Planning Complex Inspection Tasks Using Redundant Roadmaps

Brendan Englot and Franz Hover

Abstract The aim of this work is fast, automated planning of robotic inspections involving complex 3D structures. A model comprised of discrete geometric primitives is provided as input, and a feasible robot inspection path is produced as output. Our algorithm is intended for tasks in which 2.5D algorithms, which divide an inspection into multiple 2D slices, and segmentation-based approaches, which divide a structure into simpler components that are solved individually, are unsuitable. This degree of 3D complexity has been introduced by the application of autonomous in-water ship hull inspection, in which protruding structures at the stern (propellers, shafts, and rudders) are positioned in close proximity to one another and to the hull, and clearance is an issue for a mobile robot. A global, sampling-based approach is adopted, in which all the structures are simultaneously considered in planning a path. First, the obstacle-filled state space of the robot is discretized by constructing a roadmap of feasible states; construction ceases when each primitive is observed by a specified number of states. Once a roadmap is produced, the set cover problem and traveling salesman problem are approximated in sequence to build a feasible inspection tour. We analyze the performance of this procedure in solving one of the most complex inspection planning tasks to date, covering the stern of a 200-meter naval ship, using an *a priori* polygonal mesh model obtained from real sonar data. Our algorithm generates paths on a par with dual sampling, at about sixty percent the computational effort.

1 Introduction

A variety of autonomous surveillance, inspection, and distribution tasks can be solved using coverage path planning. Given an accurate model of the environment,

Brendan Englot, Franz Hover
Massachusetts Institute of Technology, 77 Massachusetts Ave, Cambridge, MA, 02139, e-mail: benglot@mit.edu, hover@mit.edu

a path is designed in which an agent sweeps its geometric footprint over 100% of a required surface area. Manufacturing operations, security and maintenance inspections, painting, plowing, cleaning, environmental monitoring and mine-sweeping are a few of the many applications in which coverage path planning enables faster task completion compared with greedy or next-best-view strategies [7], [29].

In 2D workspaces with obstacles, cellular decomposition methods divide the free space into simple, easily-covered pieces [6],[8], allowing a full sweep of the open area. Alternatively, some applications call for the inspection of structure boundaries, and both deterministic (using Voronoi diagrams) [12], [32] and randomized (sampling-based) approaches [11], [16] have been used.

In 3D workspaces, the inspection task is typically one of boundary coverage. A structure is represented by a two-dimensional closed surface embedded in $\Re^3$, and the sensor must sweep over 100% of the interior or exterior surface area. This problem is often solved by partitioning a 3D structure and planning individual inspection paths for separate components. In a 2.5D approach, the workspace is divided into 2D cross-sections, and planned paths over these cross-sections are assembled into a full 3D inspection [2], [16]. If a complex structure is comprised of distinct 3D components, one can plan individual inspection paths for each of them, assuming there is no risk of collision with neighboring components. This approach has been applied to painting the exterior surfaces of a car [3] and inspecting buildings in an urban environment [2]. In the former case, a segmentation algorithm automatically partitioned the car into topologically simple surfaces, and each was covered individually using a lawnmower-type trajectory [4]. In the latter case, each building was treated as an individual planning problem, and neighboring buildings were ignored (this required sufficient clearance between buildings). The key enabler for these modular approaches is that the plan for covering any one partition, component, or cross-section can be developed with no knowledge of the others.

Our coverage application is the autonomous in-water inspection of a ship hull, a 3D structure with challenging complexity at the stern due to shafts, propellers, and rudders in close proximity to one another and to the hull. The Bluefin-MIT Hovering Autonomous Underwater Vehicle [19], pictured in Figure 3, is tasked with inspecting 100% of the surface area at the stern using a forward-looking bathymetry sonar. Our vehicle is fully actuated and precision-maneuverable, but it cannot fit into the spaces between the component structures at the stern. If a 2.5D approach is adopted for coverage planning, it will need to be augmented with special, out-of-plane views in this problem to grant visibility of confined areas that are occluded in-plane. If a 3D modular approach is implemented, paths planned for component structures are at risk of collision with neighboring structures.

In consideration of these factors, we take a global optimization approach, in which all 3D protruding structures are considered simultaneously. The constraints are determined by the geometry of the 3D model provided as input. We use a triangle mesh, typically comprised of thousands of primitives, to accurately model a ship's running gear. Rather than explicitly optimizing robot configurations over the thousands of collision and visibility constraints posed by such geometry, sampling-based

planning is used to find feasible means for the robot to peer into the low-clearance areas from a distance [23].

Sampling-based planning was first applied to a coverage problem by Gonzalez-Baños and Latombe [16], [17], who used random sampling to construct a solution to the 2D art gallery problem [30]. This method was recently utilized to achieve 2D view-planning for a laser-equipped wheeled robot [5]. The method was also extended to path planning in work by Danner and Kavraki, who approximated the traveling salesman problem (TSP) over the solution to the art gallery problem, planning inspections for complex 2D structures and for 3D cubes and pyramids [11].

We extend this work in several ways to enable sampling-based coverage path planning over complex, real-world 3D structures. We construct and analyze computationally a *redundant roadmap*, in which each geometric primitive is observed by multiple robot states. To enable fast planning over a large roadmap, tools from multi-robot [28] and multi-goal [27] planning are utilized to enable lazy collision-checking. The roadmap construction and collision-checking procedures are discussed in Section 2.

