MIT Libraries | DSpace@MIT

MIT Open Access Articles

Sampling-Based Sweep Planning to Exploit Local Planarity in the Inspection of Complex 3d Structures

The MIT Faculty has made this article openly available. ***Please share*** how this access benefits you. Your story matters.

Citation: Englot, Brendan and Franz S. Hover. "Sampling-Based Sweep Planning to Exploit Local Planarity in the Inspection of Complex 3d Structures." IEEE/RSJ International Conference on Intelligent Robots and Systems, October 7-12, 2012, Vilamoura, Algarve, Portugal.

As Published: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=06386126

Publisher: Institute of Electrical and Electronics Engineers (IEEE)

Persistent URL: http://hdl.handle.net/1721.1/87728

Version: Author's final manuscript: final author's manuscript post peer review, without publisher's formatting or copy editing

Terms of use: Creative Commons Attribution-Noncommercial-Share Alike

Massachusetts Institute of Technology

Sampling-Based Sweep Planning to Exploit Local Planarity in the Inspection of Complex 3D Structures

Brendan Englot and Franz S. Hover

***Abstract*— We present a hybrid algorithm that plans feasible paths for 100% sensor coverage of complex 3D structures. The structures to be inspected are segmented to isolate planar areas, and back-and-forth sweep paths are generated to view as much of these planar areas as possible while avoiding collision. A randomized planning procedure fills in the remaining gaps in coverage. The problem of selecting an order to traverse the elements of the inspection is solved by reduction to the traveling salesman problem. We present results of the planning algorithm for an autonomous underwater vehicle inspecting the in-water portion of a ship hull. The randomized configurations succeed in observing confined and occluded areas, while the 2D sweep paths succeed in covering the open areas.**

I. INTRODUCTION

Coverage path planning enables fast and efficient task completion in applications that require an autonomous agent to sweep an end effector over some portion of its workspace, including sensing, cleaning, painting, and plowing [7]. Optimal coverage paths often utilize a back-and-forth sweeping motion to cover the required areas efficiently. This is achieved in obstacle-filled 2D workspaces using cell decomposition methods [6], [14], which allow areas of open floorspace to be swept with uninterrupted motions. In 3D workspaces, the coverage task typically requires a full sweep of the interior or exterior boundary of a 3D structure embedded in the workspace. Back-and-forth sweeping has achieved uniform coverage of curved surface patches [3], and circumferential looping around 2D cross-sections has been used to cover the full boundary of closed 3D structures [2], [5].

The paths planned by these algorithms contain uniform spacing between tracklines and often accumulate data slice-by-slice along a single spatial dimension of the workspace. Travel along a highly regular inspection route of this type allows a human operator to monitor task completion, and facilitates easy reading and interpretation of a sensor-based data product. To our knowledge, however, the existence of an arbitratry, collision-free coverage path is not a sufficient condition for the existence of a route with uniform spacing or a layout along a single spatial dimension.

On the other hand, covering the boundary of a 3D structure using randomly sampled view configurations, a technique that employs a discrete set of stationary views rather than a continuous sensing trajectory, [9], [10], has been shown in certain cases to be probabilistically complete [11]. This approach, which formulates coverage planning as a variant of the art gallery problem [20], is practical when a robot can reliably stabilize at a fixed waypoint, but cannot execute a continuous sweep trajectory with high precision. Unfortunately, a randomized approach lacks the desirable regularity of other methods.

This work was supported by the Office of Naval Research under Grant N00014-06-10043, monitored by Dr. T.F. Swean

B. Englot and F.S. Hover are with the Department of Mechanical Engineering, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge MA 02139 USA benglot@mit.edu hover@mit.edu

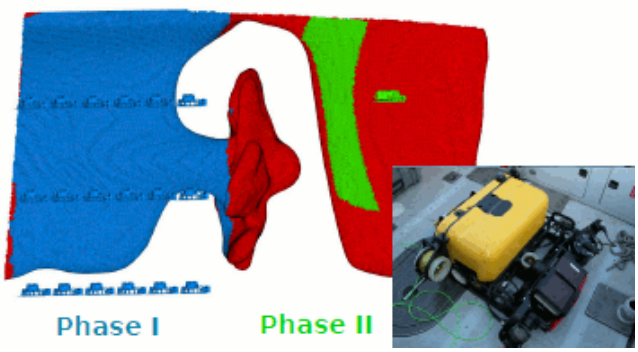


Fig. 1. At left, a low-resolution triangle mesh model of the *SS Curtiss* constructed from an HAUV survey, along with waypoints designed to cover the mesh. Illustrating Phase I of the coverage sampling problem, a waypoint grid and the surface area observed by its waypoints are plotted in blue. The grid is designed to cover a large, planar segment of the mesh. Illustrating Phase II of the problem, an individual waypoint and its observed surface area are plotted in green. At right, a photograph of the Bluefin-MIT HAUV.

Our application of interest, the autonomous in-water inspection of a ship hull, stands to benefit from both completeness and regularity. The Bluefin-MIT Hovering Autonomous Underwater Vehicle (HAUV) [13], pictured in Figure 1, is tasked with inspecting 100% of the surface area at the stern using a forward-looking bathymetry sonar. The shafts, propellers, rudders, and other protruding structures at the stern contribute to a confined and occluded environment in which quantitative bounds governing algorithm completeness are valuable. It is also desirable for as much of an inspection as possible to be executed along a regularized route. Some areas of the stern are open and easily accessible, and we propose a novel approach that reconciles these two objectives.

