Received: 24 August 2018 | Revised: 13 June 2019 | Accepted: 17 July 2019

DOI: 10.1002/rob.21905

REGULAR ARTICLE

WILEY

Multi-AUV motion planning for archeological site mapping and photogrammetric reconstruction

Jane Wu[1] | Russell C. Bingham[2] | Samantha Ting[2] | Kolton Yager[3] | Zoë J. Wood[3] | Timmy Gambin[4] | Christopher M. Clark[2]

[1]Department of Computer Science and Mathematics, Harvey Mudd College, Claremont, California

[2]Department of Engineering, Harvey Mudd College, Claremont, California

[3]Department of Computer Science, California Polytechnic State University, San Luis Obispo, California

[4]Department of Archaeology, University of Malta, Msida, Malta

Correspondence
Jane Wu, Department of Computer Science and Mathematics, Harvey Mudd College, Claremont, CA 91711.
Email: jhwu@hmc.edu

Funding information
National Science Foundation, Grant/Award Number: 1460153

Abstract

This paper presents coupled and decoupled multi-autonomous underwater vehicle (AUV) motion planning approaches for maximizing information gain. The work is motivated by applications in which multiple AUVs are tasked with obtaining video footage for the photogrammetric reconstruction of underwater archeological sites. Each AUV is equipped with a video camera and side-scan sonar. The side-scan sonar is used to initially collect low-resolution data to construct an information map of the site. Coupled and decoupled motion planning approaches with respect to this map are presented. Both planning methods seek to generate multi-AUV trajectories that capture close-up video footage of a site from a variety of different viewpoints, building on prior work in single-AUV rapidly exploring random tree (RRT) motion planning. The coupled and decoupled planners are compared in simulation. In addition, the multiple AUV trajectories constructed by each planner were executed at archeological sites located off the coast of Malta, albeit by a single-AUV due to limited resources. Specifically, each AUV trajectory for a plan was executed in sequence instead of simultaneously. Modifications are also made by both planners to a baseline RRT algorithm. The results of the paper present a number of trade-offs between the two planning approaches and demonstrate a large improvement in map coverage efficiency and runtime.

KEYWORDS
cooperative robots, mapping, planning, underwater robotics

1 | INTRODUCTION

The discovery and mapping of historical archeological sites present an ongoing challenge for marine archeologists. One common method for surveying areas of interest utilizes side-scan sonar towed from a surface vessel, which requires substantial equipment overhead and manual operation (Blondel, 2010). Recently, autonomous underwater vehicles (AUVs) have been deployed as a more efficient alternative to towed systems (Fallon, Kaess, Johannsson, & Leonard, 2011; Paull, Saeedi, Seto, & Li, 2014; Ruiz, DeRaucourt, Petillot, & Lane, 2004). AUV trajectory planning for archeological surveying typically involves a manually planned lawnmower pattern that covers the entirety of the surveying area.

Upon discovery of interesting sites in a survey area, divers or remotely operated vehicles (ROVs) are often sent to investigate the location and capture close-up video footage for three-dimensional (3D) photogrammetric reconstruction (McCarthy & Benjamin, 2014; Van Damme, 2015). Technical divers are able to explore down to 100 m below the ocean surface, while ROVs can be used in deeper waters but require a boat to be in close proximity to the site of interest. AUVs provide solutions to the difficulties posed by the aforementioned methods, as they are capable of operating in deep

waters and can be deployed far away from the site of interest, removing the need for a nearby boat.

In Viswanathan et al. (2017), a novel *single* AUV motion planning approach was presented for creating 3D photogrammetric reconstructions of shipwrecks using AUVs equipped with side-scan sonar and cameras. The full pipeline involves multiple AUV deployments. First, the AUV surveys an area of interest using sonar, which creates a coarse seafloor map. The map is then used to generate AUV trajectories with the objective of maximizing information gain, for example, the total number of unique viewpoints of the site of interest a given trajectory captures. Generation of robot trajectories is accomplished using rapidly exploring random trees (RRTs) (Kuffner & LaValle, 2000). Experiments showed the planning algorithm was able to successfully create trajectories that allowed for 3D reconstruction of several known wrecks along the coast of Malta.

This paper builds on previous work and presents a multi-AUV motion planning method for constructing 3D maps of marine archeological sites. Motion planning for multiple AUVs presents a number of challenges, including (a) collision prevention between robots, (b) viewpoint distribution across robot trajectories, and (c) runtime of generating such trajectories. The fieldwork for this study was conducted using the OceanServer Iver3 AUV (Figure 1a). The AUV was equipped with an EdgeTech 2205 side-scan sonar, as well as a GoPro HERO4 camera. Using seafloor maps generated by the side-scan sonar during initial surveys, two multirobot RRT-based approaches were implemented and tested to efficiently record video footage of various archeological sites in Malta. The motion planning models extend the single robot case through a number of significant algorithmic changes and runtime improvements. Comparisons are also presented between the coupled RRT algorithm (that searches the composite configuration space of all robots) and the decoupled RRT algorithm (that sequentially searches each individual robot's configuration space).

Specifically, this paper presents a number of contributions to the field of underwater robotics

- A coupled multi-AUV planning approach that grows a single RRT, simultaneously generating trajectories such that all robot dives are time-synced with each other.
- A decoupled multi-AUV planning approach that sequentially generates robot trajectories, removing viewpoints that prior robots have obtained from the shared map to maximize information gain.
- Simulations comparing the two planning approaches, which indicate decoupled planning is more time-efficient, covers a larger percentage of a given area of interest, and scales more effectively to large search areas.
- Validation of algorithm feasibility via AUV deployments and mapping of several underwater archeological sites, (e.g., plane wrecks and shipwrecks) located along the coast of Malta. Since only a single AUV was available, multi-AUV plans were validated by deploying the same AUV for each individual AUV trajectory of the plan that was designed to be run simultaneously.

The paper is organized as follows. Section 2 presents relevant background and prior research on photogrammetry, AUV motion planning, and multirobot motion planning (MRMP). Section 3 outlines the complete AUV 3D photogrammetric reconstruction system. Sections 4 and 5 describe the MRMP method. Sections 6 presents and analyzes the simulations and field experiments that were performed. Finally, Section 7 discusses conclusions and future work.

2 | BACKGROUND

Photogrammetry is the process of using photographs to construct 3D surface models and has been used to study underwater archeological sites (Drap et al., 2015; Gambin, 2011; von Fock et al., 2017). To create such visualizations, triangulation of still images from video data is used to make 3D measurements from a variety of viewpoints (Olague & Mohr, 2002). Since larger variations in camera angle yield more accurate range calculations from triangulation, multiple camera angles are essential for accurate photogrammetric reconstruction (Bern, Eppstein, & Gilbert, 1994). Recently, AUVs have been used to collect sensor data for reconstruction of underwater sites (Bingham et al., 2010; Johnson-Roberson et al., 2016; Ozog, Troni, Kaess, Eustice, & Johnson-Roberson, 2015; Skinner & Johnson-Roberson, 2015; von Fock et al., 2017). In Bingham et al. (2010), an AUV was used to survey a shipwreck with multiple manually programmed lawnmower patterns over 30 × 45 m of seafloor. The AUV missions

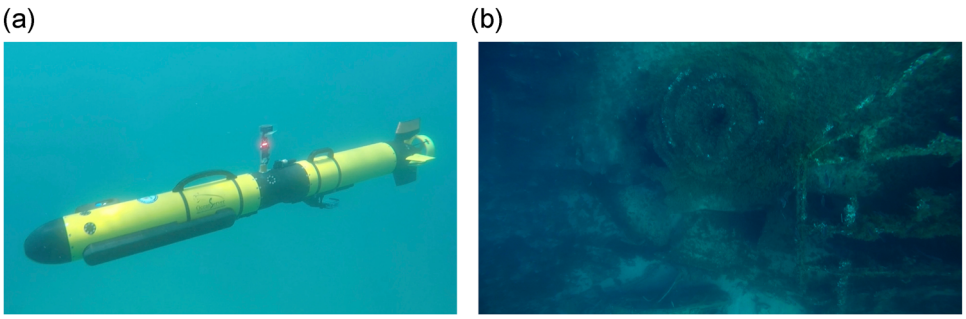


FIGURE 1 (a) Iver3 AUV with side-scan sonar at the front and a GoPro HERO4 attached under the rear handle. (b) GoPro footage of Maori wreck in Malta, taken aboard the AUV and used for photogrammetric reconstruction. AUV, autonomous underwater vehicle [Color figure can be viewed at wileyonlinelibrary.com]

took over 6 hr and much of the mission time was spent surveying parts of the ocean floor that did not contain the shipwreck.

Sampling-based planners are a common approach to generating AUV trajectories for information gain, as deterministic approaches become infeasible in high dimensional spaces (Elbanhawi & Simic, 2014). In particular, RRT-based planners are most widely employed due to the ability to quickly explore the search space (Hollinger, Englot, Hover, Mitra, & Sukhatme, 2013; Hollinger & Sukhatme, 2014). Stochastic methods such as STOMP have also been explored for AUV motion planning (Galceran et al., 2015; Kalakrishnan, Chitta, Theodorou, Pastor, & Schaal, 2011). More recent work includes that in Viswanathan et al. (2017) and Palomeras, Hurtós, Carreras, and Ridao (2018), where AUV trajectories were constructed with RRT type planners. RRTs can be used when planning trajectories for robots with dynamic constraints, as well as gathering information about an environment (Huang & Gupta, 2008; Lavalle & Kuffner, 2000; Yang, Gan, & Sukkarieh, 2013). These sampling-based methods can be directly applied to AUV motion planning, where the objective is often to collect information about a marine environment. Existing AUV information gathering planners differ from the methods presented in this paper primarily in the information objective function and method for reconstruction. While (Hollinger et al., 2013) randomly samples viewpoints in the search space and subsequently combines TSP and RRT algorithms to construct trajectories, a discretized information map is used in the coupled and decoupled planners presented. In addition, photogrammetric reconstruction necessitates the use of cameras such as GoPros to capture video footage, whereas bathymetry-based surface reconstructions are common to underwater inspection applications.

While video footage can be captured by a single AUV, it is advantageous to deploy multiple AUVs to achieve greater coverage of an area of interest in the same deployment time. RRT type algorithms have also been employed for MRMP problems as shown in (Clark, 2004, 2005), although much of this study employed the algorithms for collision avoidance with respect to obstacles as opposed to maximizing information gain. In other previous studies, MRMP algorithms were designed to maximize information gain in multirobot exploration and mapping (Burgard, Moors, Stachniss, & Schneider, 2005; Simmons et al., 2000; Visser & Slamet, 2008). In Visser and Slamet (2008), the information gain of each robot is balanced with the cost of movement in the context of multirobot frontier exploration. Similarly, Burgard et al. (2005) developed a technique for distributing multiple robots to explore a broader range of areas in their environment.

Most recent MRMP approaches employing RRTs exclusively apply to robots capable of communicating with each other. However, underwater communication between AUVs is typically unreliable and low bandwidth, making the implementation of these approaches extremely challenging (Cui, Li, & Yan, 2016; Desaraju & How, 2012; Jahn et al., 2017; Jung, Lee, Lee, Choi, & Lee, 2009; Stephan, Fink, Kumar, & Ribeiro, 2017). In Stephan et al. (2017), a decentralized multirobot system architecture was presented for MRMP, with an emphasis on maintaining network integrity between all mobile robots in the team during task execution. The authors used a variation of RRT* for the control algorithm, divided into centralized and distributed controllers. The decentralized multiagent RRT (DMA-RRT) is another framework applicable to robot teams where communication between robots is feasible (Desaraju & How, 2012). To navigate to specific locations in the shared space, each robot expands its own search tree using the Closed-loop RRT (Kuwata et al., 2008). In addition, to improve initial robot trajectories during task execution, the framework includes a merit-based token-passing strategy. When a robot wins a bid for the token, its trajectory is updated, and the resulting information is broadcasted to all other robots. Because underwater communication between AUVs is often low bandwidth and not dependable, the aforementioned MRMP systems cannot be applied to multi-AUV planning since predefined trajectories are required before task execution.

Less work has been dedicated to MRMP using RRTs for information gain. While Cui et al. (2016) introduced the MDMI-RRT* for multi-AUV scalar field sampling, the method requires an online system where robots can iteratively update shared information to reduce the uncertainty of the scalar field prediction. Similarly, Jung et al. (2009) presents an online method for exploring unknown terrains using multiple AUVs. To our knowledge, no prior work has been published in multi-AUV planning for 3D photogrammetric reconstruction. With this in mind, the research presented aims to fill this void by developing a multi-AUV system that can employ one of multiple approaches to RRT planning as applied to information gain.