In Section 3 we discuss the methods by which the set cover problem (SCP) and TSP are approximated in sequence to build an inspection tour from a redundant roadmap. We compare two fast SCP approximation algorithms, the greedy algorithm [20], [25] and the linear programming rounding algorithm [18], with the dual sampling method of Gonzalez-Baños and Latombe.

In Section 4 we examine algorithm performance over ensembles of Monte Carlo trials in which randomly-sampled primitives must be inspected by a point robot in a 3D workspace. For simplicity, this workspace is devoid of obstacles. Finally, in Section 5 we apply the inspection-planning algorithm to a large-scale, real-world task, planning the inspection of a ship hull by the HAUV.

2 Sampling-Based Planning Procedure

In developing an inspection path, we employ two sampling-based routines. First, roadmap construction, which samples robot configurations and catalogs their sensor observations, creates a discrete state space from which the inspection path will be made. Second, a point-to-point planner, capable of finding collision-free paths for multi-degree-of-freedom robots in obstacle-filled workspaces, finds feasible paths joining the configurations on the roadmap. A stateflow diagram summarizing the coverage path planning procedure from start to finish is given in Figure 1.

2.1 Roadmap Construction

The roadmap serves as a discrete proxy for the robot's continuous state space, as well as a mapping from the state space onto the surfaces and geometric primitives of

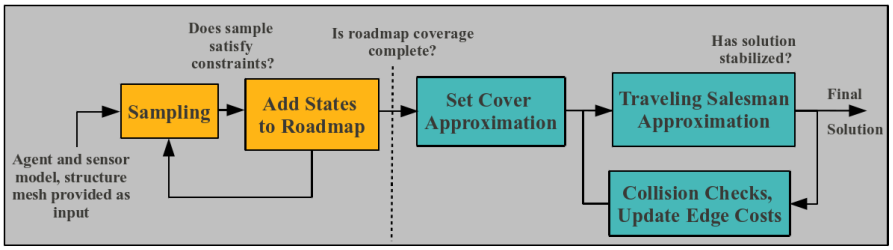

Fig. 1 A stateflow diagram illustrating the coverage path planning procedure from start to finish. Construction of a roadmap is illustrated in orange, at left, and construction of a feasible path is illustrated in blue, at right.

the structures being inspected. Rather than compute an inverse kinematic mapping from the geometric primitives to the robot's state space, which requires expensive and repeated computation of a Jacobian pseudoinverse, we rely on forward kinematics only. Robot configurations are sampled uniformly at random, and observed primitives are catalogued. To prevent the roadmap from growing too large in size, configurations are only added to the roadmap if they fall within the redundancy criteria.

Our roadmap construction procedure is detailed in Algorithm 1. The parameter *Redundancy* is an integer value which indicates the minimum number of sightings of any geometric primitive that can exist in the completed roadmap. Increased redundancy is intended to create a finely discretized state space from which a smaller covering subset of robot states will ultimately be chosen. Once a primitive has been observed a number of times equal to the specified redundancy, a new robot configuration cannot be added to the roadmap for a sighting of this primitive alone. All sensor observations of configurations added to the roadmap are stored and none are discarded, even if an individual primitive sighting exceeds the required redundancy.

This procedure stands in constrast to that of prior work in sampling-based coverage [11], [16], [17], in which a "dual sampling" approach is preferred over one that constructs a full-coverage roadmap. In these works, an unobserved geometric primitive is chosen at random, and a specified number of robot configurations is sampled from a local neighborhood. The configuration that collects the largest quantity of new sensor information is selected as a waypoint for the inspection tour. This method does not require the solution of the SCP over the roadmap, but it does require the computation of a state space region from which a specific geometric primitive is visible (which may be expensive in non-Euclidean state spaces). We will compare our approach to instances of the dual sampling method in which a solution can be obtained in equivalent computation time.

2.2 Lazy Point-to-Point Planning

Algorithm 1 $ConfigList = BuildRoadmap(Primitives, Obstacles, Redundancy)$

```
IncompletePrimitives ← Primitives
while IncompletePrimitives ≠ ∅ do
    NewConfig ← FeasibleSample(Obstacles)
    NewSightings ← Sensor(NewConfig, Primitives, Obstacles)
    NeededSightings ← NewSightings ∩ IncompletePrimitives
    if NeededSightings ≠ ∅ then
        ConfigList.add(NewCfg, NewSightings)
        for i ∈ NeededSightings do
            NeededSightings[i].incrementNumSightings()
            if NeededSightings[i].numSightings = Redundancy then
                IncompletePrimitives ← IncompletePrimitives \ NeededSightings[i]
            end if
        end for
    end if
end while
return ConfigList
```

Algorithm 2 $RobotTour = LazyTourAlgorithm(Nodes, Obstacles)$

```
AdjMat ← EuclideanDistances(Nodes)
UnclearedEdges ← GetEdgePairs(Nodes)
ClearedEdges ← ∅
while NewTourCost ≠ PreviousTourCost do
    PreviousTourCost ← NewTourCost
    NewTourCost ← 0
    LazyTour ← ComputeTour(AdjMat)
    for Edge_ij ∈ LazyTour do
        if Edge_ij ∈ UnclearedEdges then
            FeasiblePath_ij ← RRT(Edge_ij, Obstacles)
            ClearedEdges ← ClearedEdges ∪ Edge_ij
            UnclearedEdges ← UnclearedEdges \ Edge_ij
            AdjMat(i, j) ← PathCost(FeasiblePath_ij)
        end if
        NewTourCost ← NewTourCost + AdjMat(i, j)
    end for
end while
RobotTour ← LazyTour
return RobotTour
```