For this inspection task, we have developed a two-phase path planning strategy that takes advantage of the simplicity and efficiency of modular and sweep-based approaches while considering the collision and occlusion hazards in the most confined areas of a ship's stern. First, *a priori* triangle mesh models of structures are segmented to isolate planar areas using a hierarchical face-clustering algorithm [4], and a planar, sweep-based path is designed for each segment. The paths are generated using a sampling-based algorithm

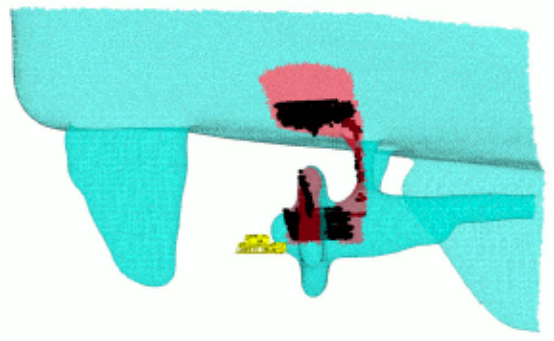


Fig. 2. A snapshot of experimental results from a February 2012 HAUV inspection at the stern of the *USCGC Seneca*. The mesh model used for planning the inspection is rendered in blue, with a single planned view of the ship depicted in red. The plan is compared with the sonar data gathered by the robot at this waypoint, plotted as a point cloud in black. The mesh is rendered with some transparency to allow visibility of data that lie inside the mesh boundaries. Inaccuracies in the hull curvature of the model are responsible for the scarcity of data in the uppermost part of the planned view. The robot is depicted to scale.

that checks all sweep paths against the entire mesh model for collisions, not just the segment being covered. This procedure comes with no guarantee of full coverage; it is simply intended to exploit the open, planar regions of a complex structure using simple and intutive paths.

Then, after designing sweep trajectories for all segments, a randomized, art-gallery-style algorithm [10] is used to fill in the gaps in coverage with individual robot configurations that observe the remaining areas of the structure. An inspection tour specifying the order of traversal among sweep paths and gap-filling configurations is computed by reduction to the traveling salesman problem (TSP), which is solved using the chained Lin-Kernighan heuristic [1].

In Section II we introduce our hybrid sweeping-and-sampling procedure used to obtain 100% structure coverage. We define the property of probabilistic completeness in the context of sweep-based path planning and analyze our algorithm's completeness and convergence to a feasible solution. In Section III the combinatorial optimization steps are presented that build a full-coverage inspection tour from our hybrid components, and in Section IV we present computational results of the algorithm.

II. OBTAINING 100% COVERAGE OF THE STRUCTURE

We obtain full coverage of the structure through a combination of back-and-forth sweep paths and individual configurations, which fill in the gaps in coverage left by the sweep paths. Unlike most coverage path planning algorithms, which assume continuous sensing by the end effector along a sweep trajectory, the sweep paths we construct are comprised of discrete, static waypoints arranged in a grid. An example of the planned and executed data acquisition at one of these waypoints is given in Figure 2. This inspection scheme allows the HAUV to accurately stabilize at each waypoint for the collection of data, something it cannot do reliably along a continuous sensing trajectory. If the HAUV is sensing continuously while translating, ocean disturbances may displace the vehicle from its planned path and missed views will need to be revisited.

The complete algorithm for generating a sweep-based 100%-coverage inspection tour is illustrated in Figure 3. In this section we address the problem of sampling a set of feasible configurations that achieves 100% coverage of a structure boundary, which we term the coverage sampling problem (CSP). In Phase I of the CSP, a waypoint grid is generated for each surface in the mesh segmentation. In Phase II, individual configurations are sampled to cover the unobserved remainder of the structure mesh. An example of the waypoints designed in each phase of the CSP is given in Figure 1. Once a full-coverage set of configurations is obtained, a set cover is solved over the configurations. The final step is solution of the multi-goal planning problem (MPP), in which the grids and other sensing configurations are connected by feasible paths, and an inspection tour is constructed by iterative solution of the TSP.

The goal of our planning procedure is to produce a feasible inspection of short duration. To this end, it is desirable to have both a small number of sensor views, and for the path connecting these views to be as short in length as possible. These objectives are addressed separately in our respective CSP and MPP sub-problems. Dividing a traveling-and-covering optimization into two sub-problems makes available a wide variety of fast, effective heuristics each. This strategy has been successful in practice [8], producing poor-quality solutions only when the robot's sensor has an infinite field of view [22]. In our problem of interest, however, the HAUV must inspect an expansive structure with a field of view limited to three meters in range.

A. Sampling-Based Sweep Paths

As mentioned above, a sweep path is not required to cover 100% of the surface segment it is inspecting; the goal is instead to exploit the open, planar areas of the structure wherever possible using a simple trajectory. Using a sampling-based method to achieve this goal reduces the amount of geometric computation required. We can avoid the explicit construction of the robot configuration space, which, for the HAUV, is comprised of four degrees of freedom, x,y,z, and yaw, and is populated with mesh models comprised of hundreds of thousands of geometric primitives. In addition, as we demonstrate below, a cell decomposition is not required to fit a long, efficient sweep path in the obstacle-free areas of configuration space; this is achieved instead using random sampling.

1) Set System Preliminaries: We will analyze Phase I of the CSP using *set systems*, a modeling framework that has proven useful in sampling-based coverage problems [15], [11]. We first consider the set system $(P, \mathcal{Q})$, in which P is a finite set of geometric primitives p_i comprising a structure that that must be covered by the robot, and $\mathcal{Q}$ is the robot configuration space. Every feasible configuration $q_j \in \mathcal{Q}$ maps to a subset of P viewed by the robot's sensor. Given a finite set of configurations $\mathcal{Q}$, the *set cover problem* calls

