Robot Area Coverage Path Planning in Aquatic Environments

by

Nare Karapetyan

Bachelor of Science
Yerevan State University, 2012

Master of Science
American University of Armenia, 2015

Submitted in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy in

Computer Science and Engineering

College of Engineering and Computing

University of South Carolina

2021

Accepted by:

Ioannis Rekleitis, Major Professor

Jason O'Kane, Committee Member

Marco Valtorta, Committee Member

Pooyan Jamshidi, Committee Member

Nikolaos Vitzilaios, Committee Member

Tracey L. Weldon, Vice Provost and Dean of Graduate School

Dedication

To my family - my driving force and my source of comfort.

To the loving memories of my grandparents, Lida and Sasha,

and my uncle, Gagik.

Acknowledgments

This thesis would not have been possible were I not surrounded by so many wonderful and supportive people. I am deeply grateful to my supervisor, Ioannis Rekleitis, for his constant guidance, for not giving up even when I was so close to calling it quits, for filling sleepless nights with conversations ranging from science to politics, and for his contagious love of robotics and diving. Thank you for your mentorship, friendship, and understanding. Thank you for pushing me to grow academically, professionally, and personally, and for exposing me to so many opportunities – field trials, conferences, and workshops.

I would like to express my gratitude to Jason M. O'Kane for his invaluable suggestions and constructive criticism that always pushed me to grow. Thank you for the critical conversations that became pivotal in my career.

I want to thank my committee members: Marco Valtorta, Pooyan Jamshidi, and Nikolaos Vitzilaios. Your valuable feedback and comments were vital to the improvement of my work.

Thanks to the current and former members of the Autonomous Field Robotics Lab (AFRL). In particular, I would like to thank Alberto Quattrini Li, Marios Xanthidis, Sharmin Rahman, and Md Modasshir for the warm welcome to AFRL and the support throughout these years. Special thanks to Jason Moulton for his contributions and help with deployments of autonomous surface vehicles. And many thanks to all of my co-authors for their major contributions to this work, in particular: Jeremy S. Lewis, Emberlynn McKinney, Adam Braude, and Alex Johnson.

Thanks to the Department of Computer Science and Engineering and International Student Services for being so helpful with the dazzling amount of paperwork throughout the past five years.

My deepest appreciation for everyone that has read and will read this dissertation. Thanks for putting all those missing articles and commas (oxford or not) back where they belong.

I want to thank my master's thesis supervisor, Perouz Taslakian, who enhanced my love of research, and without whom I would not be where I am today. Thank you for sifting through the mountain of research on your desk and handing me a paper that eventually became the cornerstone of my research.

I am extremely grateful for the local coffee shops where I would get my boost of productivity – a substantial portion of this dissertation as well as many lines of code have been produced in many that I visited. Thanks to all the magical places that I got to travel, from the mountains I hiked up to the oceans I dived in, while working on this dissertation. Thanks to Columbia, a city that eventually became a lovely and colorful hideaway, for hosting an important part of my life.

A special thanks goes to my lovely friends who made me feel at home away from home: to those who filled some miserable days with silly laughs, to those who were just there when needed, even from across continents. And a very special thanks must go to Maria Sevoyan and Shant Ananyan, who opened not only their door, but their hearts, to a stranger only days after I arrived in the United States for the first time.

Finally, I want to thank my family, to whom this thesis is dedicated. Thank you for your ever-present love, and at times, your unreasonable faith. To my mother and grandmother, Anahit and Lida – the strongest and most supportive humans I know. Thank you for sharing your enthusiasm for education, math, and science with me from an early age.

Abstract

This thesis is motivated by real world problems faced in aquatic environments. It addresses the problem of area coverage path planning with robots - the problem of moving an end-effector of a robot over all available space while avoiding existing obstacles. The problem is considered first in a 2D space with a single robot for specific environmental monitoring operations, and then with multi-robot systems — a known NP-complete problem. Next we tackle the coverage problem in 3D space - a step towards underwater mapping of shipwrecks or monitoring of coral reefs.

The first part of this thesis leverages human expertise in river exploration and data collection strategies to automate and optimize environmental monitoring and surveying operations using autonomous surface vehicles (ASVs). In particular, four deterministic algorithms for both partial and complete coverage of a river segment are proposed, providing varying path length, coverage density, and turning patterns. These strategies result in increases in accuracy and efficiency compared to manual approaches taken by scientists. The proposed methods were extensively tested in simulation using maps of real rivers of different shapes and sizes. In addition, to verify their performance in real world operations, the ASVs were deployed successfully on several parts of the Congaree River in South Carolina, USA, resulting in a total of more than 35km of coverage trajectories in the field.

In large scale coverage operations, such as marine exploration or aerial monitoring, single robot approaches are not ideal. Not only the coverage might take too long, but the robot might run out of battery charge before completing the task. In such scenarios, multi-robot approaches are preferable. Furthermore, several real world vehicles

are non-holonomic, but can be modeled using Dubins vehicle kinematics. The second part of this thesis focuses on environmental monitoring of aquatic domains using a team of Autonomous Surface Vehicles (ASVs) that have Dubins vehicle constraints. It is worth noting that both multi-robot coverage and Dubins vehicle coverage are NP-complete problems. As such, we present two heuristic methods based on a variant of the traveling salesman problem—k-TSP—formulation and clustering algorithms that efficiently solve the problem. The proposed methods are tested both in simulations and with a team of ASVs operating on a 40 000m^2 lake area to assess their ability to scale to the real world.

Finally, in the third part, a step towards solving the coverage path planning problem in a 3D environment for surveying underwater structures, employing vision-only navigation strategies, is presented. Given the challenging conditions of the underwater domain, it is very complicated to obtain accurate state estimates reliably. Consequently, it is a great challenge to extend known path planning or coverage techniques developed for aerial or ground robot controls. In this work we are investigating a navigation strategy utilizing only vision to assist in covering a complex underwater structure. We propose to use a navigation strategy akin to what a human diver will execute when circumnavigating around a region of interest, in particular when collecting data from a shipwreck. The focus of this work is a step towards enabling the autonomous operation of lightweight agile robots near underwater wrecks in order to collect data for creating photo-realistic maps and volumetric 3D models while at the same time avoiding collisions. The proposed method uses convolutional neural networks (CNNs) to learn the control commands based on the visual input. We have demonstrated the feasibility of using a system based only on vision to learn specific strategies of navigation, with 80% accuracy on the prediction of control command changes. Experimental results and a detailed overview of the proposed method are discussed.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Automation and robotic systems are becoming a crucial part of our day-to-day activities. We use them in manufacturing, hospitals, stores, at home for cleaning floors or windows, and deploy them in remote areas of the Earth and outer space. Despite the anecdotal belief that robots might eliminate human labor and jobs, automation and intelligent robotic systems help us to reach areas that are dangerous or unreachable for humans. We are able to explore the ocean, the surface of the Moon and Earth more efficiently than ever before. Nevertheless, we are far from having low maintenance and systems with high intelligence, and still rely on human operators in many dangerous or mundane tasks. For the last fifty years a large body of research has been done in robotics, automation, and artificial intelligence to address those gaps and advance the field.

1.1 Motivation

The area coverage path planning is the problem of finding a path such that, if followed by the robot, the robot's end-effector or sensor will be moved over the region of interest while avoiding existing obstacles. The area coverage problem is common in a wide range of domains and in various applications: cleaning the floor in indoor environments [17, 62, 93], window cleaning robots [12, 76, 128], in underwater environments for archaeological and environmental applications [27, 91, 115], in aerial surveillance or reconstruction of 3D objects [104, 108, 138], with surface vehicles for environmental monitoring and inspection operations [53, 54, 92].

Ocean covers 71% of Earth's surface, about which we know less than we do about the moon. Only 10% of this area has been explored. Until now, the scientific community has relied heavily on manual surveying operations to conduct their studies (see Figure 1.1). Deploying autonomous systems during those expeditions on the surface and underwater for exploring and moving the frontier of the unknown is extremely important from environmental, biological, and historical perspectives. In this domain, the problem is how to effectively navigate and collect data with a robot that moves on the surface of the rivers, lakes or underwater in the ocean or sea.

(a) (b)

Figure 1.1: (a) Scientists manually collecting bathymetric data over Congaree River, Columbia, SC. (b) A diver manually photographing a shipwreck for photogrammetric mapping.

To address the environmental monitoring problem, autonomous surface vehicles (ASVs) have been deployed on rivers and lakes, while different designs have been proposed for specific applications [8, 61, 78]. These ASVs are fuel powered, with basic autonomy; they can be deployed with a range of sensors, and can perform long term operations with up to nine hours of operational time.

In the underwater environment, different types of platforms are being used for exploration and monitoring operations [4]. Among many challenges in this environment, the operational time and computational power are key aspects. As such, remotely operated vehicles could be operated by humans through a tether [88, 110, 123], thus providing long operational time. Tether-less systems tend to be more expensive and

have limited battery life, but can be deployed in remote areas [22, 105]. Gliders are a type of autonomous systems designed to operate over large distances for a very long period of time (for example, crossing the Atlantic in over 200 days) [9, 36, 133]. Nevertheless, the motion of the glider is limited to the linear direction and thus does not provide the flexibility and agility demonstrated by six degrees of freedom robotic systems.

In this work we address the area coverage problem for aquatic domains for both surface and underwater environments. We use single and multiple ASVs for performing surface coverage of rivers and lakes. With the above discussed considerations regarding different autonomous platforms, for 3D dimensional coverage our target platform is Aqua2 robot, which has six degrees of freedom and can navigate around complex structures.

1.2 Objective of the Thesis

The objective can be stated as follows:

To solve the coverage path planning problem in aquatic environments, generating different coverage strategies for different aquatic domains to be deployed on autonomous surface or underwater vehicles, that will efficiently perform uniform sampling of the environment, taking into account obstacles, implicit geological properties of the environment and also perform effective data collection suitable to the deployed sensors.

1.3 Contributions of The Work

The following work presents different algorithms for solving area coverage problems in aquatic environments. Nevertheless, they can be generalized for ground and aerial environments as well. The contributions of this thesis can be summarized as follows:

- Riverine Coverage: four different coverage path planning patterns are designed, aiming to optimize coverage efficiency for survey operations performed on the surface of rivers [54, 58, 59].

- Multi-robot Area Coverage: two approximation algorithms are introduced for solving the area coverage problem with non-holonomic vehicles, in particular surface vehicles with Dubins constraints [53, 57, 78].

- Coverage in 3D: a coverage strategy that performs visual navigation in 3D is presented. The main contribution of this approach is the ability to perform a set of predefined navigation steps without requiring knowledge about the environment and without relying on an accurate state estimation [55, 56, 101].

1.4 Overview

The rest of this dissertation is organized in the following way. In the Chapter 2 we review the literature on area coverage path planning problems in different domains. We present first state-of-the-art approaches for the area coverage problem in two dimensional spaces with a single robot. Then we present the solutions to the same problem with multi-robot systems. Next we review the navigation problem in three dimensional space and conclude with an overview of works for solving environmental sampling and monitoring problems.

Chapter 3 discusses the problem of area coverage in a riverine environment. It presents strategies for automating methods used by the scientists for collecting data over rivers. In addition, more efficient approaches are proposed for improving efficiency of automated versions of the manual surveying operations. The proposed methods are suitable for different survey operations with different sensor suites. Parallel to the shore patterns are used for complete coverage with side scan sonar to minimize the number of turns for optimal data collection. Another complete cov-

erage algorithm that performs boustrophedon motion across the river can be used with single ping sonar. To sample the river without complete coverage, a zig and zag approach is proposed. All patterns are extensively tested in a segment of Congaree River resulting in more than $35km$ of covered trajectories.

Chapter 4 presents two approximation algorithms for solving multi-robot area coverage problems with Dubin vehicles. The first approach is finding an optimal path and then clustering it between multiple robots. The second approach is first splitting the area and then solving the generalized multiple-salesman travelling problem to find an optimal path for each robot. These methods ensure complete utilization of all robots. Experiments were performed with autonomous surface vehicles covering an area of up to 40 $000m^2$ over Lake Murray in SC, USA.

In Chapter 5, we propose our approach for three dimensional area coverage using only a camera for performing navigation in an unknown environment. We use human expertise for learning to navigate in a complex environment and collect data for 3D reconstruction. We train a network to learn predefined navigation strategies using real world data collected over shipwrecks and from the Gazebo simulation environment. The neural network achieved approximately 80% accuracy for predicting the direction in which the robot has to move. We also present navigation results acquired from simulation performed in the Gazebo environment.

The last Chapter 6 gives an overview of the presented work and summarizes the contributions of this thesis. It discusses possible research areas that can be explored to extend the presented work with a report of some preliminary results.

Finally, the Appendix describes related work performed by the author that is relevant to this dissertation. Namely, in Appendix A, the main experimental platforms are described with details only relevant to this dissertation. And Appendix B gives an overview of a framework for controlling ASVs affected by water current and wind speeds.

Chapter 2

Related Work

2.1 2D Coverage

There are numerous ways to formulate area coverage, including *static* or *dynamic* coverage, *complete* or *partial*, *offline* or *online* [14, 34]. In addition, there are many different approaches to tackle such a problem, such as defining it as a graph partitioning problem, performing region-based decomposition, or defining it as a submodular optimization problem [34, 114].

When prior information about the environment is available in the form of a map, the coverage is called offline [14]. One of the approaches widely used in offline coverage algorithms is based on area decomposition. Choset [15] proposed a cellular decomposition technique, called *boustrophedon decomposition* (BCD). In his work, the area of interest is decomposed into obstacle-free cells. A lawnmower pattern or boustrophedon motion is typically executed to cover each cell (see Figure 2.1). Several works use this type of motion as the method of choice in solving the coverage problems as a main strategy [7] or as a prior strategy to improve coverage quality [60]. Other approaches were also used for decomposing areas, such as Morse decomposition [1] or grid-based decomposition [31, 62].

Polynomial time algorithms were proposed for solving single robot coverage using a boustrophedon decomposition based approach [75, 137]. In contrast to the original algorithm, in these approaches, the problem is represented as the *Chinese postman problem (CPP)*, which ensures efficient coverage order of cells. The latter is a graph

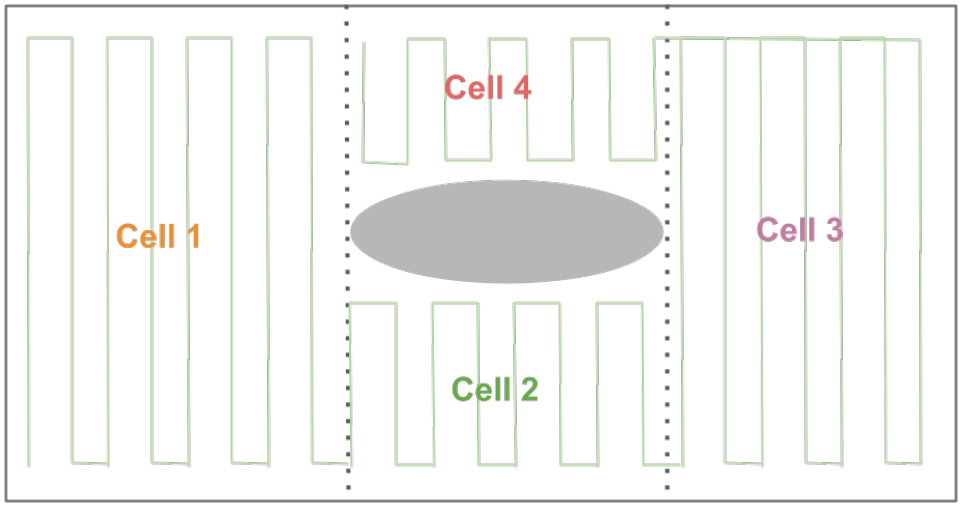


Figure 2.1: Boustrophedon cellular decomposition example with a lawn-mower pattern covering each cell.

routing problem, of finding a minimum-cost closed tour that visits each edge at least once. The solution to CPP had polynomial time complexity and was proposed by Edmonds and Johnson [24]. Sadat et al. [108, 109] address selective coverage using a tree, based on a Hilbert space-filling curve.

When considering the coverage problem for robots with turning constraints, the presented methods that use boustrophedon coverage pattern may not be the most efficient — that is, they will either spend an excess time on covering areas out of the region of interest or fail to perform complete coverage on turns because of the rotation constraints. The Dubins vehicle is a common robot model in coverage problems, and Savla, Bullo, and Frazzoli [112] consider a control-theoretic solution.

Reducing traversal time by considering motion constraints is not a new idea in coverage. Both Huang [46] and Yao [139] minimize the path length by using motion constraints in their environmental decomposition. Both of them seek to reduce the amount of rotation required by the robot.

In contrast, Lewis et al. [67] proposed to optimize the solution by carefully selecting how the robot transitions from covering to not-covering. They use boustrophedon

decomposition to split it into cells, and then each cell is decomposed further into passes, and based on those passes a routing problem is defined (see Figure 2.2). As such, the traveling salesman problem (TSP) with Dubins curve constraints — called Dubins traveling salesman problem — is utilized in their work. In [85, 113], the Dubins TSP is defined as a metric TSP with an additional constraint, such as the paths between nodes must adhere to a minimum turn radius necessary for the covering the vehicle's transition between nodes.

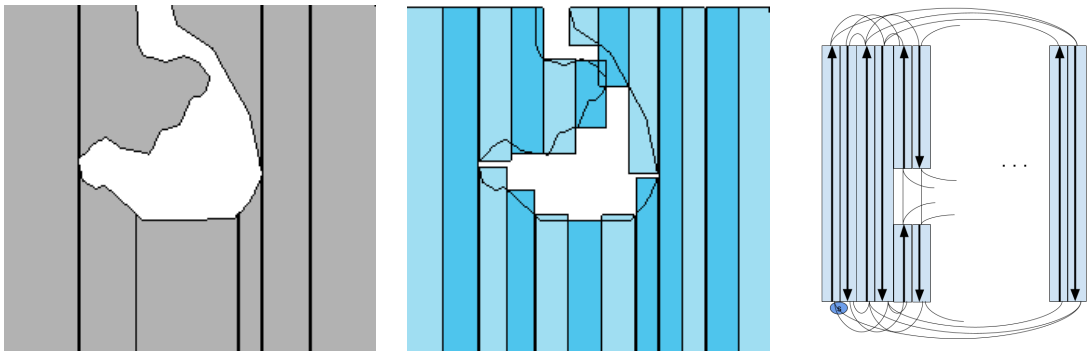


Figure 2.2: (a) Boustrophedon Decomposition; (b) Splitting each cell into passes with a width of a sensor footprint; (c) Creating a graph by connecting each pass.

2.2 Multi-robot Coverage

Some of the presented methods based on cellular decomposition were also designed for multi-robot systems [103], assuming restricted communication. Avellar et al. [6] present a multi-robot coverage approach that operates in two phases: decomposing the area into line-sweeping rows, based on which a complete graph is constructed to be used in the second phase, where the vehicle routing problem [125] is solved. Field trials with Unmanned Aerial Vehicles (UAV) showed that their proposed approach provides minimum-time coverage. However, that algorithm is only applicable for obstacle free environments.

Grid-based methods for single robot coverage were also adapted for multi-robot systems as well [3, 28, 39, 104]. The robustness and efficiency of the systems proposed by these works depend on the resolution of the input representation. Because the size of each cell is typically based on the size of the sensor footprint, the coverage becomes more challenging in environments with many obstacles, since the footprint size increases.

A large body of work in multi-robot systems assumes that there is some form of communication between the robots [136]. Some of them came up with alternative implicit communication means, such as trail of other robots [48, 49, 132]. Nevertheless, this type of communication is impractical in aquatic or aerial environments.

Graph routing problems such as TSP and CPP have also been defined for multiple routes: finding k routes that visit non overlapping vertices of the graph, such that the union of those clusters are the exact set of vertices in the TSP case. This problem is called k-TSP problem or m-TSP. When edges are considered instead of vertices, the problem is called k-CPP. Both these problems and their variations were shown to be NP-complete [30].

In our previous work [52], we presented a communication-less multi-robot coverage algorithm based on efficient single robot coverage. In that work, we proposed two approximation methods that utilize boustrophedon decomposition and the Chinese postman problem. Nevertheless only one of the proposed methods have been executed and compared with an original k-CPP solver. Moreover, the utilization of robots is dependent on the number of obstacles in the area. Since clustering is based on boustrophedon cells, a small number of obstacles will result in a small number of cells, and consequently less clusters per region. Note however, that the solution generated did not take into account any kinematic constraints of the robots.

Turn minimization is especially important when considering multiple robots. Vandermeulen et al. [128] tackle this problem by partitioning the area into long thin

rectangles as wide as the robot's sensor. Then those areas are used by a m-TSP problem solver to minimize the coverage time. The m-TSP algorithms can be also used to perform task allocation [127]. Özdemir et al. [89] consider a problem for an unknown environment with swarms of simple robots. It does not require run-time computation or memory for storage, and only relies on one bit information about the presence of another robot in the line of sight. Nevertheless this approach may not guarantee complete coverage.

2.3 3D Coverage

So far we have discussed the area coverage problem when the area of interest is assumed to be constrained by a 2D environment. However, some applications require acquiring information about 3D structures, thus anticipating a coverage path planning approach that works in higher dimensions. The field of applications of these problems is as wide as for 2D coverage, with the use of aerial, underwater robots or manipulators.

The 3D coverage problem is quite challenging because there are exponentially many solutions to choose from. Some approaches have been proposed and shown to have feasible solutions both with single and multiple robots using octomaps [19, 20]. In addition, some coverage solutions in three dimensions have been presented for ensuring the even distribution of spray paint in the automotive industry [5]. Another work has addressed the coverage of an unknown environment using a frontier-based approach [18].

Peng et al. [13] addressed the problem by representing an area through well defined 2.5D features thus reducing the complexity of structure-dependency of the 3D coverage. In that work, the coverage problem is specifically designed for an aerial vehicle with a conical field of view that can rotate around a fixed point with three degrees of freedom, thus limiting its application to other domains.

Similar to Palomeras et al. [90], Bircher et al. [10] combine the problem of covering an unknown environment with a given structure by sampling random next-best views around the target structure. It is an extension of their previous work [11] that builds a tree of next-best views and selects the best branch using the size of unexplored area covered as the metric. Like most next-best view based approaches, these works are also subject to the sensor data quality, system localization and vehicle dynamics.

The coverage problem has also been studied for underwater environments, which has valuable environmental and archaeological importance. A number of works presented seabed and underwater coverage path planning methods with Autonomous Surface Vehicles and Autonomous Underwater Vehicles [32, 44]. Behavior-based control of an underwater vehicle for coral-reef inspection was proposed by [106]. The behavior selection is implemented using both fuzzy logic and utility fusion. The behaviors ensure collision avoidance, proper distance from the reef and rope following or target following actions. However, no similar approach has been developed for surveying more complicated environments.

To ensure complete coverage without overlaps, Galceran et al. [35] suggest segmenting the environment based on similar depths. Each of these segments then is considered as an individual planning problem. The proposed algorithm extends cellular decomposition [15], performing 2.5D coverage by traveling at a constant depth from the surface. This work has been extended to also take into account the state estimation uncertainty and perform replanning as needed [33]. The above works rely on knowing the approximate depth to split the area, and it only views the environment from above.

When the environment is unknown, Vidal et al. [130] propose a next-best view approach, but it is constrained by the certainty of the state estimation. To overcome the complexity of 3D exploration, a simplified formulation of the problem is considered, such as 2D mapping of an underwater structure [131]. In these works, the authors

use a view planner and frontier-based strategies. The environment is represented as a quadtree occupancy map which is also used to generate viewpoints for exploration. It is worth noting that quadtree representations become more computationally expensive when representing large-scale complex structures. These methods also require accurate localization.

3D coverage has also been shown to have a wide range of applications for mapping historical artefacts and structures in underwater environments [38]. Most of the presented works either assume reliable localization or some type of prior information about the environment. Nevertheless, when operating in an underwater unknown environment, reliable state estimation is very challenging. In contrast, a wide range of vision based navigation methods have been used in literature for aerial robots and ground vehicles, either for goal oriented path planning or coverage. For example, Smolyanskiy et al. [121] collect real and simulation data to train a drone to navigate over a trail. When operating in 2D Lewis et al. [68] show that it is even possible to guarantee coverage with simple blind robots — with no state estimation.

Motivated by similar works, Manderson et al. [72] have built a 3D navigation deep learning framework for underwater systems by proposing a vision based navigation in an unknown environment for coral reef exploration. This work became the motivation of our proposed solution as well, for 3D coverage of shipwrecks. More recent work by the same group [71] incorporates path planning in conjunction with obstacle avoidance and a bias towards areas with corals. Their approach is trained for navigating over the coral reef, rather than for navigating around an underwater structure.

More generally, in computer vision literature, methods grouped under the terms shape-from-motion and next-best-view have been utilized for the visual mapping of an object or a structure. By definition, shape-from-motion techniques always estimate the location of the camera together with the shape of the mapped structure, thus requiring accurate state estimation of the mapping system — a difficult task as shown

in [51]. Furthermore, all next-best-view techniques require the knowledge of where the camera is and an ability to navigate to the next viewpoint.

2.4 Aquatic Exploration and Coverage

Substantial work has been done on the design and operation of autonomous surface vehicles (ASVs) in rivers. One team of researchers has shown that it is both possible and desirable to design and operate autonomous surface vehicles for the purpose of performing bathymetric surveys [29]. Significant progress has also been made on the problem of navigating a river with an ASV [122]. Additionally, another team has determined a technique for exploring and mapping a river using an unmanned aerial vehicle [47]. Though, to the best of our knowledge, there is no existing research on automated vehicles zig-zagging their way through rivers, the basic principle has been applied using an underwater autonomous vehicle [142]. The vehicle used repeated 135 degree turns to map an upwelling front underwater, covering 200 square kilometers over the course of five days without human intervention. Estimating the meanders of a river has also been studied by Qin and Shell [98], and the proposed estimator can be used for online path selection.

Of particular relevance to environmental monitoring are two works dealing with the coverage of rivers using drifters: vehicles that do not have sufficient power to travel against the current [66, 65] and another work dealing with coverage path planning for a group of energy-constrained robots [119]. One notable work breaks from the tendency to emphasize complete coverage, instead attempting to conserve time and fuel by focusing coverage on regions of interest [73]. This allowed them to create a map of a coral reef area with half the distance travelled and power used than a lawnmower-style complete coverage algorithm would have required. Another paper, in which lawnmover-style coverage is applied to a Dubins vehicle, reformulates the problem as a variant of the TSP in order to obtain an optimal solution [67].

Although the above selective coverage work phrases the problem in terms of coverage, it bears kinship with the literature for *informative motion planning*, that is, the problem of planning a path using limited resources in order to maximize the amount of information gained. Unfortunately, informative motion planning problems are usually NP-hard optimization problems. The formulation of these problems requires the definition of an information metric that can be associated with the locations or path. Since the information metric cannot be known *a priori* for a real-world scenario, approximations are done using methods such as Gaussian Processes. This means that informative motion planning can be applied to practical problems, such as mapping wireless signal strength on a lake [45], understanding salinity at a river confluence [117], or investigating algal blooms [118] and sampling areas with high chlorophyll [74]. Despite the success of these projects, qualitative considerations involved in the formulation of the riverine coverage problem mean that reformulating it as an informative motion planning problem would not necessarily produce data with the desired qualities, and it would be difficult to devise an information metric that obtains the desired result.