Efficient computation along roadmap edges is achieved with a lazy algorithm. As the roadmap is constructed, an adjacency matrix is maintained in which all entries represent the Euclidean norms among roadmap nodes. Computation of a Euclidean norm is far simpler than collision-checking and observation-checking along every edge of the roadmap. An initial inspection tour is computed over this naive adjacency matrix, and only the edges selected in the tour are collision-checked, not every edge of the roadmap. The bi-directional rapidly-exploring random tree (RRT) is utilized as the point-to-point planner [22]. Presumably, the computation of RRTs over the edges of the inspection tour increases the lengths of some edges. To address this, an iterative improvement procedure, similar to that of [27], is utilized.

After the first set of feasible paths is obtained, the costs in the adjacency matrix are updated, and the inspection tour is recomputed using the new costs. This procedure is repeated, and goal-to-goal costs are iteratively updated, until there is no further improvement in the length of the returned path. Instead of requiring $O(n^2)$ calls to the RRT, this approach requires $O(C*n)$ calls, where C is the number of iterations in which a new tour is computed. This procedure is detailed in Algorithm 2.

3 Constructing an Inspection Tour

An efficient implementation of the RRT subroutine is only useful if computations over the the adjacency matrix are fast and efficient. However, the exact problem we aim to solve, finding the shortest path that collects an element from every set (where the sets are observations of primitives obtained at each roadmap node) is an instance of the generalized traveling salesman problem (GTSP), which is NP-hard and has no constant-factor approximation. Although branch-and-cut algorithms [14] and reduction to a non-metric asymmetric TSP [24], [26] have been characterized, these are not suitable for an iterative, real-time procedure (neither is solved by an approximation algorithm with a guaranteed termination time). As of this writing, a constant-factor approximation can only be obtained if each roadmap node is limited to exactly two primitive sightings, in which case the problem reduces to a Tour Cover [1]. We have found that stripping sensor information out of the roadmap to achieve an equivalent Tour Cover undoes any benefit of a constant-factor approximation.

Our approach is similar to that suggested by Current and Schilling [10], in which a GTSP (referred to in their work as the Covering Salesman Problem, a special geometric case of the GTSP [15]) is solved by posing, in sequence, a SCP subproblem and a Euclidean TSP subproblem. Both the SCP and Euclidean TSP can be approximated to within a constant factor of optimality using fast, polynomial-time algorithms. Recent work on penalizing both viewing and traveling costs [31] addresses the possibility of an arbitrarily bad result if a global optimization is broken into separate SCP and TSP subproblems. Although this would be possible for a sensor model with infinite range, the inspection problems for which our algorithms are intended involve robots with decidedly finite sensing radii. Specifically, our application of interest employs a bathymetry sonar with a 4-meter sensing radius. The workspace is much larger than this, and thus we believe there is a strong correlation between the minimum-cardinality set cover and the minimum-cost GTSP.

3.1 Set Cover Subproblem

To solve the set cover subproblem, we rely on polynomial-time approximation algorithms that find solutions within guaranteed factors of optimality. We consider

two such algorithms, a greedy algorithm and a linear programming (LP) rounding algorithm. The greedy algorithm [20], [25] simply adds to the set cover, on each iteration, the roadmap node with the largest number of observed primitives not yet in the cover. This algorithm solves the SCP within a factor of optimality that is bounded above by $\ln(m)+1$, where m is the number of primitives required in the inspection. The rounding algorithm [18] solves the LP relaxation of the SCP, and then rounds the fractional solution according to a simple rule: if f is the largest number of roadmap nodes which share sightings of a primitive, then any roadmap node whose fractional decision variable is greater than or equal to $1/f$ is included in the cover. This method is guaranteed to return a solution within a factor f of optimality.

In the ship hull inspection example to be presented below, there are more than 10^5 primitives required in the inspection, giving a greedy algorithm approximation factor of about 12.5. At the same time, a typical value of f on a representative roadmap for this task is about twenty. Since these are both fast algorithms, and the approximation factors are of the same order, we will compare the two to assess their performance in practice.

3.2 Traveling Salesman Subproblem

To solve the TSP subproblem, we rely on another polynomial-time approximation. The algorithm of Christofides [9] computes the minimum spanning tree (MST) over a graph, and then a minimum-cost perfect matching over the odd-degree nodes of the MST, achieving an approximation factor of 1.5 when the triangle inequality holds over the roadmap. Although our lazy computation procedure may occasionally violate the triangle inequality, our use of RRT post-optimization and smoothing ensures that there are no paths from a roadmap node i to a roadmap node k such than an alternate path from i to some node j to k is dramatically shorter. This assumption has proven successful in MST-only variants (with factor-2) for single and multi-agent coverage planning [11], [13], as well as pure multi-goal planning [27].