3 | MULTI-AUV SYSTEM OVERVIEW

This paper proposes a multimission, multirobot approach for collecting the data necessary to create a photogrammetric reconstruction of an archeological site. Figure 2 shows a block diagram overview of the primary components of the pipeline. The motion planners described in

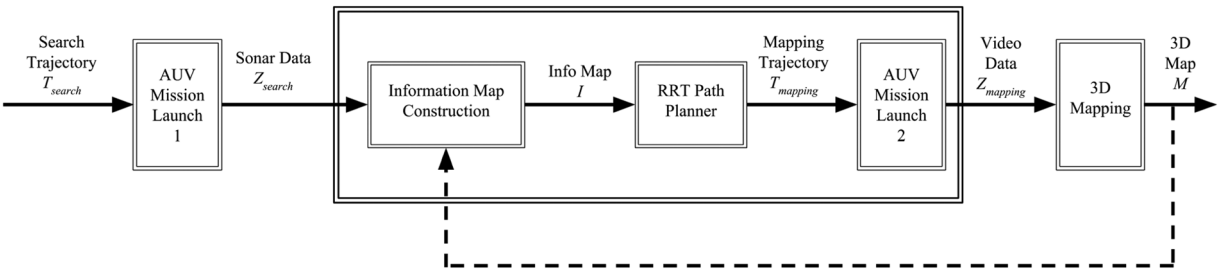


FIGURE 2 Block diagram of the experimental pipeline for photogrammetric reconstruction

FIGURE 3 Lawnmower trajectory for initial autonomous underwater vehicle search mission [Color figure can be viewed at wileyonlinelibrary.com]

Section 5 define one component of this system. During the first mission, the AUVs are equipped with side-scan sonar and deployed to follow some trajectory T_{search}, typically a coarsely spaced lawnmower pattern, with the goal of finding potential underwater archeological sites (e.g., Figure 3). The sonar data Z_{search} collected during this first mission is used to identify sites of interest. This data is also used to create an information map I that acts as the basis of the trajectory planner's objective function. I is fed in as the input to the RRT algorithm which generates a multi-AUV trajectory $T_{mapping}$ for mapping the site. This trajectory is then postprocessed to be kinematically and dynamically feasible for a physical AUV.

Once the trajectories are converted to mission waypoints, the AUVs are equipped with cameras and deployed on a second mission to follow $T_{mapping}$. The images $Z_{mapping}$ collected while following $T_{mapping}$ can be used to create a 3D photogrammetric reconstruction M of the archeological site. As indicated by the feedback loop in the block diagram, the map M can be used to create a new information map, new trajectory $T_{mapping}$, and new mission for additional data collection and improved mapping. This paper focuses on the bold box in the block diagram, and each step in the box is described in the following sections.

3.1 | Information map creation

The first stage in the trajectory planning is the creation of an information map given the side-scan sonar data. The information map I is then queried by the motion planner to generate AUV trajectories that obtain maximum information gain. I is produced in the format of a two-dimensional (2D) matrix with cell values that range from 0 to 255, where 255 represents high information gain and 0 is no information gain. Two methods for creating the information map are presented, such that I could be generated from either a sonar PNG file or a point cloud PLY file.

3.1.1 | Information map creation via side-scan sonar

Information maps can be created from sonar images of sites of interest. The information maps are created through the following localization procedure. The AUV's onboard global positioning system (GPS) records its position. The AUV takes sonar scans of information maps and uses its GPS position to georeference each sonar image. Based on a specific feature's position in the georeferenced sonar image, the feature is localized. Due to this method of localization, the accuracy of the map is reflective of the accuracy of the AUV's position system. A USBL could have been used to increase localization accuracy, but for the IVER3 AUV, a GPS is used to localize while at the surface, and a Doppler velocity log (DVL) is used below water. Since the position estimation will drift for AUV dives when using only DVL, corrections to dive position estimation are made post-deployment by linearly interpolating between the AUV's dive and resurface GPS coordinates.

The steps for creating an information map from a sonar image are shown in Figure 4 and are described as follows. First, the image resolution is decreased and the format is changed from RGB to

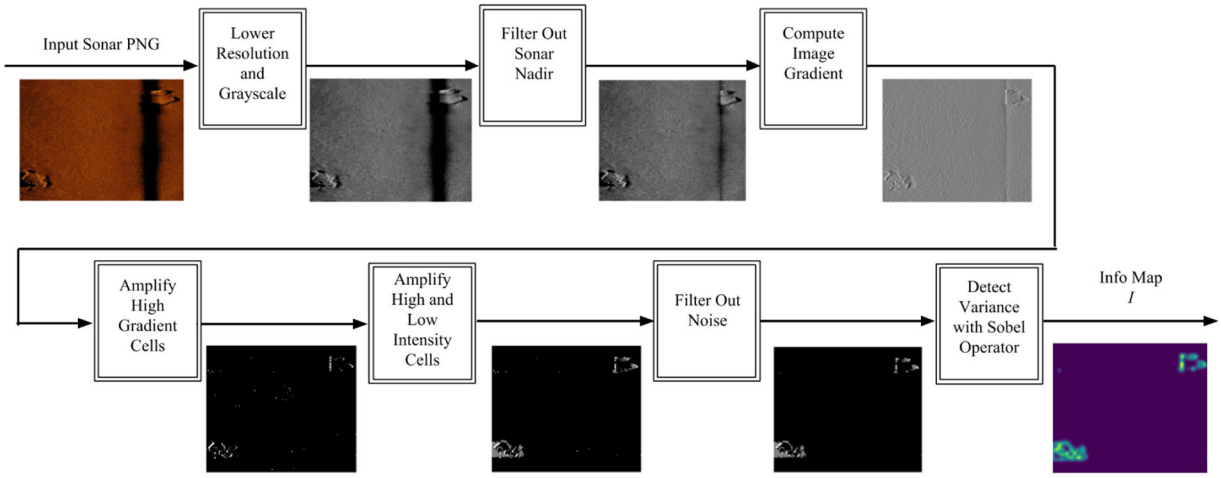


FIGURE 4 Block diagram of information map generation from the sonar PNG file [Color figure can be viewed at wileyonlinelibrary.com]

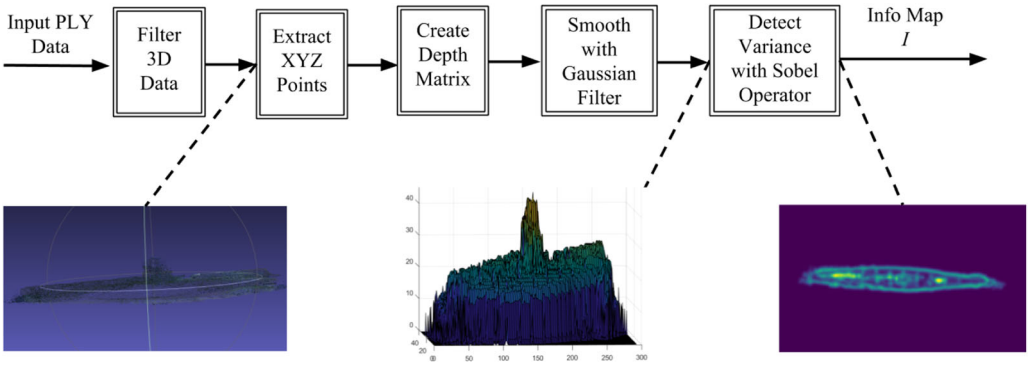


FIGURE 5 Block diagram of information map generation from PLY file [Color figure can be viewed at wileyonlinelibrary.com]

grayscale. Next, the image centerline, that is, the nadir, is removed, after which the image is passed through a gradient filter.

To increase the contrast of a site of interest, areas of high pixel intensity, low pixel intensity, and high gradient are identified on the image. First, in the *Amplify High Gradient Cells* step, all pixels with a gradient greater than the threshold $\tau_{highGrad}$ are scaled by a factor σ_{Grad}. Second, in the *Amplify High and Low Intensity Cells* step, all pixels with intensity greater than the threshold $\tau_{highIntensity}$ are scaled by a factor $\sigma_{highIntensity}$ and all pixels with intensity less than the threshold $\tau_{lowIntensity}$ are scaled by a factor $\sigma_{lowIntensity}$.

These six parameters are tuned for each image because the AUV's height from the bottom varies between sonar scans, affecting the brightness and contrast of each image. While this additional labor reduces the autonomy of the system, tuning took on the order of only 2 min for a single map.

The information map generated in Figure 5 illustrates how this method not only identifies the site of interest but also highlights important artifacts within the site, such as the edges of the ship and the tall cabin in the middle.

After amplifying the areas of interest, an additional filter is added which sets a cell to zero if all its neighbors are also zero; this is done to minimize noise. The map is then adjusted using the Sobel operator to increase information gain around areas of high gradient, amplifying the edges of the site of interest. Finally, a 2D Gaussian Blur is applied, which compensates for uncertainty in the AUV position from when the sonar scan was taken.

3.1.2 | Information map creation via point cloud

The information map I can also be constructed from a 3D point cloud. In some cases, this point cloud data would not be obtained by the AUV. For example, the point cloud data in Figure 5 was obtained from a previous reconstruction made by divers. Though this implies the wreck is diveable, using an AUV instead of divers would be favorable during constant monitoring because it would prevent repeatedly sending divers into deep water.

To create I from a 3D point cloud, it is first flattened into a 2D elevation matrix E, where $E = \{e_{ij} |, \ \ i = 1, \ldots m; \ \ j = 1, \ldots n\}$. Here, e_{ij} is the maximum height of all the occupied cells with indices i, j in the Z direction. To extract areas of elevation variation, the map is adjusted using the Sobel operator, as done in the sonar pipeline. The 2D Gaussian Blur is also applied, again due to uncertainty in state estimation.

This method assumes that both tall artifacts and areas with a large variance in height are of interest. Tall features are of interest because a large, unnatural height off bottom often indicates a site of interest. Similarly, a large change in height typically indicates either the boundary of a wreck site or a feature within the site of interest. Furthermore, these areas of the large variation in height are of interest because these areas contain complex 3D geometry that can only be reconstructed if viewed from multiple viewpoints.

3.1.3 | Extending information maps to include view angles

Planning was not performed in 3D space due to runtime constraints of the algorithm. However, the 2D information map was extended to incorporate discretized view angles as a third dimension. In this method, the planning is still performed in 2D space; however, the addition of the angle dimension incentivizes the trajectory planner to generate trajectories that view each portion of the site of interest from multiple angles, as multiple viewpoints are necessary to create an optimal photogrammetric reconstruction. Given a unique viewpoints that each cell can be viewed from, the map is correspondingly expanded such that each cell is a discrete units in the third dimension (Figure 6). With respect to the physical wreck, the a viewpoints divide a full 360-degree view of each cell location into discrete angle ranges of size $2\pi/a$ radians. Each cell in the new 3D map inherits the information gain that its respective 2D cell held, representing the value of seeing high information areas from all sides.

Hence, when the map is expanded to include view angles, the total amount of information available to be gathered on the map is multiplied by a. A visualization of the expansion of the information map to the 3D form is shown in Figure 6. In all simulations and fieldwork, the number of angle ranges is set to four unless otherwise noted.

