

Coverage Planning and Autonomous Gait Switching for Amphibious Aqua Robot

Farzaneh Entezari

Department of Electrical and Computer Engineering
McGill University, Montreal
Summer, 2022

A thesis submitted to McGill University in partial fulfillment of the
requirements of the degree of Master of Science

©Farzaneh Entezari, 2022

Abstract

In this work, we examine the literature on coverage planning problem aiming to generate an inspection plan that ensures complete coverage of the intended spatial field. We discuss cellular decomposition, grid-based, and sampling-based approaches to the coverage planning problem. We present a novel off-line sampling-based method for an autonomous underwater vehicle imaging the ocean floor. The proposed algorithm generates view configurations, differentiating between elongated areas and wide areas using the Voronoi skeleton, to achieve complete observability in the regions of interest. Next, we optimize the coverage route through the arranged viewpoints by solving a variant of the traveling salesman problem. The proposed algorithm is validated in simulation experiments and is proven to outperform previous approaches by improving coverage while reducing the number of scanning locations.

We also survey terrain identification and gait adaptation approaches in order to identify the surface characterizations and select the most adequate walking behavior. Our goal is to autonomously identify the environment in which the robot is maneuvering by capturing the pattern of sensors' data and enabling a legged robot to switch gaits accordingly. We present a classification algorithm to classify the environment based on inertial measurements and leg actuator feedback such that the system can decide between swimming or walking modes when the amphibious Aqua robot is autonomously entering or exiting a body of water. Our model is trained and tested on labeled real-world data, gathered during field trials with the Aqua robot at the lake and the ocean.

Abrégé

Dans ce travail, nous examinons la littérature sur le problème de planification de la couverture visant à générer un plan d'inspection qui assure une couverture complète du champ spatial visé. Nous discutons des approches de décomposition cellulaire, basées sur la grille et basées sur l'échantillonnage pour le problème de planification de la couverture. Nous présentons une nouvelle méthode basée sur l'échantillonnage hors ligne pour un véhicule sous-marin autonome imageant le fond de l'océan. L'algorithme proposé génère des configurations de vue, en différenciant les zones allongées et les zones larges à l'aide du squelette de Voronoi, pour obtenir une observabilité complète dans les régions d'intérêt. Ensuite, nous optimisons le parcours de couverture à travers les points de vue aménagés en résolvant une variante du problème du voyageur de commerce. L'algorithme proposé est validé dans des expériences de simulation et il est prouvé qu'il surpasse les approches précédentes en améliorant la couverture tout en réduisant le nombre d'emplacements de balayage.

Nous étudions également les approches d'identification du terrain et d'adaptation à la marche afin d'identifier les caractéristiques de surface et de sélectionner le comportement de marche le plus adéquat. Notre objectif est d'identifier de manière autonome l'environnement dans lequel le robot manœuvre en capturant le schéma des données des capteurs et en permettant à un robot à pattes de changer d'allure en conséquence. Nous présentons un algorithme de classification pour classer l'environnement en fonction des mesures inertielles et de la rétroaction de l'actionneur des jambes de sorte que le système puisse décider entre les modes de nage ou de marche lorsque le robot amphibie Aqua

entre ou sort de manière autonome d'un plan d'eau. Notre modèle est formé et testé sur des données du monde réel étiquetées, recueillies lors d'essais sur le terrain avec le robot Aqua au lac et à l'océan.

Acknowledgements

First of all, I would like to thank my supervisors, Prof. Frank Ferrie and Prof. Gregory Dudek, for their continuous support, encouragement, and valuable guidance over the course of my studies. Besides my advisors, I would like to thank the members of Artificial Perception Laboratory (APL) and Mobile Robotics Laboratory (MRL) who have always been a source of inspiration. I would like to express gratitude toward Centre for Intelligent Machines (CIM) and McGill University as a whole for making this research possible.

I would also like to thank the Autonomy group in OTTO Motors, an industrial division of Clearpath Robotics Inc., with whom I had the pleasure of collaborating. I am especially grateful to Ryan Gariepy, the CTO, for providing facilities for my research.

Finally, a special thank you to my family for their loving support; to Ali who has been by my side during the good and bad times; and to my dad who, although no longer with us, had always believed in me.

Table of Contents

Abstract	i
Abrégé	ii
Acknowledgements	iv
List of Figures	viii
List of Tables	ix
1 Introduction	1
1.1 Motivation	2
1.2 Overview of thesis	5
2 Literature Review	7
2.1 Coverage path planner	7
2.1.1 Exact cellular decomposition	10
2.1.2 Approximate cellular decomposition	14
2.1.3 Room decomposition	17
2.1.4 Sensor placement coverage	17
2.1.5 Summary	21
2.2 Autonomous gait switching	21
2.2.1 Gait planning and gait transition parameters	22
2.2.2 Terrain classification	23
2.2.3 Summary	28

3	Methodology	29
3.1	Coverage path planner	29
3.1.1	Viewpoint generation	31
3.1.2	Route generation	34
3.2	Autonomous gait switching	36
3.2.1	Robot gaits	38
3.2.2	Sensory data	38
3.2.3	Deep learning model	39
4	Results and Discussion	43
4.1	Coverage path planner	43
4.2	Autonomous gait switching	51
4.2.1	Data collection	51
4.2.2	Evaluation	53
5	Conclusion and Future Work	56
5.1	Summary	56
5.2	Future directions	57