4 Point Robot Test Case

First, we evaluate the performance of our inspection planning procedure on a point robot test case. This problem addresses algorithm performance as a function of the number of primitives, independent of collision and occlusion-checking. The unit cube is populated with a designated number of randomly sampled points, and the robot must plan a tour which observes them. Mimicking the HAUV inspection problem, the point robot has a four-dimensional state, comprised of three spatial coordinates, x, y, and z, and a yaw angle, θ. The robot's sensor footprint is also a cube, centered at the robot's location and designed to occupy one percent of the workspace

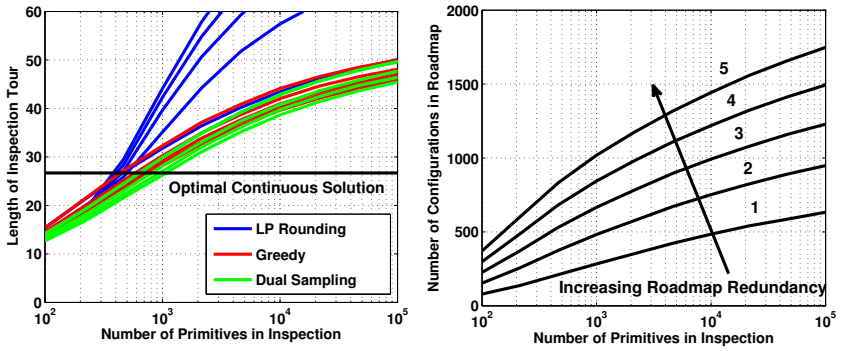

Fig. 2 Inspection planning results from a point in an obstacle-free, unit-cube workspace, in which the cube sensor is 1% of the workspace volume. Inspection tour cost and roadmap size are plotted as a function of the number of required primitives; each data point represents the mean over 100 simulations. On left, LP rounding lines represent increasing redundancy from one to five, upward on the vertical axis. Greedy rule lines have increasing redundancy moving downward. Dual sampling lines have increasing numbers of local samples, [5,10,25,50,100], also moving downward. Roadmaps on right plot refer to LP rounding and greedy rule computations.

volume, which is again representative of the parameters of a typical HAUV inspection planning problem. There are no obstacles in the point robot's workspace.

For each of several numbers of required primitives, ranging from 10^2 to 10^5, 100 instances of the planning procedure were run and the resulting tour cost and roadmap size were recorded. For each of these instances, five different redundancy requirements were explored, ranging from a roadmap in which each primitive must be observed at least once to a roadmap in which each primitive must be observed at least five times. As random robot configurations were sampled, an attempt was made to introduce each configuration into all five roadmaps, so the roadmaps of the same problem instance could share as many robot configurations as possible. Additionally, each of these five roadmaps was solved using both SCP approximation algorithms, the LP rounding algorithm and the greedy algorithm.

The same quantity of trials was run for a dual sampling implementation. A random primitive was sampled in each iteration, and robot configurations capable of viewing this primitive were sampled a designated number of times, which we refer to as the number of *local samples*. Of these, the configuration contributing the greatest quanitity of new observations was added as a waypoint in the inspection tour.

Figure 2 displays the results of this series of point-robot simulations. Increasing the redundancy of the coverage roadmap improved the quality of the greedy SCP solution, but worsened the quality of the LP rounding solution, which only works reasonably for a redundancy of one. Increasing the number of local samples in a dual sampling scheme improved the quality of the dual sampling solution, which was comparable with the results for redundant roadmaps solved by the greedy set cover algorithm. As further basis for comparison, the length of the optimal "lawnmower"

path for the point-robot's cube sensor to achieve 100% coverage of the continous workspace is plotted alongside the tour costs in Figure 2. For a low number of primitives, the sensor does not have to cover the entire volume. It is also clear from Figure 2 that an increase in roadmap redundancy has a significant impact on the size of the roadmap.

If the greedy SCP algorithm is used, improvements of approximately 10% are made by increasing the redunancy of the roadmap from at least one sighting per primitive at least five sightings per primitive. The added computational resources required to build a larger and more comprehsive roadmap would be acceptable in applications where such an improvement in robot mission time amounts to a large cost savings. A similar trend is observed if the number of local samples in the dual sampling scheme is increased from 10 to 100. Both of these strategies, dual sampling and using a redundant roadmap with the greedy SCP algorithm, however, encounter a similar asymptotic performance barrier as the discretization is set finer and finer. Despite the neighborhood optimization of the dual method, and the global scope of the redundant roadmap, a bias persists due to the greedy basis for both methods, in which robot configurations are added to the inspection one by one.

5 AUV Inspection Test Case

The inspection planning procedure is next applied to a real-world problem, the inspection of the stern of a ship by the HAUV. The inspection is planned for the SS Curtiss, a 200-meter aviation logistics support ship. The complex structures are large, with a single propeller seven meters in diameter and a shaft that is 1.5 meters in diameter. We first surveyed the area with vertical and horizontal lawnmower patterns at safe distances of around eight meters. This preliminay survey, although it did not achieve 100% coverage of all structures, was intended to build a polygonal mesh model of the stern suitable for planning a detailed inspection – employing the algorithms of this paper. For this, the Poisson reconstruction algorithm [21], which is typically applied to laser point clouds, was used to build a watertight 3D mesh from acoustic range data, pictured in Figure 4. This polygonal mesh possesses 107,712 points and 214,419 triangular faces, and has been discretized such that no triangle edge is larger than 0.1 meters, sufficient to identify a mine on the surface of the hull if all vertices are observed. Also in Figure 4, the sensor footprint represents the sonar field of view when the sonar is nodded up and down through its full 180-degree range of rotation. Although the sonar can only produce a single range scan at a time, we assume that in this planned inspection, the vehicle, at each configuration, will nod the sonar over its full range of angular motion to obtain a larger field of view. Paths for the vehicle will be planned, as before, in x, y, z, and yaw angle θ.

