numerator and denominator quantify the numbers of observed cells and unobserved cells, respectively. The final cost of each edge is a combination of the distance cost and the RTNI ratio as shown in Eq. 7. Both components are normalized values.

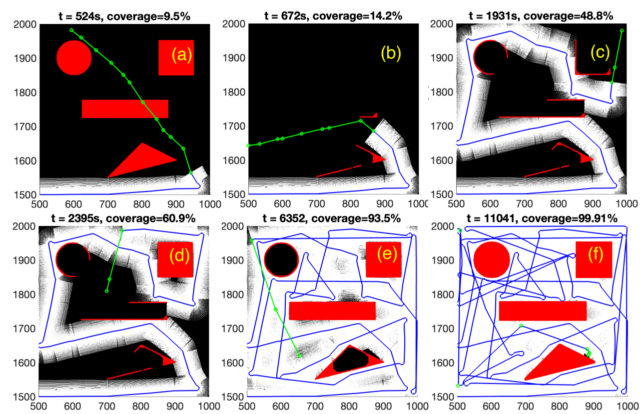

Fig. 3. A series of planning result in a single coverage mission. The red areas indicate obstacles, the blue line shows the vehicle traveled path, and the green line shows the planned path. (a) shows the a collision-free path planned at the beginning and the predefined obstacles that was incrementally detected by the AUV using a mechanically scanning sonar. (b) shows that the vehicle adjusts its path when the path to the next waypoint intersects with an obstacle detected during the mission. The designed algorithm closed the contour of the square obstacle from (c) to (d). The vehicle continued to cover the workspace and closed the rectangular obstacle. In (e) the vehicle was stuck since there is no OVP and no viable path to the vehicle's neighbor samples. Then, the algorithm dilated the detected obstacle boundary then applied the morphological closing to close the rounded and triangular obstacle. Vehicle reached the desired coverage ratio in (f) with an overall mission time of about 3 hours in the workspace (500 m by 500 m).

$$N_{i,j} = \frac{\sum |O_{x,y} - 1|}{\sum 1 - |O_{x,y} - 1|} \quad \forall x, y \in A_{i,j} \tag{6}$$

$$Cost_{i,j} = \frac{D_{i,j}}{\sum_i \sum_j D_{i,j}} + \frac{N_{i,j}}{\sum_i \sum_i N_{i,j}} \tag{7}$$

After that, the algorithm will search for the optimal viable path (OVP) from the vehicle to each sampled waypoint using a typical graphic search algorithm. Herein, we implemented the Dijkstra's algorithm with edge costs computed in Eq. 7. For each OVP, the algorithm will also compute a reward which is equals to the summation of the inverse of RTNI ratio that is normalized by the mean turning angle along the OVP. The mean turning angle is included because 1) we want to minimize turning motion which may degrade the vehicle navigation especially for AUVs [22], and 2) a large turn angle between consecutive edges will produce a large overlapped coverage area resulting in double counting $N_{i,j}$ in the overlapped region. Eventually, SO-CPP will select the OVP with the highest reward to be the best route, then send the waypoint list to the robot's guidance system.

There is a possibility that there is no OVP available for all nodes. This happens most likely in two scenarios in an obstacle-occupied environment towards the end of a coverage mission. First, the unobserved cells are sometimes separated in a small cluster. With the existence of obstacles, it is highly like that the vehicle could not connect to the samples that is further away. Secondly, the obstacle contour may not be closed due to the inefficiency of morphological closing algorithm on obstacles in a rounded or triangular shape. (see Fig.3 (e)). To handle these scenarios, SO-CPP will guide the vehicle to the nearest sampled waypoint. If the straight-line to the nearest sampled waypoint is also obstructed by an obstacle, normally happens in the second scenario, the algorithm will apply binary dilate to thicken the detected obstacle boundary, then apply the morphological closing algorithm again. By doing so, the area occupied by rounded and triangular obstacles maybe flooded (see the comparison between Fig. 3(e) and (f)). After that, the robot will repeat the planning process. This time, the robot will avoid taking samples inside an obstacle, increasing the chance of finding optimal viable paths. If there is a best route available, the robot will continue the mission with the the route. Otherwise, the algorithm will declare the end of the mission which may resulting in the final coverage slightly less than the desired value.