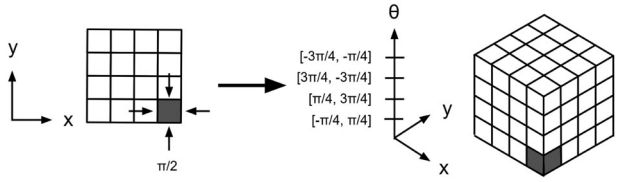


FIGURE 6 Transformation of the information map to include the third dimension of viewpoint angle. Performed on a 4x4 information map with $a = 4$. On the left, the cell at position (4, 1) has been shaded, and the four arrows demonstrate the four distinct viewpoint angles. An example angle range is labeled $\pi/2$; however, this range spans $\pm\ \pi/4$ from its center, so the effective angle range is $[\pi/4, 3\pi/4]$. On the right, the information map has been expanded to 3D to account for viewpoint angle. The shaded cell at position (4,1,1) corresponds to the information that would be gained by the AUV traveling through the 2D planar map cell at position (4,1), with the AUV having camera angle in range $[-\ \pi/4, \pi/4]$. Note the camera angle, in this case, is measured with respect to the *XY* plane, with 0 radians aligned with the *x*-axis. AUV, autonomous underwater vehicle

4 | MOTION PLANNING

After the initial search mission and information map construction, the multirobot RRT motion planner is executed to solve the following problem. Let n robots operate within a 3D bounded workspace W, where the depth boundary of W is above the object of highest elevation of the site and W is free of obstacles. At a given time step t, $X_{i,t} = [x_{i,t}, y_{i,t}, z_{i,t}, \theta_{i,t}]$ is the state vector for a specific robot and $u_{i,t}$ is the control effort for that robot. The goal is to determine each robot i's time-limited sequence of control actions $U_i = \{u_{i,0}, u_{i,1}, u_{i,2}, ...\}$ that maximizes the information gain O. To summarize, the problem is to:

Find

$$\underset{U}{max} O(U_1, \ldots U_n, X_{1,0}, \ldots X_{n,0}) \tag{1}$$

Subject to

$$O(U_1, \ldots U_n, X_{1,0}, \ldots X_{n,0}) = \sum_{X_D \in E} Info(I, X_D) \tag{2}$$

$$E(U, X_0) = \bigcup_{i=1}^{n} \bigcup_{t=0}^{p} X_{i,t,D} \tag{3}$$

$$X_{i,t,D} \in W \tag{4}$$

$$X_{i,t,D} = D(X_{i,t}) \tag{5}$$

$$X_{i,t+1} = f(u_{i,t}, X_{i,t}) \tag{6}$$

The function $Info(I, X_D)$ returns the information gain when a robot visits discretized state X_D of the information map I. The set $E(U, X_0)$ refers to the union of all discretized states visited over the course of all n of the robot trajectories that follows the p control actions of the set U.

The function $D(X_{i,t})$ returns $X_{i,t,D}$, a spatially discretized version of the AUV state vector. In this case, the first three elements of $X_{i,t,D}$ are the integer index values associated with the closest 3D cell of the information map I, and the last element corresponds with an integer index value associated with a discretization of the AUV's yaw angle. For example, if the state $X_{i,t}$ is contained within cell $c_{3,4,6}$ of I, and the yaw angle is 85 degrees, then the discretized state is $X_{i,t,D} = [3461]$ for an angle discretization resolution of 90 degrees. Finally, the function $f(u_{i,t}, X_{i,t})$ represents the robot kinematics or dynamics (specific to the application) of robot i. For both coupled and decoupled multirobot motion planners, trajectories are generated using an RRT that maximizes O as outlined in the problem definition.

5 | MOTION PLANNING SOLUTIONS

Two different multirobot RRT planners are presented in this section and are referred to as the coupled planner and the decoupled planner. The coupled planner concurrently generates trajectories for all robots, while the decoupled planner generates each robot trajectory sequentially. Several modifications to the basic RRT algorithm are made in both variants to improve runtime and information gain across robot trajectories (discussed in Section 5.3). After creating the RRT data structures modeling the potential robot trajectories, the trees are fed to an objective function that compares the score of each of the leaf nodes. The leaf node with the highest score is picked, and the best trajectory is extracted by tracing the best leaf node back to the root node.

5.1 | Coupled planning

Given m robots, the coupled planner simultaneously generates trajectories by constructing a roadmap $R(N, E)$ of node set N and edge set E. Each node $n_i \in N$ is defined by the states of all robots at a given timestamp t^i, for example, $[X_1^i, ..., X_m^i]$, where X_j^i is the state of the jth robot of node i. The state is defined as $[x_j^i, y_j^i, z_j^i, \theta_j^i]$. As well, each edge $e_{ik} \in E$ is defined by a node pair $\{n_i, n_k\}$. Since all robots have the same timestamp in a given node, each robot dives for the same length to cross an edge (assuming all robots travel at the same constant speed).

Algorithm 1: coupledMultiRobotRRT(n_0)

Input: start node n_0

Output: bestTrajectory($R(N, E)$)

1: Add n_0 to $R(N, E)$

2: **for** $i \leftarrow 1$ **to** *numExpansions* **do**

3: $n_{exp} \leftarrow$ selectNodeToExpand ($R(N, E)$)

4: $n_{new} \leftarrow$ expandNewNode(n_{exp})

5: Add n_{new} and edge (n_{exp}, n_{new}) to R

6: **if** n_{new} in endgame region **then**

7: add n_{new} to *endNodes*

8: **endif**

9: **endfor**

10: **return** bestTrajectory($R(N, E)$)

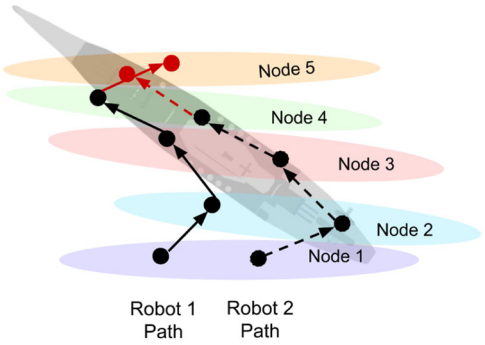


FIGURE 7 Visual representation of coupled node structure and expansion for two robots. Each node includes the state of all robots at a particular time step and is color coded accordingly. Node 5 shows the next expansion of the RRT trajectory in red; at this stage, the expansion is being checked for collisions before being added to the tree. RRT, rapidly exploring random tree [Color figure can be viewed at wileyonlinelibrary.com]

Algorithm 1 presents the steps for the planner, and Figure 7 provides a corresponding visualization. In line 1, node n_0 which includes the initial states of all robots is created and added to the roadmap R. For each expansion (line 2), an existing node n_{exp} in the tree is randomly selected in line 3 by Algorithm 5 (discussed in Section 5.3.1). In line 4, serial expansion is then used to expand each robot j from state X_j^{exp} to a new state X_j^{new}, thereby generating a new node n_{new} (see details in the following section). After a new node is expanded, it is added to the tree in line 5. If the timestamp t^{new} of the new node n_{new} reaches the maximum time allotted for a robot trajectory, n_{new} is added to *endNodes* in line 7. After *numExpansions* new nodes have been created, all trajectories that start at n_0 and end at a node in *endNodes* are evaluated to determine which trajectory maximizes information gain and should be returned (line 10).

Algorithm 2: expandNewNode_coupled(n_{exp})

Input: n_{exp}, *HICs*

Output: n_{new}

1: $m \leftarrow$ Number of robots

2: $d \leftarrow$getRandomDiveDistance()

3: Shuffle order of robots

4: **for** $j \leftarrow 1$ **to** m **do**

5: **if** $rand() < \gamma_{HIC-exp}$ **then**

6: $X_{HIC} \leftarrow$ getStateOfRandHIC(*HICs*)

7: $\theta_j^{new} \leftarrow atan2(X_{HIC}.y - X_j^{exp}.y, X_{HIC}.y - X_j^{exp}.x)$ # expand towards random HIC

8: **else**

9: $\theta_j^{new} \leftarrow rand(-\pi, \pi)$

10: **end if**

11: **while** collision $(X_j^{exp}, \theta_j^{new}, [X_1^{exp}, \theta_1^{new}: X_{j-1}^{exp}, \theta_{j-1}^{new}])$ **do**

12: $\theta_j^{new} \leftarrow rand(-\pi, \pi)$

13: **end while**

14: $X_j^{new} \leftarrow$expandState$(X_j^{exp}, \theta_j^{new}, d)$

15: **end for**

16: **return** n_{new}

(Continues)

5.1.1 | Node expansion for coupled planning

Algorithm 2 presents the steps for node expansion used by the coupled planner. Since all robots have the same timestamp at a given node and all robots are assumed to travel at the same constant speed, the distance each robot travels for a given expansion is the same for all robots and defined in line 2. The function getRandomDiveDistance() determines this distance d by sampling from a bounded uniform distribution.

To prevent any individual robot from being prioritized, the order in which the robots are expanded is randomized at every node expansion. The heading θ_j^{new} of the dive for the jth robot is determined by one of two sampling methods chosen by the parameter $\gamma_{HIC-exp} \in [0, 1]$. The first method expands in the direction of a random High Information Configuration node (line 7, discussed in Section 5.3.2), while the second samples a random heading from a uniform distribution between $-\pi$ and π (line 9). If the chosen heading causes a collision with previously expanded robots, a random heading is re-sampled from the uniform distribution until the dive is collision-free.

If the robots trajectories are not coplanar, that is, each robot dives to a different depth, no collision checking is required. Otherwise, collision checking is performed in the collision() function (line 11) and takes in the current robot state X_j^{exp}, the proposed dive heading θ_j^{new}, and the current states of previously expanded robots and their dive headings $[X_1^{exp}, \theta_1^{new}: X_{j-1}^{exp}, \theta_{j-1}^{new}]$. If two robot dives of length d are coplanar, a sufficient method for collision checking is to consider their current positions $\{X_i^{exp}, X_k^{exp}\}$ and proposed dive headings $\{\theta_i^{new}, \theta_k^{new}\}$ (assuming both travels at the same constant speed). Robots i, k will collide if they travel the same distance to reach their intersection point for the proposed dives. For instance, in Figure 7, Robots 1 and 2 will not collide in the dive from Node 4 to Node 5, as Robot 1 has reached its destination before Robot 2 intersects its trajectory.

In line 14, when the robot's new expansion is determined to be collision-free, the new state for robot j is calculated by the function expandState(), which implements the following equations.

$$X_j^{new}.x = X_j^{exp}.x + d\cos(\theta_j^{new}) \quad (7)$$

$$X_j^{new}.y = X_j^{exp}.y + d\sin(\theta_j^{new}) \quad (8)$$

$$X_j^{new}.\theta = \theta_j^{new} \quad (9)$$

5.2 | Decoupled planning

In contrast to the coupled planner, the decoupled planner creates an individual roadmap and trajectory for each of the m robots. The planner creates these robot trajectories sequentially, such that each robot's path is built to be collision-free with respect to the previous robots' trajectories. Specifically, for each robot r_j, such that $j \in [1, m]$, the planner generates a roadmap $R_j(N_j, E_j)$, where N_j and E_j are the respective node sets and edge sets. Each node $n_{j,i} \in N_j$ is defined by the state in time and space of robot j associated with node i. The state is defined as $[x_{j,i}, y_{j,i}, z_{j,i}, \theta_{j,i}]$. Similar to the coupled planner, each edge $e_{j,ik} \in E_j$ is defined by the node pair $\{n_{j,i}, n_{j,k}\}$.

Algorithm 3: decoupledMultiRobotRRT(n_0)

Input: list of start nodes n_0

Output: bestTrajectories($R_1(N_1, E_1)$: $R_m(N_m, E_m)$)

1: Shuffle order of robots
2: **for** $j \leftarrow 1$ **to** m **do**
3: Add $n_{j,0}$ to $R_j(N_j, E_j)$
4: **for** $k \leftarrow 1$ **to** *numExpansions* **do**
5: $n_{j,exp} \leftarrow$ selectNodeToExpand($R_j(N_j, E_j)$)
6: $n_{j,new} \leftarrow$ expandNewNode($n_{j,exp}$)
7: **if** $n_{j,new}$ in endgame region **then**
8: add $n_{j,new}$ to $endNodes_j$
9: **end if**
10: Add $n_{j,new}$ and edge $(n_{j,exp}, n_{j,new})$ to $R_j(N_j, E_j)$
11: **end for**
12: **end for**
13: **return** bestTrajectories($R_1(N_1, E_1)$: $R_m(N_m, E_m)$)

Algorithm 4: expandNewNode_decoupled(n_{exp})

Input: n_{exp}, *HICs*

Output: n_{new}

1: $d \leftarrow$ getRandomDiveDistance()
2: **if** $rand() < \gamma_{HIC-exp}$ **then**
3: $X_{HIC} \leftarrow$ getStateOfRandHIC(*HICs*)
4: $\theta^{new} \leftarrow atan2(X_{HIC}.y - X^{exp}.y, X_{HIC}.y - X^{exp}.x)$ # expand towards random HIC
5: **else**
6: $\theta^{new} \leftarrow rand(-\pi, \pi)$
7: **end if**
8: **while**
collision(X^{exp}, θ^{new}, $[X_0, \theta_0 ... X_{j-1}, \theta_{j-1}]$)**do**
9: $\theta^{new} \leftarrow rand(-\pi, \pi)$
10: **end while**
11: $X^{new} \leftarrow$expandState(X^{exp}, θ^{new}, d)
12: **return** n_{new}

Algorithm 3 presents the steps for the decoupled planner, and Figure 8 shows a corresponding visualization. As shown in line 2 of Algorithm 3, the algorithm loops over all m robots to make a separate trajectory for each. In line 3, the start node $n_{j,0}$ for a given robot is initialized and added to the robot's roadmap R_j. For a set number of expansions (line 4), an existing node $n_{j,exp}$ is selected from the tree as demonstrated in Algorithm 5 (discussed in Section 5.3.1). In line 6, $n_{j,exp}$ is used as the parent node for generating a new node $n_{j,new}$ (details discussed further in the following section). If the new node's corresponding time is greater than the maximum robot trajectory time, the node is added to $endNodes_j$ in line 8. In line 10, the new node is added to the robot's tree. Once *numExpansions* nodes have been generated, the trajectories for robot j that begin at $n_{j,0}$ and end at a node in $endNodes_j$ are evaluated to decide which trajectory maximizes information gain and should be returned in line 13.