Some work has been done to find a connection between river meanders and the speed of river current [82, 98] or to model the external forces [80]. The first to understand how flow affects the length of meanders and its down-flow migration was Einstein [25]. In another work, Qin and Shell [98] use the well studied model of the geometry of meanders to estimate the shape of the unseen portion of the river. Using this information as an input, a boat can adjust the speed and perform more optimal and smooth paths when performing online navigation. To the best of our knowledge, this has been first addressed by our work [59].

2.5 Vision-Based Navigation

When dealing with the real world to solve hard-to-engineer behaviors, machine learning approaches have been proven to be beneficial, particularly reinforcement learning (RL). In reinforcement learning, an agent learns the best approach to achieve the goal by interacting with the environment and receiving feedback on specific actions [63, 124].

Reinforcement learning algorithms guarantee complete exploration when any state is assumed to be reachable (ergodicity assumption). Moldovan and Abeel [77] formulated the problem of exploration through Markov Decision Processes when there is no such assumption. But in real world applications, this assumption is almost never true, e.g, safe coverage of marine environments ensuring that agents are not visiting shallow areas, navigation through steep areas and many others. In the above-mentioned work, the intended field application was Martian terrain exploration. The results showed that with safe exploration the robot is able to explore better in terms of safety and information gain when compared with the classical methods.

In the work by Zhu et al. [143], a deep reinforcement learning approach is used for solving a problem of navigation in space with an aim to find a given goal. The only data used to achieve this is images captured by the robot's cameras. More formally, given an RGB target image, the problem is to find an optimal sequence of actions to navigate towards it: the output is an action (go forward, turn left) in 3d. In this work, the authors try to address the main issues in deep RL, such as the inability to generalize to new target goals and the amount of data required for error to converge. To ensure a better generalization, a policy is defined as a function of the goal and the current state. As for generating more data, they proposed The House of interaction (AI2-THOR) framework, which allows the system to collect a huge number of training examples from agents' interaction with objects and their navigation in the environment.

Map-less target-driven navigation was studied by Pfeiffer et al. [94]. An end-to-end approach is used, when current sensor readings are translated to the action commands for the robot. To solve this problem, imitation learning (IL) is used for initially training the navigation policy and then RL is applied. The aim in their work is to combine the best practices of both approaches by covering disadvantages of each other. Namely IL is more simple and easy to train, but at the same time it tends to overfit to the training environment, whereas RL is able to generalize and encode information about desired actions (avoid obstacles, reach target). This hybrid approach should potentially speed up the training speed of RL and produce a more robust policy than with simple IL or RL.

Chapter 3

Riverine Coverage

3.1 Introduction

Bathymetric surveys, that is, surveys of the depth of a body of water, are an important tool for understanding hydro-geologic processes. Since the sensor footprint of a bathymetric sensor is significantly smaller than the width of many rivers, a complete bathymetric survey of a river requires multiple boats or passes. The usual method for performing coverage of a known two-dimensional area, boustrophedon coverage [16, 1, 2], performs poorly in tight and uneven spaces such as rivers. Fortunately, river surveyors have developed and practiced a variety of coverage techniques that are suitable for rivers. The different surveying/coverage strategies were developed based on the property needing measurement and the available resources. For example, in a fast flowing river, studying sediment transfer requires sampling across the river. Otherwise, by the time the boat returns to the same spot, the sediment will have moved significantly.[1]

In this chapter, we address the question of how to conduct such surveys using autonomous robots. Three methods, each suited to a specific scenario, are presented:

1. To produce a high precision depth map, data is required from every point on the riverbed. To accomplish this with our first method, the survey boat performs longitudinal passes, traveling roughly parallel to the river shores. This approach

[1] As Heraclitus said "Everything changes and nothing remains still ... and ... you cannot step twice into the same river." [42].

is particularly apt when the survey is being conducted using sensors, such as side-scan sonar, that are sensitive to turning motions. The method we propose, called L-cover, adapts the number of passes based on the width of each segment of the river.

2. When complete area coverage is deemed unnecessary or overly expensive, a single pass along the river is utilized. In this method, which we call Z-cover, surveyors travel down the river in a zigzag pattern, turning away from the shore each time they reach it. This allows bathymetric data reflecting the full width of the river to be collected in a single pass.

3. The third method is an alternative complete coverage approach which works by guiding the boat across the river, performing a lawn-mowing pattern in a transverse direction, in a manner quite similar to traditional boustrophedon coverage [2]. This strategy, which we term T-Cover, may be advantageous, for example, for tracking sediment transfer.

In addition, we show how the longitudinal coverage can be optimized using intrinsic river speed information provided by the river meanders. The presented Meander based coverage, so-called M-cover, algorithm relies on the information that on the outer bank of the river the speed is different from the inner banks and provides a more efficient coverage strategy.

3.2 LZT Coverage Patterns

Our objective for riverine coverage is to automate different approaches used by river surveyors and develop more efficient planning for each of them. We deploy an ASV with a variety of depth sensors to survey the riverbed. The ASV moves within a known environment, described by an occupancy grid map $M : \mathbb{R}^2 \rightarrow \{0, 1\}$, derived from Google satellite imagery. Values of 0 indicate the portion of the river we intend

to cover, while 1 indicates locations outside that region of interest, which we treat as obstacles. We assume that on a given map M, starting point v_s is located at the very border of the map, and thus we can implicitly infer the general direction of coverage. Otherwise the direction of the coverage should be explicitly defined as an input.

In this section, we describe algorithms executing three types of coverage patterns in such contexts:

- Section 3.2.1 presents a pattern, termed L-Cover, which moves in passes parallel to the shore. This pattern is particularly suitable for use with a side-scanning sonar.
- Section 3.2.2 describes a pattern, termed Z-Cover, which 'bounces' between the river shores. This approach is used for performing river surveying using a single pass, which is suitable for long term deployments.
- Section 3.2.3 proposes a T-Cover pattern, where passes are made across the river, perpendicular to the shore.

These three methods differ in the length of the paths they generate, in the density of the coverage pattern, and in the number of turns needed to execute those patterns.

3.2.1 Longitudinal Coverage (L-cover)

Our first approach, L-cover, performs coverage in a boustrophedon pattern, with passes parallel to the edges of the river. The goal of the algorithm is to split the river into subregions that can be covered with the same number of passes. The algorithm takes as input the map of the river M, the starting point v_s and a parameter s describing the desired spacing between the passes; see Algorithm 1. First, we identify the coverage direction (the red line in Figure 3.2a) and compute an ordered list, denoted C_{vec}, of contour points of the shore (Line 2-3). Next, the algorithm sequentially traverses the contour C_{vec} with a step size Δw connecting opposite edges

with straight line segments, denoted l. Step size Δw is the distance between each pair of segments l_i and l_{i+1}. Then, the river is split into clusters of subregions based on the width of the river, denoted by $len()$, and the desired spacing s (Lines 4–11). Any small clusters are merged with the nearest neighbor cluster that has similar width (Lines 13–15). Finally, parallel passes are generated for each resulting cluster Cl (Line 17). The resulting path π is a list of all sequential passes from each cluster (Line 19). Examples of the results of the algorithm, with different values of s, are presented in Figure 3.1.

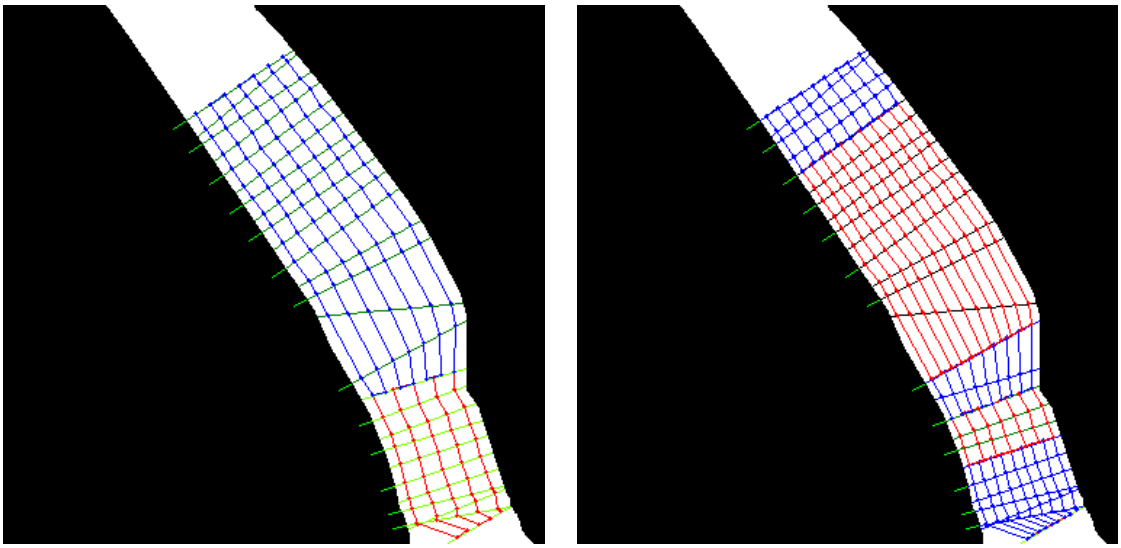
Algorithm 1 L-cover

Input: binary map of river M, starting point v_s
spacing parameter s
Output: a π path

1: $\Delta w \leftarrow$ initialize()
2: $C_{vec} \leftarrow$ getDirectionalContours(M)
3: $\theta \leftarrow$ getCoverageDirection(C_{vec}, v_s)
4: **while** the end of the river is not reached **do**
5: $l \leftarrow$ getNextSegment($\Delta w, C_{vec}, l_{prev}, \theta$)
6: **if** $len(l) - len(l_{prev}) \leq s$ **then**
7: Insert l in to Cl_{curr}
8: **else**
9: Save cluster Cl_{curr} in Cl_{vec}
10: $Cl_{curr} \leftarrow$ createNewCluster(l)
11: **end if**
12: **end while**
13: **for each** $Cl \in Cl_{vec}$ **do**
14: Merge Cl with closest neighbor within tolerance
15: **end for**
16: **for each** $Cl \in Cl_{vec}$ **do**
17: $p \leftarrow$ generatePasses(Cl, s)
18: Append p to π
19: **end for**
20: **return** π

3.2.2 Zig-Zag Coverage (Z-cover)

The Z-cover partial coverage approach is based on a zig-zag pattern which aims to cover a substantial portion of the environment in a single pass along the river.



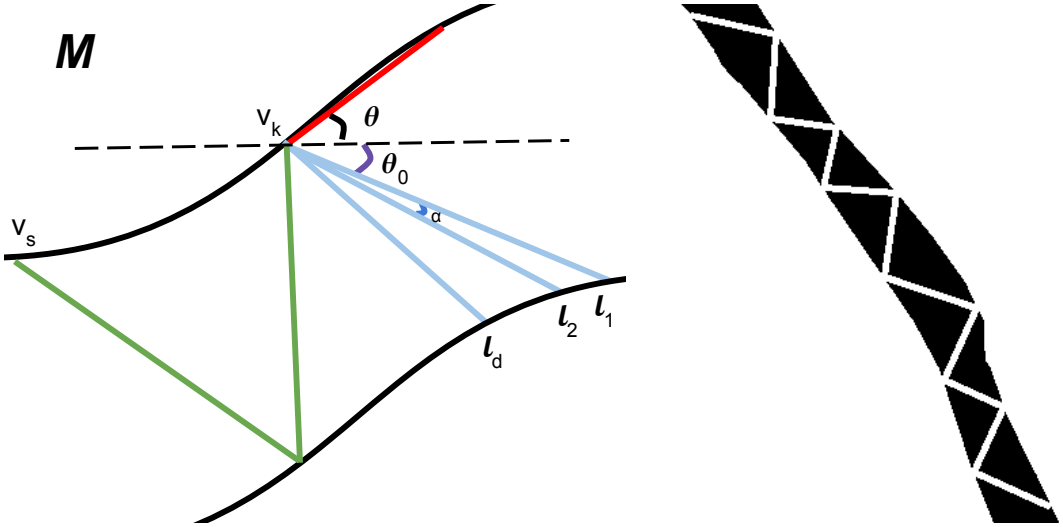
(a) (b)

Figure 3.1: An example of trajectories and clusters generated by L-cover approach on a small section of Congaree river with different coverage density values (alternating colors mark different clusters).

The core idea of the proposed algorithm is to build a coverage path that gathers information along and across the river simultaneously. By ensuring that consecutive triangles have approximately equal areas, we ensure that the ratio of the covered areas across the river area remains approximately the same.

Algorithm 2 outlines the approach. It takes as input the map M of the river and the starting point v_s. Just as in the L-cover algorithm, the C_{vec} vectors of directional contours are acquired (Line 2-3). Then, each time the algorithm searches for a next point, it does so by drawing lines from the current location towards the opposite shore. An acceptable next point is searched for among the intersections of the opposite shore with d possible lines $l_1, l_2, ...l_d$ (blue lines in Figure 3.2a) that form $\theta_0 + i\alpha$, $i = 1, 2, \ldots, d$ degree angle relative to the direction downriver. If one of these points forms a triangle with the previous two points on the path with area within tolerance of the area of the previously selected triangle (the triangle with green edges in Figure 3.2a), it will be selected. If no such point exists within d intersections, the

tolerance $\Delta\epsilon$ will be increased and the algorithm will do the same search again (Lines 13-15). The tolerance $\Delta\epsilon$ is predefined and can be tuned if necessary.



(a) (b)

Figure 3.2: (a) A sketch of the triangle selection procedure. (b) The result of the Algorithm 2 applied on a section of Congaree river.

3.2.3 Transversal Coverage (T-cover)

Finally, we consider T-cover, which performs a continuous lawn-mower motion pattern perpendicular to the shores of the river. The algorithm uses the same information as L-cover, namely the map M, the start location v_s, and the coverage spacing s. After acquiring directional contours, it generates passes, perpendicular to the shores, spaced by distances s from each other. This is similar to covering a single cell of the Boustrophedon Cellular Decomposition, albeit the direction of the coverage varies with the river's meanders. This approach is utilized when the quantities measured change rapidly over time and the transverse profile of the river bed is required.

Algorithm 2 Z-cover

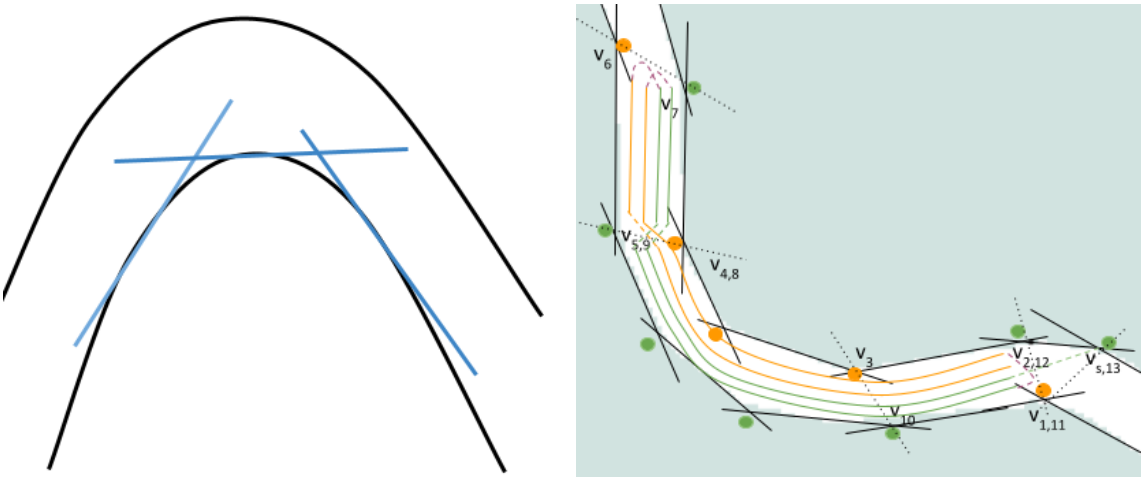
Input: binary map of river M, starting point v_s
Output: a π path

```
θ_0, d, α, Δε ← initialize()
C_vec ← getDirectionalContours(M)
θ ← getCoverageDirection(C_vec, v_s)
while the end of the river is not reached do
    for each i ∈ 1,..., d do
        v_curr ← getIntersectionPoint(l_i, C_vec, α)
        p_1, p_2 ← getPreviousTwoPoints(π)
        S_curr ← computeAreaOfTriangle(v_curr, p_1, p_2)
        if |S_curr − S_prev| ≤ Δε then
            append v_curr to π
        end if
        if i == d and π is empty then
            Δε++
            i ← 1
        end if
    end for
    v_curr ← getNextPoint(C_vec)
end while
return π
```

3.3 Meander-Based Coverage (M-cover)

To optimize the complete coverage L-cover method proposed above, we use the meanders of the river. In the meander-based coverage we rely on the fact that on the inner bend, the downriver speed of the current is slower compared to any neighbouring region closer to the outside bend of the river. As shown in Figure 3.3b, along the passes that connect green dots the water flow is faster, whereas orange ones indicate regions where the flow is slower.

To find the meanders, Algorithm 3 looks into the intersection of two consecutive tangent lines to the curve of the river contour (Figure 3.3a). If the lines intersect inside the river, an inner curve is identified (orange vertex), otherwise if the intersection is on land then an outside bend has been found (green vertex). Using this information, the M-cover algorithm depicted in Algorithm 3 finds an efficient complete coverage path.



(a) (b)

Figure 3.3: A sketch for finding the meanders: (a) The procedure of checking the intersection of a neighbor pair of tangents; (b) The order of vertices the algorithm will visit if coverage is to be performed upwards.

It takes as input the map of the river M, a starting point v_s, and the spacing width information (sensor footprint size). First, the direction of the coverage is identified implicitly from v_s and M; then, the directional contour C_{vec} is generated (Line 2). Consequently, the river is split into S_{vec} segments, utilizing the meander information, based on the above explained intuition (Line 4). Each segment is split into segments that the robot can cover in a single pass (Line 6). We decompose the area into an even number of segments in order to return back to the initial starting point v_s. Each of the passes are assigned a direction: the first pass that is closest to the inner bank of the river is getting reserved for upwards travel, whereas the ones closer to the outer side are getting reserved for downwards travel (Lines 7–14). A pass is added between each consecutive segment of meanders: from orange to the closest green on the opposite edge of the river (Lines 16–17).

The simple M-cover approach does not take into account the change in the width of the river, which can affect the number of the passes one can generate. To solve that problem we propose to adapt the clustering stage in the L-cover algorithm. With

Algorithm 3 M-cover

Input: binary map of river M, starting point v_s, spacing parameter s
Output: a π path

1: $\Delta w \leftarrow$ initialize()
2: $C_{vec} \leftarrow$ getDirectionalContours(M)
3: $\theta \leftarrow$ getCoverageDirection(C_{vec}, v_s)
4: $S_{vec} \leftarrow$ getMeanderSegments($\Delta w, C_{vec}, \theta$)
5: **for each** $S \in S_{vec}$ **do**
6: $P \leftarrow splitIntoEvenPasses(S, C_{vec}, s)$
7: $k \leftarrow |P|$
8: **for each** $p_i, p_{k/2+i} \in P$ **do**
9: **if** p_i *is on outside bend* **then**
10: $p_i \leftarrow$ *down direction*, $p_{k/2+i} \leftarrow$ *up direction*
11: **else**
12: $p_{k/2+i} \leftarrow$ *down direction*, $p_i \leftarrow$ *up direction*
13: append p_i, $p_{k/2+i}$ to π
14: **end if**
15: **end for**
16: $S_{prev} \leftarrow S$
17: $p \leftarrow createPassBetween(S_{prev}, S)$
18: append p to π
19: **end for**
20: **return** π

this modification the algorithm will perform coverage in segments that have the same width (see Algorithm 4). In the same way, the width based approach will take as an input the map M, the starting point v_s and the spacing information. In this case we simply apply the clustering step of the L-cover algorithm, to split the area into regions that have approximately the same width (Line 4), and then on each of those segments we apply the M-cover algorithm to generate the more efficient path. The Figure 3.1 shows an example of width-based clusters, denoted in alternating colors, for the same segment of river with different spacing values s.

3.4 Experiments

The performance of the proposed coverage strategies was first tested extensively on different size and shape river maps. Then, some of the generated paths were deployed

Algorithm 4 Width Based M-cover

Input: binary map of river M, starting point v_s spacing parameter s
Output: a π path

1: $C_{vec} \leftarrow$ getDirectionalContours(M)
2: $\theta \leftarrow$ getCoverageDirection(C_{vec}, v_s)
3: $Cl_{vec} \leftarrow getSameWidthClusters(C_{vec}, \theta, s)$
4: **for each** $Cl \in Cl_{vec}$ **do**
5: $\quad p \leftarrow$ M-cover(Cl, v_s, s)
6: $\quad$ append p to π
7: **end for**
8: **return** π

both in simulation using the Stage simulator [129] and in the field to perform large scale river surveying that covered a total $35.82km$ distance. In the latter case, the developed algorithms were deployed on the AFRL Jetyaks [78]. The ASVs are equipped with a PixHawk controller that performs GPS-based waypoint navigation, a Raspberry Pi computer that runs the Robot Operating System (ROS) framework [100] recording sensor data and GPS coordinates (Figure 3.4). For more details on the AFRl Jetyaks, please see Appendix A.1. In addition to the base components for the control and data logging, different types of acoustic range finding sensors were used during deployments.

3.4.1 Performance Analysis

The proposed methods were tested on a set of real maps with up to $3.3km$ long segments of river. Because Z-cover is a partial coverage method it has been compared against a fixed-angle approach used for manual surveying operations [142]. With the latter, the boat always navigates to the opposite shore by making a fixed-angle turn relative to the near shore. The qualitative results in Figure 3.5 demonstrate the motivation behind the Equal Triangle Heuristic approach for improving this operation. When automated, the fixed degree method resulted in severe overshooting and thus loss of coverage area. Meanwhile, the Equal Triangle approach ensures more even coverage.

Figure 3.4: The AFRL jetyak used during the field deployment with different depth sensors mounted on it for surveying operations.

The primary metrics considered to evaluate performance of the coverage tasks are:

- *Covered Area* (%), expressed as a percentage of the total area of the region of interest. For all algorithms, we assume that the travel path π has a width and it is proportional to the spacing value s.

- *Return Path* (%), defined as the percentage of the distance traveled to return back to the starting location v_s, after the coverage was completed, over the total travel distance. This metric is especially important for large scale operations, as returning to the initial location might be time and energy consuming.

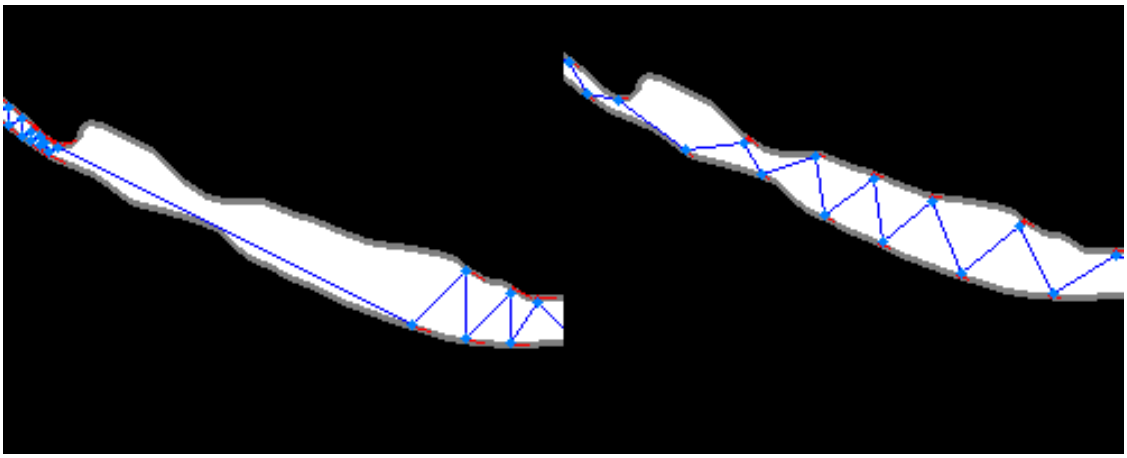


Figure 3.5: Contrasting two Z-cover methods: 45 degree heuristic zig zag method (left) with the equal triangles coverage (right) described in this work. Note, this excerpt is 5% of the original map.

The coverage is efficient if it maximizes covered area while minimizing the return path length. It is worth noting that in the classical coverage path planning problem, the robot has to return to the starting position and there are areas (dead-ends) where the robot enters covering and then has to traverse back resulting in double coverage. During riverine coverage, there is only a single segment which is covered and at the end the ASV has a single return trip to the starting point; as such, we do not use the total distance travelled metric as it is not informative.

In summary:

1. Even though T-cover and L-cover approaches show similar performance in terms of completeness, when accounting for the need for a return trip, the T-cover method is clearly outperformed by the L-cover methods in terms of efficiency of the coverage path (Table 3.2).

2. The quantitative results validate the qualitative observation regarding the differences between the Z-cover algorithm and the fixed-angle approach discussed above; see Table 3.1. The Equal Triangles Method produces paths with slightly higher coverage rate.

3. The T-cover method introduces more turns in the path compared to the L-cover. When using a side scan sonar for bathymetric mapping, this can cause loss of data.

Table 3.1: The Average results of the Z-cover approach compared against the fixed-angle approach.

	Fixed-angle Heuristic	Z-cover
Return Path (%)	43.3 %	41.4%
Area Covered	29.39%	31.05%

Table 3.2: The Average results of the L-cover and the T-cover approaches from simulation.

	L-cover	T-cover
Return Path (%)	8.9%	16.17%
Area Covered	92.65%	91.42%

3.4.2 Field trials

The main objective of the field trials is to ensure that the ASVs are able to collect adequate data when following the trajectories generated by the proposed algorithms. We deployed an ASV to execute the L-cover, T-cover and the Z-cover algorithms on a $0.25km^2$ area of the Congaree River that had an average width of $91m$. For these experiments the ASV was equipped with three different Sonar sensors (see Section 3.4.4). Note that in this work we are assuming that the footprint of the bathymetric sensor (when side-scan sonars are used) is constant and can be calculated based on the average depth of the area/river.

The boat's trajectories in Figure 3.6 and Figure 3.7 are closely aligned to their ideal mission plans, with small deviations caused by GPS error and environmental

forces (wind, current). The effect of those forces have been studied in our previous work [80] and are not the subject of this thesis. In addition, the execution of L-cover on a different and smaller region of the Congaree river with $0.1km^2$ area is presented in Figure 3.6b and Figure 3.6d. The resulting time and distance traveled during each experiment together with actual coverage distance are presented in Table 3.3.

Table 3.3: Coverage time and distance results from the field deployments.

Algorithm	Z-Cover	L-Cover	T-Cover
Time traveled	42m	1hr 45m	1hr 09m
Total Distance	5.2km	13.02km	10km
Coverage Distance	3km	11.02km	7.3km

Finally, a qualitative difference was observed when backscatter images of the riverbed were produced for both autonomous and manual coverage; see Figure 3.11. Note that the manual operation trajectory is not complete compared to the path of L-cover; see Figure 3.11b. The time of operation in both cases was similar (close to 2hr), during which the autonomous operation covered a region twice the area of the manual coverage; compare Figure 3.11a and Figure 3.11b. Moreover, with our approach, the mosaicing is both complete and cleaner because of fewer overlapping tracks and odd orientations to the lines (Figure 3.12).