One key parameter that is expected to affect the path planning result is the number of samples. A high number of samples is likely to produce a better and smoother path with a trade-off of increased computational time. We later found that the total coverage mission time will converge after the sample number has increased beyond a value. We believe such value varies with respect to the workspace size. For example, on a larger workspace, the best sample size may be a larger value. Also, we would like to avoid oversampling that may happen when there is only a small portion of the workspace left unexplored. Therefore, we designed the SO-CPP to determine the sample number and connected neighbors adaptively. When the workspace is small or the uncovered region is small, the algorithm could reduce the sample size to accelerate the computation.

$$n = \sqrt{\sum_x \sum_y (1 - |O_{x,y} - 1|)} \tag{8}$$

As shown in Eq. 8, the sample size, n, is equal to the square root of the total number of unobserved cells, and we define that the number of connected neighbors for each sample is the square root of the sample size, $\sqrt{n}$. One could set the minimum and maximum value according to their computer and workspace. Herein, we set the lower threshold of n to be 100.

IV. ALGORITHM EVALUATION

The proposed algorithm has been evaluated in a simulated environment where the seafloor topography (1-m resolution) is linearly interpolated from a 30-m grid bathymetric map of the Narragansett Bay, RI, provided by National Oceanic and Atmospheric Administration (NOAA). The AUV is modeled using a 6-DOF dynamic model with coefficients from a REMUS AUV [32]. We constrained the AUV to move at a constant depth of 10 m at the desired speed of 1 m/s. The waypoint tracking is realized using the line-of-sight guidance law [33] with an acceptance radius of 10 m. The sonars shown in Fig. 1 are modeled using a simple ray-tracing sonar model which was used in [31] without considering the multi-path effects. During the simulation, the MBS is configured with 120 beams (1 deg separation) pinging at 2 Hz with a maximum profiling range of 120 m, and the FSS is configured to ping at 5Hz with a stepping size of 1.8 deg and a maximum range of 75 m. The sonar specification represents a common MBS available from Imagenex or Kongsberg and a compact FSS available at Tritech, Ecologger or Imagenex.

TABLE I
A SUMMARY OF PERFORMED SIMULATION RUNS WITH SO-CPP

Set No.	runs	sample no.	obstacle condition	workspace [m] W	L	D
1	360	fixed	no	200	500	30-40
2	50	adapitve	yes & known	300	500	30-40
3	50	adaptive	yes& unknown	300	500	30-40
4	30	adaptive	yes& unknown	500	500	30-40

We have run four sets of simulations with the SO-CPP as summarized in Table 1. At least 30 Monte Carlo runs are applied to each setting in order to quantify the randomness induced by the sampling process. The coverage goal for all simulations is set to 99.9%, and the maximum mission time is set to 10,000 seconds.

SO-CPP vs. boustrophedon. The first set of simulation is done at different sample sizes ranged from 20 to 240 with an increment of 20. The purpose is to compare the coverage performance between SO-CPP and pre-programmed lawnmower patterns at different inter-distance ranges from 10 m to 50 m. The red and blue markers in Fig. 4(a) show the final coverage with respect to the total traveling distance from lawnmower pattern and SO-CPP runs. We observed that lawnmower pattern could reach the desired coverage ratio only when the inter-distance is less than 20 m. In contrast, all the SO-CPP runs yields a final coverage over 99.9%. In the zoom-in plot, we observed that the the traveling distances from the 360 runs vary from 4 km to 6.5 km. There are 334 runs and 26 runs on the left-hand-side and right-hand-side of the red reference curve from the lawnmower surveys, respectively.