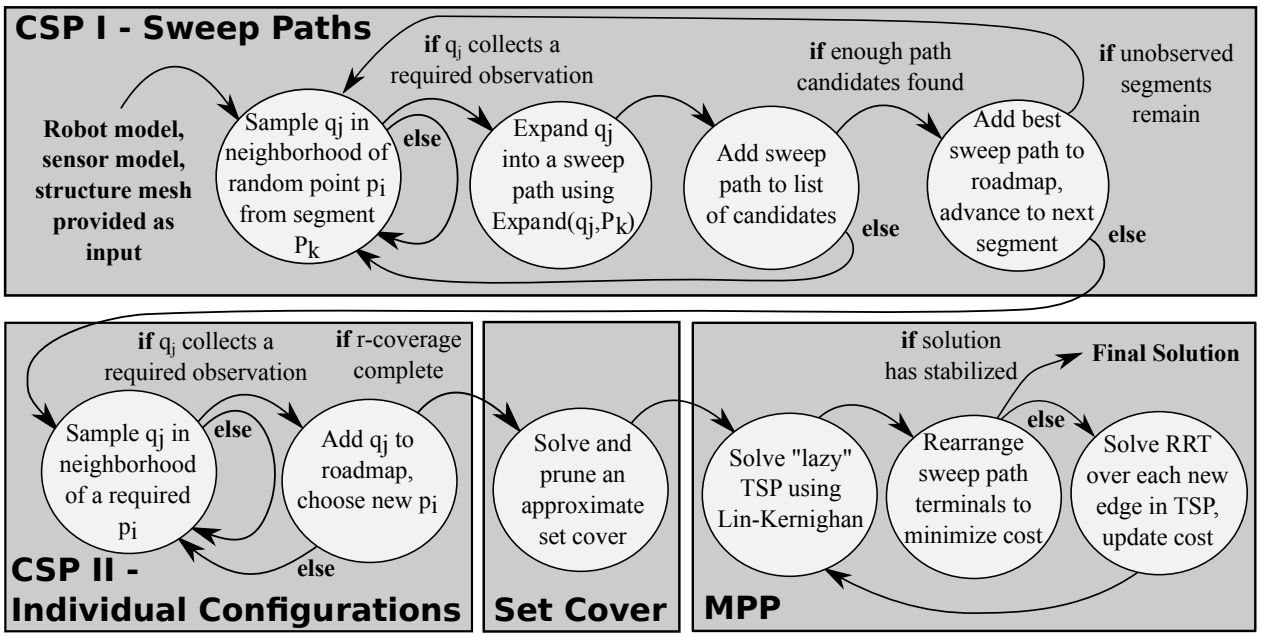


Fig. 3. A stateflow diagram illustrating the complete algorithm for sampling-based coverage path planning, comprised of a coverage sampling phase to generate sweep paths, a coverage sampling phase to fill in the remaining gaps in coverage, a set cover phase, and a multigoal planning phase. The algorithm used in CSP II was developed in prior work and has been used previously for planning full inspections of 3D structures [10].

for the minimum number of configurations q_j such that all elements $p_i \in P$ are covered.

The problem can also be modeled using the *dual set system* $(\mathcal{Q}, \mathcal{S})$, where $S_i \in \mathcal{S}$ is the set of feasible robot configurations in $\mathcal{Q}$ that obtain views of the primitive $p_i \in P$. Given a finite set of robot configurations from $\mathcal{Q}$, the *hitting set problem* calls for the minimum number of configurations q_j such that at least one configuration lies in every S_i for all $p_i \in P$. The components of the primal and dual set systems are illustrated in Figure 4, in which a subscript is added to P, $\mathcal{Q}$, and $\mathcal{S}$ to specify the mesh partition to which the primitives and configurations correspond. For a mesh segment k, P_k is the set of primitives contained in the segment, $\mathcal{Q}_k$ is the user-defined region of configuration space that is sampled to achieve views of P_k, and $\mathcal{S}_k$ is the set of all configurations that observe any $p_i \in P_k$.

2) Sweep Path Construction Algorithm: As illustrated at the top of Figure 3, after choosing a specific mesh segment k to cover, a point from this segment, p_i, is selected at random and configurations q_j are randomly sampled in a local neighborhood of p_i, such that p_i lies within the field of view of the sensor. This procedure, utilized in prior sampling-based coverage algorithms [9], [10], gives us the *seed configuration* from which a sweep path is produced. If q_j is free of collision and collects observations of segment k, then the subroutine $Expand(q_j, P_k)$ is called to expand q_j into a grid of waypoints.

Each waypoint grid is constructed in a 2D plane with a single yaw angle common to all waypoints, selected to capture mesh segment k in the sensor field of view. The plane is oriented using the distribution of points in mesh segment k, with the eigenvectors of the segment's statistical

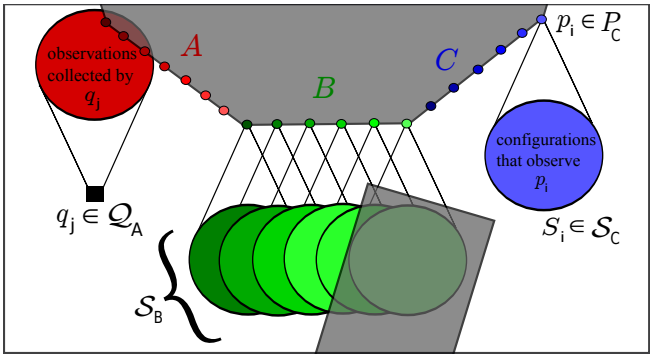


Fig. 4. An illustration of the set systems involved in the coverage sampling problem, for a robot with a circular sensor footprint capable of translational motion in $\Re^2$. In this example, the structure to be inspected is discretized and segmented into three pieces. One of the primitives in the green partition cannot be observed due to the presence of an obstacle.

covariance matrix comprising the axes for alignment. The waypoints in each grid are either depth-varying or fixed in depth depending on the orientation of the normal vectors in mesh segment k.

A user-specified spacing is enforced between waypoints when $Expand(q_j, P_k)$ expands a seed configuration into a waypoint grid. $Expand(q_j, P_k)$ is given in Algorithm 1; each subroutine attempts to add one extra row or column to the grid, separated by the designated spacing. Due to this systematic expansion procedure, the seed configuration q_j determines the layout of the entire grid.