3.4.3 M-cover Results

Field experiments were also conducted for the M-cover approach, to demonstrate that even with dynamically changing environments, it ensures more efficient coverage compared to the above-mentioned complete coverage techniques. We deployed the ASV on 4.12 km and 2.76 km segments of the Congaree river; see Table 3.4. The width of the river on average is 90m. The long segment was covered with a small

sensor width which resulted in four passes (Figure 3.9a), whereas in the smaller river segment the ASV performed only two passes (Figure 3.8a and Figure 3.8b).

In addition, we executed the L-cover algorithm with same spacing for the smaller region and similar to the M-cover two passes were generated (Figure 3.8c and Figure 3.8d). For the longer region with the same spacing value, L-cover generated segments with either three or five passes, though it resulted in a similar length of the coverage trajectory (Figure 3.9b). When the execution time of the coverage operation of M-cover is compared with L-cover for both experiments, M-cover is on average 20% more efficient. Note that these results are only based on the performed field trials. It has been observed that the river current data change even at an hourly basis [80]. Therefore, generating a graph model that will represent the approximate currents would not be a comprehensive representation of a real world scenario.

Furthermore, we have sampled small portions of the river and compared the coverage time for the two opposite banks of the river to show the effect of the current on coverage time. The results showed that when travelling upstream on the outside portion of the meander the coverage time is almost twice longer than if going downstream (approximately 47%).

Table 3.4: The experimental results of the M-cover deployments

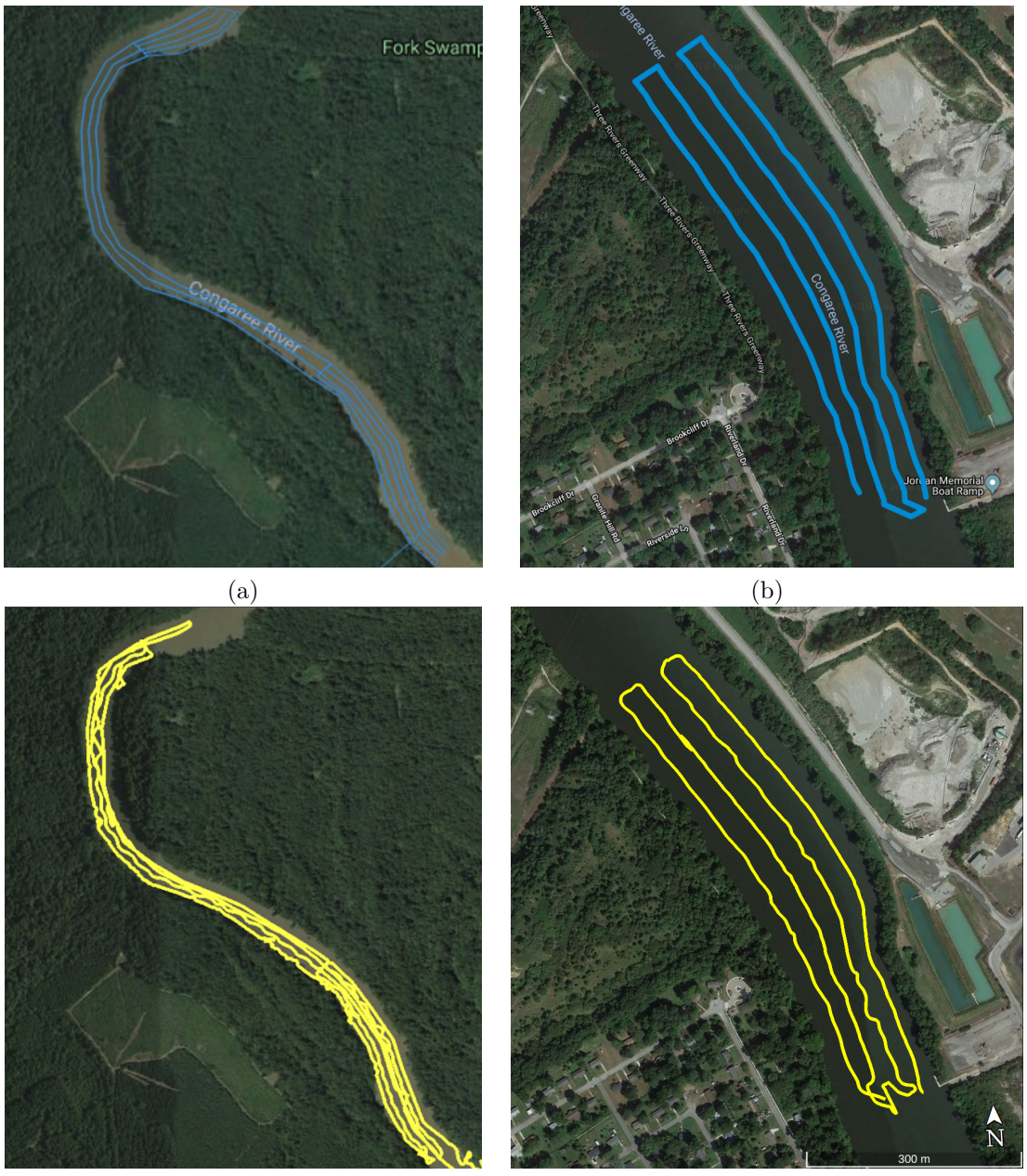
Algorithm	Total Area	Path Length	Duration	# of Passes
M-cover	4.12km x 90m	16.6km	2h 55m	4
L-cover	4.12km x 90m	16.3km	3h 35m	3 and 5
M-cover	2.76km x 90m	5,32km	47m 36s	2
L-cover	2.76km x 90m	5.13km	59m 46s	2

3.4.4 Riverbed Mapping

Different acoustic sensors have been deployed over the course of the field trials in order to evaluate their performance and to consider the effect of different coverage motions on the quality of the collected data. More specifically, a CruzPro DSP Active Depth, Temperature single ping SONAR Transducer was used for the majority of the experiments. As only a single data point is collected at the time, the coverage is sparse and an integration strategy needs to be utilized.

The depth measurements gathered from all experiments with single ping sonar were used to produce a bathymetric map of the covered area utilizing a Gaussian Process (GP) mapping technique [134]. To evaluate the performance of both algorithms for depth map generation, an uncertainty map was produced based on the root-mean-square error (RMSE). The results showed that even though the operation time is longer for the L-cover algorithm, the data collected by the ultrasonic range sensor results in a more accurate depth map. The depth map produced by data collected using L-cover, T-cover and Z-cover patterns are presented in Figure 3.13.

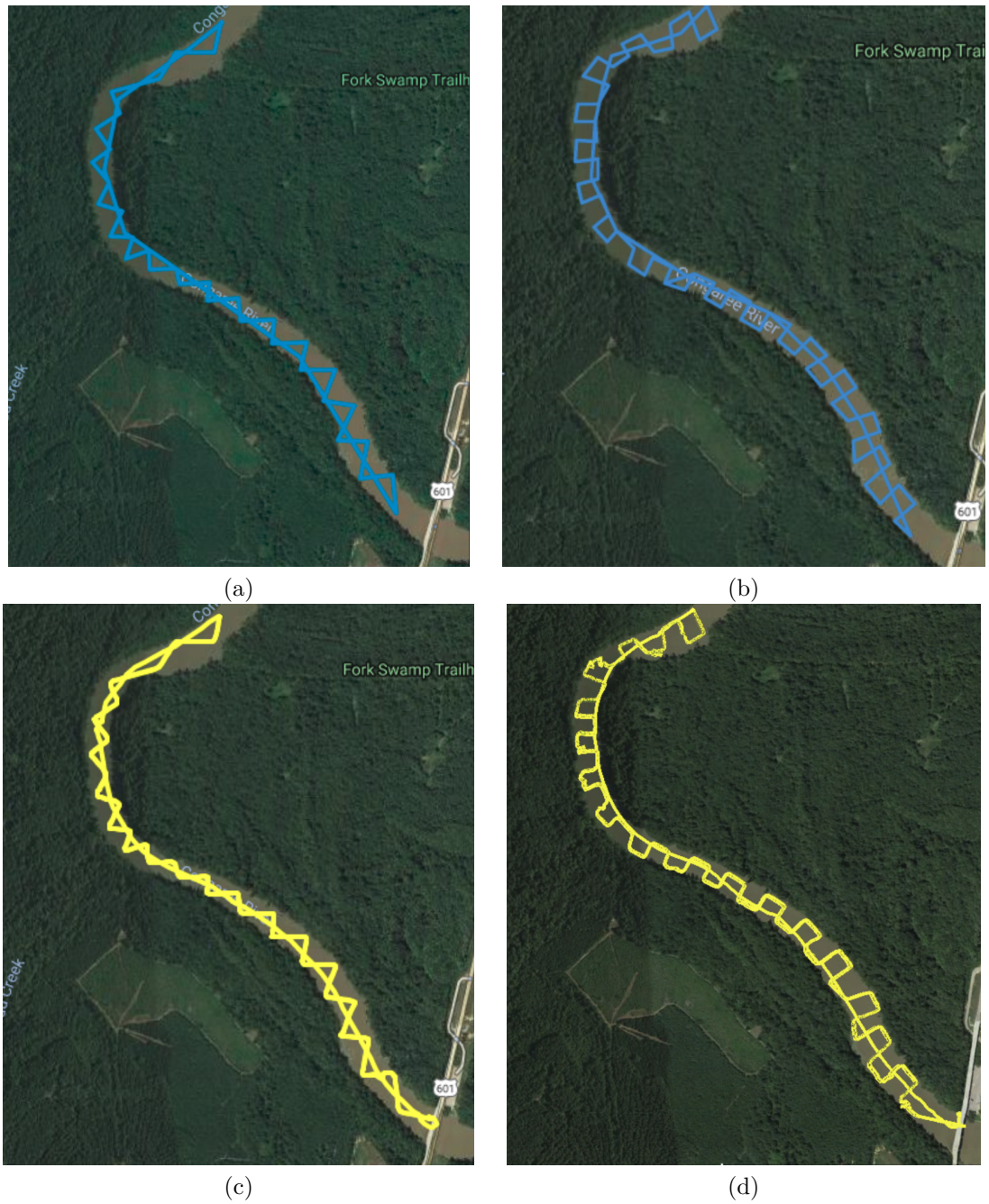
The second sensor used was the Humminbird helix 5 chirp SI GPS G2 imaging sonar. Being a low cost, proprietary sensor, all the collected data has to be post-processed. Finally, a long range 3DSS-DX-450 side scan transducer from Ping DSP[95] was deployed a limited number of times. As can be seen in Figure 3.10, rotations and repeated scans do not match very well due to the sensitivity to orientation error. Acoustic data processing is beyond the scope of this dissertation.



(a) (b)

(c) (d)

Figure 3.6: Riverine coverage on the Congaree River, SC, USA. The blue paths are the ideal paths produced by the algorithms while the yellow one is boat's GPS track: (a), (c) L-cover on a 2.15km long river segment; (b), (d) L-cover on 0.8 km long river segment.



(a) (b)

(c) (d)

Figure 3.7: Riverine coverage on the Congaree River, SC, USA. The blue paths are the ideal paths produced by the algorithms while the yellow one is boat's GPS track: (a),(c) Z-cover; (b),(d) T-cover on a 2.15km long river segment.

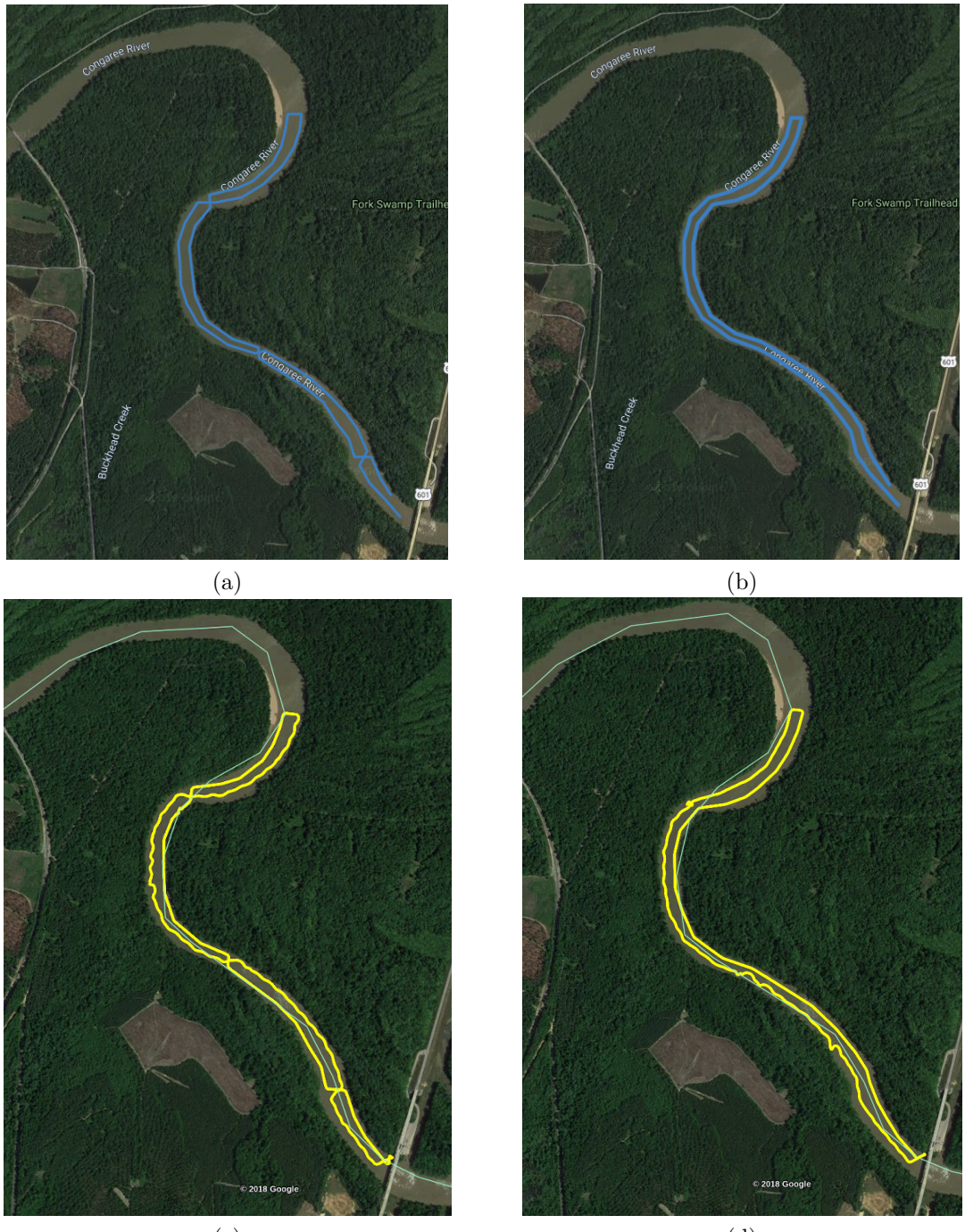
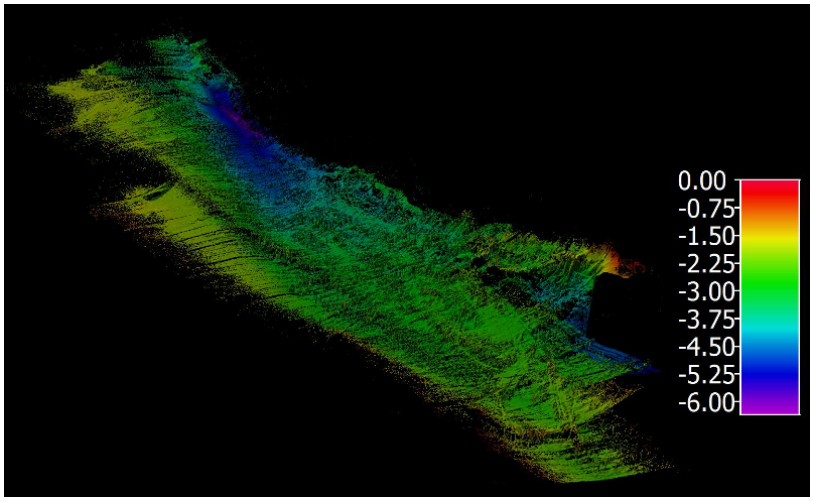


Figure 3.8: Riverine coverage on the Congaree River, SC, USA. The blue paths are the ideal paths produced by the algorithms while the yellow one is the boat's GPS track (a),(b) M-cover. (b),(d) L-cover on a 2.76km long river segment.

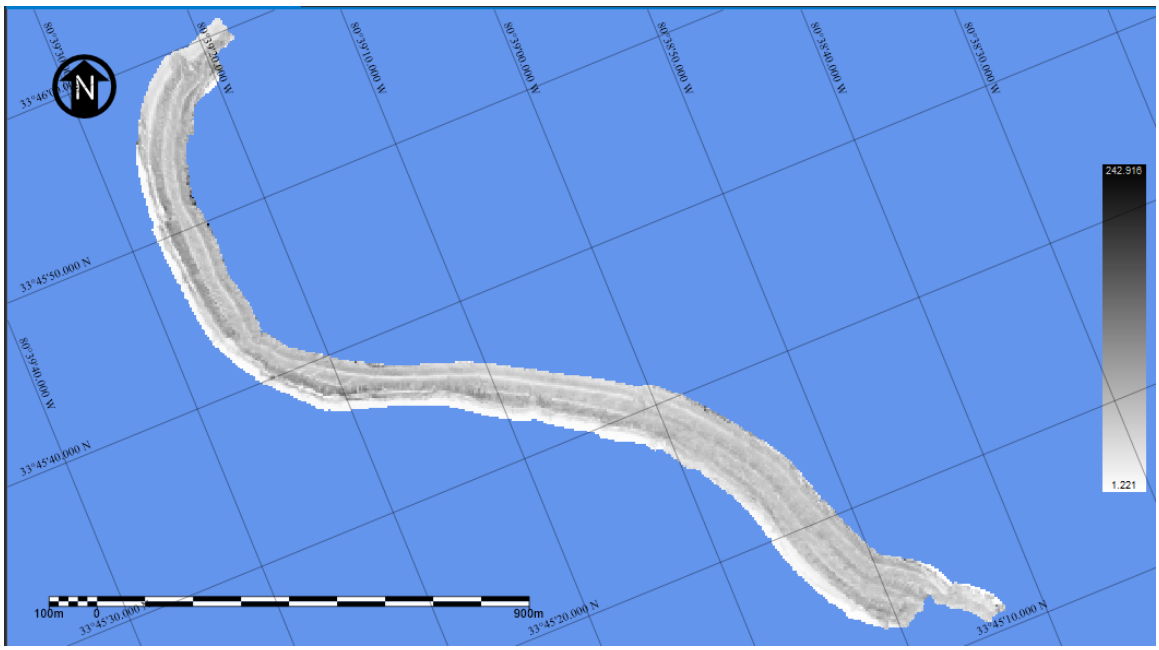
(a) (b)

Figure 3.9: Way-points of the complete coverage patterns executed on the 4.2km long segment of Congaree river, SC: (a) M-cover; (b) L-cover,

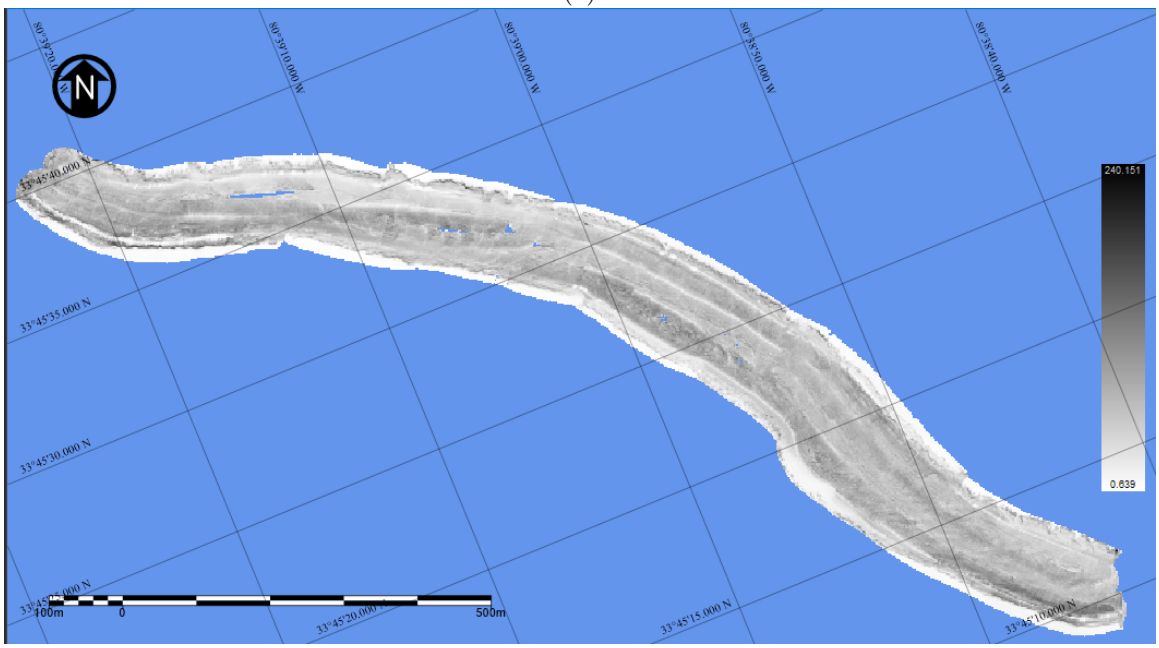


(a)

Figure 3.10: CruzPro depth pinger data integrated using a GP model collected with 3DSS-DX-450 side scan sonar.

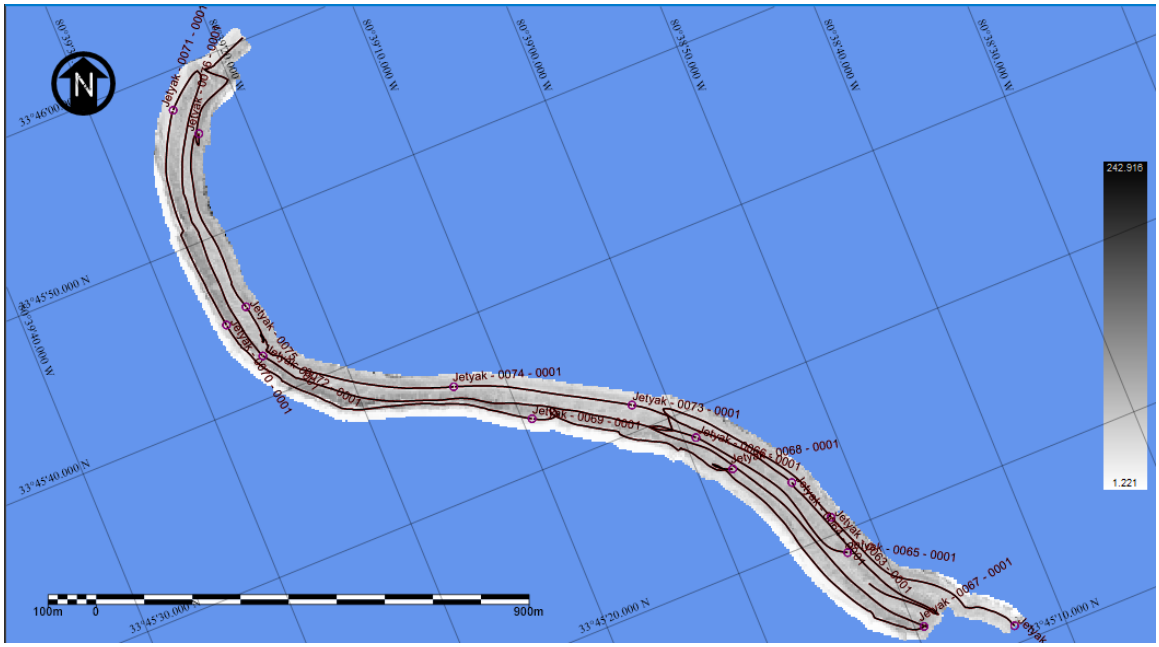


(a)

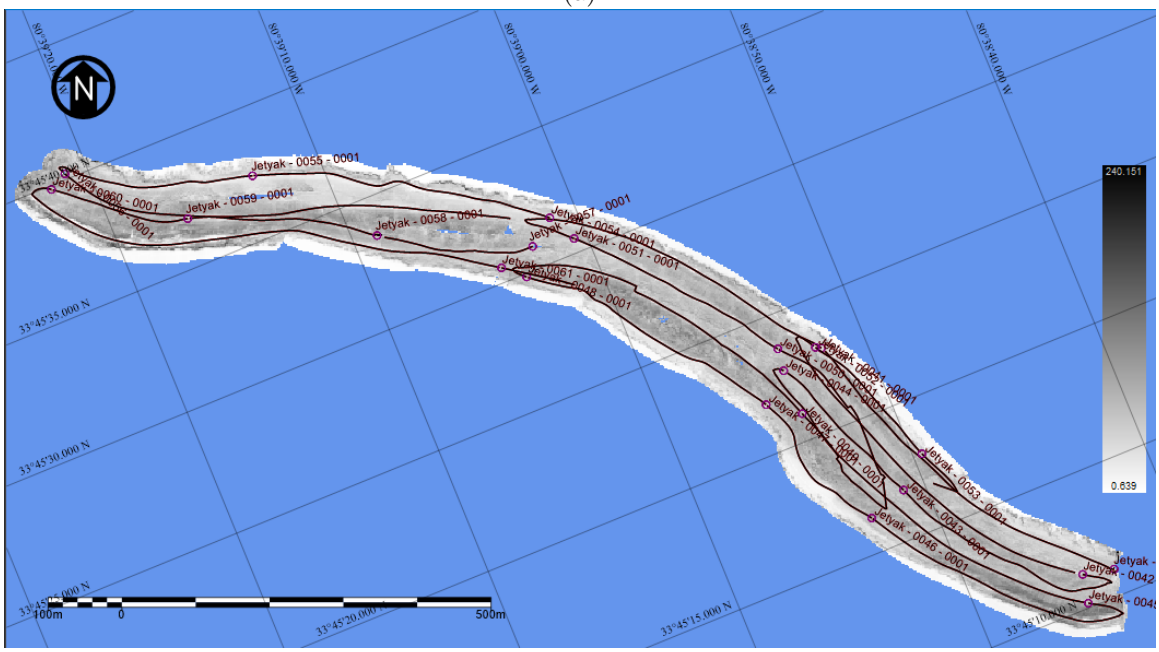


(b)

Figure 3.11: Backscatter image of riverbed, Congaree River, showing the bathymetric map compiled from the Ping DSP data collected by (a) autonomously performing L-cover coverage; (b) manually controlling the boat.



(a)



(b)

Figure 3.12: Backscatter image of riverbed, Congaree River, showing the bathymetric map from Ping DSP with the ASV's path superimposed, from (a) autonomously performing L-cover coverage; (b) manually controlling the boat.