Figure 4(b) shows the resulting traveling distance and SO-CPP computational time at different sample sizes. The mission time is observed longer with small sample size (less then 60), and the averaged mission time was slightly increased after the sample size has exceeded 100. On one hand, the long mission time with small sample size may due to under sample, which results in sub-optimal coverage path during each planning. On the other hand, the small increase occurred with larger sample sizes may due to oversampling. When the vehicle has obtained a relatively high coverage, e.g., 80%, a large sample size will result in sample clusters on separated uncovered regions. During the graph building process, connection may not be established between clusters. As a result, SO-CPP will produce a sub-optimal route for the robot. We also have shown the relation between computational time and the sample size in Fig. 4(b). The SO-CPP was implemented in MATLAB on a standard configuration laptop (dual core 3.1 GHz i-7 CPU). A moderate embedded computer, e.g., Jetson TX-2 or Raspberry PI-4, could offer a similar computational resources, and we expect the run time could be reduced if it is implemented in C^{++}. As a result, SO-CPP is feasible to implement on marine robots on a backseat computer for real-world deployments.

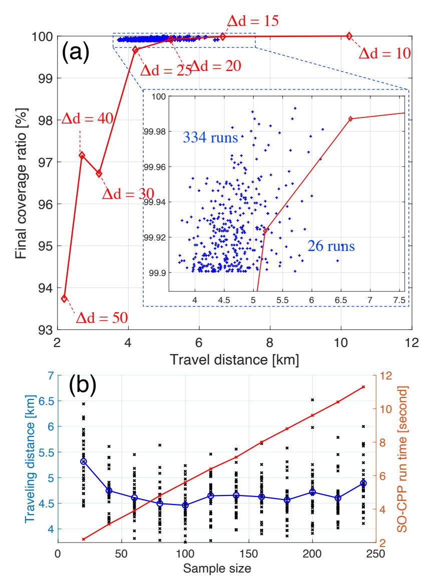

Fig. 4. (a) Final coverage ratio versus traveling distance (blue points: the SO-CPP runs, red points: lawnmower runs at different inter-distance, Δd). (b) The resulting traveling distance and computational time at different sample sizes.

Known vs. priorly unknown workspace. Simulation set No.2 and No.3 are performed on the same workspace with different initial conditions. In simulation set No. 2, the obstacles' shape and location are known. In contrast, in simulation set No. 3, the obstacles are assumed unknown initially, and the vehicle uses the FSS to gain incremental knowledge about the workspace. The CPP algorithm introduced in Section III.B have the replanning feature when the vehicle finds its path to the next programmed waypoint is not feasible due to the newly discovered obstacles from the FSS.

Figure 5(a) presents the resulting vehicle path and the observation conditions from one simulation run in set No.3. In Fig. 5(b), we present the histogram of resulting traveling distances from two simulation sets. We observed that the mean values from two distributions are similar, meaning the initial condition of the workspace has minimal effects on the overall length of the coverage mission.

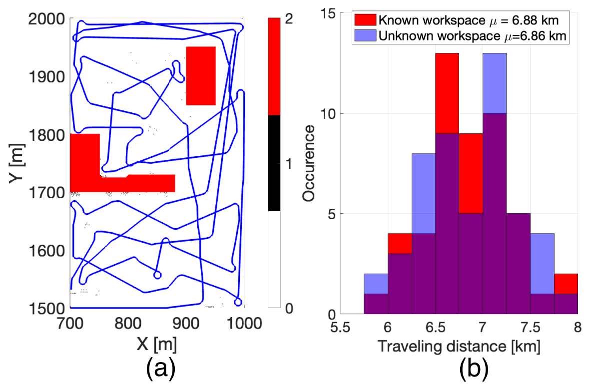

Fig. 5. (a) An example coverage mission from simulation set 6, the red region indicates obstacles, (b) histogram of the resulting vehicle traveling distance for simulation set No.2 and No. 3.

Limitation and discussion So far, we have presented promising results that the SO-CPP could be used to cover a non-convex, convex, or obstacle occupied workspace with or without prior knowledge about the obstacles and the seafloor topography. However, during our simulation runs, we discovered several limitations of the SO-CPP that will need attention during field deployments. First, the obstacle shape may be slightly larger due to the morphological closing algorithm. As shown in Fig. 6 (b) the obstacle's corners have been smoothed due to the morphological closing algorithm and the shape has been enlarged due to the image dilation. As a result, the coverage ratio maybe slightly over estimated, especially, when the occupied space is high in the workspace. One way to overcome this limitation is to implemented a more universal obstacle contour closing algorithms, e.g., image flooding methods.