Because it may not be possible for a grid to observe all primitives in segment k, we wish to identify the seed configurations whose grids, after expansion, observe the maximum-

Algorithm 1 $\mathcal{Q}_k = Expand(q_j, P_k)$

$\mathcal{Q}_k \leftarrow q_j$;
$ExpansionComplete = false$;
$SweepPlane = GeneratePlane(q_j, P_k)$;
while $!ExpansionComplete$ **do**
  $\mathcal{Q}_k^{right} \leftarrow ExpandRight(\mathcal{Q}_k, P_k, SweepPlane)$;
  $\mathcal{Q}_k \leftarrow \mathcal{Q}_k \cup \mathcal{Q}_k^{right}$;
  $\mathcal{Q}_k^{up} \leftarrow ExpandUp(\mathcal{Q}_k, P_k, SweepPlane)$;
  $\mathcal{Q}_k \leftarrow \mathcal{Q}_k \cup \mathcal{Q}_k^{up}$;
  $\mathcal{Q}_k^{left} \leftarrow ExpandLeft(\mathcal{Q}_k, P_k, SweepPlane)$;
  $\mathcal{Q}_k \leftarrow \mathcal{Q}_k \cup \mathcal{Q}_k^{left}$;
  $\mathcal{Q}_k^{down} \leftarrow ExpandDown(\mathcal{Q}_k, P_k, SweepPlane)$;
  $\mathcal{Q}_k \leftarrow \mathcal{Q}_k \cup \mathcal{Q}_k^{down}$;
  if $|\mathcal{Q}_k^{right} \cup \mathcal{Q}_k^{up} \cup \mathcal{Q}_k^{left} \cup \mathcal{Q}_k^{down}| = \emptyset$ **then**
    $ExpansionComplete = true$;
  end if
end while
return $\mathcal{Q}_k$

possible number of primitives in segment k subject to the presence of obstacles, occlusions, and the spacing enforced between waypoints. We are not concerned with growing the shortest-possible sweep path from $\mathcal{S}_k$, simply a feasible path. We use the notation $\mathcal{S}_k^*$ to describe the special subset of $\mathcal{S}_k$ from which a sampled configuration will generate a grid that satisfies the maximum-possible number of coverage constraints. $\mathcal{S}_k^*$ is depicted in Figure 5 for the coverage of segment B. It is evident that the rightmost mesh point in segment B is obscured from view by the presence of an obstacle, but any seed configuration in $\mathcal{S}_B^*$ will yield a single-row grid that observes the other five mesh primitives.

3) Probabilistic Completeness: Random sampling proves to be a powerful tool for finding a maximal-coverage feasible sweep path, and it motivates our definition of *probabilistic completeness*. This property originated in the context of point-to-point path planning [18], but we offer a definition in the context of sweep paths. We analyze probabilistic completeness with respect to a local set system, $(\mathcal{Q}_k, \mathcal{S}_k)$, that applies to a specific segment k. We define the property of probabilistic completeness for a CSP algorithm as follows.

Definition 1 (Probabilistic Completeness, CSP I): Let CSA be a proposed sweep-based coverage sampling algorithm for Phase I of the CSP. Let $\delta = \min_k \mu(\mathcal{S}_k^*)/\mu(\mathcal{Q})$ be the smallest maximal-coverage volume fraction of all segments k, where the measure μ represents the volume of the specified region of configuration space. If, when $\delta > 0$, the probability that at least one sample has landed in every $\mathcal{S}_k^*$ approaches one as the number of samples of $\mathcal{Q}$ drawn by CSA approaches infinity, then CSA is probabilistically complete.

This definition implies that a probabilistically complete CSP algorithm will, in the limit, find sweep paths that satisfy as many coverage constraints as possible while avoiding collision and obeying the rules of sweep path construction. This definition is intended to eliminate degenerate scenarios from

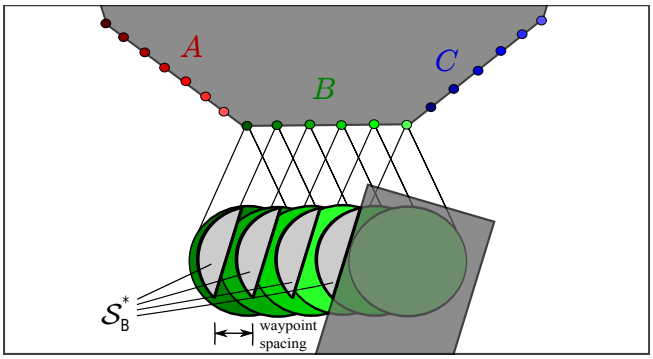


Fig. 5. An illustration of additional set system nomenclature for a robot with a circular sensor footprint capable of translational motion in $\Re^2$. The set of configurations that map to a maximally informative sweep path are depicted for segment B. One of the primitives in the green partition cannot be observed due to the presence of an obstacle.

consideration in which $\mathcal{S}_k^*$ is a manifold of lower dimension than $\mathcal{Q}$. By relaxing additional coverage constraints, it is possible that a degenerate $\mathcal{S}_k^*$ can be replaced with a new set that achieves a nonzero volume fraction of $\mathcal{Q}$. We can further analyze probabilistic completeness by studying the simple event of whether a randomly sampled configuration q_j lands in a particular set $\mathcal{S}_k^*$.

Theorem 1 (Completeness & Convergence, CSP I): Any algorithm for Phase I of the CSP that samples uniformly at random from all $\mathcal{Q}_k$ such that $\mu(\mathcal{S}_k^*)/\mu(\mathcal{Q}_k) \geq \epsilon > 0 \; \forall k$ is probabilistically complete. Additionally, the probability that a solution has not been found after m samples of each $\mathcal{Q}_k$ is bounded such that

$$Pr[FAILURE] \leq \frac{K}{e^{\,m\epsilon}} \quad , \tag{1}$$

where K is the number of partitions into which P is segmented.

Proof: The probability of m samples of each $\mathcal{Q}_k$ producing a maximal-coverage CSP solution is equivalent to the probability that at least one random sample has landed in every set $\mathcal{S}_k^*$. This fails to occur if there is at least one $\mathcal{S}_k^*$ in which no samples have landed. To model this event, we define the binomial random variable $X_k = X_{k_1} + X_{k_2} + \ldots + X_{k_m}$, which gives the number of samples that have successfully landed in $\mathcal{S}_k^*$ out of m total trials. We express the probability of CSP algorithm failure as follows:

$$\begin{aligned} Pr[FAILURE] &\leq Pr\left[\bigcup_{k=1}^{K} X_k = 0\right] \\ &\leq \sum_{k=1}^{K} Pr[X_k = 0] \\ &\leq K \cdot Pr[X_{k^*} = 0] \end{aligned} \tag{2}$$

Using the union bound, the probability that $X_k = 0$ for at least one $\mathcal{S}_k^*$ is bounded above by the sum of the probabilities of this event for all $\mathcal{S}_k^*$. This is further simplified by taking $Pr[X_{k^*} = 0]$ as an upper bound on the failures of all X_k, where X_{k^*} is the binomial random variable corresponding to the segment k that minimizes $\mu(\mathcal{S}_k^*)/\mu(\mathcal{Q}_k)$.