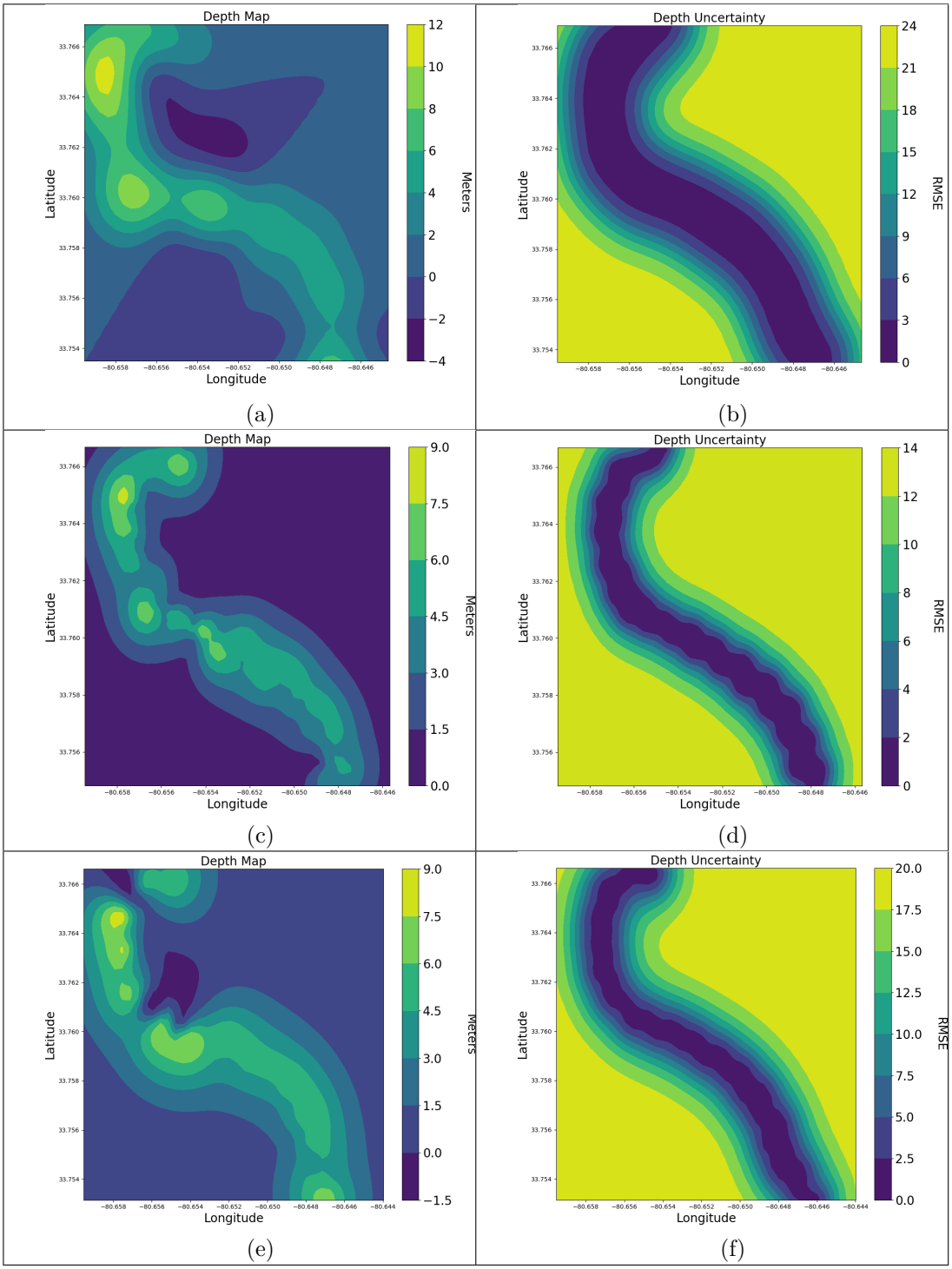


Figure 3.13: The depth map (in meters) of covered region and uncertainty map of selected method for that region expressed by RMSE: (a), (b) L-cover; (c),(d) Z-cover; (e),(f) T-cover.

Chapter 4

Multi-robot Coverage

4.1 Introduction

This chapter addresses the problem of covering a large area for environmental monitoring with multiple Dubins vehicles (Figure 4.1). Finding a solution to the coverage problem means planning a trajectory for a mobile robot in a way that an end-effector, often times the body of the robot, passes over every point in the available free space. Employing multiple robots can reduce the coverage time cost, and, in hazardous conditions such as humanitarian de-mining, increase the robustness by completing the task even in the event of accidental "robot deaths." The use of multiple robots however, increases the logistical management and the algorithmic complexity.

Figure 4.1: Three autonomous surface vehicles performing coverage on Lake Murray, SC, USA.

When covering a known environment, the focus is on performing the task as efficiently as possible [137]. The scale of the environment in conjunction with the

speed and endurance of the robot(s) classify the coverage task as small, medium, or large scale. For example, a flying vehicle with 30 min battery life and an average speed of 40 km/h can cover a trajectory of 20 km, while an autonomous surface vehicle (ASV), moving at 10 kn (5 m/s) for five hours will travel approximately 90 km.

As in the previous chapter we are focusing on the monitoring of aquatic environments. The vehicles of choice are ASVs that were custom-made at the University of South Carolina (see A.1 for more details). Aquatic environments, in general, require large scale operations. For example, one of the testing grounds used—Lake Murray—has a surface of over 40 000 m^2. Many ASVs, similar to fixed wing aircraft, are governed by Dubins vehicle kinematics [21]; i.e., Dubins vehicles cannot turn in place. More formally, a Dubins vehicle is defined as a vehicle which may only follow line segments and arcs with radius greater than some specified minimum with non-negative velocity, i.e., they may not back up.

Recent works [67, 141] presented an efficient approach to cover an area by a single Dubins vehicle. We extend the proposed algorithm to multiple robots based on the multi-robot coverage path planning algorithms proposed in [52] for holonomic robots. That work proposed two approximation algorithms, termed CAC and CRC, for solving the problem. Nevertheless, the CRC algorithm was not implemented at the time and thus was not compared with CAC within that work.

We will first show the comparison results of the proposed algorithms without the Dubins constraint. Efficiency is measured as a combination of the utilization of the robots and the reduction of the maximum coverage cost. The idea is that robots are limited by the battery life; as such, the workload should be evenly distributed. Then we will present two methods based on the same approaches utilized by the CRC and CAC algorithms. In the first one, the efficient path produced by the algorithm proposed by Lewis et al. [67] is divided into approximately equal parts, in terms of path length, and each part is assigned to a different robot. In the second method, the

target area is divided into equal parts, based on the team size, and then the single robot algorithm [67] is applied to each area.

Analytical comparison of CRC and CAC algorithms show that CRC is slightly advantageous in terms of efficiency. Experimental results of the new proposed methods for Dubins vehicles using several simulated world maps show that indeed the utilization of the robots are maximized and the maximum coverage cost is minimized compared to even the original CRC. Moreover, the approach is scalable to a large number of robots. Field trials with a single robot, a team of two, and a team of three ASVs, demonstrated the feasibility of the proposed approach with real robots executing plans generated by the planner and highlighted several practical challenges.

4.2 Problem Statement

The *Dubins multi-robot coverage problem* can be formulated as follows. We assume to have k homogeneous robots, with no communication capabilities, each equipped with a sensor with fixed-size footprint s, and with Dubins constraints—namely, the robots have a minimum turning radius r that constrains the robots to follow line segments and arcs with radius greater than or equal to r, and they cannot drive in reverse. Such robots are deployed in a 2D-bounded area of interest region $\mathcal{E} \subset \mathbb{R}^2$. The objective is to find a path π_i for each robot i, with $1 \leq i \leq k$, so that every point in the region of interest $\mathcal{E}$ is covered by at least one robot's sensor.

An efficient solution is one that minimizes the length of the trajectories for the robots, while at the same time ensuring that the workload on the robots are evenly distributed. This is motivated by the fact that homogeneous robots have the same limited battery life, and thus, to cover a big region, it is better to utilize all of them for the coverage task.

In practice, this means that an efficient algorithm finds k non-overlapping regions $\mathcal{E}_i \subset \mathcal{E}$ such that $\mathcal{E} = \bigcup_1^k(\mathcal{E}_i)$, where each robot i can perform a calculated covering

trajectories π_i. Note that π_i includes the whole path the robot has to follow: a robot starts from an initial starting point v_s, goes to a point of entry into a partition of the interest region $\mathcal{E}_i$, covers fully $\mathcal{E}_i$, and goes back to v_s. We call the coverage cost—i.e., the traveled distance—of a single robot covering $\mathcal{E}_i$ as $c(\mathcal{E}_i)$. As such, we can define the optimization problem of Dubins multi-robot coverage as a *MinMax* problem: minimizing the maximum cost $\max_1^k(c(\mathcal{E}_i))$ over all robots.

4.2.1 Terminology

In this section we introduce the terminology used in the subsequent sections. A *cell* is defined as a continuous region containing only the area of interest that one of the robots must cover entirely. The cells are the result of a boustrophedon cell decomposition (BCD) [15]. The Dubins coverage algorithm by Lewis et al. [67]—referred to as Dubins coverage solver (DCS)—is a process by which a coverage problem is mapped to a graph for which a solution to the TSP results in a single coverage path.

The DCS algorithm divides cells into a collection of *passes*, defined as the smallest unit of coverage; each of which is axis-aligned and has a width equal to the robot's sensor footprint. Each pass has an assigned direction to cover its cell and thus it is presented as a node of a directed, weighted Dubins graph $G_d = (E_d, V_d)$. The edges of G_d are defined as the Dubins path from a source node to a target node. The weight of an edge $w(u, v)$ is then the length of the segments and arcs of the Dubins path between two passes u and v. The output of the DCS algorithm is an optimal *Hamiltonian path* $R = (v_1, v_2, ..., v_n)$, where $v_i \in V_d$ and n is the number of passes, that is $n = |V_d|$.

4.3 Dubins Coverage with Route Clustering (DCRC)

Our first approach for multi-robot Dubins coverage is based on DCS and the Coverage with Route Clustering (CRC) method [67].

The CRC algorithm creates cells applying the BCD algorithm on a binary image of the area with obstacles [53]. Then, boustrophedon cells are turned into edges of a weighted graph—called Reeb Graph—on which the k-Chinese Postmen Problem (k-CPP) is solved. The result is a k sub-routes of an optimal Eulerian tour.

To address Dubins constraint in this work, we are interested in solving the k-TSP problem instead of k-CPP. The pseudocode for DCRC is presented in Algorithm 5. Line 1 gets an optimal Hamiltonian path $R = (v_1, v_2, ..., v_n)$, where the vertices are passes, by using the DCS algorithm to solve the single-robot Dubins Coverage problem with the DCS algorithm. Its cost $c(R)$ is given by the initial traveled distance to get to the region of interest $c(v_s, v_1)$, the sum of the costs $w(v_j, v_{j+1})$ to cover passes v_j,v_{j+1}, and the cost $c(v_n, v_s)$ to go back to the starting point v_s (Line 3). Note that the travel cost $c(v, u)$ is defined as Euclidean distance between midpoint coordinates of corresponding u and v passes. The resulting optimal path R is split into k subtours $\{R_1, R_2, ..., R_k\}$ (Lines 4-7). For a given starting point v_s, the cost of any tour $R_i = (v_{i_1}, v_{i_2}, ..., v_{i_m})$ is defined as the cost of traveling from the starting point to reach a designated coverage cell, the actual cost of covering that cell and the cost of traveling back to the starting point (Line 8, where m is the index of the last pass/vertex in the path). Cost $c_{\max}$ is calculated to balance travel and coverage costs between robots (Line 3). Such a clustering procedure was proposed in the k-TSP solver by Frederickson el al. [30].

The complexity of this algorithm is exponential as DCS uses an exact TSP solver. The k-CPP solver — referred as FHK — is proved to have an approximation factor of $\frac{5}{2} - \frac{1}{k}$ [30].

Algorithm 5 DCRC

Input: number of robots k, binary map of area M,
turning radius r, sensor footprint s
Output: k tours, 1 for each robot

1: $R \leftarrow \text{DCS}(M, s, r)$
2: *initialize for each i in k empty tours* R_i
3: $c(R) = c(v_s, v_j) + \sum_{j=1}^{n-1} w(v_j, v_{j+1}) + c(v_n, v_s)$
4: $c_{\max} = \max_{1 \leq i \leq n} \{c(v_1, v_i) + w(v_i, v_{i+1}) + c(v_{i+1}, v_1)\}$
5: **for each** $i \in$ *1, ..., k* **do**
6: **while** $c(R_i) <= (c(R) - 2c_{\max}) * i/k + c_{\max}$ **do**
7: *include next vertex v along R into* R_i
8: $c(R_i) = c(v_s, v_{i_1}) + \sum_{j=1}^{m-1} w(v_{i_j}, v_{i_{j+1}}) + c(v_{i_m}, v_s)$
9: **end while**
10: **end for**

4.4 Dubins Coverage with Area Clustering (DCAC)

The DCAC algorithm, similar to the CAC algorithm [53], performs clustering of the region of interest $\mathcal{E}$ and then finds the optimal route for each robot. An overview of the DCAC algorithm is presented in Algorithm 6.

In particular, the BCD algorithm is applied to decompose the environment into cells, consisting entirely of areas which should be covered (Line 1). Then, each cell is divided into passes (Line 2). A corresponding graph is created from these passes (Line 3). The graph is an undirected weighted graph $G = (V, E)$, where each vertex is located at the center of a pass; vertices (v_i, v_j) in this graph are connected with an edge e if and only if their corresponding passes share a common edge. The cost $c(e)$ of each edge $e = (v_i, v_j)$ is defined as the Euclidean distance between midpoints of passes. The vertices of graph G are clustered performing a breadth-first search (BFS) clustering (Line 4). The size of a cluster $C = \{v_1, v_2, ..., v_m\}$ is defined as $c(C) = \sum_{\{e|e=(v_i,v_{i+1}),1<i\leq m\}} c(e)$. DCS is then applied on each resulting cluster of passes (Lines 5-7).

The clustering step in the CAC algorithm [53] ensures that the cost of reaching the region of interest and the actual coverage costs per region are balanced, by assigning

more passes to cover to robots that are closer to the region of interest; while the robots that have to travel more to reach the coverage area will have less passes to cover.

Algorithm 6 DCAC

Input: number of robots k, binary map of area M,
turning radius r, sensor footprint s
Output: k tours for each robot

1: $cells \leftarrow \text{BCD}(M)$
2: $passes \leftarrow \text{GenPasses}(cell, s)$
3: $G \leftarrow \text{buildGraph}(passes, r)$
4: $C\_set \leftarrow \text{BFSClustering}(G, k)$ ▷ clusters of passes
5: **for each** $C_i \in C\_set$ **do**
6: $tour \leftarrow \text{DCS}(C_i, r, s)$
7: **end for**

As the complexity of TSP is exponential, by partitioning problem into k small TSP subproblems, the overall TSP performance is improved. Nevertheless, the complexity will still remain exponential.

4.5 Comparison of CRC and CAC Methods

The experimental validation of the proposed algorithms consists of two components: first, a large number of randomly generated Reeb graphs were tested, analysing the performance of the area partitioning stage of proposed CRC and CAC algorithms. Second, multi-robot simulations were run in the Stage mobile robot simulator [129] for different environments to validate statistical results. In both cases, we compare the CAC and CRC methods with simple equal partitioning and the original FHK algorithm with only one edge cost. We will refer to last one as FHK. As for simple equal partitioning, we partition a single optimal route using cluster cost equal to $1/k$ of the optimal route cost, where k is the number of robots. We call this naive route clustering and we will henceforth refer to it as NRC.

4.5.1 Statistical Analysis

For statistical analysis we compare the area partitioning phase performance of the proposed algorithms, i.e. the k-tour splitting FHK algorithm used in CRC algorithm with two different edge costs and the DFS-like graph clustering used in CAC. Moreover, we show the performance of these algorithms compared to the baseline, i.e. the original FHK algorithm [30] and NRC. Before presenting the actual results, the testing framework is described in detail.

Test Data

Testing data represent the same randomly generated 200 images used in the original work, with arbitrarily distributed obstacles. The boustrophedon decomposition is performed on each of these artificial environments and as a result a Reeb graph is produced. In the resulting graph, the Eulerian path is constructed by applying the algorithm solving the Chinese postman problem. The following input is required for the comparison of the algorithms: a Reeb graph with coverage cost and travel cost assigned to each edge, an Eulerian tour of that graph, and the number of tours the graph is partitioned into. Size information about the input graph and the single coverage tour costs are shown in Table 4.1. The number of robots or tours that the graph will be partitioned into have the following values $k \in \{1, 4, 8, 16, 20, 32\}$. All data examples are used with all possible k values. Note that single cell coverage is considered an atomic action. As such, it is only divided where an edge (cell) has to be duplicated (at most once).

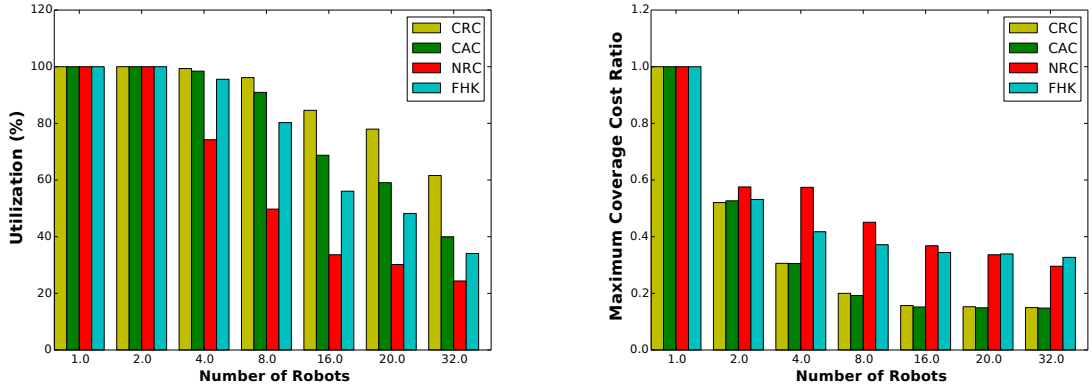
Measurement Metrics

Similar to the original work [52], for every call of the algorithm, we measure the number of idle robots and the maximum coverage cost. In Figure 4.2 the maximum coverage cost is represented as a fraction of a single optimal coverage path, and the

Table 4.1: Input Graphs Information

	MIN	MAX	MEAN
Number of Vertices	10	41	25
Number of Edges	12	57	35
Length of Eulerian Tour	16	73	43
Cost of Eulerian Tour	2414	11176	5123

utilization as the percentage of covering robots. We present average results over all input data for each number of robots in Figure 4.2. In addition, to measure overall effectiveness, we averaged results over all k robots, when $k > 1$; see Table 4.2.



(a) (b)

Figure 4.2: CRC, CAC utilization and max coverage cost average results compared to NRC and original FHK algorithms for a variable number of robots: (a) Robot utilization (highest number is better); (b) Maximum area covered (lowest number is better).

Results

The area coverage problem was defined previously as a MinMax problem [52]. The maximum coverage cost is minimized by ensuring that no robot stays idle. However, even with this definition there is a possibility to have idle robots when the size of cells is not balanced.

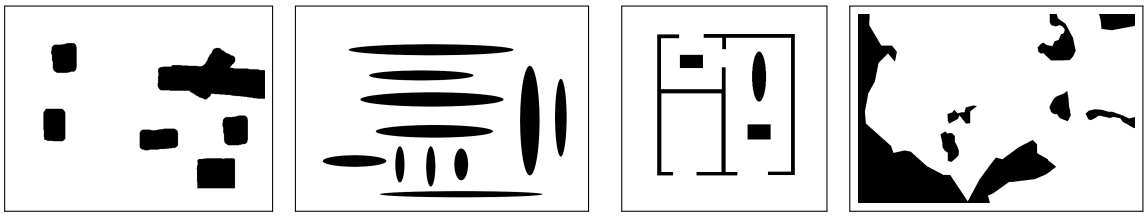
Table 4.2: Average Results for $k \in \{2, 4, 8, 16, 20, 32\}$ robots.

	Utilization (%)	Max Coverage Cost ratio
CRC	86.6	0.248
CAC	76.2	0.245
NRC	52.0	0.433
FHK	69.0	0.388

All algorithms were applied on the datasets generated as described above. The results show that the CRC and CAC algorithms demonstrate similar performance on solving the MinMax problem and on utilizing the robots; see Figure 4.2. But on average, for larger numbers of robots, CRC shows 10% better utilization compared to CAC . Meanwhile both CRC and CAC outperform NCR and FHK. On average CRC and CAC outperform NRC and FHK algorithms by increasing the utilization by 20.5% and reducing the coverage cost by 39.8%; see Table 4.2.

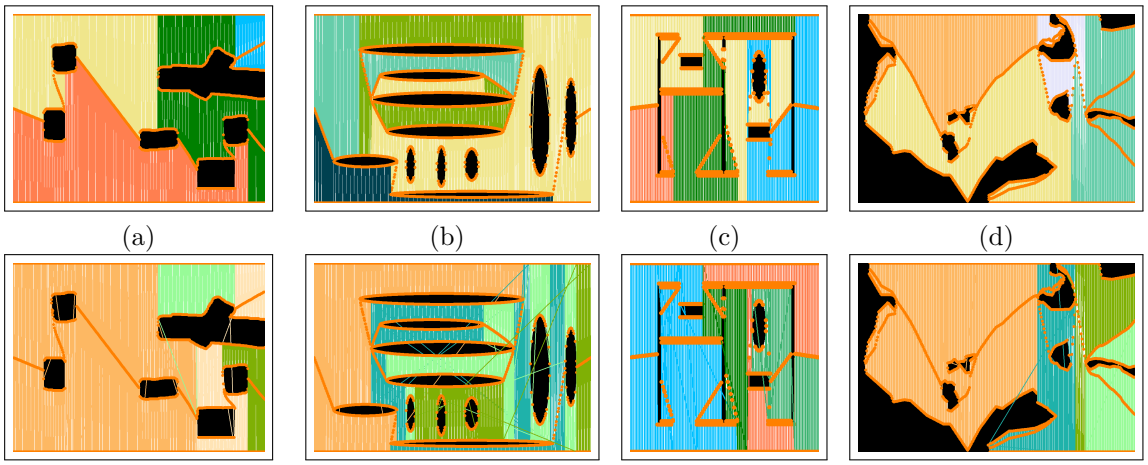
4.5.2 Experimental Validation in different environments

For testing the complete pipeline of CRC and CAC, different environments were used. In Figure 4.3(a), presents a variant of the well known cave environment from Stage [129]. A complex environment with many obstacles is presented in Figure 4.3(b). An indoor environment is shown in Figure 4.3(c), and the large environment from rural Quebec from the work of Xu et al. [137] is presented in Figure 4.3(d). The top row in Figure 4.4 presents the coverage path for four robots performing the CRC coverage algorithm, while the second row of Figure 4.4 presents the respective coverage paths utilizing the CAC algorithm. Each of the robots utilizes the Boustrophedon coverage pattern [16]. As can be seen from the results in the different environments, the distribution of areas among the robots vastly varies. To a large extent, the fact that each cell represents an atomic coverage action is responsible for the uneven distribution of tasks.



(a) (b) (c) (d)

Figure 4.3: The four environments where the proposed multi-robot coverage algorithms were tested: (a) Cave; (b) Multi-cell; (c) Indoor; (d) Rural Quebec.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 4.4: Coverage paths for four robots utilizing (a-d) CRC coverage algorithm; (e-h) CAC coverage algorithm.

To validate statistical results, the same four environments were also used to test the performance of the proposed algorithms in comparison with the NRC and FHK algorithms for different numbers of robots. The maximum coverage cost and utilization ratio are presented in Figure 4.5 and Figure 4.6, calculated the same way as it was described in the previous section, for covering each of the environments presented in Figure 4.3. Different numbers of robots ({1,2,4,8,16,20,32}) were used. The results once again show that both CRC and CAC in comparison with naive clustering (NRC) and FHK provide better minimization and utilization, even though they have the same convergence for minimizing the maximum coverage cost. It is worth noting that the scale of each environment is arbitrary and the distances measured only serve

as relative measurements. What is significant, however, is the comparison among different numbers of robots in a single environment.

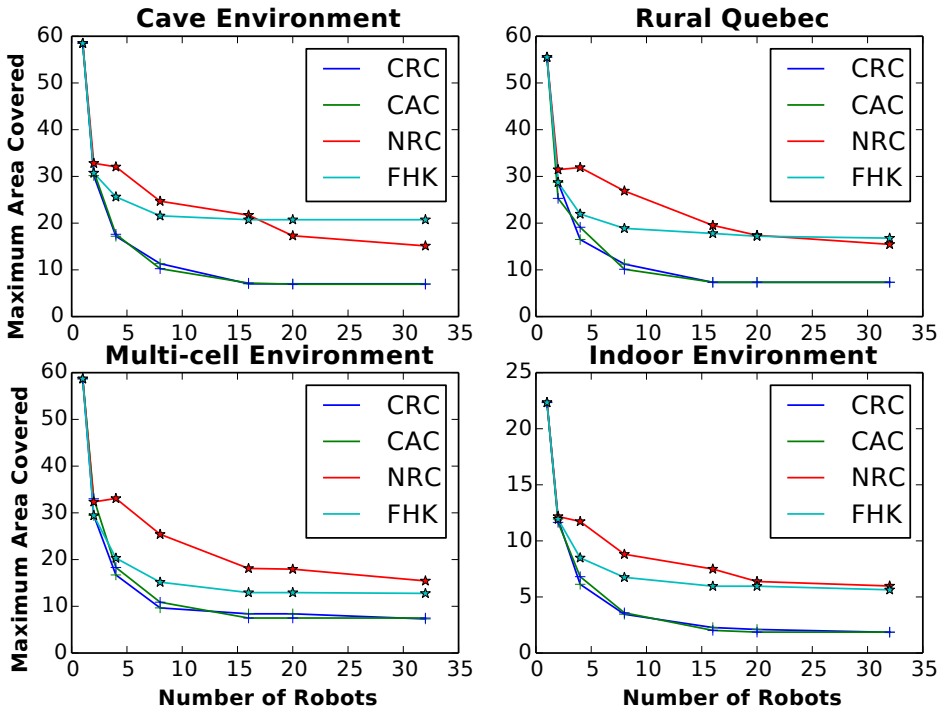


Figure 4.5: The maximum coverage cost of the multi-robot coverage algorithms in different environments and for different number of robots.

4.6 DCRC and DCAC Experiments

The proposed methods were initially evaluated with simulation tests for large environments within a custom simulator that accounts for the Dubins constraints, to test the optimality of such an approach and its scalability.