List of Figures

1.1	Schematic overview of the robotic exploration mission	4
2.1	Comparison between the two exact cellular decomposition methods	11
2.2	Morse cellular decomposition with various Morse functions	13
2.3	A complete coverage path within each cell	14
2.4	Randomized sampling approach to sensor placement problem.	19
2.5	Boundary placement approach to sensor placement problem.	19
3.1	A sample map of the seafloor environment.	30
3.2	An illustration of 2D skeletonization.	32
3.3	An illustration of free space segmentation and skeletonization.	33
3.4	The amphibious Aqua robot walking on the shoreline.	37
3.5	Overview flowchart of the gait switching algorithm.	40
3.6	The gait selection Neural Network architecture.	41
4.1	The effect of survey radius on segmentation and skeletonization of the free space.	45
4.2	Observation point placements on environment segments before optimisation.	46
4.3	Coverage enhancements in missing areas with optimisation procedure.	47
4.4	Final inspection path of the environment for several regions of interest.	48
4.5	Computational time of the algorithm for a sample experiment.	50

4.6	Gait transition experiments in the field. (a) transition from <i>swim</i> to <i>walk</i> when approaching to the shore on a rocky beach. (b) transition from <i>walk</i> to <i>swim</i> when entering the lake on a sandy beach.	52
4.7	Results of gait adaptation to environment with multiple transitions	54

List of Tables

2.1	Tabular summary of coverage planning algorithms.	20
4.1	Quantitative performance of the coverage planner algorithm.	49

Chapter 1

Introduction

In the last decade, autonomous inspection and environment monitoring has gained great attention for various applications. Imagine a robot which is asked to periodically carry out a complete inspection in an environment and document the variations along the time. Such a fully automated monitoring mission by robots arises a variety of different challenges including coverage planning, motion planning, gait planning, localization, obstacle avoidance, data capturing, image processing, model construction, etc. In this thesis, we address two of the fundamental challenges in autonomous outdoor environment exploration with a focus on mobility challenges that might arise on a shoreline.

The first problem is coverage planning for robotic platforms to efficiently inspect a spatial field. We explore coverage planning tasks that are intended to generate mission plans in order to collect data from two-dimensional spatial fields. The objective is to prepare a detailed survey mission plan consisting of the sensing locations and the overall robot traveling path, such that the intended survey area is fully observed by the robot's sensor. Our ultimate goal is to survey the seabed and monitor the environmental processes in the ocean.

The second problem tackled in this thesis is gait planning for an amphibious robot. While performing a survey mission, a robot is inevitably required to traverse a variety of environments. For a walking robot, different terrains imply different walking behavior

to enhance stability and facilitate locomotion. Similarly, for legged amphibious robots, environment-specific gait changes are necessary to adapt their gait to the particular environment in which they are operating. For example, an amphibious robot should learn to differentiate sand from water to effectively switch between swimming and walking when entering or exiting the water. Our goal is to enable an amphibious robot to identify its environments and switch gaits on its own.

1.1 Motivation

Our emphasis on efficient coverage planning is motivated by the growing importance of the health of marine environments which need to be monitored repeatedly and consistently. In marine environments, coral reefs have great environmental and scientific value as they are essential pieces of a balanced ecosystem and home to diverse species of marine organisms. However, the coral reefs are threatened by local and global risks such as water temperature increase, solar radiation, ocean acidification, chemical pollution, destructive fishing, etc. [13,45]. Scientists believe that global warming has caused the mass extinction of coral reefs, a phenomenon also known as coral bleaching, at an ever-increasing rate. They argue that corals could dramatically disappear if we do not act quickly [46]. Hence, researchers are highly interested in persistently monitoring the response of these ecosystems to climate change and the remediation efforts.

Conducting coral reef surveys is critical to understand and analyze the impact of different factors on this phenomenon. however, repeated manual data collection by human operators is expensive and risk-prone, especially in open oceans. Therefore, deploying underwater robots to perform exploration missions and collect effective and efficient data from the reefs substantially facilitates marine environment monitoring. There has been extensive research on automating seabed exploration and navigation in the literature [36,61,63,67,114].

Our work addresses the automated collection of survey data to efficiently cover the area of interest. We explore complete coverage planning algorithms to generate a survey plan for seabed inspection by an Autonomous Underwater Vehicle (AUV). The task is to determine an optimal path that allows the robot to cover the entirety of the target area. In these applications, usually a known map of the environment is available [28, 30] that may contain the bathymetric map, location, size, etc. In a similar work [67], bathymetric mapping of the sea floor is performed on-site using an Autonomous Surface Vehicle (ASV) to provide the surveyor AUV with a depth map of the environment. We seek offline coverage algorithms to generate a mission plan for an already mapped underwater environment. The area where we are interested in covering is where the coral reefs exist. For our experiments, the reefs are annotated on the map by an expert, indicating the area to be surveyed. The AUV navigates at a constant altitude above the seafloor surface, carrying a down-looking sensor, to capture as much data as possible from the marine environment.