Since we are sampling uniformly at random, $Pr[X_{k^*} = 0]$ can be expressed and bounded in the following way:

$$Pr[X_{k^*} = 0] = (1 - \epsilon)^m \leq e^{-m\epsilon}, \quad 0 \leq \epsilon \leq 1 \qquad (3)$$

Combining the result of (3) with (2), we obtain the desired relationship between m and the probability of failure:

$$Pr[FAILURE] \leq \frac{K}{e^{\,m\epsilon}}, \quad \lim_{m \to \infty} \frac{K}{e^{\,m\epsilon}} = 0 \qquad (4)$$

Since $\mu(\mathcal{S}_k^*)/\mu(\mathcal{Q}_k) > 0 \; \forall k$, $\epsilon > 0$ and the limit behaves as indicated in (4). ■

As a direct consequence of Theorem 1, our algorithm for Phase I of the CSP illustrated in Figure 3 is probabilistically complete if the $\mathcal{Q}_k$ are selected to allow $\epsilon > 0$ whenever $\delta > 0$. By iteratively choosing a random $p_i \in P_k$ and sampling from the region of $\mathcal{Q}$ in which p_i lies in the sensor's geometric footprint, we are sampling from a subset of $\mathcal{Q}$ which fully includes $\mathcal{S}_k^*$ and the condition on ϵ and δ is always satisfied. The bounding methods used in this analysis were used previously in the proof of completeness of the probabilistic roadmap (PRM) [16] to analyze the failure of m samples of a common configuration space to construct a collision-free path between two configurations. We have applied the same bounds here to analyze the failure of m samples of each $\mathcal{Q}_k$ to land at least once in every set $\mathcal{S}_k^*$.

B. Filling in the Gaps

To fill in the remaining gaps in coverage left by the sweep paths, we rely on individual robot configurations rather than waypoint grids. This sub-problem comprises Phase II of the CSP as illustrated in Figure 3. To solve this problem, we utilize the sampling method of the redundant roadmap coverage path planning algorithm [10], which samples robot configurations until a set is obtained that views each required geometric primitive from r distinct configurations, termed r-coverage in Figure 3. From these configurations, a subset is selected for traversal by approximation of the minimum-cardinality set cover. In Phase II of the CSP, we apply the sampling routine of the redundant roadmap algorithm only to primitives left unobserved by the sweep paths designed in Phase I.

This sampling routine is also probabilistically complete. If a feasible, 100%-coverage set of configurations exists for the remaining primitives, then the redundant roadmap algorithm will find such a solution with probability that tends to one as the number of samples tends to infinity. We omit formal definitions and proofs for Phase II of the CSP since the definitions and proof of completeness in [11] apply directly to this sub-problem. Using this prior result, we state the convergence of algorithm failure probability as a function of the number of samples m, the volume fraction ϵ of the configuration space that is sampled, and $|P|_{gaps}$, the number of primitives comprising the gaps in coverage remaining at the beginning of Phase II.

$$Pr[FAILURE] < |P|_{gaps} \cdot \frac{e^r}{e^{\,m\epsilon/2}} \qquad (5)$$

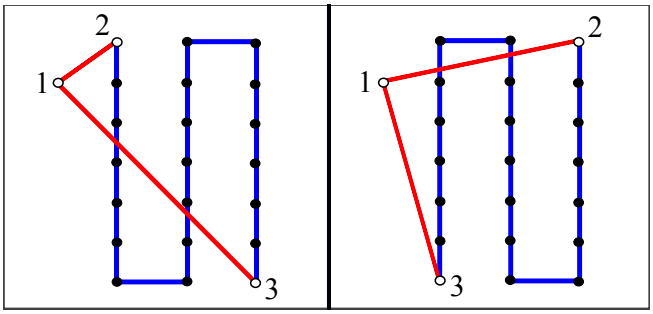


Fig. 6. A diagram illustrating the integration of back-and-forth sweep paths into the TSP. At left, one possible choice of sweep path is depicted, and at right, the alternate choice is depicted. Each choice results in a different set of terminals being used to connect the sweep path to the rest of the inspection tour. For each choice of terminals, the red lines and numbered points represent edges and nodes that are introduced into the TSP. The blue lines represent the sweep path, which is omitted from the TSP and represented by a zero-cost edge between the two terminals.

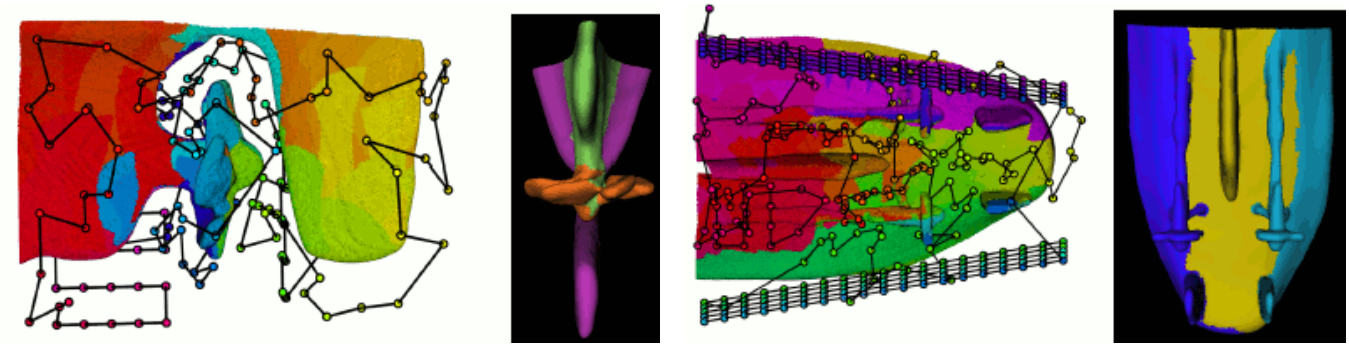
The coefficients in (5) differ slightly from (1) because the Phase II sampling process must achieve r-coverage, as opposed to single-coverage. Despite the minor differences between (1) and (5), both phases of the coverage sampling problem are solved by algorithms for which the probability of failure plunges toward zero exponentially fast in the number of robot configurations sampled.