Because the robot trajectories are generated sequentially, the optimal trajectory for each robot is also chosen sequentially. To optimize

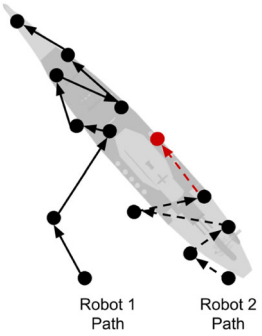


FIGURE 8 Visual representation of decoupled node structure and expansion. Each node is represented by a black dot on each trajectory. Node 6 of Robot 2 shows the next expansion of the Robot 2 RRT trajectory in red; at this stage, the expansion is being checked for collisions with the previously planned Robot 1 trajectory before being added to the tree. RRT, rapidly exploring random tree [Color figure can be viewed at wileyonlinelibrary.com]

(Continues)

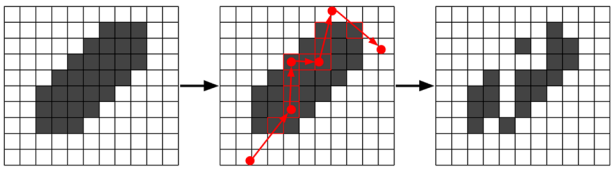


FIGURE 9 Visual representation of information map adjustment performed after each robot is planned for decoupled planner. The original information map, robot trajectory over the information map, and final information map after adjustment are shown in respective order. Gray cells represent cells containing information, while white cells represent areas with no info. Note this figure assumes the planner is being run without angles (Section 3.1.3) or scanning spread (see Section 5.3.3) enabled [Color figure can be viewed at wileyonlinelibrary.com]

the trajectories, the decoupled planner adjusts the information map after each robot trajectory is planned to remove the information observed by the previous robot from the information map. This adjustment is demonstrated visually in Figure 9, which shows the information map before and after a planned robot trajectory. The adjustment helps the decoupled planner converge to 100% map coverage, as each robot is incentivized to gain new information on the map that has not yet been seen by other robots. A disadvantage to this method is that the information gain is not equally dispersed between all robots, as robots that are planned first attain most of the information, leaving little information gain remaining for the robots planned later.

5.2.1 | Node expansion for decoupled planning

Algorithm 4 describes the expansion step that takes place on line 6 of the overall decoupled algorithm (Algorithm 3). The algorithm selects a random dive distance within the minimum and maximum dive lengths on line 1. On lines 2 through 6, the dive heading is determined by the same logic seen in Algorithm 2 and is discussed further in Section 5.3.2. On line 8, collision checking is performed.

The collision checking function takes in the new node state $X_{j,new}$, the proposed heading $\theta_{j,new}$, and the list of previously planned robot trajectories; this list is composed of robot states and headings, represented by the list $[X_0, \theta_0 ..., X_{j-1}, \theta_{j-1}]$.

The function checks to ensure that the proposed new node state does not collide with any previously planned trajectories by calculating potential intersections between them. The function uses several cases to optimize collision checking runtime and minimize the number of parameterized line-intersection collision checks required. If the function detects a collision, a random new heading for the new node is chosen by repeating the call to the expandNewNode function until the expansion is collision-free. Once collision-free, the new state is calculated in line 11 using Equations (7)–(9) , as with the coupled planner. Finally, the corresponding new node n_{new} is returned.

5.3 | Improvements common to coupled and decoupled planners

To improve map coverage and decrease runtime, several modifications are made to the general RRT algorithm. First, a novel node selection framework stores nodes into time-discretized cells, simplifying the selection process by choosing a node in a random time cell rather than a random space cell as done by other researchers (Clark, 2004; Kindel, Hsu, Latombe, & Rock, 2000). Second, high information configuration (HIC) nodes are identified and stored to guide the planner toward high information areas in the map. Third, a modification called scanning spread is included to better model the information gain attained by a given robot trajectory.

5.3.1 | Node selection for expansion

In sampling-based motion planning, it is often beneficial to discretize the search space for the sake of node declustering. Previously, motion planners have discretized the configuration space in several ways. In Kindel et al. (2000), the planner divides the configuration space into grid cells, selecting a random cell, then randomly selecting a node within that cell. This method prevents node clustering by giving nodes in less populated grid cells a higher probability of being expanded upon. Bohlin (2001) presents a method of reducing runtime by using both a local and global planner to search a grid-discretized configuration space. In this approach, the global planner restricts the portion of the grid the local planner is allowed to search and expands this area until a solution connecting the start to the goal is found. In Lindemann and LaValle (2003), a similar multidimensional grid is developed; the algorithm recursively divides each grid cell of the configuration space grid into smaller cells until a trajectory to the goal point is found.

The node selection method presented in this paper discretizes the search space based on time rather than the configuration space. When the RRT planner is called, a set number (typically 5–10) of time-indexed node containers are initialized in a dictionary to organize the nodes as the tree is created. This structure makes it easy to place a node into one of the discrete, time-indexed cells after it is created.

To initialize the time-indexed cells, the root node is placed in the lowest time cell and all other cells are initialized as empty lists. An extra cell is created at the end of the dictionary that holds all nodes over the maximum time; this cell is called *endNodes* and is never chosen from for node expansion. In addition, a secondary list f tracks which cells already contain nodes to ensure the algorithm does not expand from an empty cell. During each expansion, the new node is sorted into the appropriate cell based on its timestamp, and f is modified accordingly.

Algorithm 5: RRT Node Selection

1: **if** $len(H) \neq 0$ **and** $rand() < \gamma_{HIC-sel}$ **then**

2: $n_{exp} = H[randIndex]$

3: **else**

4: randCell = cells $[rand() \in f]$

5: $n_{exp} = rand() \in dict$ [randCell]

6: **end if**

Algorithm 5 shows the process used to select a node to expand upon. The planner utilizes one of two selection methods for each node selection, and the relative use probability of the methods is controlled by the parameter $\gamma_{HIC-sel}$. If the first method is chosen, a random HIC node (discussed in the following section) from the list of HICs H is chosen to expand from (line 1). Otherwise in line 3, a random cell is chosen from the list of filled cells f, and a random node n_{exp} is selected from that cell.

After node selection in both planners, the coupled and decoupled methods follow the steps shown in Algorithms 1 and 3, respectively, and a new node n_{new} is created and added to the RRT. After n_{new} is created, the planner stores the new node in the appropriate cell. To do so, the planner checks if the new node's time $n_{new,t}$ is greater than the max time t_{max}; if so, the node is appended to the last cell. Otherwise, the node is placed in the appropriate bin by calculating the cell index c. This value is computed from $n_{new,t}$, t_{max}, and the total number of discrete time cells *numCells*; the calculation is shown in Equation (10) . To ensure all filled cells are marked as such, the algorithm checks if c is in the filled cells list f, and if not adds c to f. Finally, n_{new} is added to the cell with index c.

$$c = int(n_{new,t} \times numCells / t_{max}). \tag{10}$$

This method provides declustering by discretizing the workspace in time. Similar to how Kindel et al. (2000) shows that discretizing the search space spatially provides node declustering, dividing time into temporal chunks causes the node selection process to be more likely to expand from nodes with a higher timestamp. This is because the probability of randomly picking a particular node in a bin with fewer other nodes is higher than the probability of picking a node in a more filled (and typically shorter-time) bin. The method also benefits algorithm runtime; by using fewer discrete node containers than spatial binning, the algorithm does fewer computations when performing operations over the entire set of node containers.

5.3.2 | High information configuration nodes

To optimize information gain, the planner is motivated to sometimes create new expansions based on nodes previously marked as containing high information. The HIC nodes are stored in a list H, composed of all nodes in the tree for which the immediately preceding edge gathers an amount of information above a specified threshold. Both the node selection and expansion components of the general RRT algorithm were modified to encourage the creation of new nodes around HICs.

In the node selection algorithm, a random HIC node is chosen for expansion with probability $\gamma_{HIC-sel}$, where $\gamma_{HIC-sel}$ is a tuning parameter. As outlined in Algorithm 5, if a random number chosen from a uniform distribution between 0 and 1 is less than the threshold $\gamma_{HIC-sel}$, a random HIC is picked from H as the node to expand upon. This encourages the creation of trajectories in information rich areas. In the node expansion algorithms for both coupled and decoupled planners (Algorithms 2 and 4, respectively), a robot dives toward a random HIC with probability $\gamma_{HIC-exp}$. If a random number is less than $\gamma_{HIC-exp}$, the expansion algorithm picks its heading by expanding toward a random HIC. This incentivizes the algorithm to expand nodes toward areas of high information.

To evaluate the utility of HICs, tests were run comparing the map coverage and runtime of the motion planning algorithm using HICs in both node selection and expansion. Figure 10a shows that using HICs in selection provides an increase of approximately 5% in coverage in comparison to having no HICs, while using HICs in expansion doubles map coverage. Further, combining both methods yields map coverage percentages that are almost triple the original amount without HICs. Figure 10b indicates that incorporating HICs in the planning algorithm does not increase runtime significantly.

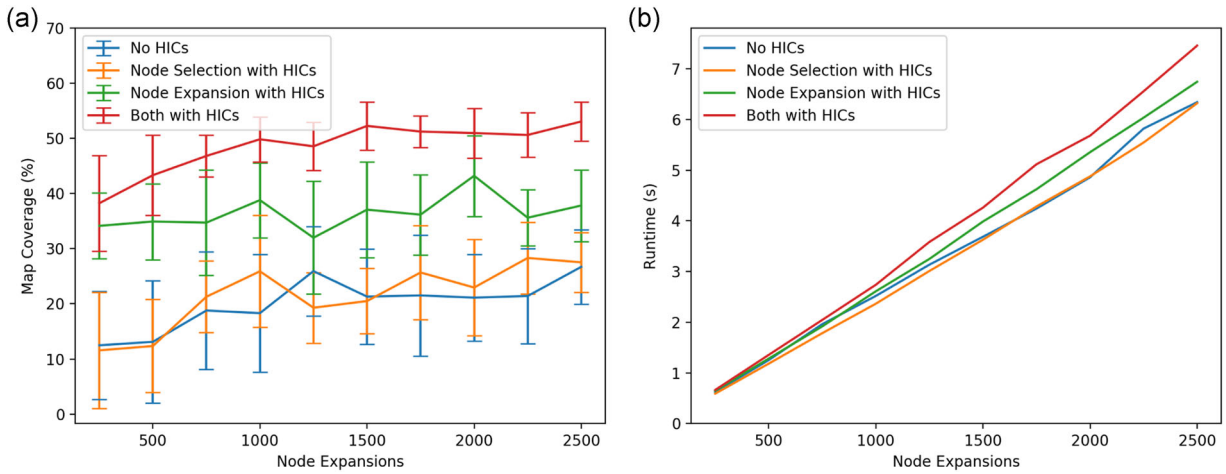


FIGURE 10 Plots of (a) coverage and (b) runtime as a function of the number of node expansions for a single robot, with $\gamma_{HIC-sel} = \gamma_{HIC-exp} = 0.5$. The results indicate that when HICs are incorporated in both node selection and node expansion, map coverage improves by approximately 25%

5.3.3 | Ray casting and scanning spread

To accurately and efficiently calculate the information captured by the edge created in each expansion step of the planner, a new method was used to construct the set of visited cells stored by each node. Ray casting is essential for reducing the runtime of the cell aggregation process, while scanning spread allows the planner to approximate the reality of the viewing breadth of the AUV's onboard video camera.

Ray casting is implemented using vectorization to efficiently find all cells visited by the edges of the robot trajectories. Using this method helps reduce the computational cost of finding the trajectory's visited cells, as it removes the need to iterate over the entire information map. This was the case in previous work (Viswanathan et al., 2017). However, vectorization alone is insufficient because only cells directly intersecting the raycasted trajectory are marked as "visited cells." To account for the wide-angle of the onboard video camera, a scanning spread function is implemented such that the set of visited cells more accurately represents all reconstruction data captured. The scanning spread function marks additional cells surrounding each cell on the AUV trajectory as visited, and the number of additional cells marked is set by a constant variable σ. For example, when σ is set to 1, the AUV trajectory marks a 3×3 grid centered at each visited cell as visited. The impact of varying σ values is demonstrated in Figure 11.