We examine recent coverage planner algorithms in the literature and propose a novel method for generating a survey plan. Considering that data collection by precise sensors on the AUV can not be done while the robot is moving, we focus on stationary sensor placement methods. We aim to determine the best set of viewpoints that fully covers the target area and find the optimal robot path that visits all the viewpoints and covers the entirety of the area. Our goal is to perform the survey with a shorter trajectory and fewer sample points while ensuring full coverage.

In this thesis, we also address the problem of environment identification and gait adaptation for a legged robot. Legged robots that possess a variety of potential gait patterns are usually able to navigate over different terrains and environments. Humans intuitively know how to adjust their walking behavior to the surface, they are walking on, in order to walk more stable and efficient. For example, we tend to bend slightly forward and take smaller steps on the ice, compared to when we walk on the ground. Similarly, legged robots should utilize different gait patterns to ensure stability and energy-

efficiency. For an amphibious robot, it is even more critical to select the appropriate gaits for its environment to navigate safely. For example, a robot that moves from a sand beach to deep water should have the ability to switch from walking to swimming as it enters the water. Failure to switch gaits in response to environmental changes may cause destructive damage to the robot.

We believe that in order for the robots to perform smooth transitions between different gaits, they need to first differentiate various types of environments and terrains. With the ability to identify its surrounding, the robot can select the most adequate gait for its current environment. Hence, we explore terrain classification algorithms focusing on gait adaptation to the environment. We then propose a gait controller to decide on the desire to switch gaits by analyzing the inertial and actuator information. We argue that interaction forces between the robot legs and the ground are a promising indicator of the surface mechanical properties. Therefore, we utilize inertial sensor and actuator feedback to probe and estimate the environment and switch gaits accordingly when entering and exiting the body of water.

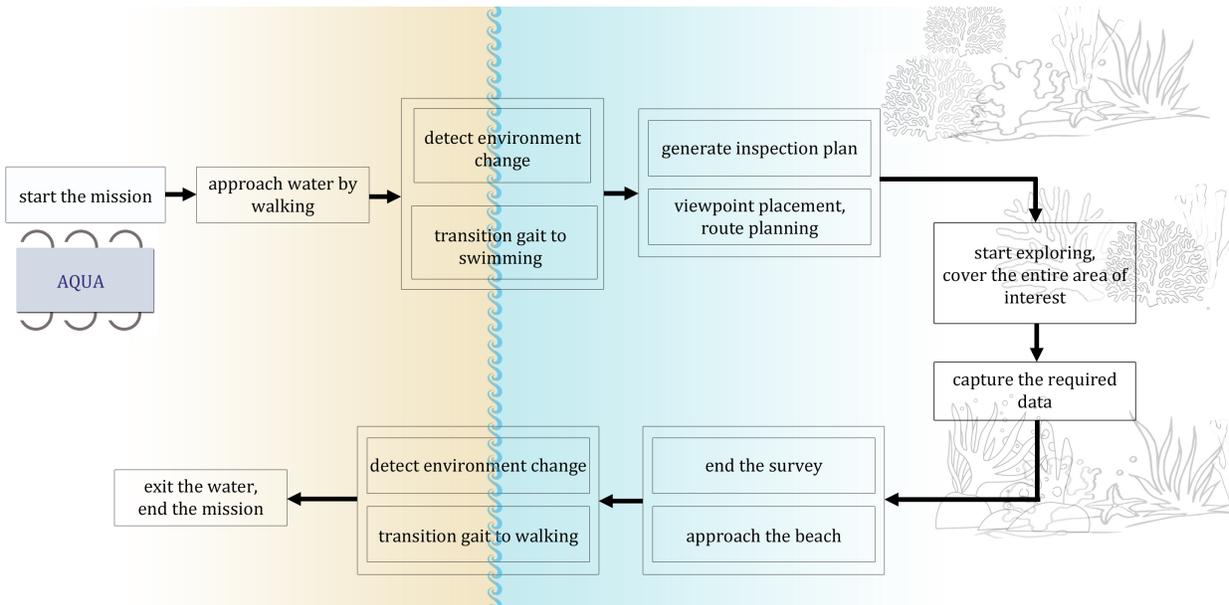


Figure 1.1: Schematic overview of the robotic exploration mission and the two sub-problems in hand: gait adaptation and coverage planning.

Figure 1.1 illustrates a schematic overview of the robotic exploration mission on land and water, that we wish our amphibious robot to carry out autonomously. Such an autonomous monitoring task involves numerous components and challenges. The schematic demonstrates the steps related to the two sub-problems addressed in this thesis and clarifies how the two sub-problems are connected and contribute to our ultimate goal.

1.2 Overview of thesis

In this thesis, we present two novel algorithms to overcome two of the challenges in fully automated outdoor environment exploration. We explore complete coverage planning and autonomous gait adaptation with a focus on mobility challenges on a shoreline for the amphibious Aqua robot [24]. Due to the differences in the two domains, in every chapter, we include relevant sections for each of the two main objectives.