Next, we used our autonomous fleet of jet-driven Mokai ES-Kape sport kayaks (see A.1) for validating the proposed approach with real robots. The goal includes checking if the assumptions hold in the real world. The ASVs in addition to the baseline electronics — a PixHawk controller for waypoint navigation and safety behaviors, and a Raspberry Pi with the Robot Operating System (ROS) framework [100, 86]

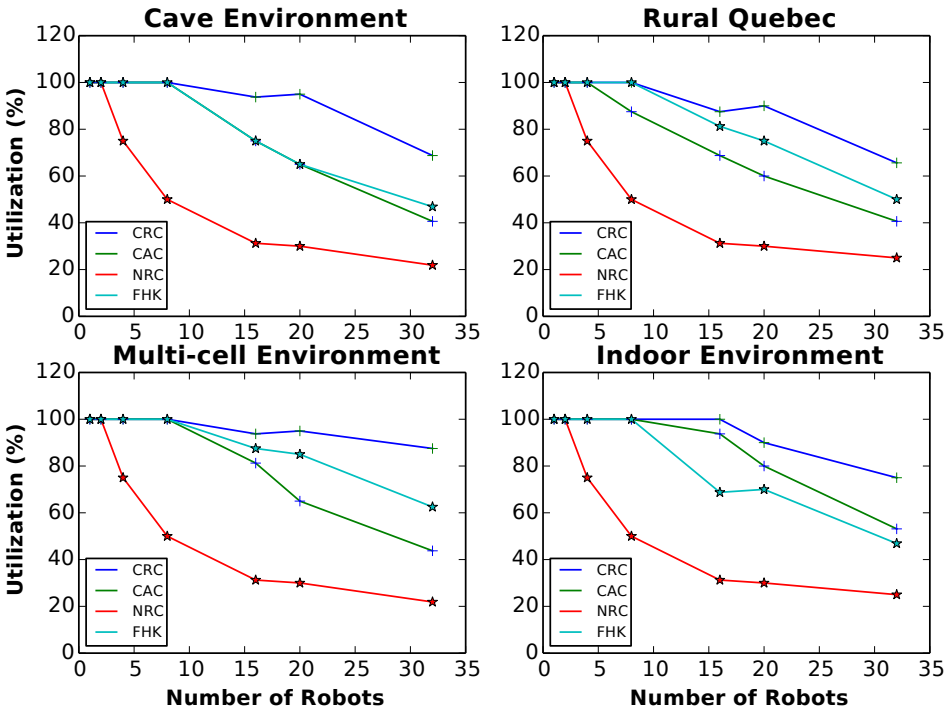


Figure 4.6: The utilization factor comparison the multi-robot coverage algorithms in different environments and for different number of robots.

to record GPS and depth data – are equipped with a SONAR transducer collecting depth measurements with a frequency of 1 Hz.

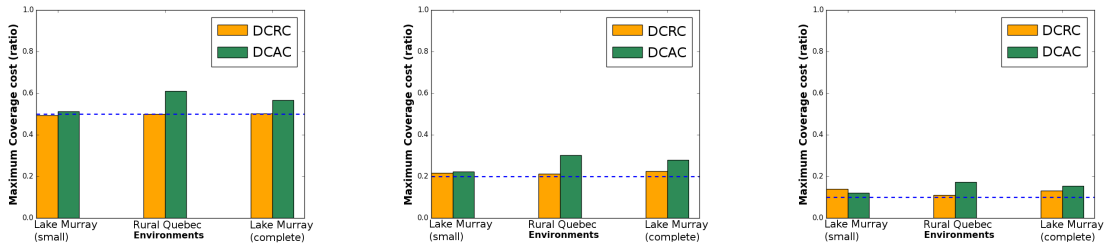
4.6.1 Simulated Results

The simulation was performed for three large input maps taken from Lake Murray [67] and a rural Quebec area [137]. The maps differ in terms of size and shape complexity. We have evaluated both DCRC and DCAC with different numbers of robots, that is $k \in \{1, 2, 5, 10\}$ robots. The baseline for comparing the costs of each tour is the cost of the optimal route produced by the TSP algorithm. As the problem is defined as a MinMax problem, we consider the value of the maximum cost per robot along with the ideal cost as a metric. The ideal cost is defined by dividing the single optimal route cost by the number of robots. Another metric considered in this paper is the

utilization of robots, that is the ratio between the number of robots used and the total number of robots available. However, in the following, results with the robots' utilization are not reported: in all the experiments, the robots' utilization is 100%, a difference from the results obtained in [53]. The way we designed our algorithms, namely with the additional decomposition of the obstacle free cells in passes, results in a more even distribution of cells to the robots.

Figure 4.10 shows the paths followed by 5 robots in the three environments considered, using both algorithms. Qualitatively, it can be observed that DCAC produces paths where robots mostly transition to adjacent passes, while with DCRC, robots go from one pass to another that is typically not adjacent. This fact makes the robots follow the paths generated by DCAC going outwards from the region of interest because of the minimum turning radius—compare for example Figure 4.11 (a) and (b). Those tighter turns contribute to an increase in the overall cost.

Indeed, as illustrated in Figure 4.7—which shows the ratio between maximum coverage cost and ideal cost—DCRC has better performance. For example, in the Rural Quebec environment with 5 robots, DCRC has a maximum coverage cost ratio of 0.2, while for DCAC is 0.3.



(a) (b) (c)

Figure 4.7: The comparison of actual maximum coverage cost and the ideal cost for three different environments for (a) $k = 2$, (b) $k = 5$, and (c) $k = 10$ robots.

4.6.2 Field trials

Given the better performance of DCRC, we validated the approach using DCRC with the ASVs, in a 200 m × 200 m area in Lake Murray, SC. The sensor footprint used had 4.5 m and the turning radius of the ASV is 5 m. A path, in the form of a waypoint sequence, was generated with the ASVs starting just outside the area of interest. A description of the experiments performed and the results obtained follows.

The main objective of the field trials was to ensure that the assumptions hold with real robots, so that the ASVs are able to follow the trajectories generated by the proposed algorithms.

Single Robot Coverage baseline

Similar to the simulation experiments, the single robot coverage for Dubins Vehicles algorithm [67] is used here as a baseline for comparison with the multi-robot approach.

Figure 4.9a and Figure 4.9b present the ideal path and the path followed by the ASV, respectively, as recorded GPS points overlaid on Google Maps. The depth measurements were combined using a Gaussian Process (GP) mapping technique [134] to reconstruct the floor map of that part of the lake (see Figure 4.8a).

Multi-Robot Coverage Experiments

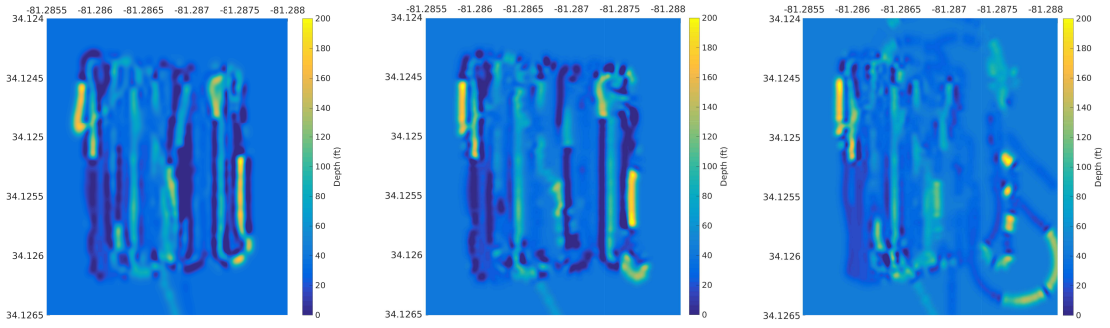
A variety of experiments were performed using teams of two or three robots in different areas of Lake Murray.

The resulting multi-robot coverage is comparable to the single-robot coverage trajectory, where only small areas were left uncovered. Indeed, the bathymetric maps resulting from the single and multi-robot coverage are similar.

The maximum traveled distances per experiment with a different number of robots are presented in Table 4.3 along with the ideal traveled distance. As in the case of the

Table 4.3: The maximum distance traveled per robot and the cost of perfect division for multi-robot coverage experiments with the deployed ASVs.

Number of Robots	1	2	3
Max Distance	6863 m	2905 m	3356 m
Ideal Distance	6863 m	3431.5 m	2287.7 m



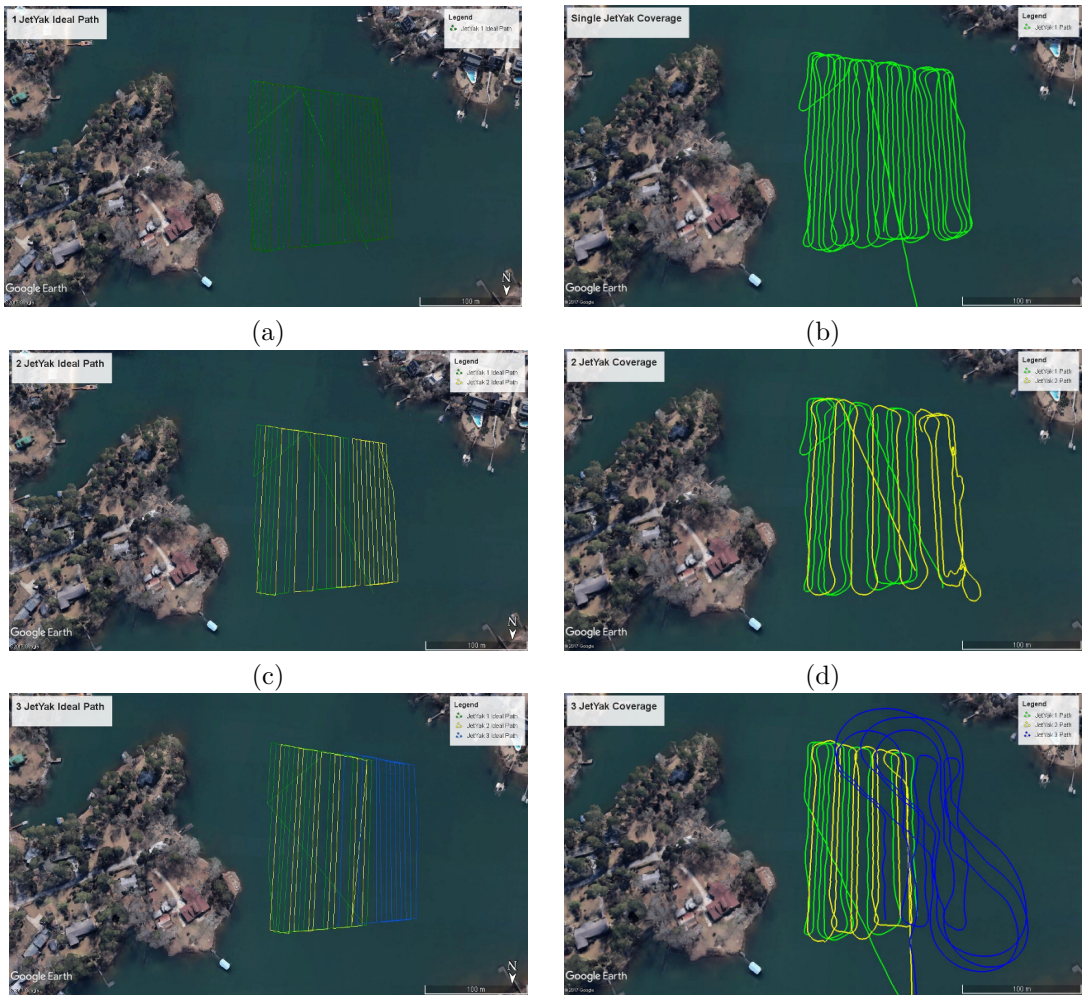
(a) (b) (c)

Figure 4.8: Depth map of Lake Murray produced using a GP-based mapping using data collected with (a) a single robot, (b) two robots, (c) three robots.

simulation, the ideal path length is the size of the sub area if the tasks were exactly divided into equal parts.

Figures 4.9c and 4.9e show the ideal path for two and three ASVs as generated by DCRC; while Figures 4.9d and 4.9f show the actual path followed by two and three robots, respectively. As can be seen qualitatively in Figures 4.9c and 4.9e, the paths followed by the ASVs are pretty much in line with the ideal path. The small deviations are due to GPS error, current, wind, and waves from other vessels. As such, the proposed methods can be applied for coverage using ASVs with Dubins constraints.

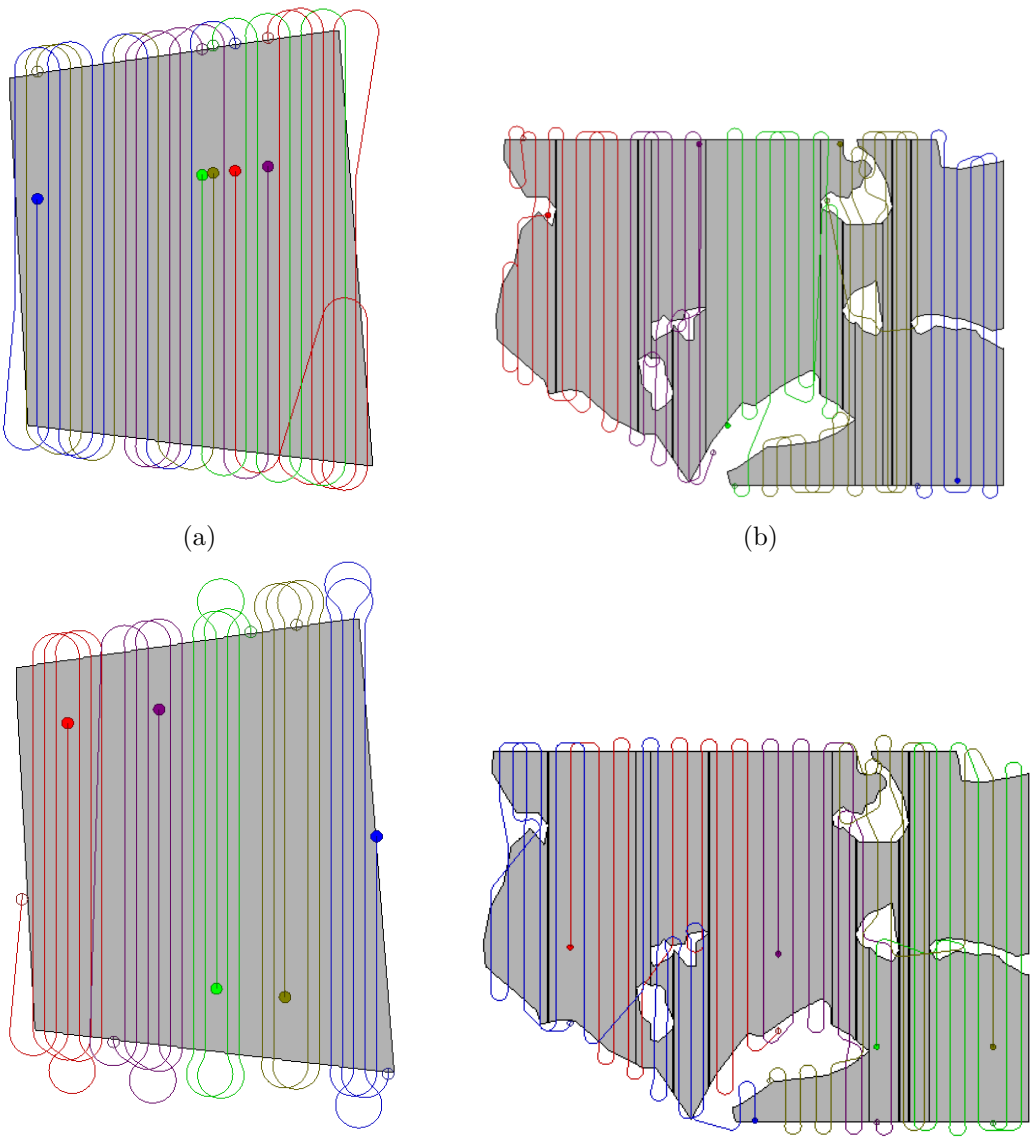
Note that the ill-structured path of one of the robots (with the blue trajectory in Figure 4.9f) is a result of a hardware failure and a hysteresis of its on-board PID controller. This illustrates the real world challenges with field trials: even with



(a) (b) (c) (d) (e) (f)

Figure 4.9: Multi-robot experiments at Lake Murray, SC, USA: (c) Ideal path produced for two and (e) three robots; (d) GPS track of the actual coverage path for two and (f) three robots.

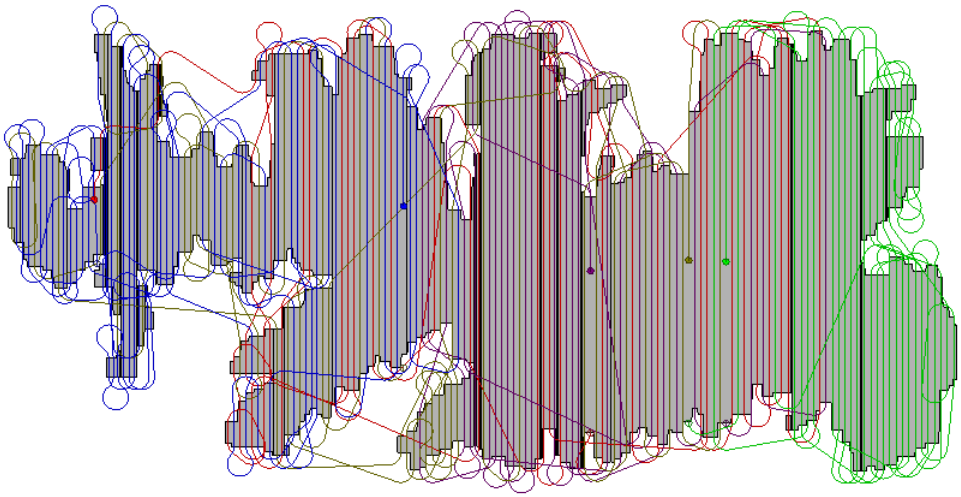
presumably identical boats, each of them should undergo an initial tuning phase of the different operational parameters. Such an issue opens interesting research directions on robust multi-robot coverage, including recovery mechanisms to adapt the algorithms to the new minimum turn radius and accounting for heterogeneity.



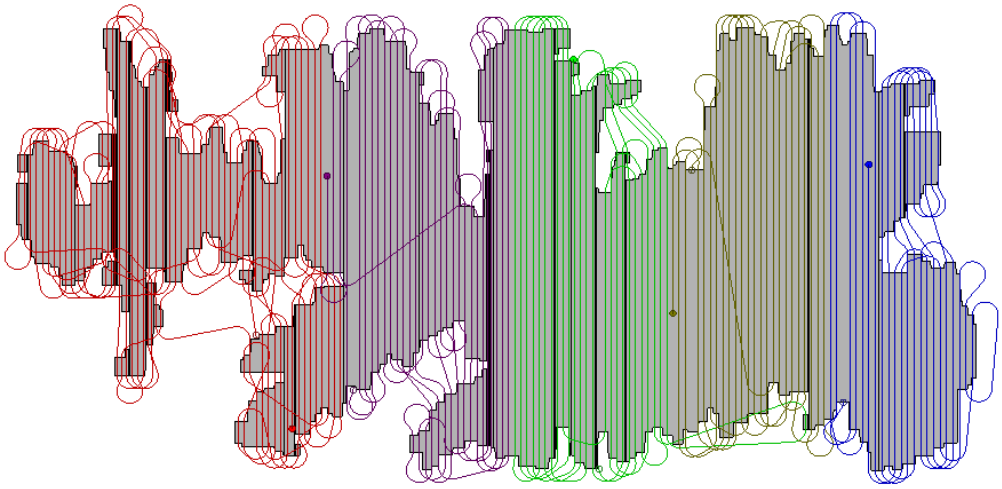
(a) (b)

(c) (d)

Figure 4.10: A simulation instance of DCRC (first row) and DCAC (second row) algorithms with 5 robots performing coverage over the area of interest indicated in gray, where the first column shows a small segment in Lake Murray (200 m × 200 m); second column, Rural Quebec (13 km × 10 km)



(a)



(b)

Figure 4.11: A simulation instance of algorithms on the complete Lake Murray (25 km × 25 km): (a) DCRC; (b) DCAC .

Chapter 5

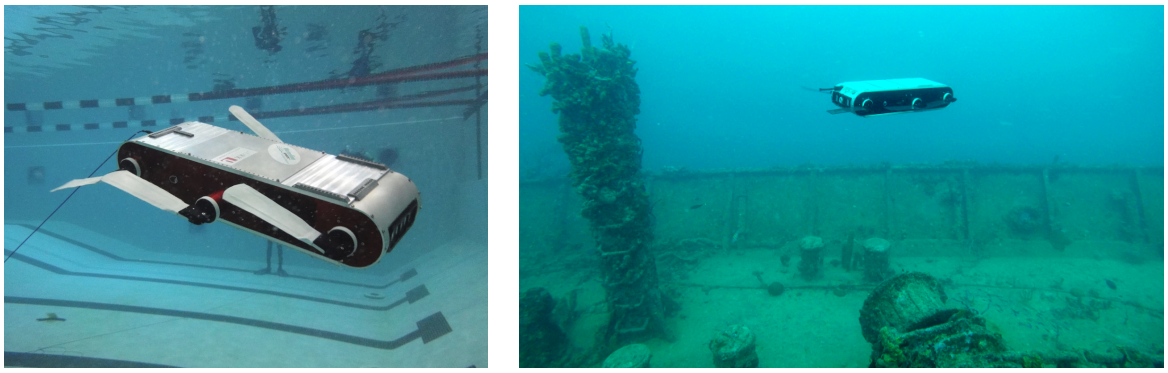
Underwater Coverage

5.1 Introduction

Mapping and inspection of underwater structures is an important task found in environmental studies, marine archaeology, resource acquisition, and infrastructure maintenance, that potentially be performed by underwater vehicles(Figure 5.1b). To inspect a structure, an underwater vehicle has to move over most (ideally all) of the region of interest, thus performing coverage.

The coverage problem in an underwater environment inherits all the challenges faced by planning for an aerial robot in three dimensions, due to the increased dimensionality of the planning space compared to the planning in planar environments. The coverage path planning problem's complexity exponentially increases when moving from two to three dimensions. In addition, the underwater environment presents novel challenges both from the coverage and the navigation perspective. Underwater vision presents unique challenges such as hazing, color attenuation [107, 120], and lack of good features [116, 99]. The dynamics of the water [80] and visibility constraints contribute to instability, drifting, and error in the localization of an autonomous underwater vehicle (AUV); for details on the challenges of underwater sensing please refer to the comparison studies in [51, 69, 99].

The focus of this work is visual mapping of a shipwreck. Historical shipwrecks tell an important part of history and at the same time have a special allure for most humans, as exemplified by the plethora of movies and artworks of the Titanic; see for



(a) (b)

Figure 5.1: (a) AQUA underwater robot in a pool. (b) AQUA performing coverage over Stavronikita Shipwreck in Barbados

example the work of [26] for the visual mapping of the Titanic. Shipwrecks are also one of the top scuba diving attractions all over the world. The historical shipwrecks are deteriorating due to warm, salt water, human interference, and in some places extreme weather (frequent tropical storms). Reconstructing accurate models of these sites will be of high importance not only for the historical study of the shipwrecks, but also for monitoring subsequent deterioration [81, 83]; see Figure 5.2 for the different floors of a shipwreck exposed after a partial collapse.

Currently, limited mapping efforts are performed by divers who take measurements manually using a grid and measuring tape, or using hand-held sensors [41] — a slow and sometimes dangerous task; see Figure 5.3 for a diver collecting data, manually, to be used in training the navigation model. While acoustic sensing (SONAR) is common, the resulting maps do not contain the details vision can provide. Vision has been utilized successfully to map underwater structures [43] or even underwater caves [102]. This method presents a methodology for training a learning system to guide an underwater vehicle covering a shipwreck using only vision. The aim of the trained system is to perform coverage in a completely unknown space by learning a set of motion strategies based on the previous visual observations of different environments.

Figure 5.2: The different levels of the Stavronikita shipwreck, Barbados, after a partial collapse.

To perform coverage or reconstruction of 3D structures, reliable state estimation is traditionally required [10]. But even with ideal state estimation, performing coverage in an unknown underwater environment exposes new and very different challenges. The aim of this work is to answer the question: Is it possible to achieve autonomous behavior that will provide meaningful coverage decisions relying only on vision? In contrast to the works presented in the literature on 3D coverage [34], this work is solely based on vision and does not rely on state estimation nor a map of the environment for navigation.

The main contribution of this chapter is a deep learning framework for learning the motions for navigating around a shipwreck. The proposed system trained on data collected by human operators learns to guide an autonomous system in a similar manner around different shipwrecks using only vision. Results demonstrate the accuracy of the learned system in validation datasets.

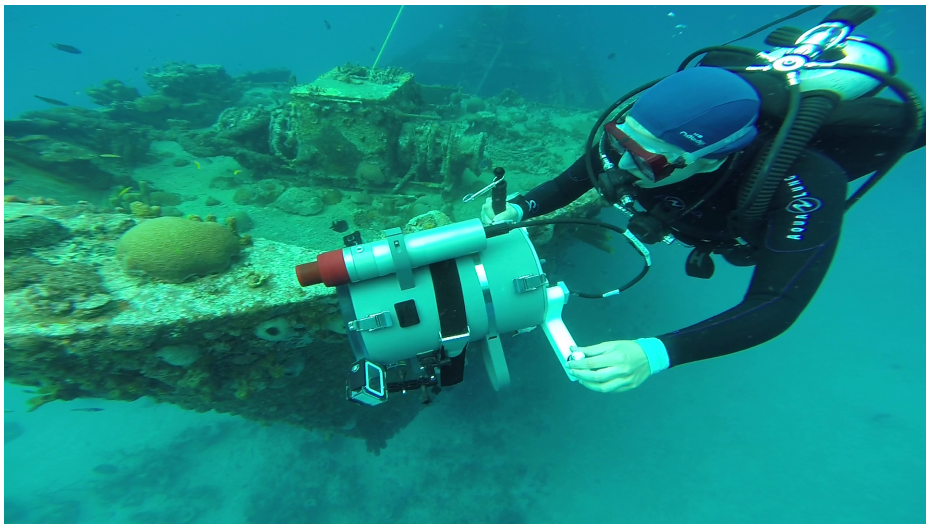


Figure 5.3: A diver collecting data over Pamir Shipwreck, Barbados.

5.2 Problem Statement

In general terms, the coverage path planning problem (CPP) is concerned with finding a trajectory that moves an end-effector / sensor of the robot over each available point of the environment (area or volume) while avoiding obstacles. The target platform in this work is an autonomous tether-less robot with six degrees of freedom from which controllable are the three orientations — yaw (rotation around the z-axis), pitch (rotation around the y-axis), and roll (rotation around the x-axis) in a robot-centric frame of reference — with up and down (depth change) and forward motions. There is no prior information about the environment and it is deployed in a 3D-bounded area of interest $\mathcal{E} \subset \mathbb{R}^3$. The robot follows a path π that will result in the acquisition of a sequence of camera frames $V = \{f_1, f_2, ..., f_n\}$, where f_i is the i-th image frame. The objective of the coverage path planning problem is to ensure that π is obstacle free and that the 3D reconstruction resulting from V covers the entire surface of the $\mathcal{E}$ object.

The objective of this work is to build a system that will use only the current frame V to guide the robot and build a path (π), in such a way that it will imitate

the behaviors of a human diver when surveying a shipwreck. As such the main strategies, that will be the building blocks for coverage path planning, considered within this work, are:

1. keep the shipwreck in the field of view,
2. follow the shipwreck's side,
3. turn around the bow and stern of the ship,
4. circumnavigate the mast.

The ultimate goal will be to ensure all of these strategies are executed at the same time, but within this thesis our aim is to achieve these behaviors one at a time.