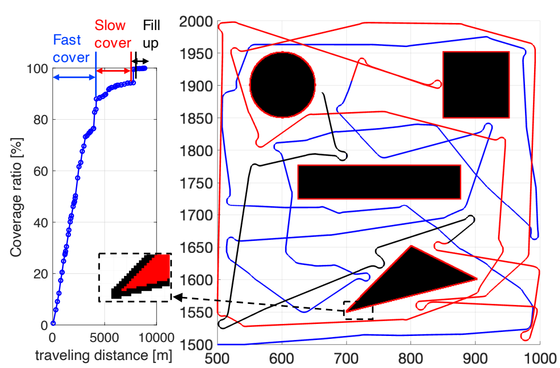

Fig. 6. A coverage mission from set No. 4 Left: coverage ratio vs. traveling distance. Right: overall path and obstacles. Zoom-in: the difference between the actual obstacle (red) and the SO-CPP described obstacle (black). The vehicle trajectory is displayed in three stages (blue, red, and black) corresponding to the coverage stage indicated in the left panel.

To validate that the SO-CPP is capable of guaranteeing the coverage goal, we performed additional 30 runs on a larger domain with more obstacles. A time series example of the result is already shown in Fig. 3, and a final result is also shown in Fig. 6. From the 30 runs, we observed that there were 2 runs aborted with a final ratio (98.75% and 99.84%). During these runs, the robot could not find a viable route after it has dilated the obstacles' boundary. For the remaining successful runs, the resulting traveling distance is 9.78 km (max. 10.18 km and min. 8.73 km). To overcome this limitation, one may increase the lower threshold of the sample size such that more samples could be drawn to increase the possibility of producing a viable path, or implement another planning logic (e.g., move towards a safe direction for a certain distance) to handle this scenario. However, the first approach will increase the computation load while the later approach may increase the mission time.

For real world deployment, there are two more factors needed to be account for. First, we assumed that the vehicle knows its position during the mapping mission. This assumption is applicable to autonomous surface vehicles (ASVs) equppied with a GNSS module. On the AUV, localization is a known problem. In order to compliment the assumption and make the SO-CPP applicable, approaches could be made besides implementing sophisticated SLAM algorithm. Since the path planning are performed intermittently, the AUV could surface to obtain GPS fixes then use the information to back propagate its path for the best localization result. Meanwhile, the sensing confidence map could be updated based on the updated vehicle trajectory. Alternatively, for deep water operation, acoustic localization could be used to derive the geo-referenced location for AUV for trajectory back propagation. Moreover, environmental factors, such as waves, tides, and currents are not considered in our simulation as we focus on evaluating the systematic performance of the algorithm. The environmental disturbance has two major effects that require attentions during field deployments. First, the disturbance will make our predicted area coverage less accurate, affecting the best route selection. Secondly, the disturbance will affect the moving cost of the robot along a viable path. To this end, current speed and attack angles could be integrated into our cost function shown in Eq. 7.

V. CONCLUSION AND FUTURE WORK

In this paper, we presented a sampling-based online coverage path planning (SO-CPP) algorithm for robot-based seafloor mapping applications. The SO-CPP utilizes a guided probabilistic sampling process to generate candidate waypoints, then applies two-stage optimization to select the most-rewarding waypoint combination that will be updated in robot's guidance system intermittently. The SO-CPP was validated in a realistic simulation environment. A series of Monte Carlo runs was conducted to investigate the mission performance due to different sample sizes and workspaces. The new algorithm guarantees the overall coverage per user's request at a shorter mission time compared to preprogrammed lawnmower surveys. The algorithm has also been tested in a obstacle-occupied workspace. The desired coverage is highly achievable while several limitations has to be considered during the actual field deployments.