Chapter 1 provides a brief insight into the main objective of this thesis and discusses the two challenges that we tackle to achieve the overarching goal. Section 1.1 further describes the motivation behind the project, the importance of marine environment monitoring, and the necessity of gait adaptation for fully automated outdoor missions by legged robots.

Chapter 2 is dedicated to a discussion of related literature. The background on coverage planning algorithms is provided in Section 2.1, exploring different techniques used in exact, approximate, room decomposition, and sensor placement coverage. We review the requirements and applications of each of the presented algorithms and decide on the most suitable approaches for our specific robotic platform and environment. In Section 2.2, we study the earlier publications on gait learning and transition. We demonstrate how gait adaptation algorithms are closely related to terrain classification problem. Section 2.2.2 lists and compares various sensing techniques and learning techniques for autonomous gait switching.

Chapter 3 presents the methods that we proposed for each of the two problems. In Section 3.1, we explain our coverage path planning algorithm to completely explore an arbitrary shape target area and the algorithm's constraints and requirements for our specific application. The viewpoint generation and route generation steps are discussed in detail with illustrations of the maps. The proposed method for automatic gait switching is provided in Section 3.2. We discuss the robot's gaits, the sensory data, and the deep learning model to detect the desire to transition gaits at the time of entry and exit on the beach.

We validate the effectiveness of the proposed methods in Chapter 4. The implementation details, simulator details, and the comparative results for our coverage planner algorithm are reported in Section 4.1. In Section 4.2, the data collection process, our experimental testbed, and the evaluation findings for the proposed gait switch controller are explained.

Finally, Chapter 5 concludes the thesis with discussions about exploration and gait adaptation strategies and potential future directions.

Chapter 2

Literature Review

In this chapter, we explore the existing literature that addresses the two main problems which we are tackling, namely coverage planning and gait adaptation, for our ultimate goal of a fully autonomous environmental monitoring mission. We will start by briefly describing different solutions to the coverage planning problem, along with their use-cases, strength, and limitations, and outlining their suitability for our specific problem. The subsequent section reviews the previous research on terrain identification and gait adaptation techniques to approach our second problem of autonomous gait switching for legged robots. We study various sensing and learning strategies to come up with an idea of the most suitable approach for our robot.

2.1 Coverage path planner

This section presents a summarized description of various coverage planning algorithms in the literature, aiming to provide an insight into the features, suitability, and applications of each method.

Coverage Path Planning (CPP) in robotics aims to generate a route that allows a robot to entirely cover all the reachable points in an area of interest while avoiding obstacles. With the growing number of applications of automation and robotics, the application of

CPP algorithms has extended to the ground, aerial, and underwater robotics. This task is fundamental to many applications including floor cleaning robots [47], seabed inspection [44], farming [41], and demining [3]. Various CPP algorithms have been studied in the literature and discussed in the surveys of Choset [20] and Galceran and Carreras [31].

The Coverage Path Planning problem has several variants to satisfy various requirements and conditions such as finding the shortest path, having prior information, covering with the robot's footprint or the sensor's Field of View (FoV), and dealing with stationary or mobile sensors. One of the problem variants is the Art Gallery Problem [91] which aims to place a minimum number of stationary sensors in an area in a way that every point in the specified area is observable from at least one of the sensors. It is also related to the Zookeeper problem [104] and Watchman Tour Problem [16] which try to find the minimum length round tour for a watchman to guard an entire area with maximum observability. These two are most suitable for surveillance and inspection tasks such as gas leak sensing [4] and guarding a museum.

The coverage problem is also closely related to Traveling Salesman Problem (TSP) [109] which tries to find the shortest possible closed route that visits all the cities in a given set exactly once, knowing the distance between every pair of cities. The Covering Salesman Problem (CSP) [22], a generalization of TSP, is the problem of identifying the minimum cost tour of a subset of cities such that each not visited city is within a predetermined covering distance of at least one visited city. The CSP is also a common approach to solving the CPP problems. In Coverage algorithms, TSP and CSP are usually employed to find the efficient visiting order of the room segments or stationary sensor locations. Faigl et al. [27] decompose the mobile robot inspection planning problem into a sensor placement problem and a multi-goal path planning problem, and utilize TSP to discover the optimal route through arranged viewpoints. This decoupled approach is more applicable when the view cost is greater than the travel cost, for example in scenarios where high-quality measurements cannot be taken while the robot is moving or the image processing, image registration and model construction for each viewpoint are expensive. Although

Wang [101] argues that solving the two sub-problems independently may not lead us to the globally optimum solution, it's a common feasible approach for inspection planning.

Literature classifies coverage algorithms as either heuristic or complete [56]. Heuristic algorithms do not provably guarantee to cover the entirety of the space, however, complete algorithms ensure finding a path that passes through all the points in the region of interest [20]. Provably complete coverage has significant importance for some applications such as mine clearance where clearing the whole minefield is necessary. However, complete approaches require more sensor measurements and higher computational power and can be impractical for large outdoor environments. Adaptive sampling approaches, on the other hand, consider a trade-off between completeness and efficiency. These approaches are more suitable for coverage tasks in environments where not all the target area is uniformly important; however, the desired features are concentrated in a few hot-spot regions [63]. Low et al. [59] presents an adaptive multi-robot exploration strategy for both wide-area coverage and hotspot sampling using non-myopic path planning. Manjanna et al. [65] also address adaptive coverage of a spatial field without prior knowledge by following a multi-scale path to produce a variable resolution map of the field.