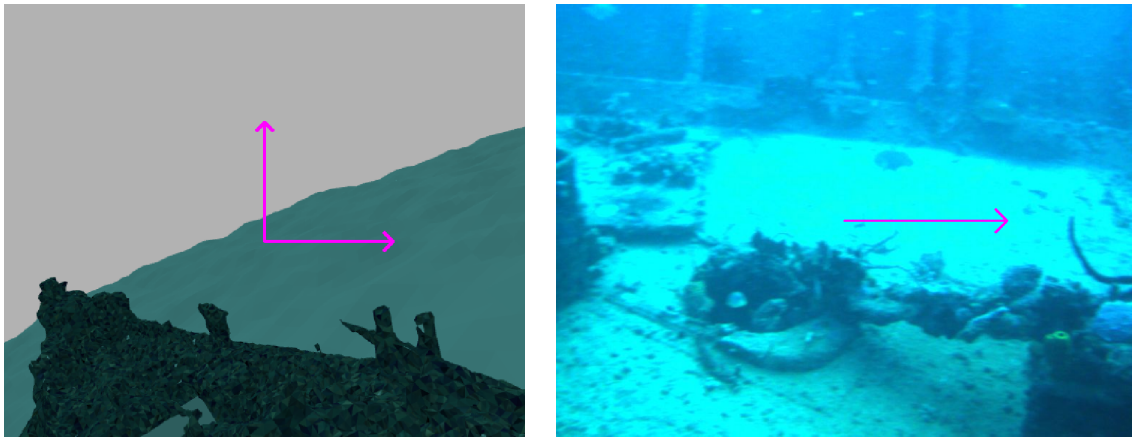
5.3 Towards Coverage Path Planning of Shipwrecks

To create a vision based navigation system we need to have a large dataset of different shipwrecks. We begin with 3D meshes of shipwrecks - the data consists of Gazebo [64] models of shipwrecks provided by the National Oceanic and Atmospheric Administration (NOAA)[1]. In addition, we have generated test data from the coverage of the Stavronikita shipwreck in Barbados where we have collected shipwreck images utilizing an underwater Aqua2 robot [23], a stereo-rig sensor (see A.2), and a GoPro camera. The Aqua2 vehicle is capable of autonomous operations [111] up to a depth of 30 meters.

A diver was asked to label data based on the action that they would take if they were to perform coverage around a shipwreck. The possible values that the diver selects are the directions in a 2D image - the label window is illustrated in Figure 5.4. The labeling performed is based on a diver's strategy of data collection, consistent with the above-mentioned four behavioral strategies. Following the same strategies,

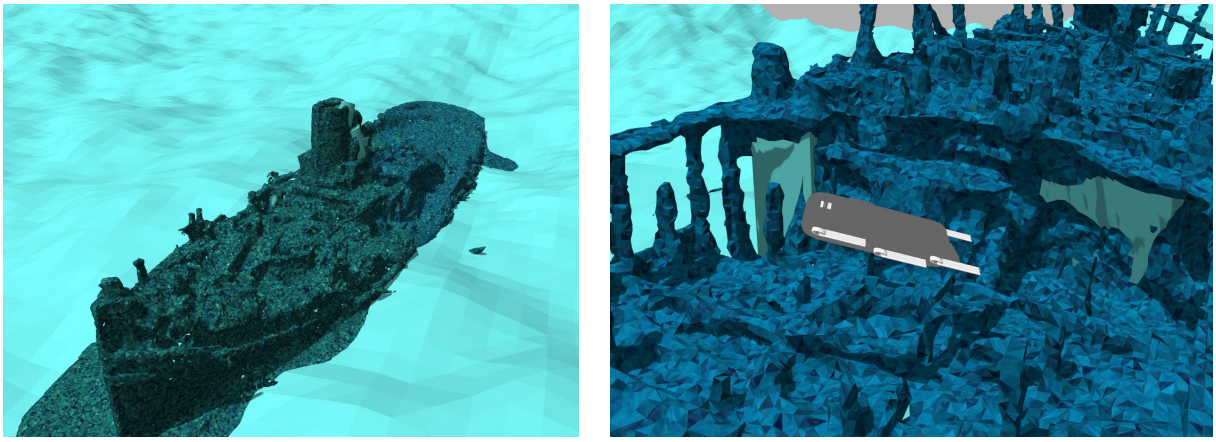
[1] `https://nauticalcharts.noaa.gov/data/wrecks-and-obstructions.html`

simulation data has been collected by driving an Aqua2 robot in the Gazebo simulator and recording the yaw and pitch changes corresponding to the current camera view (Figure 5.5).



(a) (b)

Figure 5.4: The labeled data, showing the desired change in orientation along the yaw and pitch angles: (a) The gazebo simulation, presenting changes in both the yaw and pitch angles; (b) The underwater video data, where there is change only along the yaw angle.



(a) (b)

Figure 5.5: (a) Gazebo Model of a Schipwreck used for training; (b) Aqua2 robot navigating over shipwreck in Gazebo.

The labeled data is fed to an 18 layer residual network with similar architecture to the one proposed by [121] and adopted later by Manderson et al. [72] (see Figure 5.6).

The network is a variation of standard ResNet-18 [40]. The first convolution layer has 7x7 and the rest are 3x3, all are downsized using strides of 2. After each convolutional layer dropout layer is added with 0.2 rate. To avoid the dead neuron problem, we use the LeakyReLU activation function. We used the Momentum gradient descent optimizer with a learning rate of 0.001 and a momentum of 0.1. Similar to Smolyanskiy et al. [121], we also treat this as a classification problem rather than regression, to ensure we have a consistent data labelling process. As such, similar to Manderson et al. [72], the change in yaw and pitch have seven possible values for each. Thus our network operates on each frame of the incoming video, classifying each image into one of possible 49 classes. Each of these classes consists of a different yaw/pitch command composed of two integers in the [-3,3] range. The classification indicates the chosen action for the robot. After training the network classification output on the input images is directly used as control commands of the Aqua2 robot.

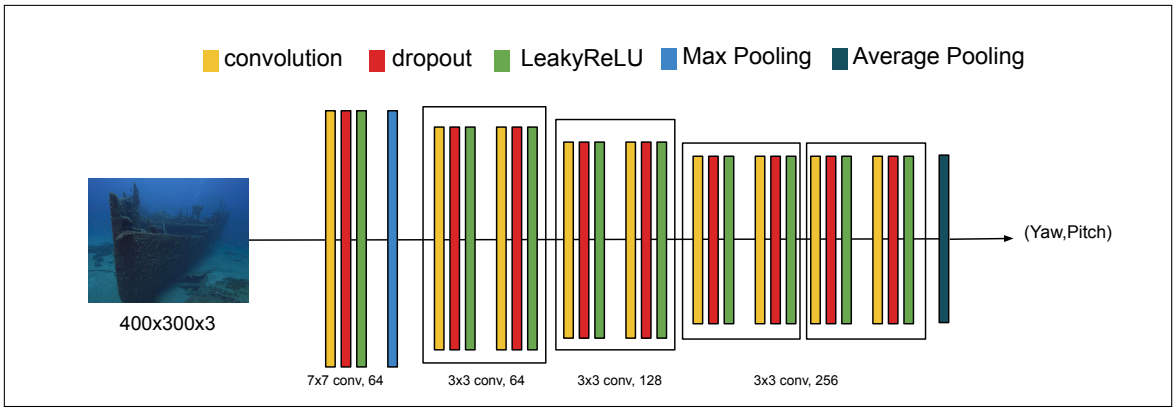


Figure 5.6: The overview of neural network architecture.

In training, the network used a batch size of 32 and a dropout rate of 0.2 over 2048 epochs, and a stochastic gradient descent method optimized the network with a categorical cross entropy loss function. These variables were tuned until the expected behavior could be observed. We ended up with a system with 10 layers that did not result in data overfitting (see Figure 5.7) and achieved increased accuracy in both

the testing and the validation of data. The output of the system consists of direction commands that are then converted to the yaw, pitch and roll commands to control the Aqua2 robot.

5.4 Experiments

The experiments were performed using the Aqua2 simulator in Gazebo. It emulates the real dynamics of an underwater environment and allows control of the robot. The Aqua2 robot used in simulation uses the motion from six flippers, each independently actuated by an electric motor, to swim. It has 6 degrees of freedom, of which five are controllable. The robot's primary sensing modality is vision. It is equipped with three iDS USB 3.0 UEye cameras: two facing forward and one in the back. The front-facing cameras are used for navigation and data collection.

The training has been performed separately on simulation data and real world data, as well as on combined data samples. Both simulation and real world datasets consisted of over 25,000 labeled images. The simulation dataset was collected by driving Aqua2 in Gazebo using the keyboard, which produced automatically labeled data. Data from the Stavronikita shipwreck was collected with handheld cameras and later was labeled by a human with the above mentioned approach.

The focus of this chapter is on answering the question: can a deep neural network learn the motion controls of an underwater vehicle using human driven commands informed solely by the visual input? A key observation is that during training with a simulator, the human user only sees the view from the robot's camera, while in the footage taken by a human diver the diver has higher awareness as they swim around the structure. In order to answer the above question, we evaluate how well the learned system is able to predict the motions similarly to the human annotations. Thus, within this work, we are concerned with the performance of the classifier and

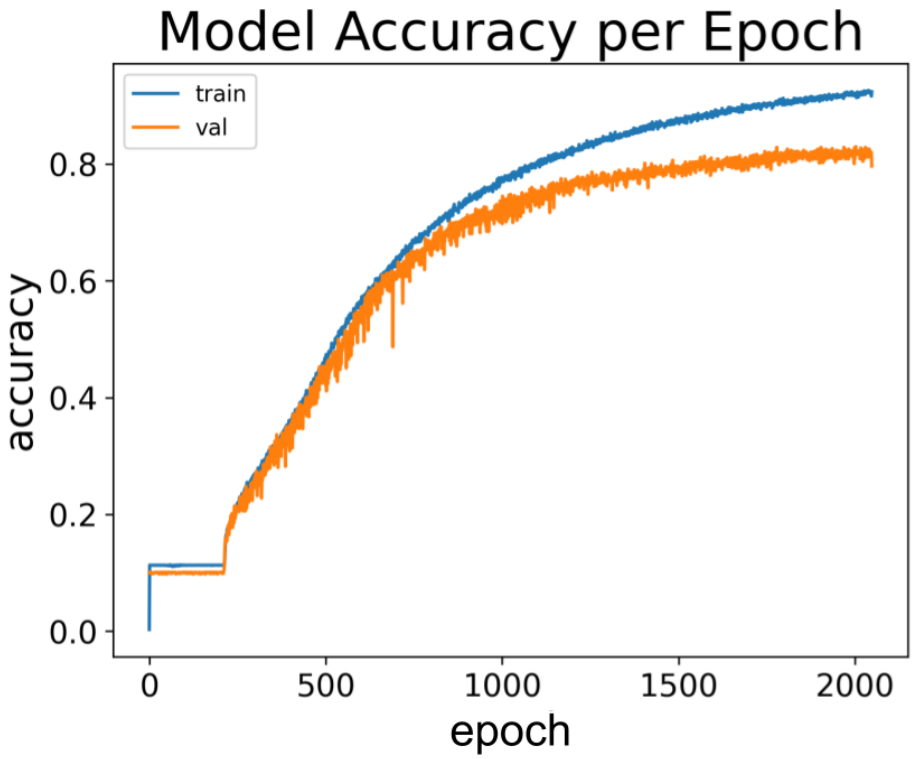


Figure 5.7: The accuracy per epoch plot of the proposed method trained only on real data.

how different datasets affect the prediction accuracy. The training data, model test and validation test data was selected with a proportion of $6 : 2 : 2$.

The model accuracy results using different datasets are depicted in Table 5.1. We performed separate training using only simulation data, then only real world data collected from shipwrecks and finally we performed training on combined data. Test accuracy is reported on the same type of data, whereas additionally we performed validation tests using a model trained in a real environment on simulation and vice versa. We also report the number of epochs after which the model is converging, e.g. when the more iterations of training do not result in improvement of model. When the CNN is trained on separate datasets the validation accuracy converges close to 80%; see Figure 5.7. In contrast, when the training happens on the combined dataset

the validation accuracy drops to 34% (see Table 5.1). This discrepancy between simulation and real world datasets is indicative of the different visual appearances between the simulator and the real world. In future work, the simulation data will be enhanced by using generative adversarial networks (GAN) [37] to resemble the real world imagery, similar to the approach used for training vision based cooperative localization [50]. It is worth noting that, in the above-mentioned model, a separate GAN was necessary for pool and ocean images.

Nevertheless, the quantitative results on separate datasets highlight the feasibility of the method. The resulting control commands have been used to navigate the Aqua2 robot in a Gazebo simulation over different shipwreck models. From the earlier mentioned target behaviors, the Aqua2 performed in simulation the "keeping the shipwreck in the field of view" and "following the side of shipwreck" strategies (see Figure 5.8), which are critical for the majority of the exploration. These experiments demonstrate the capabilities of the proposed system which is trained only on one real shipwreck dataset using vision as the only sensor.

Table 5.1: Comparison of Accuracy of training using Simulation and Real world data, and the convergence speed (accuracy values are approximations). Note, the validation is given for combined data, hence reported as the same.

Training Dataset	Simulation Based	Real World Collected	Simulation and Real World
Validation Accuracy on Simulation	82%	12%	34%
Validation Accuracy on Real World	6%	80%	34%
Test Accuracy	78%	80%	47%
Convergence on Epochs	500	1000	1000
Size of Complete Dataset	29180	26817	56097

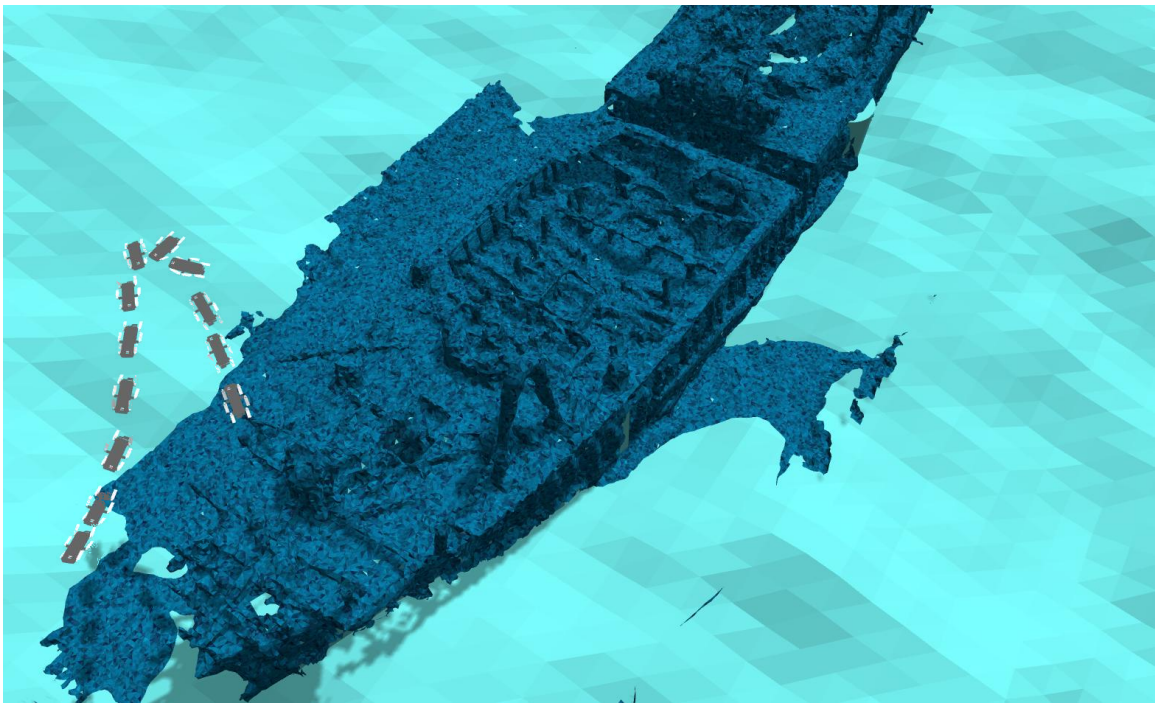


Figure 5.8: Portion of a trajectory of the robot in simulation produced by a prediction based controller.

Chapter 6

Conclusions

This chapter summarizes the contributions of this dissertation, with short overviews of each chapter in Section 6.1. Following, Section 6.2 details some preliminary findings and highlights possible future directions of research in environmental area coverage.

6.1 Summary

This thesis addresses area coverage problems in aquatic environments. It addresses the problem in 2D for monitoring rivers, in large-scale operations with applications in lake monitoring using multiple robots and finally in 3D for mapping underwater structures. The dissertation started with a comprehensive overview of the literature presented in Chapter 2, which discussed the coverage path planning problem from different perspectives, for two dimensional and three dimensional areas. Multi-robot area coverage problems were also discussed. Of particular interest for this thesis was also to present advances in the area coverage field for environmental monitoring and exploration in aquatic environments. In addition, vision based navigation was briefly discussed to motivate the proposed vision based coverage approach. The presented literature review highlighted the importance of this thesis.

In Chapter 3, different strategies were presented for performing partial and complete coverage. Both complete coverage algorithms outperform boustrophedon coverage. L-cover performs coverage parallel to the shores of the river and takes into account the width of the river for generating the passes, while T-cover performs coverage perpendicular to the shores of the river. The M-cover algorithm takes into

account meander-based velocity difference of the water current, proved to be a more efficient approach for longitudinal coverage. The Z-cover algorithm shows improvements over the fixed-angle based approach used in practice by scientists for manual survey operations. The performance of the algorithms is validated on a large number of river maps with different size and shape of shores. In addition, the algorithms were tested in simulation and in the real world. The field trials were performed on 40 000m^2 and 10 000m^2 regions of the Congaree River, SC.

Chapter 4 presents a novel approach for multi-robot coverage utilizing multiple ASVs governed by Dubins vehicle kinematics. Two algorithms were discussed, both of which extend our previous work on efficient multi-robot coverage to accept for Dubins constraints. Additionally, an experimental comparison of two algorithms presented in our previous work without Dubins constraints were performed. As a result we show in this work that further clustering of the area ensures 100% utilization of the robots when Dubins coverage methods are used. The experiments assert that both algorithms result in almost optimal solutions. Nevertheless, the DCRC algorithm demonstrated a slight advantage over the DCAC algorithm in terms of coverage cost. As a result, our algorithm of choice was DCRC for performing field trials on Lake Murray, SC, USA.

Finally, we concluded with three dimensional coverage presented in Chapter 5. With this work, we proposed a new approach to tackle underwater navigation around complex structures based solely on visual input. The method is based on the diver's expertise in performing navigation while collecting data around a structure. Reported preliminary results of 80% accuracy achieved from training on separate sets of simulated and real underwater data showed that this approach is feasible and capable of performing some of the predefined behaviors. Thus, this is a step towards a vision-based navigation system for coverage of shipwreck structures.

6.2 Preliminary Applications

The area coverage path planning problem has a large spectrum of applications. In this dissertation, our main emphasis was geared towards environmental monitoring and surveying in the aquatic domain. This field represents a rich source of research for coverage path planning. One interesting and important application for coverage path planning is sampling lakes with ASVs(Figure 6.2a) to measure water quality, with a focus on detecting and predicting the spread of Harmful Cyanobacterial Blooms (HCBs)(Figure 6.1). In such applications, coverage patterns will be constrained by the sensor setup, the possible predictive pattern of the spread of the bloom or the morphology of the environment.

(a)

Figure 6.1: Harmful algal bloom in the western basin of Lake Erie.[1]

6.2.1 Lake Monitoring Setup and Basic Pattern Comparison

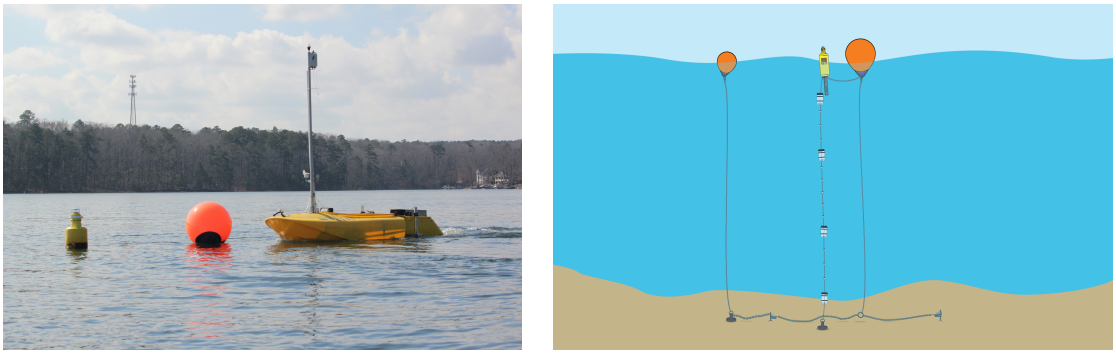
An initial multi-modal monitoring system has been set up for monitoring Harmful Algal Blooms (HABs) on the surface of fresh waters, such as lakes and reservoirs.

[1]A European Space Agency (ESA) Envisat satellite image taken on Oct. 8, 2011, using its MERIS sensor.

Table 6.1: Comparison of the coverage metrics for different ASV patterns. Spiral and Bsd refer to spiral and boustrophedon patterns covering an 100 m×100 m area, while LSpiral and LBsd refer to the same patterns covering an 151 m×151 m area respectively.

	Spiral	Bsd	LSpiral	LBsd
Time	6m 29s	7m 46s	15m 58s	17m 55s
Length	745 m	869 m	1915 m	2173 m
Area	$7451m^2$	$8689m^2$	$19150m^2$	$21730m^2$

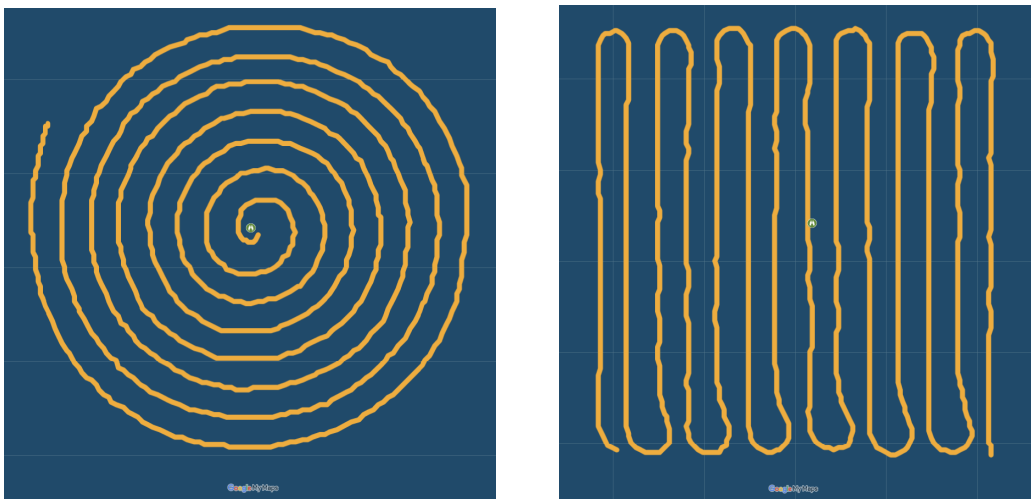
The stations utilize the miniDOT [97] sensor in conjunction with an anti-fouling wiper [96] to measure dissolved oxygen and temperature (Figure 6.2). Additionally, miniature one-channel temperature data loggers (HOBO 64K Pendant sensors from Onset) [87] are also deployed. The YSI EMM25 buoy is used to secure the string of sensors on one end, at the other end a 4.53 kg mushroom anchor keeps the line from drifting(Figure 6.2b). In addition to the sensor station, the ASV has been equipped with a YSI EXO2 multiparameter sonde [140] to collect water quality samples near the surface (at 0.5 m depth) utilizing different (horizontal) sampling patterns (Figure 6.3).



(a) (b)

Figure 6.2: (a) ASV sampling next to the deep water station at Lake Murray. (b) The station setup ensures stability, even during extreme weather events.[2]

[2]Credit: Gevorg Dallkyan, https://sites.google.com/view/dallage



(a) (b)

Figure 6.3: GPS coordinates of deployed patterns recorded by PixHawk with the deep station in the center (green landmark): (a) Spiral and (b) Boustrophedon.

Different trajectories are employed in order to guide the ASV. Of particular interest are the regions around the buoy deployments, across each lake, and the data variance near shore and in the middle of the lake. Two different strategies, a spiral pattern and a boustrophedon pattern have been utilized to investigate the variation in the chlorophyll values around the testing stations; see Figure 6.3 for two sample trajectories along. The patterns have been deployed several times on 100 m×100 m and 151 m×151 m areas around the deep station with an ASV's sensor footprint of 10 m (Figure 6.3). A comparison of the distance traveled and area covered in the two patterns is presented in Table 6.1. The experiments show that there is only a slight difference between boustrophedon and spiral patterns in terms of distance traveled, area covered and time. This can be explained by the fact that, in a curved trajectory, what is gained in the wider end is lost from the narrower end. For example, in the coverage of a part of an annulus, the inner circle is shorter by the same amount that the outer circle is longer, as compared to the middle. Note when deploying to cover a small area around a point of interest, the spiral pattern is more applicable. When

a rectangular patch needs to be covered, given the ASV turning radius constraints, for smaller footprints, the Dubins vehicle coverage method is required [67].

With the initial setup of the sensor system and comparison of simple coverage patterns, the next step is to integrate those patterns in larger surveying operations — such as an informative sampling strategy with altering spiral and boustrophedon patterns around the locations of interest.

6.2.2 Skeleton Pattern: Towards Better Coverage of Lakes

The large areas, targeted together with the slow varying of parameters in water bodies, make exhaustive complete coverage impractical and unnecessary. In addition, the morphology of the lakes such as Lake Murray, SC (Figure 6.4), with many narrow inlets, can potentially pose difficulties for classical boustrophedon decomposition based methods. We want to explore a set of strategies for the efficient data collection of water quality in such environments when deploying an ASV.

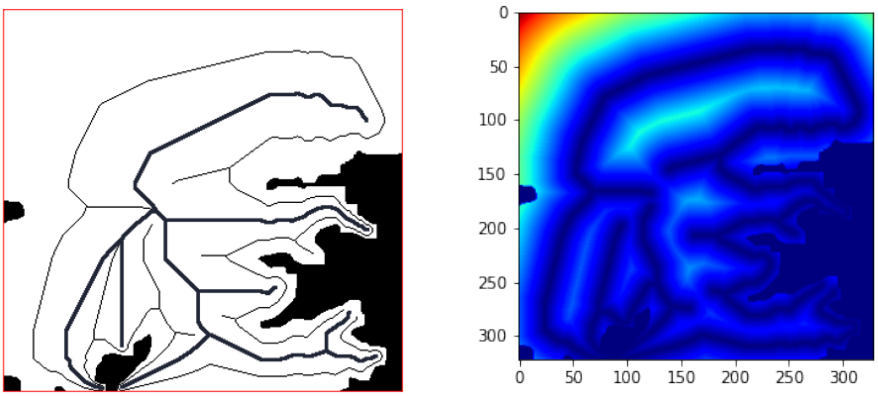
(a)

Figure 6.4: View of Lake Murray's inlets.

A possible approach for generating an ASV trajectory that visits a representative area of the environment can be expressed through utilizing the medial axis of the

water bodies and by pruning redundant branches. This will ensure that each collected water sample is equidistant from the closest shores. Taking this into consideration, we have developed two strategies for performing a coverage with the above-mentioned constraints:

- Skeleton-based trajectory, for maximally reaching all inlets and collecting data that can give an initial representation about the area (Figure 6.5).
- Skeleton-based zig-zag trajectory, for performing more detailed coverage and possibly better than boustrophedon based approaches due to the flexibility of reaching inlets of the lake (Figure 6.6).