III. COMPUTING A HYBRID INSPECTION TOUR

Phases I and II of the CSP yield a set of feasible robot configurations that observe 100% of the structure boundary. Part of this set is comprised of waypoint grids, which form the basis for back-and-forth sweep paths. The remainder of the set is comprised of individual robot configurations that fill in the gaps in coverage left by the waypoint grids. Before constructing an inspection route among these configurations, we apply a set cover approximation to both the sweep-path subset and gap-filling subset, followed by iterative pruning of each set cover solution. After the set cover step, an order of traversal among waypoints is computed by reduction to a symmetric instance of the TSP, and collision-free paths are iteratively computed among the waypoints in the inspection route.

A. Set Cover Sub-Problem

The set cover problem is solved twice; once over the K sweep paths and once over the individual configurations that fill the remaining gaps in coverage. In the former case, each sweep path is treated as an individual set, and in the latter case, each robot configuration is treated as an individual set. Each set cover is posed over the specific group of geometric primitives required in the respective phase of the CSP. In both cases, the greedy algorithm [21] is used to give a polynomial-time approximation to the minimum-cardinality set cover. The greedy algorithm returns a feasible solution, but this solution may contain sets that can be eliminated completely while preserving feasibility. A polynomial-time pruning procedure [10] is implemented to remove sets which cover no elements uniquely. In the case of the sweep paths,



(a) Segments: 3, Configurations: 112 random, 57 sweep, Length: 241m

(b) Segments: 3, Configurations: 145 random, 202 sweep, Length: 402m

(c) Segments: 10, Configurations: 36 random, 282 sweep, Length: 383m

(d) Segments: 20, Configurations: 84 random, 377 sweep, Length: 526m

Fig. 7. Examples of planned inspection tours for both the *SS Curtiss* and *USCGC Seneca*, each for two different segmentation test cases. For the *SS Curtiss*, three-segment and ten-segment tours are shown, and for the *USCGC Seneca*, three-segment and twenty-segment tours are shown. In each subfigure, the image at right illustrates the segmentation only, and the image at left illustrates the full-coverage inspection tour. We list the number of randomized configurations in each tour, the number of sweep configurations, and the total tour length. The waypoints are color-coded and map to the colored patches on the mesh surface; these patches represent the sensor observations collected at each waypoint. The changes in color occur gradually and follow the sequence of the inspection tour.

the pruning procedure is also applied to the individual rows and columns of each waypoint grid, and in each iteration the obsolete row or column with the largest number of waypoints is eliminated. This allows redundant waypoints to be eliminated from the sweep paths while preserving their rectangular structure.

B. Traveling Salesman Sub-Problem

Our aim is to solve the TSP over a graph containing sweep paths without re-computing the order of traversal within the sweep paths themselves. After choosing an entry and exit point, the order of traversal within a sweep path is trivial, as depicted in Figure 6. Consequently, we reduce each sweep path in the set cover to a single pair of graph nodes in the TSP, representing the points of entry and exit. To ensure that this pair of sweep path *terminals* appear adjacent to one another in the final TSP tour, the costs of travel between other configurations are augmented. A cost of zero is assigned to every edge connecting a pair of terminals, and all other node-to-node costs are initialized using the Euclidean distances between robot configurations. A large number is then added to the costs of all Euclidean-distance edges. This large number, selected to be larger than any true path length that will be returned as a solution to the TSP, will ensure that pairs of terminals remain adjacent in the final TSP tour. We are not aware of prior work on the topic of forcing a pair of TSP nodes to be adjacent. After this initialization, the TSP is solved using the chained Lin-Kernghan heuristic [1].

Even though a pair of terminals is selected for each sweep path in advance of solving the TSP, it is possible that the alternate pair of entry and exit terminals will yield a shorter inspection tour, as demonstrated in Figure 6. To address this possibility, we consider alternate choices of sweep path terminals and switch them after solution of the TSP if the alternate terminals lower the cost of the tour. We iterate through the sweep paths in round robin fashion, and stop once a single pair of terminals is adjusted. This adjustment requires an update of the node-to-node distances in the adjacency matrix used for TSP computations.

Once a sequence of waypoints is selected, every pair of adjacent nodes in the traveling salesman tour is fed to a bi-directional rapidly-exploring random tree (RRT) [17], which finds collision-free paths among the waypoints in the TSP candidate solution. The node-to-node distances in the TSP adjacency matrix are updated to reflect the cost of collision-free paths between waypoints. The full TSP computation is then repeated, and node-to-node distances are iteratively updated until a stable solution is found. The multi-goal planning problem (MPP) iterative procedure is summarized

in Figure 3. This procedure, which relies on lazy collision-checking, has been used previously to solve the TSP in the presence of obstacles [19], [10].

Our terminal-switching heuristic is intended to avoid the complexity of examining, in every iteration of the MPP procedure, all 2^K combinations of terminals over K sweep paths. Despite our simplification of the problem, even the proposed heuristic risks the worst-case scenario of an MPP procedure that marches through every one of these 2^K combinations while approaching a stable solution. However, this would only occur in the unlikely scenario that every combination makes incremental progress toward a single optimal solution. We have found the heuristic MPP procedure to converge quickly in practice, but a time limit, a ceiling on the allowed number of MPP iterations, or a stopping criterion based on the cost improvement of the MPP procedure could easily be imposed.