Independently, Coverage Planning algorithms are also categorized as offline and online depending on having a priori knowledge about the area such as the layout of the environment. Since online algorithms rely on real-time sensor measurements to cover the target area, they are also called sensor-based coverage algorithms. The focus of the present work is on complete offline algorithms, generating a complete coverage path for a $2D$ known workspace.

For a coverage task, introducing multiple robots usually provides more efficiency and robustness at the cost of complexity [83]. Rekleitis et al. [85] have presented multi-robot approaches to complete coverage. They introduced distributed coverage algorithms for a team of robots exploring an unknown environment with unlimited communication between the robots. They have also tackled the coverage task with multi robots that have

limited communication, restricted to the line-of-sight. Later, Shkurti et al. [94] expanded the environment monitoring task to a multi-domain task performed by a team of an aerial vehicle, an autonomous airboat, and an agile legged underwater robot.

Many coverage path planners partition the target environment into multiple non-overlapping cells to reduce the problem for complex areas, where determining the optimal coverage path is infeasible due to NP-Hardness of the problem [5]. Cellular decomposition methods are classified as exact, approximate, and semi-approximate by Choset [20].

2.1.1 Exact cellular decomposition

Exact cellular decomposition methods divide the free space within the environment into non-overlapping regions called cells which can be covered using simple motions such as back-and-forth motion or spiral motion. The union of all the cells completely fills the free space in the target environment. An adjacency graph represents the decomposed environment where the nodes represent the regions to be covered and the edges represent whether they share a common boundary. The planner utilizes the adjacency graph to generate an exhaustive walk through nodes and compute the sequence in which cells are visited. A major limitation of exact cellular decomposition is that it can result in unnecessary small cells that could be merged with other cells and still be covered with the simple in-cell motions [52]. Next, we discuss three popular exact cellular decomposition approaches and summarize the pros and cons for each.

Trapezoidal decomposition

One of the most popular exact cellular decomposition methods for offline coverage planning is the trapezoidal decomposition [7], which breaks the free space into trapezoidal shape cells. Within each trapezoidal cell, coverage can be achieved with back and forth motions over lines parallel to one of the edges, and the coverage is shown to be provably complete.

Trapezoidal decomposition is applicable to a 2D polygonal environment consisting of non-intersecting polygonal obstacles and a polygonal boundary. Considering the vertex points of polygons, an edge segment is drawn at each vertex and is extended to the upper and lower in a vertical line to create trapezoidal cells. One can think of it as sweeping a vertical line in the environment and placing an edge every time the sweep line meets a vertex point. See Figure 2.1.

However, sweep direction highly influences the optimality of the generated cells by creating regions of different shapes and sizes and is a challenging criterion in trapezoidal decomposition. In order to optimize the coverage for an agricultural field machine, Ok-sanen et al. [75] proposed applying decomposition with six sweep lines inclined at 30° intervals and repeating the process for half-size intervals around the most favorable directions. The process continues until reaching a certain threshold of the path cost function or reaching 1° accuracy. Another drawback of trapezoidal decomposition is that it creates only convex cells, resulting in a larger number of cells that could be merged together. Allowing non-convex cells, that can still be covered by simple motions, creates fewer numbers of cells and therefore shortens the final coverage path.

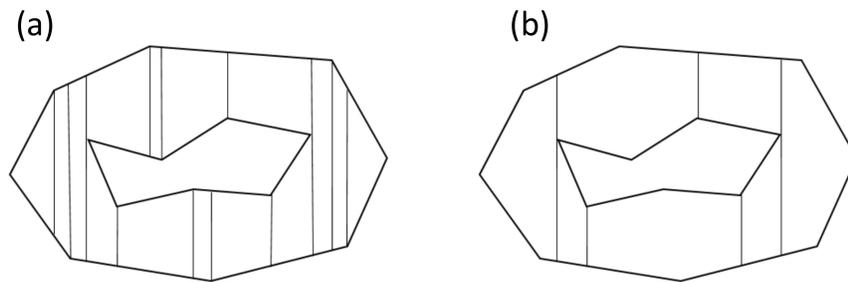


Figure 2.1: Comparison between the two exact cellular decomposition methods: (a) trapezoidal decomposition, (b) Boustrophedon decomposition. From [71] with permission of owner.

Boustrophedon decomposition

Similar to trapezoidal decomposition, boustrophedon decomposition, also known as the lawn mover algorithm, uses a sweep line to determine the breakpoints. As opposed to trapezoidal, it partitions the environment by inserting an edge only on vertices where the extension can be made both above and below the vertex, rather than every vertex [19]. These vertices are called critical points. Hence, the boustrophedon decomposition method reduces the number of cells by only opening and closing a cell only when an obstacle is encountered or is passed over when sweeping the line. See Figure 2.1. In addition to the decomposition of known environments, the boustrophedon decomposition method can be adapted to online applications for which the obstacle map is not known in advance [1]. Rekleitis et al. [84] presented a multi-robot coverage algorithm, with limited within line-of-sight communication between robots, that uses online cell-based Boustrophedon decomposition for complete coverage.