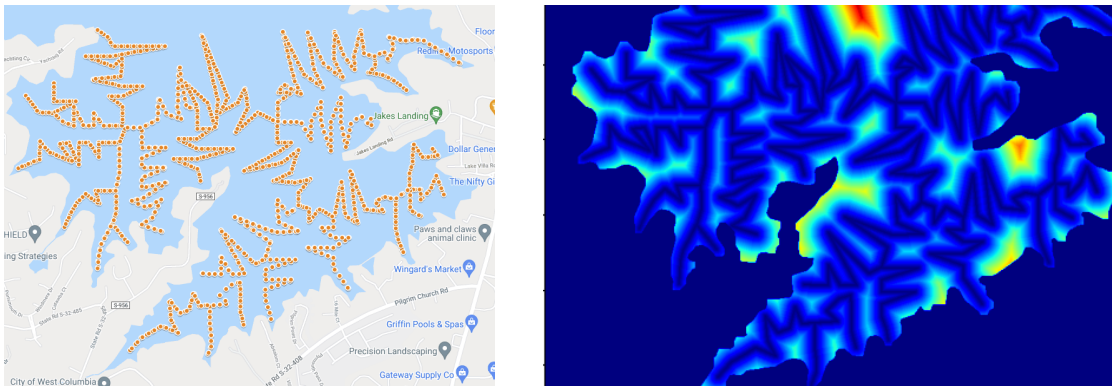


(a) (b)

Figure 6.5: Skeleton-based approach.[3] (a) The trajectory along with the skeleton in thick black. (b) The heatmap of trajectory.

The preliminary results show a possible advantage over the boustrophedon decomposition based (BCD) method (Figure 6.7). When the sensor footprint is large, the BCD-based approach sometimes fails to generate non-overlapping and evenly spaced out passes. Also, since the BCD algorithm has to choose one single coverage direction, it lacks the same ability of the skeleton-based approach to have coverage

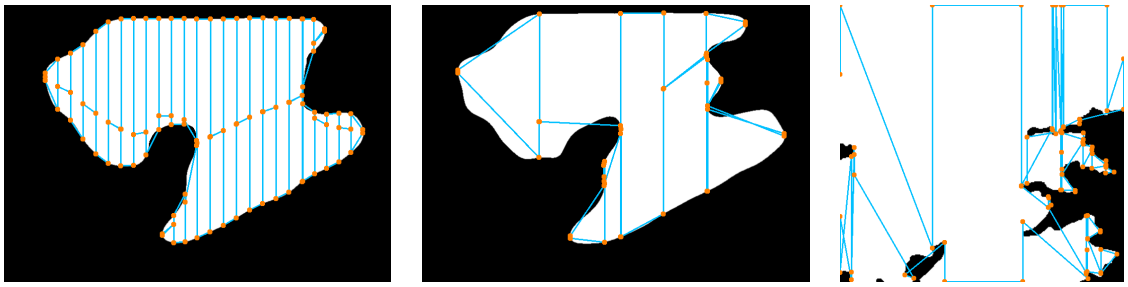
[3]Credit: Jason Raiti for preliminary implementation of methods during Research Undergraduate Experience (REU) summer program.



(a) (b)

Figure 6.6: Zig-zag coverage pattern. (a) Waypoints of the pattern on google maps. (b) The heatmap of the pattern.

trajectories spread out in different directions throughout the area of interest, thus providing better access to the inlets of the lake.



(a) (b) (c)

Figure 6.7: Boustrophedon area decomposition based efficient single robot coverage with smaller sensor footprint in (a) and 4× larger in (b),(c).

More experiments have to be conducted to compare the proposed two new patterns with BCD-based efficient coverage path planning techniques. In addition, they should be also compared against Dubins coverage path planning algorithms. For future consideration, the input to the algorithm should include resource constraints as well, such as time allotment or maximal path length.

6.3 Future Work

Several other aspects of this work are of high importance and can be addressed in future. Taking into account the challenges encountered during the field deployments with autonomous surface vehicles, obstacle avoidance strategies must be implemented for both underwater and above water obstacles. During the multi-robot coverage experiments, in a few instances, two vehicles came too close to each other. We are currently investigating an automated arbitration mechanism following the rules of the sea [126] to avoid collisions.

Another topic that we are interested in for 2D environments is planning the coverage path taking into account a model of the current in the river [80]. This model can be used to associate different cost values depending on the direction of travel with respect to the direction of the current.

Key challenges of the 3D coverage path planning work are related to the complexity of understanding the underwater scene from limited amounts of data. To address this, more restrictions and more systematic data labeling must be performed. Given the extremely challenging circumstances of data collection of underwater shipwrecks, the introduction of GAN-generated artificial training data is necessary for enabling smooth transitions between real and simulated environments.

When working in an underwater domain, we are limited by the technical constraints of the autonomous platform to a greater extent than on the surface. The constraints of underwater vehicles include but are not limited to short battery life, limited computational power and high cost. To successfully execute online learning based methods, such as the one presented in this thesis for 3D coverage, our main underwater AQUA2 robot must be upgraded to include Jetson TX2 Module for computations. In addition, a 3D reconstruction of the underwater structure should be generated using the proposed method and a state of the art 3D coverage method to show qualitative differences.

Bibliography

[1] Ercan U Acar and Howie Choset. "Sensor-based coverage of unknown environments: Incremental construction of morse decompositions". In: *The International Journal of Robotics Research* 21.4 (2002), pp. 345–366.

[2] Ercan U Acar, Howie Choset, Alfred A Rizzi, Prasad N Atkar, and Douglas Hull. "Morse decompositions for coverage tasks". In: *The international journal of robotics research* 21.4 (2002), pp. 331–344.

[3] Noa Agmon, Noam Hazon, and Gal A Kaminka. "The giving tree: constructing trees for efficient offline and online multi-robot coverage". In: *Annals of Mathematics and Artificial Intelligence* 52.2 (2008), pp. 143–168.

[4] Adam Aili and Erik Ekelund. "Model-based design, development, and control of an underwater vehicle". PhD thesis. Master's thesis, Department of Electrical Engineering, Linkoping University ..., 2016.

[5] Prasad N Atkar, Aaron Greenfield, David C Conner, Howie Choset, and Alfred A Rizzi. "Uniform coverage of automotive surface patches". In: *The International Journal of Robotics Research* 24.11 (2005), pp. 883–898.

[6] Gustavo SC Avellar, Guilherme AS Pereira, Luciano CA Pimenta, and Paulo Iscold. "Multi-UAV routing for area coverage and remote sensing with minimum time". In: *Sensors* 15.11 (2015), pp. 27783–27803.

[7] Rik Bähnemann, Nicholas Lawrance, Jen Jen Chung, Michael Pantic, Roland Siegwart, and Juan Nieto. "Revisiting boustrophedon coverage path planning as a generalized traveling salesman problem". In: *Field and Service Robotics*. Springer. 2021, pp. 277–290.

[8] Jose Balbuena, Diego Quiroz, Rui Song, Richard Bucknall, and Francisco Cuellar. "Design and implementation of an USV for large bodies of fresh waters at the highlands of Peru". In: *OCEANS 2017-Anchorage*. IEEE. 2017, pp. 1–8.

[9] Pradeep Bhatta, Edward Fiorelli, Francois Lekien, Naomi Ehrich Leonard, Derek Paley, Fumin Zhang, Ralf Bachmayer, Russ E Davis, David M Fratantoni, and Rodolphe Sepulchre. "Coordination of an underwater glider fleet for

adaptive ocean sampling". In: *Proc. International Workshop on Underwater Robotics, Int. Advanced Robotics Programmed (IARP), Genoa, Italy*. 2005.

[10] Andreas Bircher, Mina Kamel, Kostas Alexis, Helen Oleynikova, and Roland Siegwart. "Receding horizon path planning for 3D exploration and surface inspection". In: *Autonomous Robots* 42.2 (2018), pp. 291–306.

[11] Andreas Bircher, Mina Kamel, Kostas Alexis, Helen Oleynikova, and Roland Siegwart. "Receding horizon" next-best-view" planner for 3d exploration". In: *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2016, pp. 1462–1468.

[12] Wen Yuan Chen, Shih Sung Cheng, Ching Te Wang, Chin Fu Tsai, and Chiou Kou Tung. "The Cleaning Machine Path Design Using Image Recognition Techniques". In: *2014 Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*. IEEE. 2014, pp. 439–442.

[13] Peng Cheng, James Keller, and Vijay Kumar. "Time-optimal UAV trajectory planning for 3D urban structure coverage". In: *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2008, pp. 2750–2757.

[14] Howie Choset. "Coverage for robotics–a survey of recent results". In: *Annals of mathematics and artificial intelligence* 31.1 (2001), pp. 113–126.

[15] Howie Choset. "Coverage of known spaces: The boustrophedon cellular decomposition". In: *Autonomous Robots* 9.3 (2000), pp. 247–253.

[16] Howie Choset and Philippe Pignon. "Coverage path planning: The boustrophedon cellular decomposition". In: *fsr*. 1998.

[17] R Neumann De Carvalho, HA Vidal, P Vieira, and MI Ribeiro. "Complete coverage path planning and guidance for cleaning robots". In: *ISIE'97 Proceeding of the IEEE International Symposium on Industrial Electronics*. Vol. 2. IEEE. 1997, pp. 677–682.

[18] Christian Dornhege and Alexander Kleiner. "A frontier-void-based approach for autonomous exploration in 3d". In: *Advanced Robotics* 27.6 (2013), pp. 459–468.

[19] Christian Dornhege, Alexander Kleiner, Andreas Hertle, and Andreas Kolling. "Multirobot Coverage Search in Three Dimensions". In: *Journal of Field Robotics* 4.33 (2016), pp. 537–558.

[20] Christian Dornhege, Alexander Kleiner, and Andreas Kolling. "Coverage search in 3D". In: *2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. IEEE. 2013, pp. 1–8.

[21] Lester E Dubins. "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents". In: *American Journal of mathematics* 79.3 (1957), pp. 497–516.

[22] Gregory Dudek, Philippe Giguere, Chris Prahacs, Shane Saunderson, Junaed Sattar, Luz-Abril Torres-Mendez, Michael Jenkin, Andrew German, Andrew Hogue, Arlene Ripsman, et al. "Aqua: An amphibious autonomous robot". In: *Computer* 40.1 (2007), pp. 46–53.

[23] Gregory Dudek et al. "A Visually Guided Swimming Robot". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Edmonton AB, Canada, 2005, pp. 1749–1754.

[24] Jack Edmonds and Ellis L Johnson. "Matching, Euler tours and the Chinese postman". In: *Mathematical programming* 5.1 (1973), pp. 88–124.

[25] Albert Einstein. "The cause of the formation of meanders in the courses of rivers and of the so-called Baer's law". In: *Die Naturwissenschaften* 14.11 (1926), pp. 223–224.

[26] Ryan M. Eustice, Hanumant Singh, John J. Leonard, and Matthew R. Walter. "Visually mapping the RMS Titanic: Conservative covariance estimates for SLAM information filters". In: *The Int. Journal of Robotics Research* 25.12 (2006), pp. 1223–1242.

[27] Cheng Fang and Stuart Anstee. "Coverage path planning for harbour seabed surveys using an autonomous underwater vehicle". In: *OCEANS'10 IEEE SYDNEY*. IEEE. 2010, pp. 1–8.

[28] Pooyan Fazli, Alireza Davoodi, Philippe Pasquier, and Alan K Mackworth. "Complete and robust cooperative robot area coverage with limited range". In: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2010, pp. 5577–5582.

[29] Hugo Ferreira, C Almeida, A Martins, J Almeida, N Dias, A Dias, and E Silva. "Autonomous bathymetry for risk assessment with ROAZ robotic surface vehicle". In: *MTS/IEEE Oceans-Europe*. 2009.

[30] Greg N. Frederickson, Matthew S. Hecht, and Chul E. Kim. "Approximation Algorithms for Some Routing Problems". In: *Annual Symposium on Foundations of Computer Science (SFCS)*. IEEE Computer Society, 1976.

[31] Yoav Gabriely and Elon Rimon. “Spanning-tree based coverage of continuous areas by a mobile robot”. In: *Annals of mathematics and artificial intelligence* 31.1 (2001), pp. 77–98.

[32] Enric Galceran, Ricard Campos, Narcís Palomeras, Marc Carreras, and Pere Ridao. “Coverage path planning with realtime replanning for inspection of 3d underwater structures”. In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2014, pp. 6586–6591.

[33] Enric Galceran, Ricard Campos, Narcís Palomeras, David Ribas, Marc Carreras, and Pere Ridao. “Coverage path planning with real-time replanning and surface reconstruction for inspection of three-dimensional underwater structures using autonomous underwater vehicles”. In: *Journal of Field Robotics* 32.7 (2015), pp. 952–983.

[34] Enric Galceran and Marc Carreras. “A survey on coverage path planning for robotics”. In: *Robotics and Autonomous systems* 61.12 (2013), pp. 1258–1276.

[35] Enric Galceran and Marc Carreras. “Efficient seabed coverage path planning for ASVs and AUVs”. In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2012, pp. 88–93.

[36] Scott Glenn, Oscar Schofield, Josh Kohut, Janice McDonnell, Richard Ludescher, Dena Seidel, David Aragon, Tina Haskins, Ethan Handel, Clinton Haldeman, et al. “The Trans-Atlantic Slocum glider expeditions: A catalyst for undergraduate participation in ocean science and technology”. In: *Marine Technology Society Journal* 45.1 (2011), pp. 52–67.

[37] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. “Generative adversarial networks”. In: *Communications of the ACM* 63.11 (2020), pp. 139–144.

[38] Nuno Gracias, Pere Ridao, Rafael Garcia, Javier Escartín, Michel l’Hour, Franca Cibecchini, Ricard Campos, Marc Carreras, David Ribas, Narcís Palomeras, et al. “Mapping the Moon: Using a lightweight AUV to survey the site of the 17th century ship ‘La Lune’”. In: *2013 MTS/IEEE OCEANS-Bergen*. IEEE. 2013, pp. 1–8.

[39] Noam Hazon and Gal A Kaminka. “Redundancy, efficiency and robustness in multi-robot coverage”. In: *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. IEEE. 2005, pp. 735–741.

[40] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[41] Jon Henderson, Oscar Pizarro, Matthew Johnson-Roberson, and Ian Mahon. “Mapping submerged archaeological sites using stereo-vision photogrammetry”. In: *International Journal of Nautical Archaeology* 42.2 (2013), pp. 243–256.

[42] Heraclitus of Ephesus. *As quoted by Plato in Cratylus*. 402a, 535 BC – 475 BC.

[43] Andrew Hogue, Andrew German, and Michael Jenkin. “Underwater environment reconstruction using stereo and inertial data”. In: *IEEE International Conference on Systems, Man and Cybernetics*. IEEE. 2007, pp. 2372–2377.

[44] Geoffrey A Hollinger, Brendan Englot, Franz S Hover, Urbashi Mitra, and Gaurav S Sukhatme. “Active planning for underwater inspection and the benefit of adaptivity”. In: *The International Journal of Robotics Research* 32.1 (2013), pp. 3–18.

[45] Geoffrey A Hollinger and Gaurav S Sukhatme. “Sampling-based robotic information gathering algorithms”. In: *The International Journal of Robotics Research* 33.9 (2014), pp. 1271–1287.

[46] Wesley H Huang. “Optimal line-sweep-based decompositions for coverage algorithms”. In: *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*. Vol. 1. IEEE. 2001, pp. 27–32.

[47] Sezal Jain, Stephen Nuske, Andrew Chambers, Luke Yoder, Hugh Cover, Lyle Chamberlain, Sebastian Scherer, and Sanjiv Singh. “Autonomous river exploration”. In: *Field and Service Robotics*. Springer. 2015.

[48] Alireza Janani, Lyuba Alboul, and Jacques Penders. “Multi Robot Cooperative Area Coverage, Case Study: Spraying”. In: *Towards Autonomous Robotic Systems: 17th Annual Conf., TAROS*. Ed. by Lyuba Alboul, Dana Damian, and M. Jonathan Aitken. Springer Int. Publishing, 2016.

[49] Alireza Janani, Lyuba Alboul, and Jacques Penders. “Multi-agent cooperative area coverage: Case study ploughing”. In: *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*. 2016, pp. 1397–1398.

[50] Bharat Joshi, Md Modasshir, Travis Manderson, Hunter Damron, Marios Xanthidis, Alberto Quattrini Li, Ioannis Rekleitis, and Gregory Dudek. “Deep-

URL: Deep Pose Estimation Framework for Underwater Relative Localization". In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2020, pp. 1777–1784.

[51] Bharat Joshi, Sharmin Rahman, Michail Kalaitzakis, Brennan Cain, James Johnson, Marios Xanthidis, Nare Karapetyan, Alan Hernandez, Alberto Quattrini Li, Nikolaos Vitzilaios, et al. "Experimental comparison of open source visual-inertial-based state estimation algorithms in the underwater domain". In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2019, pp. 7227–7233.

[52] Nare Karapetyan. "Multi-Robot Area Coverage". [Masters thesis]. MA thesis. Yerevan, Armenia: College of Science and Engineering (CSE), American University of Armenia, May 2015.

[53] Nare Karapetyan, Kelly Benson, Chris McKinney, Perouz Taslakian, and Ioannis Rekleitis. "Efficient multi-robot coverage of a known environment". In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2017, pp. 1846–1852.

[54] Nare Karapetyan, Adam Braude, Jason Moulton, Joshua A Burstein, Scott White, Jason M O'Kane, and Ioannis Rekleitis. "Riverine coverage with an autonomous surface vehicle over known environments". In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2019, pp. 3098–3104.

[55] Nare Karapetyan, James V Johnson, and Ioannis Rekleitis. "Coverage Path Planning for Mapping of Underwater Structures". In: *Global Oceans 2020: Singapore–US Gulf Coast*. IEEE. 2020, pp. 1–6.

[56] Nare Karapetyan, James V Johnson, and Ioannis Rekleitis. "Human Diver-Inspired Visual Navigation: Towards Coverage Path Planning of Shipwrecks". In: *Marine Technology Society Journal* 55.4 (2021), pp. 24–32.

[57] Nare Karapetyan, Jason Moulton, Jeremy S Lewis, Alberto Quattrini Li, Jason M O'Kane, and Ioannis Rekleitis. "Multi-robot dubins coverage with autonomous surface vehicles". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 2373–2379.

[58] Nare Karapetyan, Jason Moulton, and Ioannis Rekleitis. "Dynamic autonomous surface vehicle control and applications in environmental monitoring". In: *OCEANS 2019 MTS/IEEE SEATTLE*. IEEE. 2019.

[59] Nare Karapetyan, Jason Moulton, and Ioannis Rekleitis. “Meander-Based River Coverage by an Autonomous Surface Vehicle”. In: *Field and Service Robotics*. Springer. 2021, pp. 353–364.

[60] Amna Khan, Iram Noreen, Hyejeong Ryu, Nakju Lett Doh, and Zulfiqar Habib. “Online complete coverage path planning using two-way proximity search”. In: *Intelligent Service Robotics* 10.3 (2017), pp. 229–240.

[61] Peter Kimball, John Bailey, Sarah Das, Rocky Geyer, Trevor Harrison, Clay Kunz, Kevin Manganini, Ken Mankoff, Katie Samuelson, Thomas Sayre-McCord, et al. “The whoi jetyak: An autonomous surface vehicle for oceanographic research in shallow or dangerous waters”. In: *2014 IEEE/OES Autonomous Underwater Vehicles (AUV)*. IEEE. 2014, pp. 1–7.

[62] Alexander Kleiner, Rodrigo Baravalle, Andreas Kolling, Pablo Pilotti, and Mario Munich. “A solution to room-by-room coverage for autonomous cleaning robots”. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2017, pp. 5346–5352.

[63] Jens Kober, J Andrew Bagnell, and Jan Peters. “Reinforcement learning in robotics: A survey”. In: *The International Journal of Robotics Research* 32.11 (2013), pp. 1238–1274.

[64] Nathan Koenig and Andrew Howard. “Design and use paradigms for gazebo, an open-source multi-robot simulator”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Vol. 3. IEEE. 2004, pp. 2149–2154.

[65] Andrew Kwok and Sonia Martínez. “A coverage algorithm for drifters in a river environment”. In: *Proceedings of the 2010 American Control Conference*. IEEE. 2010, pp. 6436–6441.

[66] Andrew Kwok and Sonia Martínez. “Deployment of drifters in a piecewise-constant flow environment”. In: *49th IEEE Conference on Decision and Control (CDC)*. IEEE. 2010, pp. 6584–6589.

[67] Jeremy S Lewis, William Edwards, Kelly Benson, Ioannis Rekleitis, and Jason M O’Kane. “Semi-boustrophedon coverage with a dubins vehicle”. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2017, pp. 5630–5637.

[68] Jeremy S Lewis, Daniel A Feshbach, and Jason M O’Kane. “Guaranteed coverage with a blind unreliable robot”. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 7383–7390.

[69] Alberto Quattrini Li, Adem Coskun, Sean M Doherty, Shervin Ghasemlou, Apoorv S Jagtap, Md Modasshir, Sharmin Rahman, A Singh, Marios Xanthidis, Jason M O'Kane, et al. "Experimental comparison of open source vision-based state estimation algorithms". In: *International Symposium on Experimental Robotics*. Springer. 2016, pp. 775–786.

[70] Randy Mackay. *ArduPilot*. `https://github.com/ArduPilot/ardupilot_wiki/blob/master/rover/source/docs/gettit.rst`. 2016.

[71] Travis Manderson, Juan Camilo Gamboa Higuera, Stefan Wapnick, Jean-François Tremblay, Florian Shkurti, Dave Meger, and Gregory Dudek. "Vision-based goal-conditioned policies for underwater navigation in the presence of obstacles". In: *Robotics: Science and Systems XVI* (2020).

[72] Travis Manderson, Juan Camilo Gamboa Higuera, Ran Cheng, and Gregory Dudek. "Vision-based autonomous underwater swimming in dense coral for combined collision avoidance and target selection". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 1885–1891.

[73] Sandeep Manjanna, Nikhil Kakodkar, Malika Meghjani, and Gregory Dudek. "Efficient terrain driven coral coverage using gaussian processes for mosaic synthesis". In: *2016 13th Conference on Computer and Robot Vision (CRV)*. IEEE. 2016, pp. 448–455.

[74] Sandeep Manjanna, Alberto Quattrini Li, Ryan N Smith, Ioannis Rekleitis, and Gregory Dudek. "Heterogeneous multi-robot system for exploration and strategic water sampling". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 4873–4880.

[75] Raphael Mannadiar and Ioannis Rekleitis. "Optimal coverage of a known arbitrary environment". In: *2010 IEEE International conference on robotics and automation*. IEEE. 2010, pp. 5525–5530.

[76] Wang Meiting, Tan Shili, Ding Junjian, and Yan Liwen. "Complete coverage path planning of wall-cleaning robot using visual sensor". In: *2007 8th International Conference on Electronic Measurement and Instruments*. IEEE. 2007, pp. 4–159.

[77] Teodor Mihai Moldovan and Pieter Abbeel. "Safe exploration in Markov decision processes". In: *Proceedings of the 29th International Coference on International Conference on Machine Learning*. 2012, pp. 1451–1458.

[78] Jason Moulton, Nare Karapetyan, Sharon Bukhsbaum, Chris McKinney, Sharaf Malebary, George Sophocleous, Alberto Quattrini Li, and Ioannis Rekleitis.

"An Autonomous Surface Vehicle for Long Term Operations". In: *OCEANS 2018 MTS/IEEE Charleston*. IEEE. 2018, pp. 1–10.

[79] Jason Moulton, Nare Karapetyan, Michail Kalaitzakis, Alberto Quattrini Li, Nikolaos Vitzilaios, and Ioannis Rekleitis. "Dynamic Autonomous Surface Vehicle Controls Under Changing Environmental Forces". In: *Field and Service Robotics*. Springer. 2021, pp. 381–394.

[80] Jason Moulton, Nare Karapetyan, Alberto Quattrini Li, and Ioannis Rekleitis. "External force field modeling for autonomous surface vehicles". In: *International Symposium on Experimental Robotics*. Springer. 2018, pp. 328–338.

[81] National Park Service – U.S. Department of the Interior. *Abandoned Shipwreck Act Guidelines*. `https://www.nps.gov/archeology/submerged/document.htm`. (Accessed 02/25/2020). 2020.

[82] Kaguchwa John Njenga, JK Kwanza, and Patricia Wanjiru Gathia. "Velocity distributions and Meander Formation of River Channels". In: *International Journal of Applied* 2.9 (2012).

[83] NOAA National Marine Sanctuaries. *The Worldś Underwater Cultural Heritage*. `https://sanctuaries.noaa.gov/science/monitoring/mi_mnms.html`. (Accessed 02/25/2020). 2020.

[84] Contributors of nuttx.org. *NuttX Real-Time Operating System — NuttX Real-Time Operating System*. [Online; accessed 14-July-2018]. 2018. URL: `http://nuttx.org/doku.php?id=nuttx&rev=1531430789`.

[85] Jerome Ny, Eric Feron, and Emilio Frazzoli. "On the Dubins traveling salesman problem". In: *IEEE Transactions on Automatic Control* 57.1 (2011), pp. 265–270.

[86] Jason M. O'Kane. *A Gentle Introduction to ROS*. Available at `http://www.cse.sc.edu/~jokane/agitr/`. Independently published, 2013. ISBN: 978-1492143239.

[87] Onset Computer Corporation. *HOBO 64K Pendant® Temperature/Alarm (Waterproof) Data Logger*. `https://www.pme.com/products/miniwiper`. 2021.

[88] Ottar L Osen, Rolf-Inge Sandvik, Vegard Rogne, Houxiang Zhang, et al. "A novel low cost ROV for aquaculture application". In: *OCEANS 2017-Anchorage*. IEEE. 2017, pp. 1–7.

[89] Anıl Özdemir, Melvin Gauci, Andreas Kolling, Matthew D Hall, and Roderich Groß. "Spatial Coverage Without Computation". In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 9674–9680.

[90] Narcís Palomeras, Natalia Hurts, Eduard Vidal, and Marc Carreras. "Autonomous Exploration of Complex Underwater Environments Using a Probabilistic Next-Best-View Planner". In: *IEEE Robotics and Automation Letters* 4.2 (2019), pp. 1619–1625.

[91] Liam Paull, Sajad Saeedi, Mae Seto, and Howard Li. "Sensor-driven online coverage planning for autonomous underwater vehicles". In: *IEEE/ASME Transactions on Mechatronics* 18.6 (2012), pp. 1827–1838.

[92] Liam Paull, Mae Seto, and Howard Li. "Area coverage planning that accounts for pose uncertainty with an AUV seabed surveying application". In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2014, pp. 6592–6599.

[93] Ehud Peless, Shai Abramson, Ronen Friedman, and Ilan Peleg. *Area Coverage With An Autonomous Robot*. US Patent App. 12/054,123. 2008.

[94] Mark Pfeiffer, Samarth Shukla, Matteo Turchetta, Cesar Cadena, Andreas Krause, Roland Siegwart, and Juan Nieto. "Reinforced imitation: Sample efficient deep reinforcement learning for mapless navigation by leveraging prior demonstrations". In: *IEEE Robotics and Automation Letters* 3.4 (2018), pp. 4423–4430.

[95] *Ping DSP Inc.* `http://www.pingdsp.com/3DSS-DX-450`. Accessed: 2018-05-11.

[96] PME Inc. *An anti-fouling wiper for submersible water sensors*. `https://www.pme.com/products/miniwiper`. 2021.

[97] PME Inc. *miniDOT logger*. `https://www.pme.com/products/minidot`. 2021.