Because of the increased size and complexity of this inspection task, coverage roadmaps were not built to the same redundacy as in the point robot test case. Only three redundancies were tested: one, two, and three. For each redundancy, coverage roadmaps were constructed in 100 separate trials, and once again, each sampled

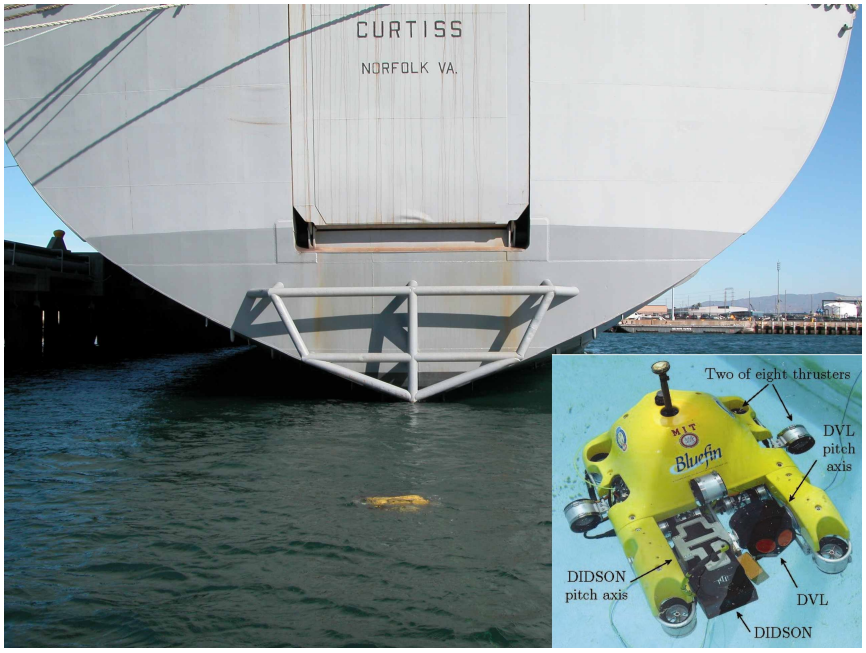

Fig. 3 An HAUV survey in progress at the SS Curtiss, a 200-meter aviation logistics support ship. At bottom right, an annotated diagram of the HAUV, model 1B.

configuration was introduced into as many of the roadmaps of varying redundancy as possible. Both the LP rounding SCP algorithm and the greedy SCP algorithm were again tested on these roadmaps. Because the ship mesh itself comprises a large, non-convex obstacle, the inspection planning procedure was used in its entirety, including the bi-directional RRT to perform lazy inquiries of point-to-point paths.

As is displayed in Figure 5, roadmap redundancy led to an improvement in tour length when the greedy SCP algorithm was used. Requiring each primitive to be observed twice shortened the average tour length by 25 meters, and reduced the worst-case tour length by nearly 50 meters. Requiring that each configuration must be observed three times yielded diminishing returns, as the mean tour length worsened. LP rounding in this problem was not competitive for any redundancy setting. As the right portion of Figure 5 indicates, an increase in redundancy causes a significant increase in the size of the roadmap. Despite this increase in roadmap size, all combinatorial approximation algorithms are solved very fast in this problem, whose runtime is largely dominated by the sampling phase.

To obtain a clearer picture of the impact of increased redundancy, Figure 6 displays histograms showing the coverage topology. It is clear that increased redundancy both increases the size of the roadmap and increases the mean and variance of the number of times a primitive is sighted.

In applying dual sampling to this planning problem, only two quantities of local samples were tested, five and ten. Beyond a value of ten, computation time grew

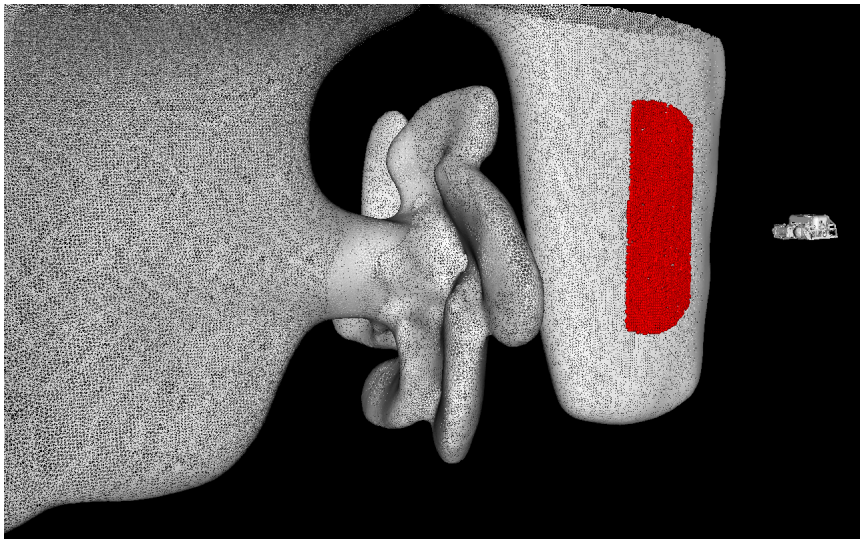

Fig. 4 A polygonal mesh obtained from our original, safe-distance survey of the SS Curtiss is depicted. The HAUV is illustrated at a configuration from which it observes a portion of the ship's rudder, at inspection range of about three meters. The primitives observed by the HAUV at this configuration are plotted in red; it is assumed that the HAUV has nodded the sonar. The ship mesh contains 107,712 points and 214,419 triangular faces. The propeller is approximately 7 meters in diameter.