Visiting the achieved polygonal non-convex cells in an order obtained from the adjacency graph, and covering each cell with back and forth motions, named boustrophedon motions, guarantees complete coverage. However, the sweep line-based methods have several limitations. First, it is difficult to determine an optimal sweep direction. Second, it can not handle more than one vertex intersecting the sweep line [71].

Morse decomposition

Acar et al. [2] introduced the Morse decomposition method which is a generalization of the boustrophedon method and uses the Morse function to determine the critical points instead of using vertices. It is inspired by Canny's roadmap method for start to goal path planning [15]. Unlike previous methods, Morse decomposition is not limited to polygonal obstacles and moreover, it can generate different cell shapes such as spiked, spiral, or square cells by utilizing different Morse functions. Theoretically, it can handle any n-dimensional space.

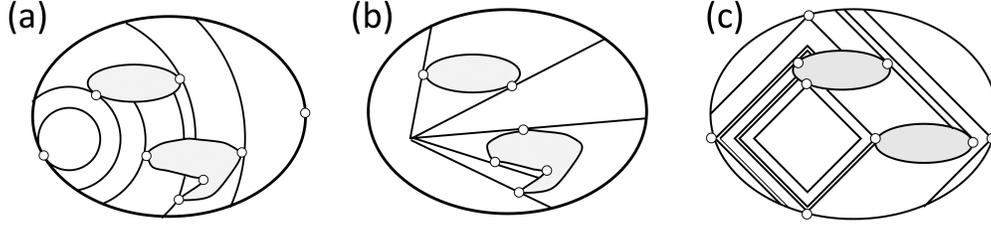


Figure 2.2: Morse cellular decomposition with various Morse functions result in different slice and cell patterns: (a) spiral cell pattern, (b) spiked cell pattern and (c) square cell pattern. From [2] with permission of owner.

From calculus, for a real-valued function $h : \mathbb{R}^m \rightarrow \mathbb{R}$ the differential at a point $p \in \mathbb{R}^m$ is $dh_p = [\frac{\partial h}{\partial x_1}(p) \dots \frac{\partial h}{\partial x_m}(p)]$. A point $p \in \mathbb{R}^m$ is called a critical point of h where either the function is not differentiable or all its partial derivatives are $\frac{\partial h}{\partial x_1}(p) = \dots = \frac{\partial h}{\partial x_m}(p) = 0$. If all the critical points of a function are non-degenerate, meaning it's Hessian ($\frac{\partial^2 h}{\partial x_i \partial x_j}(p)$) is non-singular, then the function h is a Morse function.

Acar et al. [2] define a slice function as the pre-image of a real-valued function. To create the Morse cells, the connectivity changes of the slice, swept in the workspace, are analyzed. They showed that connectivity changes happen at critical points of the Morse function restricted to the obstacle boundaries. In other words, the sweep line is perpendicular to the surface normal of the obstacle at the critical point [31]. The cell decomposition is constructed by splitting or merging the cells when the slice intersects with the obstacle or leaves the obstacle at the critical point.

In Morse decomposition, different cell shapes can be obtained by utilizing different Morse functions and hence different slice functions. For instance, in a planar case where the robot's workspace is $W = \mathbb{R}^2$, the pre-image of the Morse function $h(x, y) = x$ defines the slice function as a vertical line $W_y = h^{-1}(\lambda)$ and is effectively a boustrophedon decomposition. λ is the parameter that specifies the location of the slice in the workspace while sweeping. To illustrate a few other patterns, a spiral decomposition can be achieved with the Morse function $h(x, y) = \sqrt{x^2 + y^2}$, a spiked decomposition with the function $h(x) = \tan(\frac{x_2}{x_1})$ and a square cell pattern with $h(x) = |x|$, demonstrated in Figure 2.2.

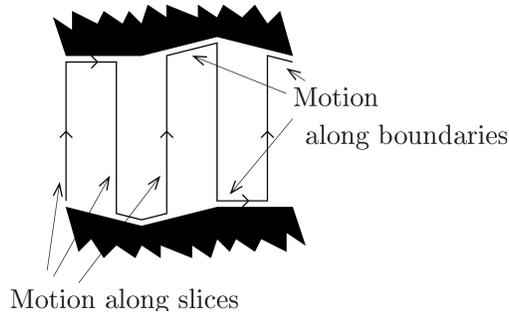


Figure 2.3: Motion along the slice and motion along the obstacle to generate complete coverage path within a cell in Morse cellular decomposition. From [2] with permission of owner.

Once the cells are constructed, the planner determines the order in which cells are visited and the explicit coverage path within each cell. The former is achieved by planning an exhaustive walk through the cells using its associated adjacency graph. A complete coverage path within each cell desires moving along the slice and moving along the boundary of obstacles. As shown in Figure 2.3, the robot moves along one slice until it encounters an obstacle, then follows the obstacle’s boundary until it has moved by an inter-lap distance equal to the robot’s sensor range, and repeats the process along a new slice to accomplish complete coverage.

Galceran [31] noted that a limitation of Morse decomposition is its inability to handle rectilinear environments where the critical points are degenerate.