IV. COMPUTATIONAL RESULTS

We now give computational results of the sampling-based sweep path algorithm as applied to the HAUV. Two mesh models obtained from HAUV experimental data are used in support of this coverage planning task. The first model represents the stern of the *SS Curtiss*, an aviation logistics support ship with a single propeller seven meters in diameter. The second model represents the stern of the *USCGC Seneca*, a Coast Guard Cutter with two propellers that are 2.5 meters in diameter. Although smaller than the *Curtiss*, the protruding structures of the *Seneca* pose a more confined and occluded coverage problem. Both meshes are pictured in Figure 7. They are discretized finely enough that observation of all mesh vertices will yield coverage sufficient to detect a 0.1-m mine anywhere on the surface. The HAUV will sweep the structures using a sensor viewing range of 1-3m, which gives high-resolution range scans within ± 15 degrees of vehicle-relative bearing, pitched up and down through ± 90 degrees. This is a small sensing volume relative to the size of the structures being inspected, and conservative waypoint spacing must be used to prevent the occurence of gaps in the data collected while sweeping over open and planar areas.

In addition to the need for heuristic design of waypoint spacing, we must decide how many partitions are appropriate in the segmentation of both structures. To explore the effect of this parameter, we have computed planned inspection paths over both ships for a variety of segmentations, from the trivial case of a fully randomized inspection (an order-zero segmentation), to a segmentation of order twenty. This was performed using EfPiSoft (`http://efpisoft.sourceforge.net/`), an implementation of a hierarchical face clustering algorithm [4] that we have employed to select segments based on their quality of fit to a plane. It is also possible to select segments based on their quality of fit to spherical and cylindrical primitives, but we have found spherical and cylindrical sweep paths to be less suitable for generalized inspection of the open areas of man-made structures. Given a mesh segmentation as input, our sweep path algorithm carries out random sampling until ten feasible

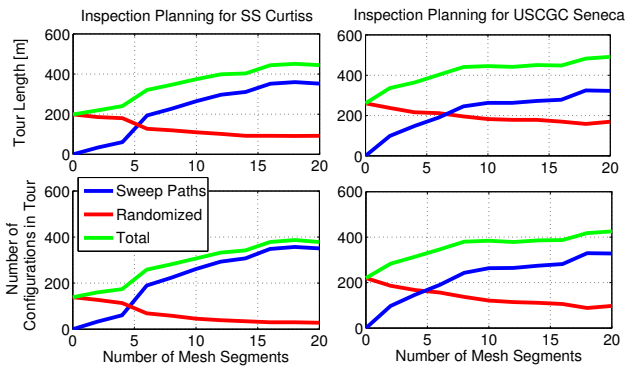


Fig. 8. Results of sweep-based inspection planning on two vessels, the *SS Curtiss* and *USCGC Seneca*, over different segmentation cases. These range from the trivial case of a fully randomized inspection (zero segments) to nontrivial cases with up to twenty partitions. The results give the mean inspection tour length over 25 trials and the mean number of configurations (waypoints) in the inspection for each test case. In blue, we plot the length of the tour contributed internally by all sweep paths. Blue also represents the number of sweep path configurations. In red, we plot the length of the tour required for interconnections among separate sweep paths and single configurations. Red also represents the number of single configurations. The sum total of these quantities is plotted in green.

candidate sweep paths are achieved for each segment, and the paths offering the most comprehensive coverage of their respective segments are used in the inspection. We proceed this way in practice since we do not know exectly when the maximal-coverage set $\mathcal{S}_k^*$ is reached.

After the sweep paths are generated, the remaining gaps in coverage are filled and an inspection tour is planned using the redundant roadmap algorithm [10] and the software tools cited in this publication, which are also used for collision checking and ray shooting in our sweep path algorithm. The gaps in coverage are filled using redundancy-three roadmaps, which must observe all required geometric primitives from three distinct sampled configurations. In each iteration of the MPP, a TSP tour is initialized using the nearest-neighbor heuristic and the chained Lin-Kernighan software provided with the concorde TSP solver (`http://www.tsp.gatech.edu/concorde.html`) is applied for one second, although sometimes only milliseconds are needed to make significant improvements to the TSP tour. All computations were performed on a computer with a 3.20 GHz processor and 24 GB RAM running the Linux operating system, and no single instance of the full planning algorithm required more than ten minutes of computation time for the structures tested.

Due to the uniform spacing and fixed orientation of sweep path waypoints, HAUV trajectories that utilize sweep paths will suffer a loss in efficiency to exchange randomized inspection routes, which accomodate every unique twist and turn in the structure, for highly regularized inspection routes. This loss in efficiency impacts both the number of configurations used in the inspection and the distance traveled by the vehicle along the inspection tour. By planning for HAUV coverage of a large trivially-segmented cube, the loss of efficiency was determined to be a factor of two for inspection

tour length and a factor of 2.5 for the number of waypoints in an idealized inspection route for which nearly 100% of wayponts lie in sweep paths. These losses were matched and exceeded in some cases by the planned coverage paths for the *Curtiss* and *Seneca*, which were planned over a wide range of mesh segmentations. Figure 8 demonstrates these results, which illustrate the proportion of each planned inspection comprised of regularized and randomized configurations. As the quantity of segments increases, larger proportions of the tour are solved by sweep paths. This is accompanied by a net increase in length of the tours and the number of total configurations, with a decrease in the number of randomized configurations. The effect of higher-order segmentations on the decrease in randomized configurations is observed to diminish as the number of segments increases.

This diminishing-returns effect is especially evident for the *Curtiss*, which is covered almost entirely by sweep paths using an order-ten segmentation, pictured in Figure 7. As illustrated in Figure 8, increasing the order of the segmentation beyond ten has only a minor effect on the number of randomized configurations, while it increases the total length of the tour significantly. The *Seneca*, on the other hand, still requires a significant number of randomized configurations for an order-twenty segmentation. The *Seneca* has a larger number of protruding component structures, and many additional planes are needed to observe these structures from all sides. The coverage path planned for the order-twenty segmentation pictured in Figure 6 uses planar sweep paths to observe both sides of the keel, both sides of each rudder, one side of each shaft, and the faces of the propellers. Although there is no single, correct choice of an optimal segmentation, it is clear that different structures will require subdivisions of differing complexity to approach full coverage with regularized sweep paths.