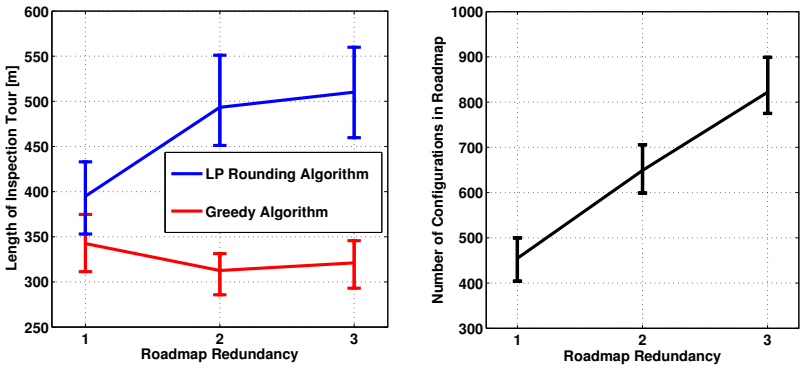

Fig. 5 Results of a series of inspection planning computations for the HAUV are depicted. Three different roadmap redundancies are examined, and the data for each redundancy represents the mean, the min, and the max (indicated by the error bars) over 100 trials. Inspection tour length is depicted at left, and roadmap size is depicted at right.

prohibitively high. For each quantity of local samples, 100 separate trials of the dual sampling algorithm were run. Table 1 displays the costs of the best-performing algorithms, and Table 2 displays the average computation times and ray shooting calls of selected algorithms. The dual sampling algorithms, in general, required more

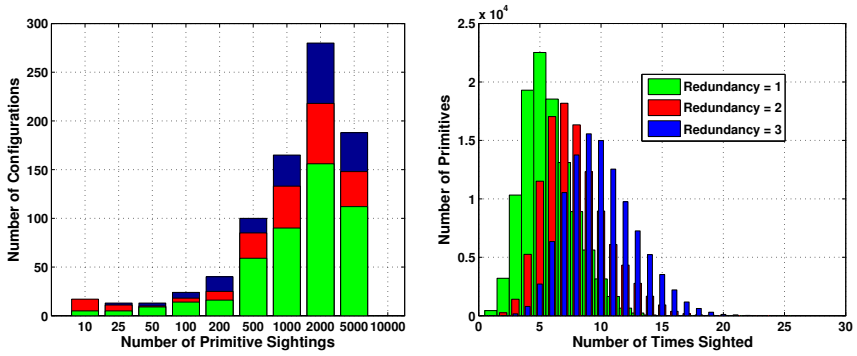

Fig. 6 Histograms display the coverage topology of a typical set of roadmaps for an instance of the ship hull inspection task. The quantities of targets observed by roadmap configurations are illustrated at left, and the quantities of shared sightings of geometric primitives are illustrated at right.

Table 1 Selected Tour Costs over 100 HAUV Inspection Planning Trials

	Mean Cost [m]	Min. Cost [m]	Max. Cost [m]
Dual Sampling with 5 Local Samples	343.2	317.7	369.0
Dual Sampling with 10 Local Samples	315.2	289.3	333.4
Roadmap with Redundancy = 1, Greedy Alg.	342.5	311.4	374.7
Roadmap with Redundancy = 2, Greedy Alg.	312.7	285.8	331.4
Roadmap with Redundancy = 3, Greedy Alg.	321.1	293.1	345.7

Table 2 Avg. Computation Time and Avg. Number of Ray Shooting Calls Required for HAUV Inspection Planning by Dual Sampling Algorithm (number of local samples indicated by *L.S.*) and Redundant Roadmap Algorithm with Greedy Set Cover (redundancy indicated by *R*)

	$L.S. = 5$	$L.S. = 10$	$R = 1$	$R = 2$	$R = 3$
Path Solution Time [sec]	112	189	79	111	126
Number of Ray Shooting Calls	6.1×10^6	1.0×10^7	4.8×10^6	6.0×10^6	6.9×10^6

computation time to achieve a result comparable in cost to that of a faster solution computed using a redundant roadmap and the greedy set cover algorithm. In particular, for equal-cost paths, dual sampling is between forty and seventy percent more expensive than are redundant roadmaps.

For all algorithms tested here, the vast majority of computational resources were spent dealing with the complex geometry of this planning problem, realized in the selection and checking of robot configurations. Ray shooting to compile sensor observations and check for occlusions at each of the sampled robot configurations was the single most time-consuming task in the planning procedure. Every algorithm made several million ray shooting calls to solve the HAUV planning problem, but the dual sampling scheme, to match the tour length of a redundancy-two roadmap, required four million more calls than the roadmap method. This data is evidence

that the selectivity of the dual sampling method becomes a burden for 3D problems in which every sampled configuration must be checked against a large number of geometric primitives.