2.1.2 Approximate cellular decomposition

Approximate cellular decomposition, also called grid-based cellular decomposition, represents the environment as a set of uniform cells and was first proposed by Moravec et al. [69]. As opposed to the exact cellular decomposition, all the cells are of the same size and shape and the union of cells only approximates the shape of the target environment and its obstacles. Grid cells can have any shape, although square and hexagonal cells are the most popular types of grid-based decomposition in the literature. The environment can then be represented as an array where the associated value to each cell contains the

occupancy information for that cell. However, exponential growth in memory with size of the map makes it inefficient for relatively large size environments [52], thus grid-based approaches to coverage tasks are more suitable for indoor coverage operations.

Typically the cells are the size of the robot's footprint or its visibility polygon such that when the robot enters a cell, the whole cell area can be marked as covered. Hence, the complete coverage path planning problem can be described as finding a path that visits all the cells in the decomposition [20].

Wavefront algorithm

The Wavefront algorithm was introduced by Zelinsky et al. [111] to find a path of complete coverage from a start cell to a goal cell with minimal secondary visits to grid cells. The algorithm propagates a wavefront from the goal position throughout the entire free space, calculating the distance transform by assigning integer values to each grid cell. The wavefront assigns 0 to the goal, and 1 to the unmarked neighboring cells and repeats with incremental values until reaching the start. The robot can then perform pseudo-gradient ascent on the computed numeric potential function to find a path from start to goal, visiting the highest value unvisited neighboring cell at each step. A remarkable advantage of this approach is that the distance transform can be modified to impose different robot navigation behaviors; such as reducing turns or considering path safety.

Spanning trees

Another approach to finding optimal coverage paths in a grid-based representation of the area was proposed by Gabriely et al. [29], known as Spiral Spanning Tree Covering (Spiral-STC). Their approach splits each grid cell into four smaller cells that are the same size as the robot. Recursively at each step, the robot selects the first cell in an anti-clockwise direction in the space [31], following the spanning tree of the grid map. Upon reaching the end of the tree, the robot turns around and crosses over to the other side of the tree until arriving at the start cell.

Later, Lee et al. [57] extended the spiral path tracking algorithm with wall following and a new path linking method, coarse-to-fine CIDT, to generate smoother and cheaper coverage paths for mobile robots.

Traveling salesman problem

Once the target area is decomposed into grid cells which are the same size as the robot or its end effector, the solution to the coverage path planning problem is a path that visits all the grid cells in the free space. The Traveling Salesman Problem (TSP) seeks the globally shortest route that passes through all the accessible grid cells. A distance graph G is computed where the nodes are the grid cells to be visited and the edges represent the distance between each two pairs of cells. The TSP is a combinatorial optimization problem and has proven to be NP-hard, hence a heuristic sub-optimal solution is usually acceptable for coverage tasks. Concorde TSP solver [21] can generate the exact optimal solution to the problem. There are also several other approximate solvers such as greedy algorithm, genetic algorithm [81], ant colony optimization [95], 2-opt moves, and Lin-Kernighan-Helsgaun [43] algorithm.

Lin-Kernighan is a heuristic approach, first proposed by Lin and Kernighan [58], to find the shortest tour in the graph using 2-opt and 3-opt moves with exceptions. A k -opt move in a distance graph is defined as selecting up to k nodes in a tour and finding k better edges to form a better tour. The idea behind Lin-Kernighan-Helsgaun, an extension of the Lin-Kernighan algorithm, is that any k -optimal solution is also l -optimal for $l < k$. Therefore, the authors suggested every time we find a promising k -opt move, we try to extend it to a $(k + 1)$ -opt move by finding another edge to exclude. Lin-Kernighan-Helsgaun tries to find k -opt moves for ascending values of k until finding the highest possible value that gives promising improvement. Lin-Kernighan-Helsgaun uses 5-opt moves as its basis for optimization. This approach is one of the most successful methods for generating optimal or near-optimal solutions for the symmetric traveling salesman problem.

2.1.3 Room decomposition

There are a number of indoor coverage tasks where we might prefer the robot to fully perform coverage in one room before traversing to the next room for the sake of robot scheduling or warehouse management. In these cases, environment decomposition requires breaking the free space into rooms considering walls and edges. In the literature, room decomposition approaches are categorized as automatic and interactive segmentation [10]. Interactive methods such as [72] require some degree of human input, while automatic methods such as [26] solely rely on the provided environment layout to segment the space.

2.1.4 Sensor placement coverage

The sensor placement coverage planner, also known as sampling-based coverage, is another approach to the coverage planning problem that generally consists of two steps. First a set of view configurations is generated that completely covers the target environment. The sensor placement problem is related to the famous art gallery problem which tries to select as few guards as possible to completely observe a known area. The second step seeks the optimal path that goes through every viewpoint with a minimum total length of the path. The multi-goal path planning problem, usually referred to as the Traveling Salesman Problem, is a combinatorial optimization problem. Both problems are proven to be NP-hard, hence, the algorithms attempt to find an approximate sub-optimal solution.