There is still room to improve the SO-CPP. First, robot's motion constraint and environmental influence is currently not considered in the SO-CPP. Implementing a track smooth procedure (e.g., the optimization mentioned in [34]) could greatly improve the SO-CPP integration for underactuated vehicles,. Meanwhile, additional environment-related costs could be integrated into the multi-variable cost function as mentioned earlier. Moreover, the vehicle is currently constrained moving in a 2D horizontal plane. In reality, the underwater robot could be controlled at different desired depths to provide mapping data at different resolutions. As we observed in the simulation, a survey from a further distance doesn't always result in a shorter coverage mission as voids may be left in a swath. It would be interesting to expand the SO-CPP to a 3D planing space where the sensing confidence shown in Eq. 4 should be modified to account for the sensing confidence at different altitudes. Besides covering a 2.5D seafloor, the SO-CPP should be generalized into 3D environment to fulfill infrastructure inspection missions both on land and in underwater environments. Currently, the authors are integrating a sonar system onto an ASV. Field experiments are planned in the Fall 2022 to demonstrate the SO-CPP in a real-world environment.

REFERENCES

[1] L. Mayer, M. Jakobsson, G. Allen, B. Dorschel, R. Falconer, V. Ferrini, G. Lamarche, H. Snaith, and P. Weatherall, "The Nippon Foundation-GEBCO seabed 2030 project: The quest to see the world's oceans completely mapped by 2030," *Geosciences*, vol. 8, no. 2, 2018.

[2] D. W. Caress, H. Thomas, W. Kirkwood, R. McEwen, R. Henthorn, D. A. Clague, C. Paull, J. Paduan, and K. L. Maier, "High-resolution multibeam, sidescan, and subbottom surveys using the mbari auv d. allan b," *Marine habitat mapping technology for Alaska*, pp. 47–69, 2008.

[3] N. Tolimieri, M. Clarke, H. Singh, and C. Goldfinger, "Evaluating the seabed auv for monitoring groundfish in untrawlable habitat," *Marine Habitat Mapping Technology for Alaska*, pp. 129–142, 2008.

[4] S. Sariel, T. Balch, and N. Erdogan, "Naval mine countermeasure missions," *IEEE Robotics Automation Magazine*, vol. 15, no. 1, pp. 45–52, 2008.

[5] K. Zwolak and A. Felski, "Current State of Deep Ocean Bathymetric Exploration," *Annual of Navigation*, vol. 24, no. 1, pp. 257–267, 2018.

[6] C. Roman and H. Singh, "Consistency based error evaluation for deep sea bathymetric mapping with robotic vehicles," *Proceedings 2006 IEEE International Conference on Robotics and Automation*, pp. 3568–3574, 2006.

[7] G. A. Hollinger, U. Mitra, and G. S. Sukhatme, "Active and Adaptive Dive Planning for Dense Bathymetric Mapping," *Experimental Robotics*, vol. 88, pp. 803–817, 2013.

[8] J. E. Manley, "Unmanned surface vehicles, 15 years of development," *IEEE/MTS Oceans 2008*, 2008.

[9] E. Galceran and M. Carreras, "Planning coverage paths on bathymetric maps for in-detail inspection of the ocean floor," *IEEE International Conference on Robotics and Automation*, pp. 4159–4164, 2013.

[10] H. Choset and P. Pignon, "Coverage path planning: The boustrophedon cellular decomposition," *Field and Service Robotics*, pp. 203–209, 1998.

[11] B. Englot and F. S. Hover, "Sampling-based coverage path planning for inspection of complex structures," *The 22nd International Conference on Automated Planning and Scheduling*, pp. 29–37, 2014.

[12] E. Galceran and M. Carreras, "Efficient seabed coverage path planning for asvs and auvs," *IEEE International Conference on Intelligent Robots and Systems*, pp. 88–93, 2012.

[13] N. Karapetyan, J. Moulton, J. S. Lewis, A. Q. Li, J. M. Okane, and I. Rekleitis, "Multi-robot dubins coverage with autonomous surface vehicles," *IEEE International Conference on Robotics and Automation*, pp. 2373–2379, 2018.

[14] E. Galceran, R. Campos, D. Ribas, M. Carreras, and P. Ridao, "Coverage path planning with real-time replanning and surface reconstruction for inspection of three-dimensional underwater structures using autonomous underwater vehicles," *Journal of Field Robotics*, vol. 32, no. 7, pp. 952–983, 2015.