In addition to allowing the algorithm to more accurately assess the map coverage of a planned trajectory, this method also increases the runtime of the planner significantly for higher σ values. This is because the number of cells marked by the scanning spread increases with $O(\sigma^2)$; thus, the number of operations per node expansion also increases by σ^2. Figure 11c verifies that the planner is in fact $O(N^2)$ with respect to σ. To obtain optimal photogrammetry data, some scanning spread is necessary to accurately capture which cells are "seen" by a given trajectory. As such, the tests for this paper were run with a σ value of two, unless indicated otherwise.

5.3.4 | Comparisons of basic versus improved planners

To test the effectiveness of the improvements made to the basic RRT algorithm, map coverage and algorithm runtime tests were run comparing a basic RRT to the improved algorithm. These tests were run with one robot, so that the tests were not characteristic of features specific to the decoupled or coupled planner. The tests were run on a low resolution 45×45 cell information map of the Bristol Beaufighter plane wreck. Data for coverage and runtime was taken, and plots for coverage per runtime were created to compare the efficiency of the algorithms at obtaining information. Figure 12 demonstrates the results, which show that the improved algorithm was much more efficient at gaining map coverage than the basic algorithm (note the difference in z-axis scales between a and b). In Figure 12a, the plot for the improved algorithm is shown, which ranges from efficiency values of 0.3297–3.6905 with an average efficiency of 1.1480% coverage per second. Contrarily, the basic algorithm efficiency values ranged from 0.0025 to 0.0397 with an average value of only 0.0133% coverage per second. Overall results showed how the modifications to the general RRT algorithm vastly improved the algorithm's effectiveness.

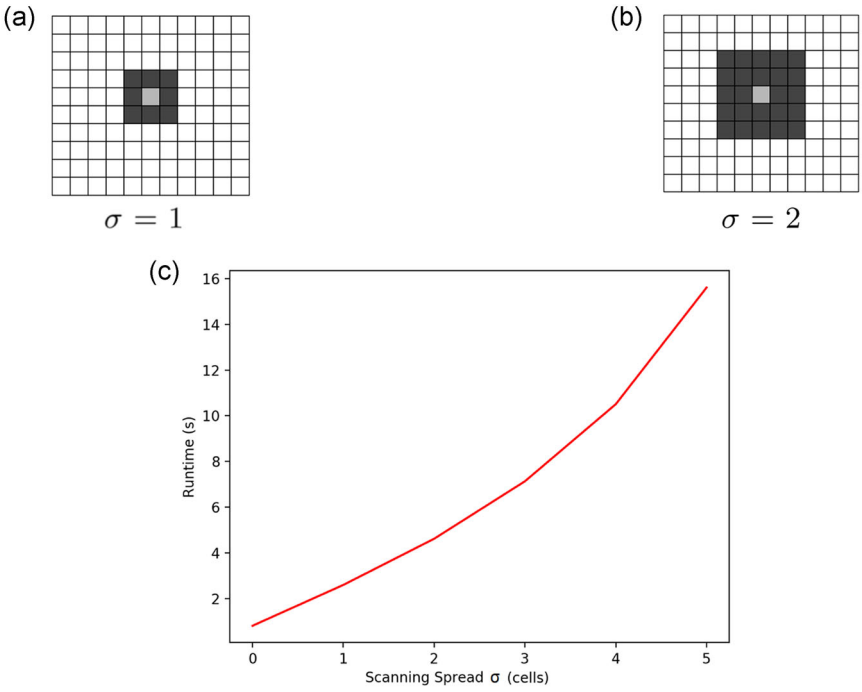


FIGURE 11 Cells marked as visited by varying σ values. The center light gray cell indicates the cell intersected by the autonomous underwater vehicle trajectory, and the surrounding dark gray cells indicate additional cells marked as visited by the scanning spread. (c) shows the runtime of the scanning spread functionality [Color figure can be viewed at wileyonlinelibrary.com]

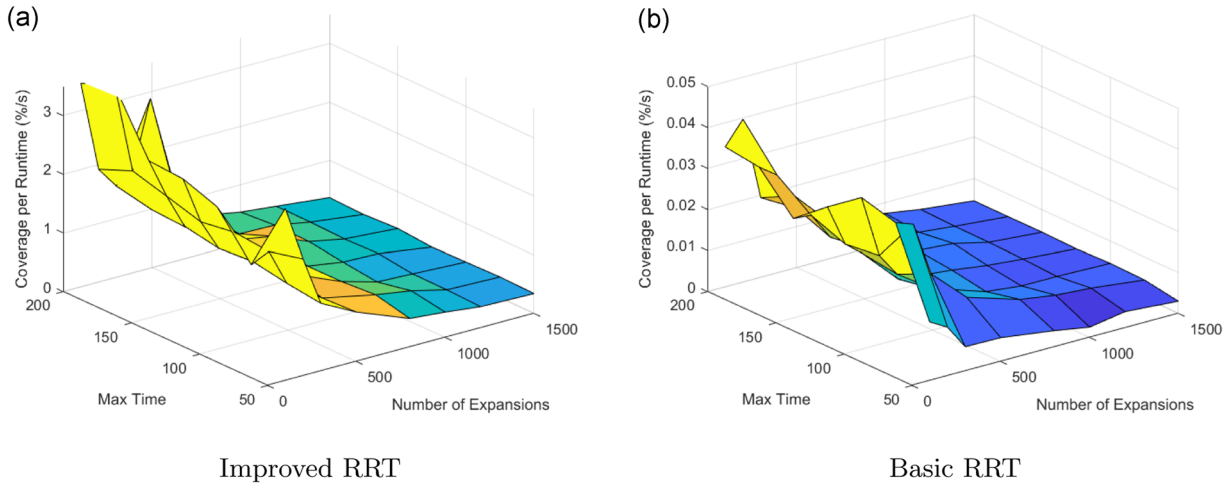


FIGURE 12 Comparison plots for basic and improved planners for one robot. Plots for the improved planner were generated with the angle map feature disabled to ensure a fair coverage comparison. Note: the z-axes use drastically different scales [Color figure can be viewed at wileyonlinelibrary.com]

6 | VALIDATION EXPERIMENTS

To test the performance of the coupled and decoupled planners, four types of validation experiments were carried out:

1. Generation of example trajectories over several different maps to demonstrate planner effectiveness on both single and multiple site maps.
2. Simulations comparing how information map coverage and runtime of each planner differed when varying the number of robots and the number of discrete sites of interest on the information map.
3. Simulations comparing the coupled and decoupled planners, testing how their information map coverage and runtime differed when run with the same constants on the same information map.
4. AUV deployments using an Oceanserver Iver3 AUV to validate the generated trajectories' effectiveness at collecting photogrammetry data on real sites of interest.

6.1 | Example trajectories

The decoupled planner was tested using information maps of different individual wrecks, as well as maps containing multiple wrecks in close proximity to each other. First, the planner was tested in simulation for 1–4 robots on the five following real underwater archeological sites: the Bristol Beaufighter, Schnellboot S-31, HMS Stubborn, HMS Maori (all located along Malta's coast), and an unnamed shipwreck located off the coast of Catalina Island. These five maps contained one primary site of interest. Example trajectories from the decoupled planner are shown for the five sites in Figure 13 . For visual clarity, an exhaustive time horizon was not used to distinguish between individual robot paths. The resultant trajectories show that the planners were able to successfully navigate the site of interest within the information map and obtain coverage of the site from several different angles. There is also a higher density of trajectory nodes near areas of high value on the information maps.

Additional tests were run on maps with *multiple* sites of interest. The map shown in Figure 14a was created from sonar data containing two sites of interest. A second multiple site map was *artificially* created by duplicating the sites of interest from the map shown in Figure 14a and distributing them randomly throughout the map. Example decoupled planner trajectories for these two maps are shown in Figure 14.

The generated trajectories shown in Figure 14a demonstrate how the planner was able to utilize multiple robots to visit both sites of the two site map and get better coverage of the map than one robot could have. This is also seen in Figure 14b; by increasing the number of robots when mission time is limited, coverage is increased because individual robots can visit unique areas of the map. A further discussion of the effects of number of robots and number of sites of interest on map coverage is presented in Section 6.2.

6.2 | Map coverage of varying sites and robots

Experiments were conducted for both planners testing the relationship between map coverage, the number of sites of interest on a given information map, and the number of robots used to plan a given mission. Trials for 1–10 robots were run on a sequence of 10 maps which ranged from 1 to 10 sites of interest. In every trial, the amount of total information in the map was kept constant; this was necessary to ensure a fair comparison of information covered between maps. In addition, all cells with non-zero value contained the same amount of information, such that robots were equally incentivized to visit any cells of interest. Example information maps are shown in Figure 15 .

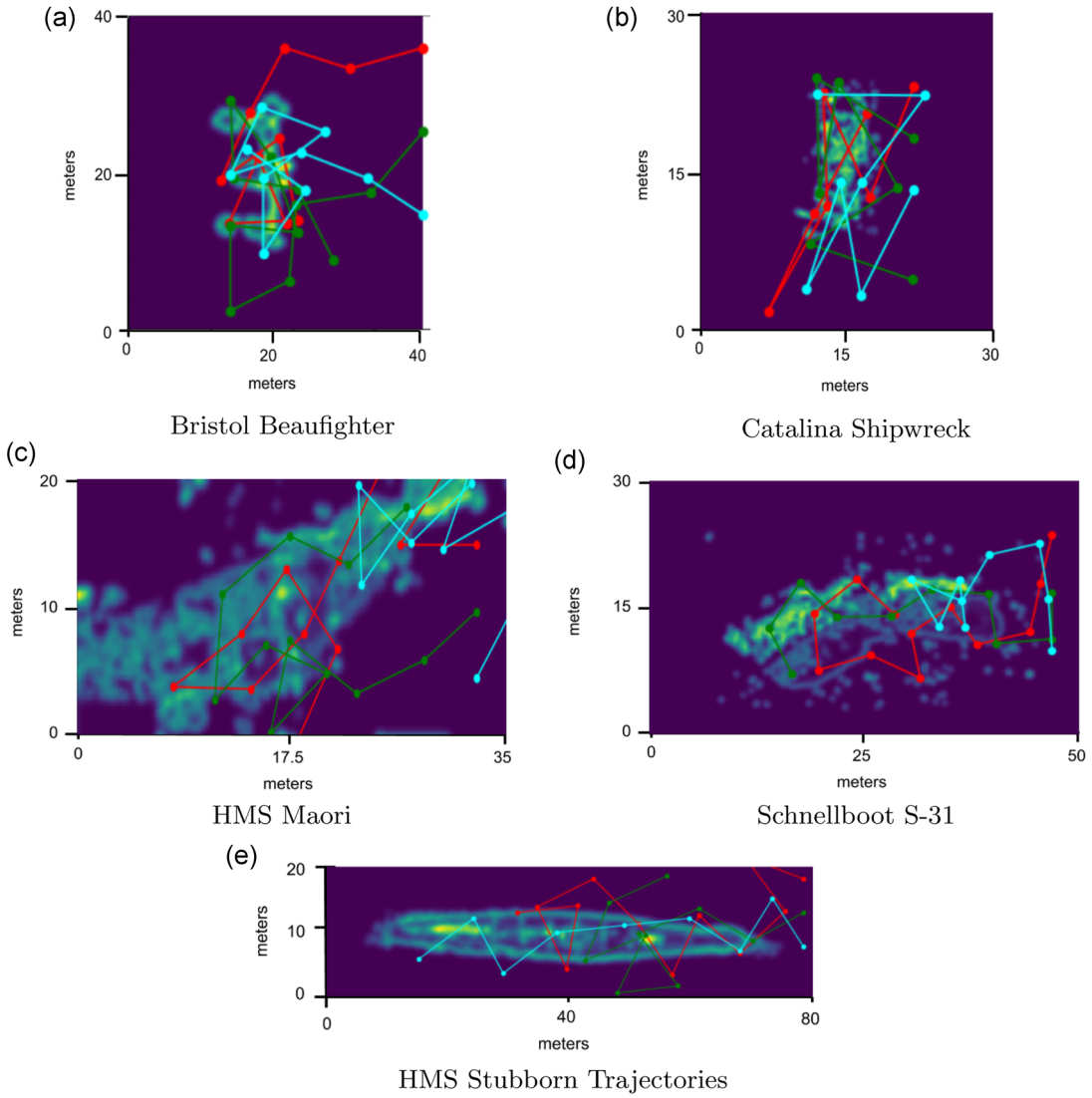


FIGURE 13 Example mission plans for decoupled planner on single site information maps; trajectories are shown in red, green, and blue. The plans shown were not run to exhaustive time horizons for visual clarity. (a) Bristol Beaufighter, (b) Catalina Shipwreck, (c) HMS Maori, (d) Schnellboot S-31, and (e) HMS Stubborn Trajectories [Color figure can be viewed at wileyonlinelibrary.com]

Four different versions of these tests were conducted. Both the decoupled and coupled planners were tested, and each of those planners was tested with and without the view angles feature described earlier in Section 3.1.3. Figure 16 shows the results of these tests. Figure 16a shows that for the decoupled planner, the map coverage converges to almost 100% for all maps when $m > 8$. By contrast, the coupled planner (Figure 16b) coverges to 100% coverage only on the single site map with 10 robots.