We also note that we have developed a sampling-based improvement procedure for randomized coverage paths that is omitted from this study [11]. This procedure would further improve the relative efficiency of the randomized portions of the inspection paths, but it requires significant added computation time to achieve high-quality smoothing and was not feasible to include in an ensemble trial of the type depicted in Figure 8.

V. CONCLUSION

We have presented an algorithm which, to our knowledge, is the first coverage planning algorithm that utilizes both randomized and regularized component paths to achieve coverage of complex 3D structures. The component paths are joined seamlessly into a single contiguous inspection tour. Given a segmented structure as input, a back-and-forth sweep path is designed for coverage of each segment. A probabilistically complete sampling procedure establishes the origin of each sweep path. This procedure is designed to cover the open, easily accessible areas of a structure using simple paths that yield easy-to-interpret sensor data. Randomized paths are used to inspect the confined, highly-occluded areas of a structure that elude the sweep paths. To minimize the number of random configurations, a loss in efficiency must be accepted in the substitution of uniformly spaced waypoint grids for individually designed single waypoints. This tradeoff is often desirable when the ability to monitor, interpret, and intervene in an inspection-in-progress is a key requirement, and our algorithm offers the flexibility to "trade" for increased regularity as needed.

REFERENCES

[1] D. Applegate, W. Cook, and A. Rowe, "Chained Lin-Kernighan for Large Traveling Salesman Problems," *INFORMS J. Computing*, vol. 15(1), 2003, pp. 82-92.

[2] P.N. Atkar, H. Choset, A.A. Rizzi, and E.U. Acar, "Exact Cellular Decomposition of Closed Orientable Surfaces Embedded in $\Re^3$", *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, Seoul, South Korea, vol. 1, 2001, pp. 699-704.

[3] P.N. Atkar, A. Greenfield, D.C. Conner, H. Choset, and A.A. Rizzi, "Uniform Coverage of Automotive Surface Patches," *Int. J. Robotics Research*, vol. 24(11), 2005, pp. 883-898.

[4] M. Attene, B. Falcidieno, and M. Spagnuolo, "Hierarchical Mesh Segmentation Based on Fitting Primitives," *The Visual Computer*, vol. 22(3), 2006, pp. 181-193.

[5] P. Cheng, J. Keller, and V. Kumar, "Time-Optimal UAV Trajectory Planning for 3D Urban Structure Coverage," *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Nice, France, 2008, pp. 2750-2757.

[6] H. Choset and P. Pignon, "Coverage Path Planning: The Boustrophedon Decomposition," *Proc. Int. Conf. on Field and Service Robotics (FSR)*, Canberra, Australia, 1997.

[7] H. Choset, "Coverage for Robotics: A Survey of Recent Results," *Annals of Mathematics and Artificial Intelligence*, vol. 31, 2001, pp. 113-126.

[8] J.R. Current and D.A. Schilling, "The Covering Salesman Problem," *Transportation Science*, vol. 23(3), 1989, pp. 208-213.

[9] T. Danner and L. Kavraki, "Randomized Planning for Short Inspection Paths," *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, San Francisco, CA, 2000, pp. 971-976.

[10] B. Englot and F. Hover, "Planning Complex Inspection Tasks Using Redundant Roadmaps," *Proc. Int. Symp. on Robotics Research (ISRR)*, Flagstaff, AZ, 2011.

[11] B. Englot and F.S. Hover, "Sampling-Based Coverage Path Planning for Inspection of Complex Structures," *Proc. Int. Conf. on Automated Planning and Scheduling (ICAPS)*, Sau Paulo, Brazil, 2012, pp. 29-37.

[12] F.S. Hover, J. Vaganay, M. Elkins, S. Wilcox, V. Polidoro, J. Morash, R. Damus, and S. Desset, "A Vehicle System for Autonomous Relative Survey of In-Water Ships," *Marine Technology Society Journal*, vol. 41(2), 2007, pp. 44-55.

[13] W.H. Huang, "Optimal Line-Sweep-Based Decompositions for Coverage Algorithms," *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, Seoul, South Korea, vol. 1, 2001, pp. 27-32.

[14] V. Isler, S. Kannan, and K. Daniilidis, "Sampling-Based Sensor Network Deployment," *Proc. IEEE/RSJ Int. Conf. on Intelligent Robotics and Systems (IROS)*, Sendai, Japan, vol. 2, 2004, pp. 1780-1785.

[15] L.E. Kavraki, M.N. Kolountzakis, and J.-C. Latombe, "Analysis of Probabilistic Roadmaps for Path Planning," *IEEE Trans. on Robotics and Automation*, vol. 14(1), 1998, pp. 166-171.

[16] J.J. Kuffner and S.M. LaValle, "RRT-Connect: An Efficient Approach to Single-Query Path Planning," *IEEE Int. Conf. on Robotics and Automation (ICRA)*, San Francisco, CA, vol. 2, 2000, pp. 995-1001.

[17] F. Lamiraux and J.P. Laumond, "On the Expected Complexity of Random Path Planning," *IEEE Int. Conf. on Robotics and Automation (ICRA)*, Minneapolis, MN, vol. 4, 1996, pp. 3014-3019.

[18] M. Saha, T. Roughgarden, J.-C. Latombe, and G. Sanchez-Ante, "Planning Tours of Robotic Arms Among Partitioned Goals," *Int. J. Robotics Research*, vol. 25(3), 2006, pp. 207-223.

[19] T.C. Shermer, "Recent Results in Art Galleries," *Proc. IEEE*, vol. 80(9), 1992, pp. 1384-1399.

[20] V.V. Vazirani, *Approximation Algorithms*, Springer, Berlin, Germany, 2001.

[21] P. Wang, R. Krishnamurti, and K. Gupta, "View Planning Problem with Combined View and Traveling Cost," IEEE Int. Conf. on Robotics and Automation (ICRA), Rome, Italy, 2007, pp. 711-716.