Table 3 Resources Used for Coverage Path Planning Software Implementation

Software	Use	Link
OpenSceneGraph	kd-tree data structure for triangle mesh, ray shooting	http://www.openscenegraph.org
FLANN	kd-tree data structure for nearest-neighbor queries	http://www.cs.ubc.ca/~mariusm/index.php/FLANN/FLANN
OMPL	RRT Implementation	http://ompl.kavrakilab.org/index.html
PQP	Collision Checking	http://gamma.cs.unc.edu/SSV
Boost Graph Library	Minimum Spanning Tree	http://www.boost.org/doc/libs/1_46_1/libs/graph/doc/index.html
Blossom IV	Min-Cost Perfect Matching	http://www2.isye.gatech.edu/~wcook/blossom4

To ensure these results were obtained using the very best data structures and computational geometry tools available, a variety of high-performance open-source software tools were utilized. A list of these software tools is availabe in Table 3, which also includes software for the combinatorial optimization problems in this work.

Representative HAUV inspection paths are depicted in Figure 7. An interesting (and generally observed) result is the dual algorithm's selection of many robot configurations that are relatively close to the ship structure, and the greedy SCP algorithm's selection of robot configurations which are further from the structure by comparison. This goes hand in hand with the observed trend that redundant roadmap tours achieved comparable length to dual sampling tours, but with a smaller number of configurations in the tour on average. This likely results from the bias of sampling the boundary of the structure in the dual method versus sampling a larger workspace in the primal method, which extended beyond the visibility range of the HAUV.

6 Conclusion

In this work we presented an algorithm which plans fast, feasible inspection paths giving 100% sensor coverage of required geometric primitives. Key developments are the use of redundancy in a roadmap for coverage path planning, and the implementation of an integrated solution procedure for sampling-based coverage planning over complex 3D structures with many thousands of primitives (using highly developed, open-source routines wherever possible).

First, a state space discretization is chosen. The level of "resolution" of this discretization is determined by the redundancy of the roadmap. Second, a set of states

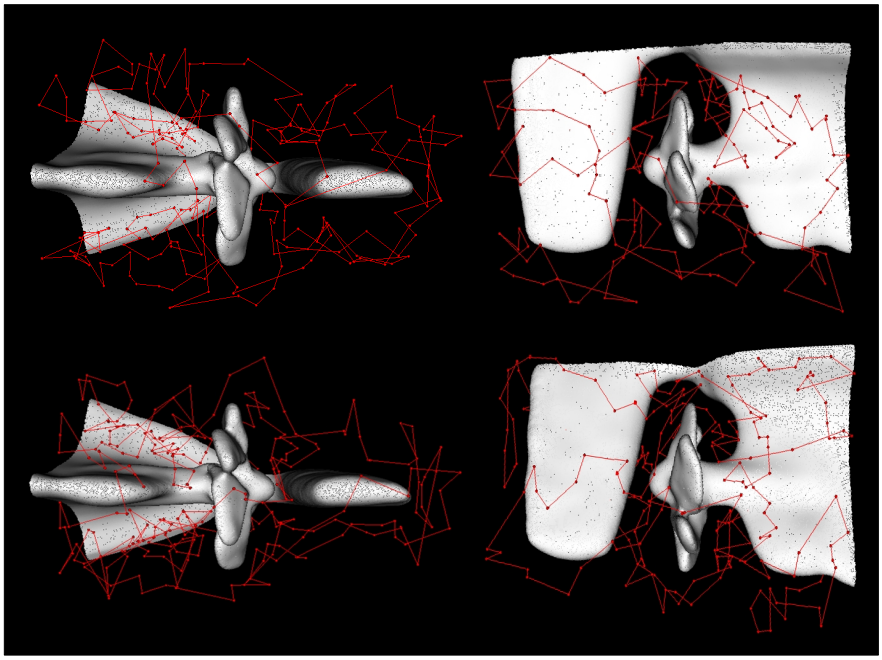

Fig. 7 Two representative examples of planned HAUV inspection paths, with views from beneath the ship and from the side. The upper half of the figure represents a roadmap solved using the greedy SCP algorithm, in which each primitive was observed at least twice by the configurations in the roadmap. The tour depicted is 310 meters in length, and contains 208 distinct configurations. The lower half of the figure represents a dual sampling solution, in which configurations were sampled in the vicinity of specific geometric primitives (ten local samples per primitive). The tour depicted is also 310 meters in length, but is comprised of 242 distinct configurations.

is chosen. A set cover approximation algorithm is applied to the roadmap, and the group of states to be used in the inspection is selected. Third, a sequence of states is chosen, using a polynomial-time approximation for the traveling salesman problem. And throughout this procedure, point-to-point sampling-based planning is used (in a "lazy" format for the sake of computational tractability) to ensure that the selected states and paths are feasible.

We have identified that this redundant roadmap method, in comparison to a dual sampling procedure, yields a computational advantage in a large-scale, real-world coverage problem. Less time is spent checking and cataloging sensor observations, and a comparable if not superior planned path is produced as a result.