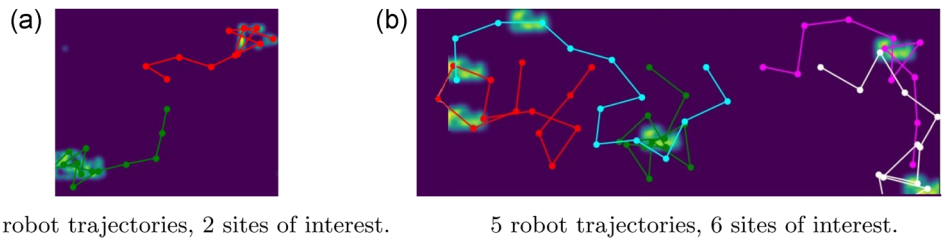


FIGURE 14 Example mission plans for decoupled planner on multiple site information maps generated from sonar data. Note that axis scales are omitted as the sites were synthetically generated. (a) 2 robot trajectories, 2 sites of interest, (b) 5 robot trajectories, 6 sites of interest [Color figure can be viewed at wileyonlinelibrary.com]

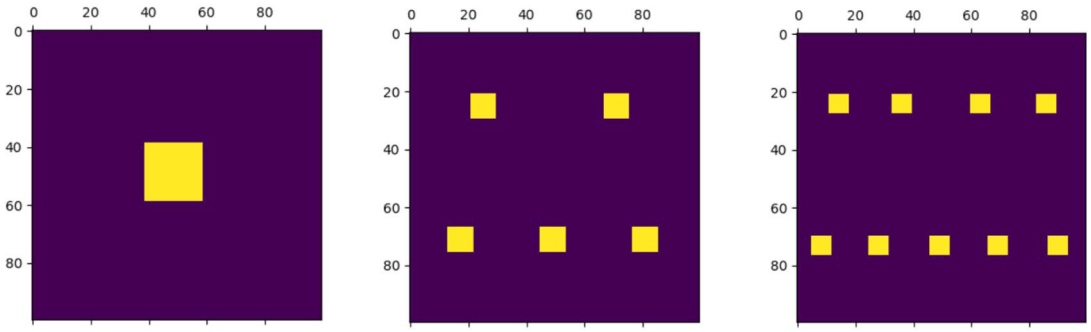


FIGURE 15 Examples of the information maps used on robots vs. sites tests [Color figure can be viewed at wileyonlinelibrary.com]

6.3 | Comparisons of coupled versus decoupled planners

To evaluate the trade-offs between the coupled and decoupled planners for use in fieldwork, performance plots were generated in simulation to compare both information map coverage and runtime. Figure 17a compares what percentage of the information map is covered as the number of robots increase. The decoupled planner asymptotically approaches 100% coverage of a 45×45 cell map of the Bristol Beaufighter wreck, while the coupled planner converges to about 70% coverage. Figure 17b shows that the runtimes of both planners scale linearly with the number of robots, with the decoupled planner being about 20% more efficient.

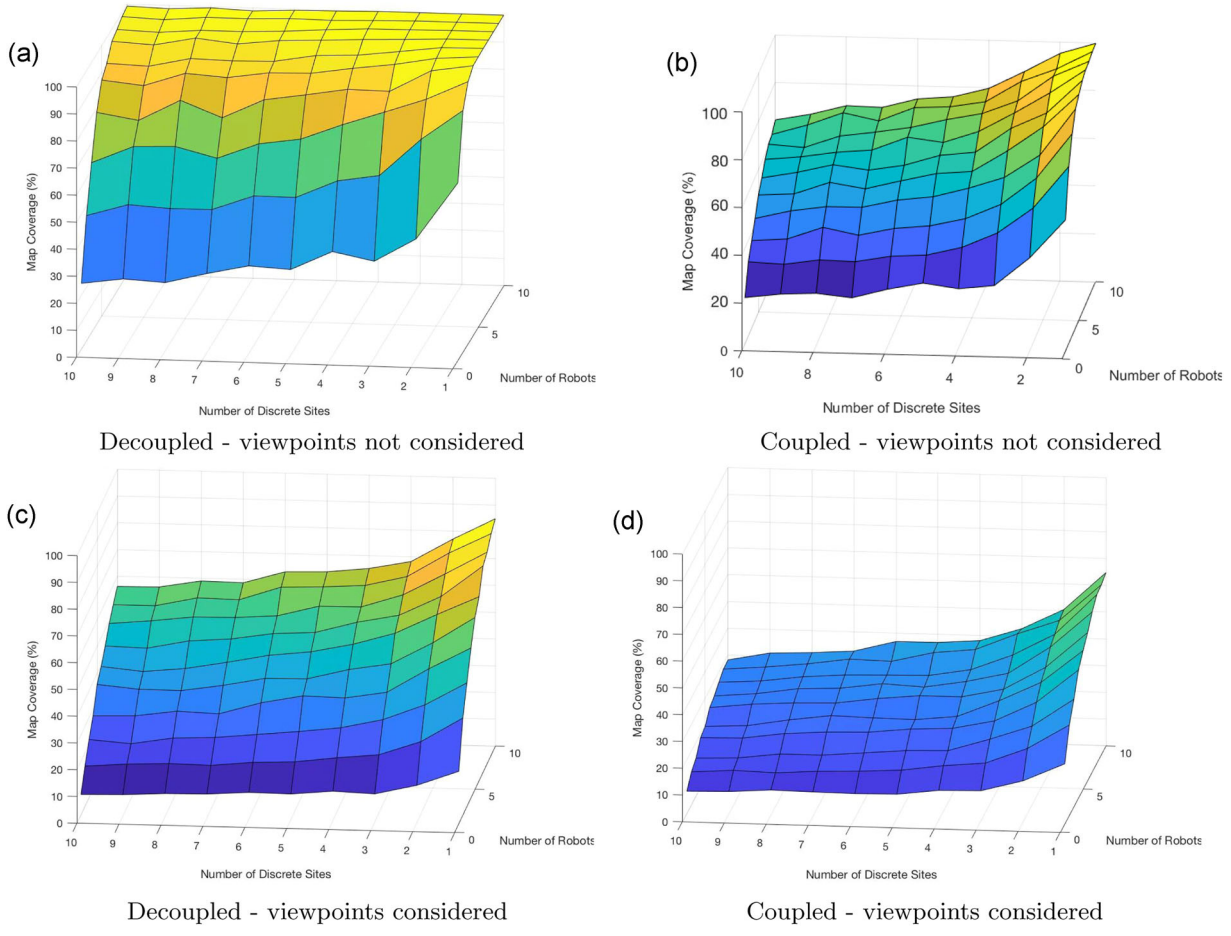
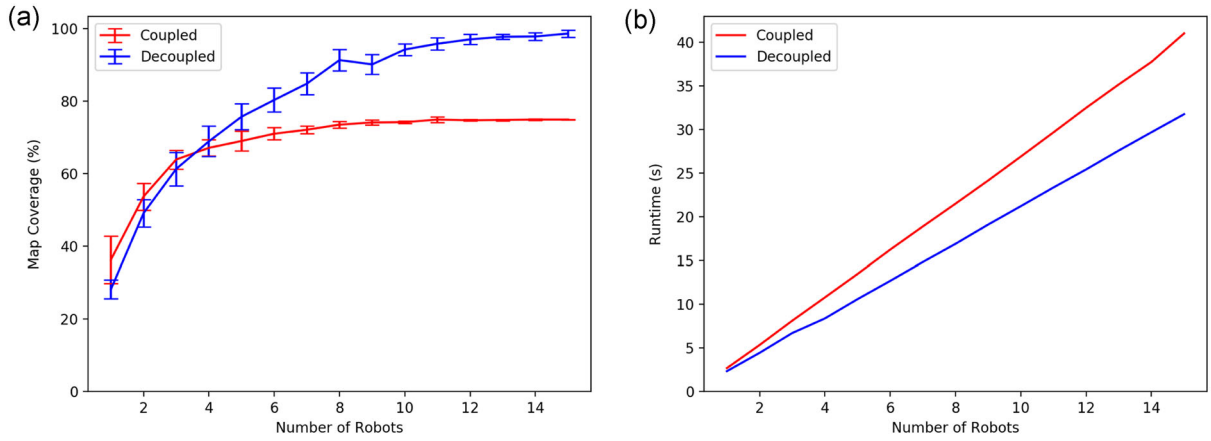


FIGURE 16 Plots of map coverage as a function of the number of robots and the number of sites of interest on a given information map. Trials were run with and without angles feature enabled: (a) is coverage for the decoupled planner without angles enabled, (b) is coverage for the coupled planner without angles enabled, (c) is coverage for the decoupled planner with angles enabled, and (d) is coverage for the coupled planner with angles enabled. Each data point is averaged over 10 trials. (a) Decoupled—viewpoints not considered, (b) Coupled—viewpoints not considered, (c) Decoupled—viewpoints considered, (d) Coupled—viewpoints considered [Color figure can be viewed at wileyonlinelibrary.com]



Information map coverage as a function of number of robots.

Runtime as a function of number of robots.

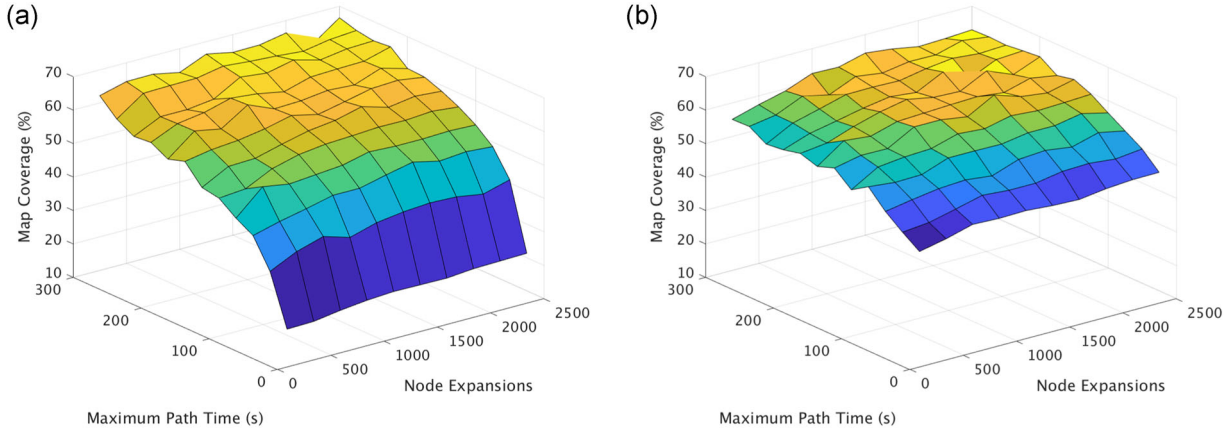
FIGURE 17 Information map coverage and runtime plots comparing the coupled and decoupled planners. Each data point is averaged over 10 trials. (a) Information map coverage as a function of the number of robots. (b) Runtime as a function of the number of robots [Color figure can be viewed at wileyonlinelibrary.com]

The convergence behavior of the two planners can be accounted for by two different phenomena. In the decoupled planner, the information map is adjusted after each robot is planned to reflect only the information that has yet to be covered by any robot. Thus, as more and more robots are planned, the planner converges to full coverage because the robots will eventually find almost every cell with information. In the coupled planner, the coverage percentage does not reach 100% because as the number of robots increases, it becomes highly unlikely that every single robot in a given node will pick a trajectory in a reasonably advantageous direction. Statistically, the highest-scoring individual nodes in the coupled planner, even for many robots, will likely contain robots that do not contribute significantly to the total information gain.