[98] Kai Qin and Dylan A Shell. "Robots going round the bend — a comparative study of estimators for anticipating river meanders". In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2017, pp. 4934–4940.

[99] Alberto Quattrini Li et al. "Vision-Based Shipwreck Mapping: on Evaluating Features Quality and Open Source State Estimation Packages". In: *MTS/IEEE OCEANS - Monterrey*. 2016, pp. 1–10.

[100] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, Andrew Y Ng, et al. "ROS: an open-source Robot Operating System". In: *ICRA workshop on open source software*. Vol. 3. 3.2. Kobe, Japan. 2009, p. 5.

[101] Sharmin Rahman, Nare Karapetyan, Alberto Quattrini Li, and Ioannis Rekleitis. "A Modular Sensor Suite for Underwater Reconstruction". In: *MTS/IEEE OCEANS - Charleston*. IEEE. 2018, pp. 1–6.

[102] Sharmin Rahman, Alberto Quattrini Li, and Ioannis Rekleitis. "Contour based Reconstruction of Underwater Structures Using Sonar, Visual, Inertial, and Depth Sensor". In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2019, pp. 8054–8059.

[103] Ioannis Rekleitis, Ai Peng New, Edward Samuel Rankin, and Howie Choset. "Efficient boustrophedon multi-robot coverage: an algorithmic approach". In: *Annals of Mathematics and Artificial Intelligence* 52.2 (2008), pp. 109–142.

[104] Alessandro Renzaglia, Lefteris Doitsidis, Savvas A Chatzichristofis, Agostino Martinelli, and Elias B Kosmatopoulos. "Distributed multi-robot coverage using micro aerial vehicles". In: *21st Mediterranean Conference on Control and Automation*. IEEE. 2013, pp. 963–968.

[105] David Ribas, Narcis Palomeras, Pere Ridao, Marc Carreras, and Angelos Mallios. "Girona 500 auv: From survey to intervention". In: *IEEE/ASME Transactions on mechatronics* 17.1 (2011), pp. 46–53.

[106] Julio Rosenblatt, Stefan Williams, and Hugh Durrant-Whyte. "A behavior-based architecture for autonomous underwater exploration". In: *Information Sciences* 145.1-2 (2002), pp. 69–87.

[107] Monika Roznere and Alberto Quattrini Li. "Real-time model-based image color correction for underwater robots". In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2019, pp. 7191–7196.

[108] Seyed Abbas Sadat, Jens Wawerla, and Richard Vaughan. "Fractal trajectories for online non-uniform aerial coverage". In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2015, pp. 2971–2976.

[109] Seyed Abbas Sadat, Jens Wawerla, and Richard T Vaughan. "Recursive non-uniform coverage of unknown terrains for uavs". In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2014, pp. 1742–1747.

[110] Pedro J Sanz, Pere Ridao, Gabriel Oliver, Claudio Melchiorri, Giuseppe Casalino, Carlos Silvestre, Yvan Petillot, and Alessio Turetta. "TRIDENT: A framework for autonomous underwater intervention missions with dexterous manipulation capabilities". In: *IFAC Proceedings Volumes* 43.16 (2010), pp. 187–192.

[111] Junaed Sattar et al. "Enabling Autonomous Capabilities in Underwater Robotics". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Nice, France, 2008, pp. 3628–3634.

[112] Ketan Savla, Francesco Bullo, and Emilio Frazzoli. "The coverage problem for loitering Dubins vehicles". In: *2007 46th IEEE Conference on Decision and Control*. IEEE. 2007, pp. 1398–1403.

[113] Ketan Savla, Emilio Frazzoli, and Francesco Bullo. "On the point-to-point and traveling salesperson problems for Dubins' vehicle". In: *Proceedings of the 2005, American Control Conference, 2005.* IEEE. 2005, pp. 786–791.

[114] Pau Segui-Gasco, Hyo-Sang Shin, Antonios Tsourdos, and VJ Segui. "Decentralised submodular multi-robot task allocation". In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2015, pp. 2829–2834.

[115] Magnus Selin, Mattias Tiger, Daniel Duberg, Fredrik Heintz, and Patric Jensfelt. "Efficient Autonomous Exploration Planning of Large-Scale 3-D Environments". In: *IEEE Robotics and Automation Letters* 4.2 (2019), pp. 1699–1706.

[116] Florian Shkurti, Ioannis Rekleitis, and Gregory Dudek. "Feature tracking evaluation for pose estimation in underwater environments". In: *2011 Canadian Conference on Computer and Robot Vision*. IEEE. 2011, pp. 160–167.

[117] Amarjeet Singh, Maxim A Batalin, Michael Stealey, Victor Chen, Mark H Hansen, Thomas C Harmon, Gaurav S Sukhatme, and William J Kaiser. "Mobile robot sensing for environmental applications". In: *Field and service robotics*. Springer. 2008, pp. 125–135.

[118] Amarjeet Singh, Andreas Krause, Carlos Guestrin, and William J Kaiser. "Efficient informative sensing using multiple robots". In: *Journal of Artificial Intelligence Research* 34 (2009), pp. 707–755.

[119] Aydin Sipahioglu, Gokhan Kirlik, Osman Parlaktuna, and Ahmet Yazici. "Energy constrained multi-robot sensor-based coverage path planning using capacitated arc routing approach". In: *Robotics and Autonomous Systems* 58.5 (2010), pp. 529–538.

[120] Sandra Skaff, James J. Clark, and Ioannis Rekleitis. “Estimating Surface Reflectance Spectra for Underwater Color Vision”. In: *British Machine Vision Conference (BMVC)*. Leeds, U.K., 2008, pp. 1015–1024.

[121] Nikolai Smolyanskiy, Alexey Kamenev, Jeffrey Smith, and Stan Birchfield. “Toward low-flying autonomous MAV trail navigation using deep neural networks for environmental awareness”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2017, pp. 4241–4247.

[122] Franklin D Snyder, Daniel D Morris, Paul H Haley, Robert T Collins, and Andrea M Okerholm. “Autonomous river navigation”. In: *Mobile robots XVII*. International Society for Optics and Photonics. 2004.

[123] Eric Stackpole and David Lang. “OpenROV”. In: *URL http://openrov. com* (2012).

[124] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[125] Paolo Toth and Daniele Vigo. *The vehicle routing problem*. SIAM, 2002.

[126] U.S. Department of Homeland Security, United States Coast Guard. *Navigation Rules: International-Inland*.
https://www.navcen.uscg.gov/pdf/navrules/navrules.pdf.
COMDTINST M16672.2D.

[127] Isaac Vandermeulen, Roderich Groß, and Andreas Kolling. “Balanced task allocation by partitioning the multiple traveling salesperson problem”. In: *2019 International Conference on Autonomous Agents and Multiagent Systems*. ACM. 2019, pp. 1479–1487.

[128] Isaac Vandermeulen, Roderich Groß, and Andreas Kolling. “Turn-minimizing multirobot coverage”. In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 1014–1020.

[129] Richard Vaughan. “Massively multi-robot simulation in stage”. In: *Swarm intelligence* 2.2 (2008), pp. 189–208.

[130] Eduard Vidal, Juan David Hernández, Klemen Istenič, and Marc Carreras. “Online view planning for inspecting unexplored underwater structures”. In: *IEEE Robotics and Automation Letters* 2.3 (2017), pp. 1436–1443.

[131] Eduard Vidal, Narcís Palomeras, Klemen Istenič, Juan David Hernández, and Marc Carreras. “Two-Dimensional frontier-based viewpoint generation for

exploring and mapping underwater environments". In: *Sensors* 19.6 (2019), p. 1460.

[132] Israel A Wagner, Michael Lindenbaum, and Alfred M Bruckstein. "Distributed covering by ant-robots using evaporating traces". In: *IEEE Transactions on Robotics and Automation* 15.5 (1999), pp. 918–933.

[133] Yao Wang, Amir M Anvar, Amir Parsa Anvar, and Eric Hu. "A feasibility study on the design, development and operation of an automated oceanic wave surface glider robot". In: *20th International Congress on Modelling and Simulation*. 2013, pp. 1–6.

[134] Christopher K Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*. Vol. 2. 3. MIT press Cambridge, MA, 2006.

[135] Xian Wu, Rachel E Stuck, Ioannis Rekleitis, and Jenay M Beer. "Towards a framework for human factors in underwater robotics". In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. Vol. 59. 1. SAGE Publications Sage CA: Los Angeles, CA. 2015, pp. 1115–1119.

[136] Bin Xin, Yang-Guang Zhu, Yu-Long Ding, and Guan-Qiang Gao. "Coordinated motion planning of multiple robots in multi-point dynamic aggregation task". In: *2016 12th IEEE International Conference on Control and Automation (ICCA)*. IEEE. 2016, pp. 933–938.

[137] Anqi Xu, Chatavut Viriyasuthee, and Ioannis Rekleitis. "Efficient complete coverage of a known arbitrary environment with applications to aerial operations". In: *Autonomous Robots* 36.4 (2014), pp. 365–381.

[138] Ling Xu. *Graph planning for environmental coverage*. Carnegie Mellon University, 2011.

[139] Zhiyang Yao. "Finding efficient robot path for the complete coverage of a known space". In: *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2006, pp. 3369–3374.

[140] YSI a xylem brand. *EXO2 Multiparameter Water Quality Sonde with 7 sensor ports*. `https://www.ysi.com/exo2`. 2021.

[141] Xin Yu and John Y Hung. "Coverage path planning based on a multiple sweep line decomposition". In: *IECON 2015-41st Annual Conference of the IEEE Industrial Electronics Society*. IEEE. 2015, pp. 004052–004058.

[142] Yanwu Zhang, James G Bellingham, John P Ryan, Brian Kieft, and Michael J Stanway. "Two-dimensional mapping and tracking of a coastal upwelling front

by an autonomous underwater vehicle". In: *2013 OCEANS-San Diego*. IEEE. 2013, pp. 1–4.

[143] Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. "Target-driven visual navigation in indoor scenes using deep reinforcement learning". In: *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE. 2017, pp. 3357–3364.

Appendix A

Experimental Platforms

A.1 Autonomous Surface Vehicles: AFRL Jetyak

The main experimental platform used in this dissertation for deployment of 2D coverage path planning algorithms were Autonomous Field Robotics Lab (AFRL) Jetyaks — ASVs modeled after the Woods Hole Oceanographic Institution (WHOI) Jetyak [61] (see Figure A.1). Our objective in developing this fleet was to to ensure moderately long range surveying and sampling operations on aquatic surfaces with low cost and lightweight vehicles [78].

Each jetyak is controlled using a Pixhawk PX4 micro-controller, and is capable of communicating using 900 MHz radio modems, 2.4 GHz remote control radios, and a 2.4 GHz WiFi connection. The communication capabilities enable connectivity with: a remote control transmitter, a remote computer termed Ground Control Station (GCS), and other ASVs using an ad-hoc network. The AFRL Jetyak design enables both manual and autonomous velocity changing operation on and off board, in addition to autonomous waypoint navigation capability off-board.

At the heart of AFRL Jetyak design, 3DR Pixhawk 1 is selected that runs PX4 on the NuttX[84] operating system, along with the ArduPilot Software Suite to enable teleoperation and way-point navigation capabilities. All algorithms presented in this dissertation whenever deployed on the AFRL jetyak, used way-point navigation uploded offline through mission planner on the board or Pixhawk. This version of the Pixhawk includes an internal compass and external I2C (inter-integrated circuit)

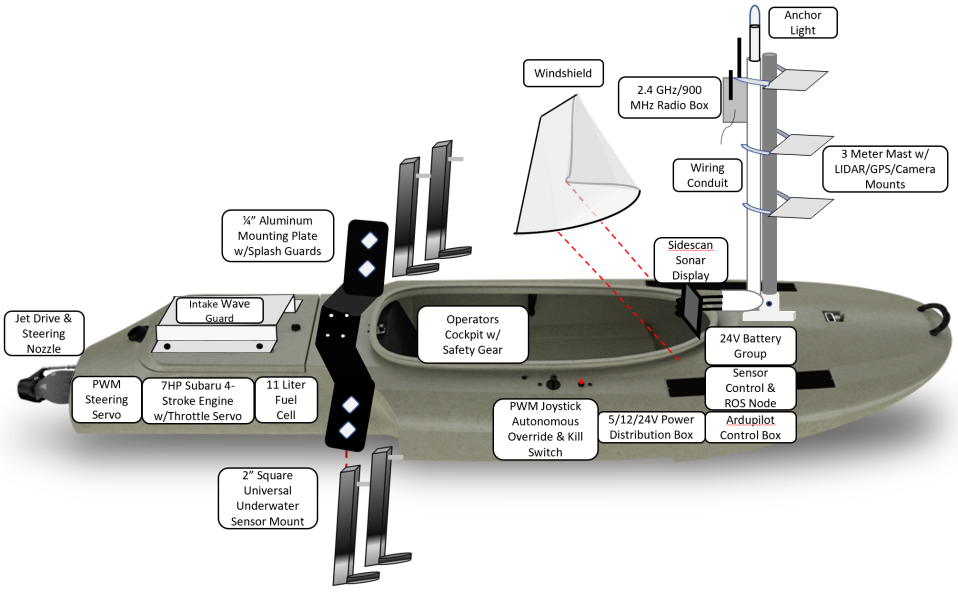


Figure A.1: The AFRL jetyak used during the field deployment with different depth sensors mounted on it for surveying operations.

compass port for an external compass, included with most GPS antennas. Configuration of the PixHawk as a Rover [70] allowed us to start from a point where the throttle and steering servos on the stock Mokai Jetyak are directly imitable. Both the Hitec HS-5485HB[1] throttle linkage servo and the Torxis i04903[2] steering servo are controlled by modifying the Pulse Width Modulation (PWM) values in the PixHawk to match their operating specifications. The latter steering servo is internally controlled by the Polulu Jrk21V3 USB motor controller, which allows the user to access the PWM cutoffs and allows direct calibration between the mechanical steering angle and the input signal.

One of the main objectives of our design is to develop a highly modular platform capable of deploying all types of sensors without the need to make structural changes to the base fleet. Our design allows us to mount different sensors using strong and

[1] `https://hitecrcd.com/products/servos/discontinued-servos-servo-accessories/hs-5485hb-standard-karbonite-digital-servo/product`

[2] `https://gearwurx.com/product/torxis-industrial-outdoor/`

lightweight universal outboard mounting plate which is permanently attached to the Jetyak. For the purposes of the coverage path planning research work discussed in this dissertation, we used the following set of sensors:

- For bathymetric mapping of the lake and river three different type of depth sensors: CruzPro DSP Active Depth, Temperature singleping SONAR Transducer; Humminbird helix 5 chirp SI GPS G2 imaging sonar; long range 3DSS-DX-450 side scan transducer from Ping DSP[95].

- For enhancing coverage pattern discussed in Appendix B environmental force receivers: Sparkfun weather station anemometer for wind speed measurements and ST800/P120 paddle wheel sensor from Raymarine for water current measurements.

All data has been recorded onboard of the Raspberry Pi and sometimes on an Intel NUC processor that runs the Robot Operating System (ROS) framework [100]. The latter provides a package management environment enabling add-on packages such as MAVROS to interface with the PixHawk controller. This allows access and integration with IMU, GPS, heading, velocity, pose and several other Pixhawk telemetry topics. We have included depth, wind, and current sensors as a standard component to our Jetyak design, enabling operation in highly dynamic environments. Finally, in order for the Jetyak to use sensor measurements for on-line path planning, the ROS framework provides an integration of sensing and acting commands, namely providing topics for sending general navigation as well as channel-level steering and throttle control commands directly to Pixhawk using MAVROS and the MAVLink protocol.

A.2 Stereo-rig Suite

The data collected in Chapter 5 were partially collected by a stereo-rig sensor suite hardware (see Figure A.2b), designed at AFRL lab, with underwater cave map-

ping [102] as the target application to be used by divers during cave exploration operations. Nevertheless, this stereo-rig has been used in almost all data collection experiments carried in our lab and it can be used for mapping a variety of underwater structures and objects.

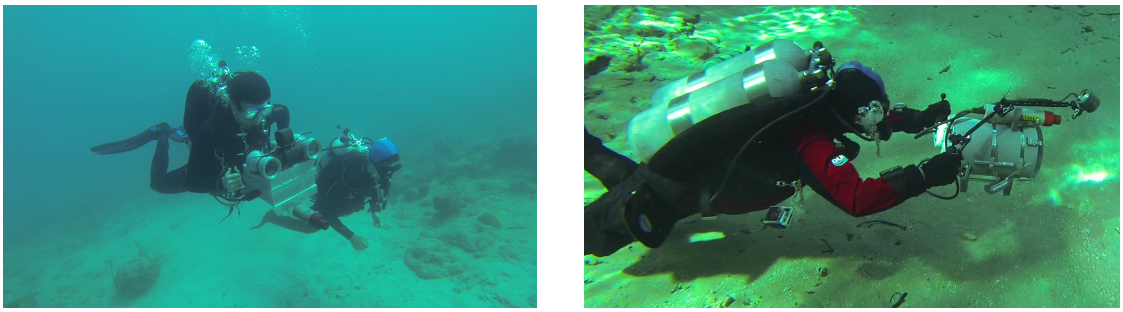
To assist vision-based state estimation, we employ an Inertial Measurement Unit (IMU), a pressure sensor, and an acoustic sensor for accurate state estimation in underwater environments. The specific sensors and electronics of the sensor suite were selected for compatibility with the Aqua2 Autonomous Underwater Vehicles (AUVs) [23]. In particular, the electronics consists of:

- two IDS UI-3251LE cameras in a stereo configuration,
- Microstrain 3DM-GX4-15 IMU,
- Bluerobotics Bar30 pressure sensor,
- Intel NUC as the computing unit,
- IMAGENEX 831L Sonar.

The two cameras are synchronized via a TinyLily, an Arduino-compatible board, and are capable of capturing images of 1600×1200 resolution at 20 Hz. The IMU produces linear accelerations and angular velocities in three axis at a frequency of 100 Hz. Finally, the depth sensor produces depth measurements at 1 Hz. To enable the easy processing of data, the Robot Operating System (ROS) framework [100] has been utilized for the sensor drivers and for recording time stamped data.

A 5 inch LED display has been added to provide visual feedback to the diver together with a system based on AR tags that is used for changing parameters and to start/stop the recording underwater [135].

In the first design (see Figure A.2a) the main unit, a square shaped aluminum box – composed of two parts tighten together by screws – contained the computer,



(a) (b)

Figure A.2: (a) First version of the stereo vision setup, where the two cameras are mounted externally to the main unit. (b) Second version of the sensor suite, where the stereo camera is inside the main unit.

sensors, and other related electronics. The two cameras were sealed in aluminum tubes with tempered glass in front of the camera lenses. The stereo camera and display were mounted on the top of the main unit whereas the sonar was on the bottom of it. Both the cameras and sonar were connected to the main unit by underwater cables. The rationale behind such a design was to allow for an adjustable stereo baseline. Unfortunately, the USB 3.0 interfacing standard used by the cameras is not compatible with the underwater cables available in the market, resulting in highly degraded performance for the cameras with multiple dropped frames.

In the second design (see Fig. A.3), we took into account the lessons learned from the first design. In particular, a PVC tube was used instead of the aluminum box. This made the enclosure lighter and positively buoyant. Rails at the bottom allow for additional weights for ballasting. Furthermore, the main enclosure hosted the two cameras as well. In this way, the cameras can be directly connected to the computer with standard USB 3.0 cables, to avoid unnecessary transmission of data over underwater cables as in the first design. In addition, the second design of the sensor suite allows for modularity in terms of electronics used: a Plexiglas plate inside the enclosure was used to mount all the electronics and can be easily removed for troubleshooting or for changing broken or not up-to date parts: different computer, cameras, or IMU.

Figure A.3: Front top view of the assembled sensor suite.

Within the framework of 3D coverage strategy, we only used the camera to extract image frames for our training. We manually moved the stereo-rig around the shipwreck to mimic a coverage pattern and recorded all sensor data (see Figure A.2b). For future applications, IMU and possibly depth sensor information can be also useful contributions to our planning approach on predicting the robot's next action.

Appendix B

Enhancing Coverage Trajectory with Control

When operating in dynamic aquatic environments precise and efficient control is important especially for small vehicles. While wind or small changes in current might not drastically affect the trajectory of large ships, it adds significant noise in the execution of mission plans of small Autonomous Surface Vehicles (ASVs); (see Figure B.1a). In order to achieve optimal path planning for the coverage of an area, an accurate point to point navigation is a key component. In this section we present an overview of a complete pipeline of achieving adaptive control to enhance coverage in 2D which takes into account the error in displacement caused by the forces such as water currents and wind in order to increase the accuracy [58].

(a) (b)

Figure B.1: The trajectory of ASV way-point navigator: (a) only using the PID controller on Pixhawk; (b) with augmented waypoint navigation.

In order to execute any decisions in dynamically changing environments we collected extensive data along different trajectories during different environmental conditions. In particular water current, wind sensor data, along with the boat's GPS

coordinates and heading information were recorded. The control can be adjusted in two modes: based on predictive values of water current and wind speed or for the current readings of the sensors. In the first scenario a model of the environment is represented by Gaussian Processes [134]. In the second scenario each measurement is processed as it is received during the execution. After this step the strategy is the same for both scenarios: the speed and orientation of the ASV is used to determine the absolute values of each measurement which is used as an input for the linear regression to predict the effect of the forces on the speed and direction of ASV. While the target way-point is not reached, an intermediate way-point is calculated based on the effect values. The speed is also adjusted based on the predicted error. Finally, the ASV is sent to the newly calculated way-point. When the new target position is processed by the navigation controller, it results in a smoother and more accurate path, following the planned trajectory(see Figure B.1b). And finally the enhanced control can be used to improve accuracy of different coverage techniques presented in this dissertation for lake and river monitoring operations.

B.1 Coverage Enhancement in Dynamically Changing Conditions

The following framework presented in Jason Moulton's Ph.D. [111] can be utilized to address the problem of coverage operations in the presence of adversarial forces (wind, currents, waves). It uses the onboard sensors to predict the effect of the force on the navigation correctness and perform correction by introducing intermediate waypoints. The proposed approach is using an ASV equipped with waypoint navigation, wind and water current sensing capabilities. The main sensors used on the vehicle are an NMEA 0183 depth sonar, a Sparkfun anemometer for wind, and the RayMarine ST 800 paddle wheel speed sensors for current measurements. For analog to digital conversion of current and wind sensors the drivers are provided by ArduinoMega and Weathershield microcontrollers. The experiments were performed both on lake

Murray and the Congaree river with different possible patterns to demonstrate the effect of the forces on the trajectories.

The framework has the following components: modelling the environmental forces, performing prediction of the effect of the forces on the navigation, augmentation of waypoints or a feed-forward PID controller. This framework can be used after the offline generation of waypoints for complete or partial coverage, to compensate for any disturbances caused to the trajectories passing through the goal waypoints.

Modelling Environmental Forces When performing waypoint navigation it is sometimes desirable to perform corrections beforehand rather than reactively by current measurements of sensors. Collecting wind and water current data, we build a map of the external forces speed and direction using Gaussian Processes with the Matern 3/2 kernel (see Figure B.2). For more details on the method we direct readers to consult the work by Moulton et al. [80].

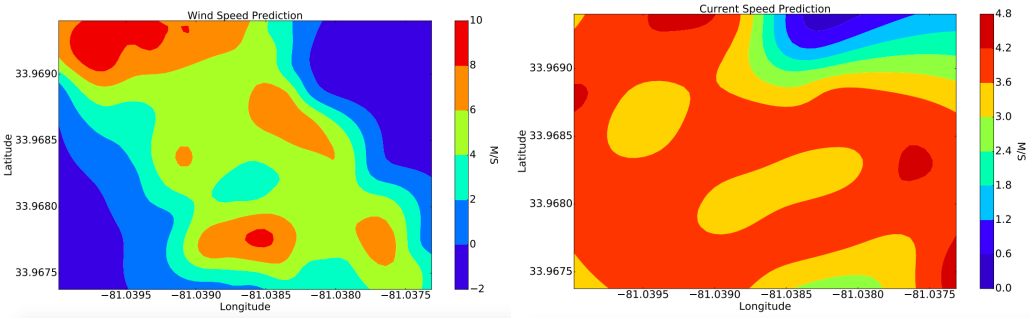


Figure B.2: Force Maps of Congaree River: (a) wind speed map, (b)water current speed map.

Prediction of the Force Affect For reactive response the direct measurements are taken from the sensors. First, comprehensive data are collected to train a linear model for predicting the displacement effect from the current sensor readings. The displacement effect is calculated as x and y components of the error distance from

desired location and the actual location of the boat. The trained linear model is used online for prediction of this effect.

Augmentation of waypoints When the linear model is available either using the current sensor readings or the predicted values from Gaussian Processes we perform a waypoint augmentation. By changing the target global pose based on the measurements and effects of the external forces an intermediate waypoint is generated for smoothing the trajectory. This represents a version of a feed-forward controller where instead of changing the velocity and heading, the corrections are performed by the introduction of waypoints. Figure B.3 illustrates the main idea of this stage: black solid line and position points denote the desired path that should be maintained. Blue arrows represent the wind and current force vector acting on the ASV. Red points and arrows represent the intermediate way-points provided to the Pixhawk navigator and their associated target headings. For more details see Moulton et al. [79].

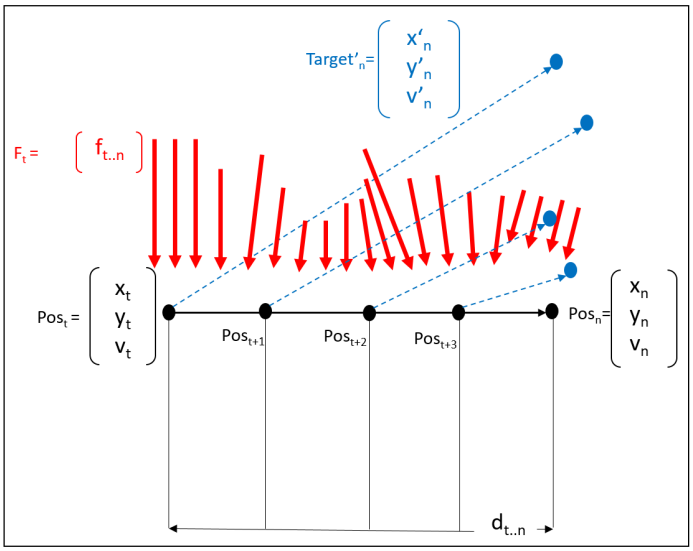


Figure B.3: High-level illustration of way-point navigation augmentation method.

Experiments Figure B.1a shows the effect of the environmental forces on the planned trajectory when the speed of the ASV relative to the ground is increased. This resulted in too much error accumulation in the PID controller and made it unable to overcome the external forces. This will usually result in an overshoot scenario where the ASV harmonically oscillates back and forth over the desired trajectory.

When the augmentation stage is applied as can be observed from Figure B.1b a more precise path-following strategy has been achieved. These results serve as proof of concept for a feed-forward controller with waypoint augmentation that can improve the optimlaity of the coverage pattern. As shown in Figure B.1, path following in currents in all orientations to the ASV is qualitatively improved. Qualitative results have been also reported by Moulton et al [80], which shows improvements in both maximum error and percentage of the path that is more than a meter from the target trajectory. Those results are confirming the intuition gained from qualitative results: the augmented control algorithm provides a better waypoint and thus path following strategy.