[15] I. Rekleitis, A. Peng, N. Edward, and H. Choset, "Efficient boustrophedon multi-robot coverage : an algorithmic approach," *Annals of Mathematics and Artificial Intelligence*, vol. 52, pp. 109–142, 2008.

[16] R. P. Hodges, *Underwater acoustics: Analysis, design and performance of sonar*. John Wiley & Sons, 2011.

[17] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1258–1276, 2013.

[18] T. M. Cabreira, L. B. Brisolara, and P. R. Ferreira Jr., "Survey on coverage path planning with unmanned aerial vehicles," *Drones*, vol. 3, no. 1, 2019.

[19] A. Khan, I. Noreen, and Z. Habib, "On complete coverage path planning algorithms for non-holonomic mobile robots: Survey and challenges," *J. Inf. Sci. Eng.*, vol. 33, pp. 101–121, 2017.

[20] A. Bagnitckii, A. Inzartsev, and A. M. Pavin, "Planning and correction of the auv coverage path in real time," *2017 IEEE Underwater Technology (UT)*, pp. 1–6, 2017.

[21] J. Binney, A. Krause, and G. S. Sukhatme, "Informative path planning for an autonomous underwater vehicle," *2010 IEEE International Conference on Robotics and Automation*, pp. 4791–4796, 2010.

[22] L. Paull, S. Saeedi, M. Seto, and H. Li, "Sensor-driven online coverage planning for autonomous underwater vehicles," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 6, pp. 1827–1838, 2013.

[23] A. Khan, I. Noreen, H. Ryu, N. L. Doh, and Z. Habib, "Online complete coverage path planning using two-way proximity search," *Intelligent Service Robotics*, vol. 10, no. 3, pp. 229–240, 2017.

[24] A. Yazici, G. Kirlik, O. Parlaktuna, and A. Sipahioglu, "A dynamic path planning approach for multirobot sensor-based coverage considering energy constraints," *IEEE transactions on cybernetics*, vol. 44, no. 3, pp. 305–314, 2013.

[25] M. Yan, D. Zhu, and S. X. Yang, "Complete coverage path planning in an unknown underwater environment based on D-s data fusion real-time map building," *International Journal of Distributed Sensor Networks*, vol. 2012, 2012.

[26] D. Zhu, C. Tian, B. Sun, and C. Luo, "Complete Coverage Path Planning of Autonomous Underwater Vehicle Based on GBNN Algorithm," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 94, no. 1, pp. 237–249, 2019.

[27] S. Karaman and E. Frazzili, "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research*, vol. 30, pp. 846–894, 2011.

[28] K. Belghith, F. Kabanza, L. Hartman, and R. Nkambou, "Any- time dynamic path-planning with flexible probabilistic roadmaps," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2372–2377, 2006.

[29] B. Englot and F. Hover, "Planning complex inspection tasks using redundant roadmaps," *Robotics Research. Springer Tracts in Advanced Robotics*, vol. 100, pp. 327–343, 2017.

[30] E. Vidal, N. Palomeras, K. Isteni, and G. Marc, "Multisensor online 3D view planning for autonomous underwater exploration," vol. 37, no. 6, pp. 1123–1147, 2020.

[31] M. Zhou, R. Bachmayer, and B. De Young, "Initial performance analysis on underside iceberg profiling with autonomous underwater vehicle," 2015.

[32] T. Prestero, "Verification of a six-degree of freedom simulation model for the remus autonomous underwater vehicle," *Ph.D. Thesis, Massachusetts Institute of Technology*, 2001.

[33] K. Shojaei and M. Dolatshahi, "Line-of-sight target tracking control of underactuated autonomous underwater vehicles," *2017*, vol. 133, pp. 244–252.

[34] S. E. Houts, S. M. Rock, and R. McEwen, "Aggressive terrain following for motion-constrained AUVs," *2012 IEEE/OES Autonomous Underwater Vehicles, AUV 2012*, 2012.