In Figure 18, the two planners exhibit the largest difference in performance when trajectory time limits were small. For example, when the maximum trajectory time was 250 s, the decoupled planner reached 40% coverage, while the coupled planner remained at 10% coverage. This can again be attributed to the fact that the decoupled planner adjusts the information map after each robot is planned, allowing subsequent robots to expand more reliably toward previously unobtained information.

Further testing was conducted to assess how efficiently each of the planners obtained coverage of the site of interest. To test path time-efficiency, each planner's performance was measured as a function of the maximum path time and number of robots, shown in Figure 19.

Figure 19 shows that regardless of the maximum allowed trajectory time, the coupled planner still eventually converges to about 70% coverage given increasing numbers of robots. By contrast, additional robots give the decoupled planner additional performance



Coupled Planner

Decoupled Planner

FIGURE 18 Plots of information map coverage as a function of maximum mission duration and number of node expansions for 3 robots. Each data point is averaged over 10 trials. (a) Coupled planner and (b) decoupled planner [Color figure can be viewed at wileyonlinelibrary.com]

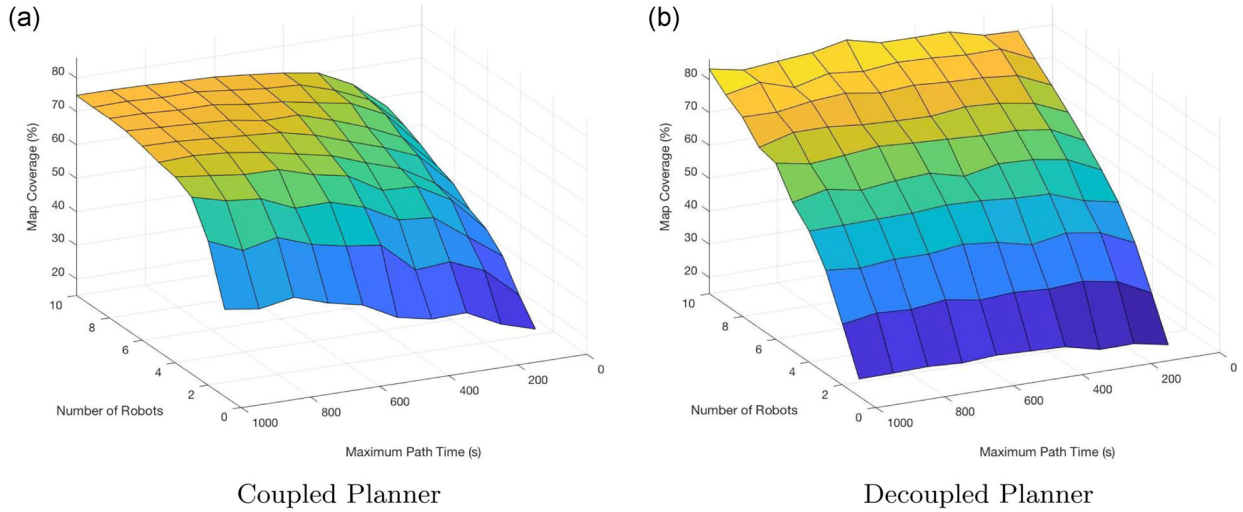


FIGURE 19 Plots of information map coverage as a function of maximum mission duration and number of robots. Each data point is averaged over 10 trials. (a) Coupled planner and (b) decoupled planner [Color figure can be viewed at wileyonlinelibrary.com]

at every value of maximum allowed time. This behavior is consistent with the convergence results in Figure 17. However, for a maximum path time of 700 s or greater and four or fewer robots, the coupled planner attains 20% more coverage than the decoupled planner. This shows that the coupled planner can be advantageous when mission time is a secondary concern and the supply of AUVs to deploy is limited.

For fieldwork purposes, the simulated performance behavior of the two algorithms defines situations in which each might be most useful for researchers with particular equipment limitations. Because each robot collects as much information as is possible in the allotted time, the decoupled approach tends toward full coverage of any given map, no matter the size or information density. For the map used to generate Figure 17a, the decoupled planner distributes the information gathering workload efficiently up to seven robots, where it reaches 85% coverage. Above seven robots, each individual robot collects a decreasing amount of information. However, for a larger and more complex map, the decoupled planner would continue to distribute workload efficiently for as many robots would be necessary to reach a similar threshold. Thus, the decoupled planner would be more efficient than the coupled planner for large sites and when the number of available AUVs is relatively large (particularly more than four). However, the gains in coverage for small numbers of robots and long trajectories seen in Figure 19 combined with its even distribution of workload mean that the coupled planner could be advantageous for researchers in certain fieldwork situations. Specifically, the coupled planner archetype would be advantageous whenever the target site is relatively small and dense, long trajectories are permissible, and four or fewer robots are available.

6.4 | Fieldwork results

The coupled and decoupled planners were tested in the field through AUV deployments off the coast of Malta to reconstruct the HMS Maori shipwreck. Because only one Iver3 AUV was available for fieldwork, the multirobot trajectories were executed sequentially to best approximate the results of simultaneously deploying multiple AUVs.

6.4.1 | Implementation details

For field deployments, the trajectories returned by the motion planners were converted to Iver3 AUV mission files through a series of steps, as illustrated in Figure 20. This process includes the addition of surface waypoints to the trajectories, scaling of the

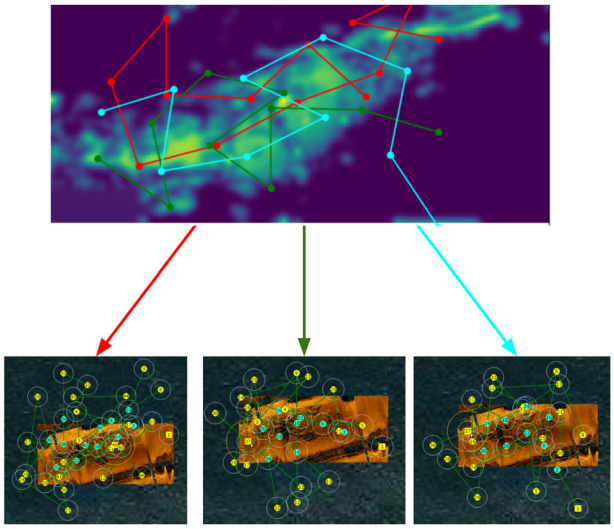


FIGURE 20 Flowchart demonstrating pipeline from path generation to scaled, usable AUV trajectories for three robots in VectorMap. Each final robot trajectory is projected over a sonar scan of the HMS Maori shipwreck for reference [Color figure can be viewed at wileyonlinelibrary.com]

trajectories to fit the actual wreck size, and converting the waypoints from Cartesian to GPS coordinates. After the planners are executed, the edges of each trajectory are modified to include a surface waypoint before and after each dive. For each edge in a robot trajectory, the original edge length d_{dive} is replaced with d_{total} by $d_{\text{total}} = d_{\text{dive}} + 2d_{\text{surf}}$.

Thus, the new edge length is the distance required to travel from the surface of the water to the desired depth (above the highest point of the target wreck), the distance traveled at that depth, and the distance required to resurface. The surface waypoints allow the robot to relocalize with GPS measurements between dives, as well as maximize the amount of photogrammetry data collected at the desired depth from the surface.

Once surface waypoints are added, the scaling of the trajectories is adjusted such that each cell on the information map corresponds to a 1×1 m space on the physical shipwreck. To make this adjustment, the conversion factor between cells and meters is calculated and denoted by C.

To determine C, the length and width of the shipwreck in meters is extracted from the sonar file using the SonarWiz software. Then, the same measurements are taken on the corresponding information map in the unit of map cells. The two values are used to determine the meters per cell ratio, for example, C. Trajectory lengths and their related variables are multiplied by C to either stretch or shrink the entire trajectory and achieve the desired 1×1 m cell scaling. After being properly scaled, the GPS coordinates of the upper left corner of the information map are used to output the trajectory's waypoints converted to GPS coordinates. The coordinates for the upper left corner of the map are determined using the georeferenced sonar files used to create the information map. Through these steps, simulated plans are converted into missions containing GPS coordinates that can be directly executed by the Iver3 AUV.

The depths of each dive can also be adjusted to improve the quality of photogrammetry data collected. Initially, the trajectories used a constant dive depth for every dive, regardless of the dive's location on the information map. However, the robot trajectories should dive as close to the shipwreck as possible to capture ideal photogrammetry data. Thus, a depth feature was added to the planner, such that a depth map can be provided to the planner in addition to the information map. This map is used to decide the robot's dive depth on each dive, based on the elevation of the wreck. On each dive of the trajectory, the depth is set by ray casting between a given node n and its parent n_{parent}. The minimum depth from the surface of any cell intersected by this ray casting process is assigned to be the depth of the edge defined by n and n_{parent}. This process ensures that any given dive will not collide with the wreck and that the planner can accurately account for collision checking at the ends of dive segments.

TABLE 1 HMS Maori AUV deployments

Planner	# of robot trajectories	Varied depth?	Depth from surface (m)	Camera angle (deg)
Coupled	1	No	6	45
Coupled	2	No	6	45
Decoupled	1	No	4	90
Decoupled	3	No	5	45
Decoupled	3	No	5.5	45
Decoupled	1	Yes	5.5–6.5	90
Decoupled	2	Yes	5.5–6	90

Note: All trajectories were executed sequentially with a single AUV.

6.4.2 | Fieldwork trials

For all missions, the AUV was equipped with a GoPro HERO4 to collect video data of the HMS Maori shipwreck. The camera was angled either directly downward, downward at a 45 degree angle to the surface, or facing directly sideways to the left. Different camera orientations were used based on the trajectory to capture optimal overhead and side-view footage of the Maori. An advantage of

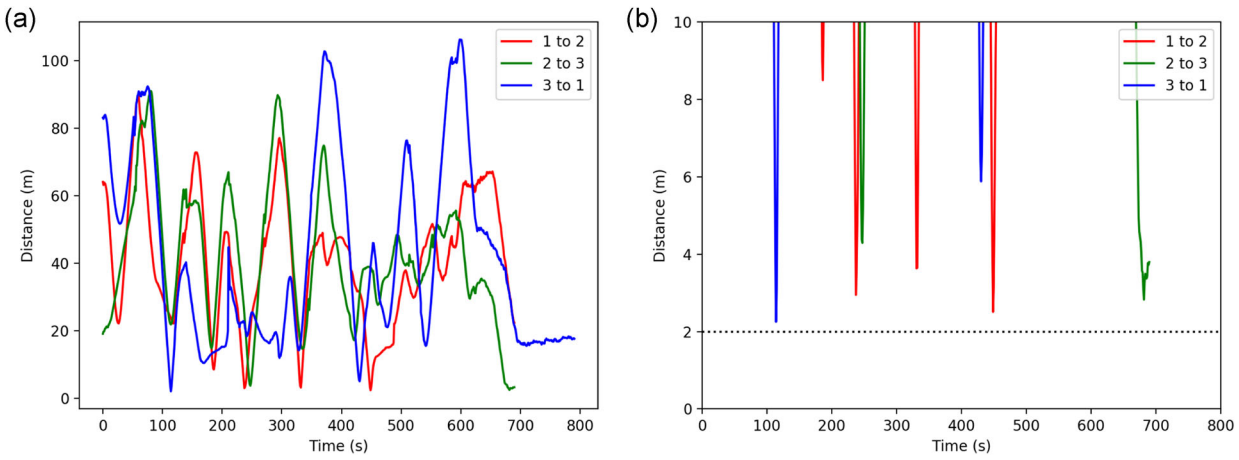


FIGURE 21 (a) Plot of pairwise distances between a set of three single AUV deployments (time-synced) corresponding to an output of the decoupled planner for three robots, for example, the fourth row of Table 1. (b) Zoomed in version of (a) to highlight the minimum distance between AUVs. AUV, autonomous underwater vehicle [Color figure can be viewed at wileyonlinelibrary.com]

FIGURE 22 HMS Maori reconstruction images [Color figure can be viewed at wileyonlinelibrary.com]

conducting collision checking in a 2D horizontal plane is that no AUV will ever be below another if deployed simultaneously and thus an AUV will not appear in another AUV's downward-facing camera footage. Table 1 summarizes the missions from which video data was used to generate a photogrammetric reconstruction of the shipwreck. For deployments with multiple robot trajectories planned, such as in row 2 of Table 1 where 2 trajectories were planned, the same AUV was deployed to sequentially follow each of the robot trajectories. For each set of multirobot trajectories generated (e.g., each row), a single information map was first initialized with zero information gain then updated as the planner constructed the different trajectories. However, an alternative approach that could be used in the future, for large areas of interest that require multiple deployments, might be to only initialize the information map once for a series of multirobot deployments. Figure 20 shows trajectories planned by the decoupled planner for three robots, which was executed by the AUV at 5 m and then again at 5.5 m from the surface of the water. To generate missions with varied dive depths, a crude depth map was created based on literature about the bay and knowledge of the shipwreck's structure.