References

1. E.M. Arkin, M.M. Halldorsson, and R. Hassin. Approximating the tree and tour covers of a graph. *Information Processing Letters*, 47:275–282 (1993)
2. P. Cheng, J. Keller, and V. Kumar. Time-optimal UAV trajectory planning for 3D urban structure coverage. *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2750–2757 (2008)
3. P. Atkar, A.L. Greenfield, D.C. Conner, H. Choset, and A. Rizzi. Uniform coverage of automotive surface patches. *Int. J. Robotics Research*, 24(11):883–898 (2005)
4. P. Atkar, A.L. Greenfield, D.C. Conner, H. Choset, and A. Rizzi. Hierarchical segmentation of surfaces embedded in $\Re^3$ for auto-body painting. *Proc. IEEE Int. Conf. on Robotics and Automation*, 572–577 (2005)
5. P.S. Blaer and P.K. Allen. View planning and automated data acquisition for three-dimensional modeling of complex sites. *J. Field Robotics*, 26(11-12):865–891 (2009)
6. H. Choset and P. Pignon. Coverage path planning: the boustrophedon decomposition. *Proc. Int. Conf. on Field and Service Robotics*, (1997)
7. H. Choset. Coverage for robotics - a survey of recent results. *Annals of Mathematics and Artificial Intelligence*, 31:113–126 (2001)
8. H. Choset, K.M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L.E. Kavraki, and S. Thrun. *Principles of robot motion: theory, algorithms, and applications.* Cambridge, MA: MIT Press (2005)
9. N. Christofides. Worst-case analysis of a new heuristic for the traveling salesman problem. Technical Report CS-93-13, Carnegie Mellon University (1976)
10. J.R. Current and D.A. Schilling. The Covering Salesman Problem. *Transportation Science*, 23(3):208–213 (1989)
11. T. Danner and L. Kavraki. Randomized planning for short inspection paths. *Proc. IEEE Int. Conf. on Robotics and Automation*, 2:971–976 (2000)
12. K. Easton and J. Burdick. A coverage algorithm for multi-robot boundary inspection. *Proc. IEEE Int. Conf. on Robotics and Automation*, 727–734 (2005)
13. P. Fazli, A. Davoodi, P. Pasquier, and A.K. Mackworth. Complete and robust cooperative robot area coverage with limited range. *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 5577–5582 (2010)
14. M. Fischetti, J.J.S. Gonzalez, and P. Toth. A branch-and-cut algorithm for the symmetric generalized traveling salesman problem. *Operations Research*, 45(3):378–394 (1997)
15. M. Gendreau, G. LaPorte, and F. Semet. The covering tour problem. *Operations Research*, 45(4):568–576 (1997)
16. H. Gonzalez-Baños and J.-C. Latombe. Planning robot motions for range-image acquisition and automatic 3D model construction. *Proc. AAAI Fall Symp.* (1998)
17. H. Gonzalez-Baños and J.-C. Latombe. A randomized art gallery algorithm for sensor placement. *Proc. 17th Annual ACM Symp. on Computational Geometry*, 232–240 (2001)
18. D.S. Hochbaum. Approximation algorithms for the set covering and vertex cover problems. *SIAM J. Comput.*, 11(3):(1982)
19. F. Hover, et al. A vehicle system for autonomous relative survey of in-water ships. *Marine Technology Society Journal*, 41(2):44–55 (2007)
20. D.S. Johnson. Approximation algorithms for combinatorial problems. *J. Comput. System Sci.*, 9:256–278 (1974)
21. M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. *Proc. Fourth Eurographics Symposium on Geometry* (2006)
22. S. LaValle and J. Kuffner. Rapidly-exploring random trees: progress and prospects. *Proc. Workshop on the Algorithmic Foundations of Robotics*, 293–308 (2000)
23. S. LaValle. *Planning Algorithms.* Cambridge, UK: Cambridge University Press (2006)
24. Y.N. Lien and E. Ma. Transformation of the generalized traveling salesman problem into the standard traveling salesman problem. *Information Sciences*, 74:177–189 (1993)

25. L. Lovasz. On the ratio of optimal integral and fractional covers. *Discrete Math.*, 13:383–390 (1975)
26. C.E. Noon and J.C. Bean. An efficient transformation of the generalized traveling salesman problem. Technical Report 89-36, Department of Industrial and Operations Engineering, The University of Michigan, Ann Arbor, (1989)
27. M. Saha, T. Roughgarden, J.-C. Latombe, and G. Sanchez-Ante. Planning tours of robotic arms among partitioned goals. *International Journal of Robotics Research*, 25(3):207–223 (2006)
28. G. Sanchez and J.-C. Latombe. On delaying collision checking in PRM planning: application to multi-robot coordination. *Int. J. Robotics Research*, 21(1):5–26 (2002)
29. W. Scott, G. Roth, and J. Rivest. View planning for automated three-dimensional object reconstruction and inspection. *ACM Computing Surveys*, 35(1):64–96 (2003)
30. T. Shermer. Recent results in art galleries. *Proc. IEEE*, 80(9):1384–1399 (1992)
31. P. Wang, R. Krishnamurti, and K. Gupta. View planning problem with combined view and traveling cost. *Proc. IEEE Int. Conf. on Robotics and Automation*, 711–716 (2007)
32. K. Williams and J. Burdick. Multi-robot boundary coverage with plan revision. *Proc. IEEE Int. Conf. on Robotics and Automation*, 1716–1723 (2006)