Fieldwork results showed successful runs for the decoupled planner, coupled planner, and decoupled planner with the depth map functionality. Both motion planners created trajectories which allowed the AUV to obtain valuable footage of the shipwreck with a closeup, over-top view. Decoupled depth map trajectories also obtained closeup footage of the shipwreck, helping to better capture the 3D geometry.

Although it was infeasible to execute simultaneous multi-AUV missions, it was possible to verify the efficacy of the multirobot collision checking procedure. For each set of multirobot trajectories output by the decoupled planner, the AUV was deployed sequentially to follow each individual robot trajectory. In Figure 21, the robot telemetry logs from one such set of missions visiting the HMS Maori is used to illustrate that if three robots had executed these trajectories simultaneously, no collisions would have occurred. Specifically, the minimum distance between any two robots at a given timestamp was greater than 2 m.

Using the video data gathered from the field deployments, a 3D reconstruction of the HMS Maori shipwreck was created using Agisoft PhotoScan. PhotoScan was chosen as it is widely used by marine archeologists and does not require camera calibration (Yamafune, Torres, & Castro, 2017). Images of the resulting reconstruction are shown in Figure 22. The reconstruction was generated from 2,701 video frames, which were manually chosen from approximately 11 min of combined video footage from all deployments. This was done to achieve the best reconstruction quality, by a researcher experienced with photogrammetry. Investigating the expected number of images required for adequate reconstruction of a typical wreck is an avenue for future work.

7 | CONCLUSION

This paper presents coupled and decoupled multi-AUV motion planning methods for photogrammetric reconstruction, improving upon existing methods for AUV information gathering. The planners implement a number of novel contributions to the baseline RRT algorithm. First, the objective function is formulated in terms of a discretized information map with discrete viewpoints for each grid cell. Considering all viewpoints in discrete space is favorable to randomly sampling a continuous space because it ensures the robustness of the planner to reliably visiting unique viewpoints around the wreck. Second, the planners incorporate the dive pattern of the AUV in both trajectory generation and collision checking, which leads to an atypical expansion type over other RRT planning problems. This means the time to dive, surface, and turn are directly considered in the trajectory generation and collision checking algorithms.

An analysis of performance trade-offs between the two planning approaches is also presented. The coupled planner is able to achieve greater map coverage given fewer robots and longer path times, demonstrating up to a 20% increase in coverage over the decoupled planner. However, for large numbers of robots, the decoupled planner is shown to be advantageous, since robot trajectories are generated sequentially and can thus take into account viewpoints that have yet to be visited. The decoupled planner is also able to converge to 100% map coverage, while the coupled planner stabilizes at 70% coverage. Finally, the decoupled planner had a consistently lower runtime than the coupled planner, due to fewer collision checks. These observations were verified both in simulation and in fieldwork. To validate the effectiveness of the planners, the Iver3 AUV was deployed over the HMS Maori shipwreck in Malta to collect video data for photogrammetric reconstruction of the wreck.

Future work for this project includes the simultaneous deployment of multiple AUVs to validate the effectiveness of collision checking as well as further comparing the two MRMP approaches discussed. Improvements to the coupled and decoupled planners can be explored, as well as new methods that may involve combining elements of both approaches. Future work could also include the usage of a multibeam sonar technology to obtain an elevation map of a given shipwreck; this could then be used to test the previously mentioned depth map functionality.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 1460153. This study was performed in part at the Claremont Colleges Robert J. Bernard Biological Field Station.

ORCID

Jane Wu http://orcid.org/0000-0003-2700-0732

REFERENCES

Bern, M., Eppstein, D., & Gilbert, J. (1994). Provably good mesh generation. *Journal of Computer and System Sciences*, *48*(3), 384–409.

Bingham, B., Foley, B., Singh, H., Camilli, R., Delaporta, K., Eustice, R., & Sakellariou, D. (2010). Robotic tools for deep water archaeology: Surveying an ancient shipwreck with an autonomous underwater vehicle. *Journal of Field Robotics*, *27*(6), 702–717.

Blondel, P. (2010). *The handbook of sidescan sonar*. Berlin-Heidelberg: Springer Science & Business Media.

Bohlin, R. (2001). Path planning in practice; lazy evaluation on a multi-resolution grid. *2001 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 49–54). IEEE/RSJ.

Burgard, W., Moors, M., Stachniss, C., & Schneider, F. (2005). Coordinated multi-robot exploration. *IEEE Transactions on Robotics*, *21*, 376–386.

Clark, C. M. (2004). *Dynamic robot networks: A coordination platform for multi-robotsystems* (PhD thesis). Stanford University.

Clark, C. M. (2005). Probabilistic road map sampling strategies for multi-robot motion planning. *Journal of Robotics and Autonomous Systems*, *53*, 244–264.

Cui, R., Li, Y., & Yan, W. (2016). Mutual information-based multi-AUV path planning for scalar field sampling using multidimensional RRT. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *46*(7), 993–1004.

Desaraju, V. R., & How, J. P. (2012). Decentralized path planning for multi-agent teams with complex constraints. *Autonomous Robots*, *32*(4), 385–403.

Drap, P., Merad, D., Hijazi, B., Gaoua, L., Nawaf, M. M., Saccone, M., & Castro, F. (2015). Underwater photogrammetry and object modeling: A case study of xlendi wreck in malta. *Sensors*, *15*(12), 30351–30384.

Elbanhawi, M., & Simic, M. (2014). Sampling-based robot motion planning: A review. *IEEE Access*, *2*, 56–77.

Fallon, M. F., Kaess, M., Johannsson, H., & Leonard, J. J. (2011). Efficient AUV navigation fusing acoustic ranging and side-scan sonar. *2011 IEEE International Conference on Robotics and Automation (ICRA)* . IEEE Computer Society.

Galceran, E., Campos, R., Palomeras, N., Ribas, D., Carreras, M., & Ridao, P. (2015). 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*, *32*(7), 952–983.

Gambin, T. (2011). A Phoenician shipwreck off Gozo, Malta. *Malta Archaeological Review*, *10*, 69–71.

Hollinger, G. A., Englot, B., Hover, F. S., Mitra, U., & Sukhatme, G. S. (2013). Active planning for underwater inspection and the benefit of adaptivity. *The International Journal of Robotics Research*, *32*(1), 3–18.

Hollinger, G. A., & Sukhatme, G. S. (2014). Sampling-based robotic information gathering algorithms. *The International Journal of Robotics Research*, *33*(9), 1271–1287.

Huang, Y., & Gupta, K. (2008). Rrt-slam for motion planning with motion and map uncertainty for robot exploration. *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Jahn, A., Alitappeh, R. J., Saldaña, D., Pimenta, L. C., Santos, A. G., & Campos, M. F. (2017). Distributed multi-robot coordination for dynamic perimeter surveillance in uncertain environments. *2017 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 273–278). IEEE.

Johnson-Roberson, M., Bryson, M., Friedman, A., Pizarro, O., Troni, G., Ozog, P., & Henderson, J. C. (2016). High-resolution underwater robotic vision-based mapping and 3d reconstruction for archaeology. Journal of Field Robotics.

Jung, Y.-S., Lee, K.-W., Lee, S.-Y., Choi, M. H., & Lee, B.-H. (2009). An efficient underwater coverage method for multi-auv with sea current disturbances. *International Journal of Control, Automation and Systems*, *7*(4), 615–629.

Kalakrishnan, M., Chitta, S., Theodorou, E., Pastor, P., & Schaal, S. (2011). Stomp: Stochastic trajectory optimization for motion planning. *2011 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 4569-4574). IEEE.

Kindel, R., Hsu, D., Latombe, J.-C., & Rock, S. (2000). Kinodynamic motion planning amidst moving obstacles. 2000 IEEE International Conference on Robotics and Automation (ICRA) (pp. 537–543). IEEE.

Kuffner, J. J., & LaValle, S. M. (2000). Rrt-connect: An efficient approach to single-query path planning. Proceedings of IEEE International Conference on Robotics and Automation, 2000 (ICRA'00) (Vol. 2, pp. 995–1001). IEEE.

Kuwata, Y., Teo, J., Karaman, S., Fiore, G., Frazzoli, E., & How, J. (2008). Motion planning in complex environments using closed-loop prediction. *AIAA Guidance, Navigation and Control Conference and Exhibit*, 7166.

Lavalle, S. M., & Kuffner, J. J., Jr. (2000). Rapidly-exploring random trees: Progress and prospects. *Algorithmic and Computational Robotics: New Directions*, 293–308.

Lindemann, S. R., & LaValle, S. M. (2003). Incremental low-discrepancy lattice methods for motion planning. *2003 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 2920–2927). IEEE.

McCarthy, J., & Benjamin, J. (2014). Multi-image photogrammetry for underwater archaeological site recording: An accessible, diver-based approach. *Journal of Maritime Archaeology*, *9*(1), 95–114.

Olague, G., & Mohr, R. (2002). Optimal camera placement for accurate reconstruction. *Pattern Recognition*, *35*(4), 927–944.

Ozog, P., Troni, G., Kaess, M., Eustice, R., & Johnson-Roberson, M. (2015). Building 3d mosaics from an autonomous underwater vehicle and 2d imaging sonar. *IEEE International Conference on Robotics and Automation* (pp. 1–8).

Palomeras, N., Hurtós, N., Carreras, M., & Ridao, P. (2018). Autonomous mapping of underwater 3-d structures: From view planning to execution. *IEEE Robotics and Automation Letters*, *3*(3), 1965–1971.

Paull, L., Saeedi, S., Seto, M., & Li, H. (2014). Auv navigation and localization: A review. *IEEE Journal of Oceanic Engineering*, *39*(1), 131–149.

Ruiz, I. T., DeRaucourt, S., Petillot, Y., & Lane, D. M. (2004). Concurrent mapping and localization using sidescan sonar. *IEEE Journal of Oceanic Engineering*, *29*(2), 442–456.

Simmons, R., Apfelbaum, D., Burgard, W., Fox, D., Moors, M., Thrun, S., & Younes, H. (2000). Coordination for multi-robot exploration and mapping. *2000 AAAI Conference on Artificial Intelligence*.

Skinner, K., & Johnson-Roberson, M. (2015). Detection and segmentation of underwater archaeological sites surveyed with stereo-vision platforms. *MTS/IEEE OCEANS*.

Stephan, J., Fink, J., Kumar, V., & Ribeiro, A. (2017). Concurrent control of mobility and communication in multirobot systems. *IEEE Transactions on Robotics*, *33*(5), 1248–1254.

Van Damme, T. (2015). Computer vision photogrammetry for underwater archaeological site recording in a low-visibility environment. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*.

Visser, A., & Slamet, B. (2008). Balancing the information gain against the movement cost for multi-robot frontier exploration. *Springer Tracts in Advanced Robotics*, *44*, 43–52.

Viswanathan, V. K., Lobo, Z., Lupanow, J., von Fock, S. S., Wood, Z., Gambin, T., & Clark, C. (2017). AUV motion-planning for photogrammetric reconstruction of marine archaeological sites. *2017 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 5096–5103). IEEE.

von Fock, S. M. T. S., Bilich, S., Davis, K., Viswanthan, V. K., Lobo, Z., Lupanow, J., & Wood, Z. (2017). Pipeline for reconstruction and visualization of underwater archaeology sites using photogrammetry. *Proceedings of the 2017 ISCA International Conference on Computers and Their Applications*.

Yamafune, K., Torres, R., & Castro, F. (2017). Multi-image photogrammetry to record and reconstruct underwater shipwreck sites. *Journal of Archaeological Method and Theory*, *24*, 703–725.

Yang, K., Gan, S., & Sukkarieh, S. (2013). A gaussian process-based rrt planner for the exploration of an unknown and cluttered environment with a uav. *Advanced Robotics*, *27*, 431–443.

How to cite this article: Wu J, Bingham RC, Ting S, et al. Multi-AUV motion planning for archeological site mapping and photogrammetric reconstruction. *J Field Robotics*. 2019;36:1250–1269. https://doi.org/10.1002/rob.21905