The decoupled approach is mostly beneficial for inspection planning problems where sensor measurements cannot be taken during the robot movement, such as 3D image acquisition, hence requires determining a set of stationary sensing locations from where we observe the whole area. Therefore, the sensor placement coverage approaches aim to cover the target area with visibility polygons of the sensor, considering the visibility constraints such as field of view and sensor range, rather than covering with the robot's

footprint. Gas sensing with a mobile robot [4], surveillance with distributed sensor networks [17,23] and inspection of the surfaces of an object [102] are among the applications of sensor placement coverage in the literature.

Polygon partitioning

Kazazakis et al. [50] proposed a polygon partitioning algorithm to select a sufficient number of viewpoints for inspecting a non-convex polygonal workspace with polygonal obstacles. He assumes the robot is equipped with a panoramic camera with 360° field of view, therefore it is capable of covering a convex polygon with only one viewpoint if the distance of the point to the vertices is no larger than the visibility range. The algorithm decomposes the original non-convex polygon into a set of convex polygons and divides them into smaller sub-polygons successively until each of them can be completely covered by one viewpoint.

Randomized sampling

Another sensor placement algorithm, proposed by Gonzalez-Banos et al. [39], positions the viewpoints based on sampling random points and selecting a subset of minimum cardinality to fully inspect the boundary of the environment. The algorithm is extended by Faigl [27] to address the covering of the interior.

The randomized sampling is demonstrated in figure 2.4. First, a point p is sampled from the boundary of the workspace W and the visibility polygon V from the point p is computed. Then, m number of points are sampled from within the visibility polygon and the one which has the largest intersection between its visibility polygon and V is selected as a new viewpoint. The uncovered space is updated according to the new set of viewpoints and the process is repeated until complete coverage is achieved. [27] proves that the overall complexity is $O(mn_v n_g \log(n_v n_g))$ where m , n_v , and n_g are the number of samples, the number of vertices of the original polygon and the number of found viewpoints respectively.

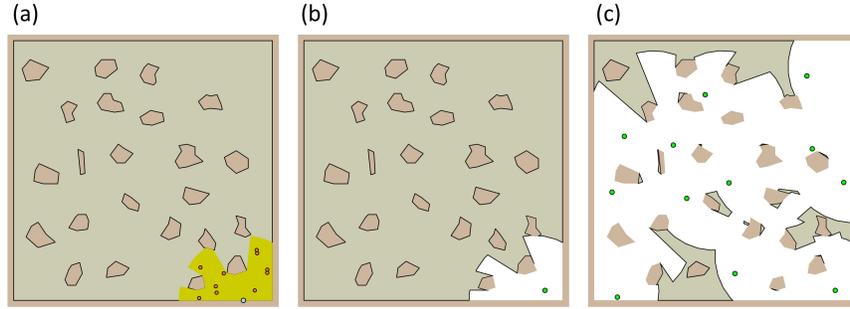


Figure 2.4: Randomized sampling approach to sensor placement problem from [27]. (a) initial random point in green at the border and a set of random candidate points, (b) The covered area in white after placing the first point in green, (c) coverage map after 12 iterations of placing points. From [27] with permission of owner.

Boundary placement

Faigl et al. [27] argue that the previous algorithms do not benefit from as much area as possible of the sensor's visibility polygon because they may place viewpoints close to boundaries or obstacles, occluding the sensor's visibility, thus leading to a larger number of guards. They believe that it is unnecessary to place guards or viewpoints closer to obstacles than the visibility range of the sensor.

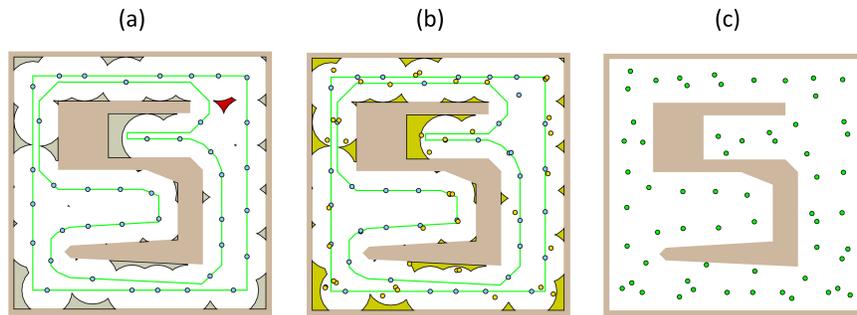


Figure 2.5: Boundary placement approach to sensor placement problem from [27]. (a) coverage map for the points placed on the boundary of shrunk space, (b) coverage map after placing additional points in uncovered areas with randomized sampling method, (c) The final set of points after optimization procedure. From [27] with permission of owner.

Therefore, they suggest positioning the viewpoints in three phases: First, the algorithm places viewpoints at a specified distance from the obstacles by shrinking the free

Category	Approach	Environment	Ref.	on/off-line	Notes
Exact cellular decomposition	Trapezoidal decomposition	Polygonal	[7]	off-line	Breaks the free space into trapezoidal shape cells; creates only convex cells, resulting in a larger number of cells
	Boustrophedon decomposition	Polygonal	[19,71]	off-line	Reduces the number of cells and the length of the overall coverage path in comparison with Trapezoidal decomposition
	Morse decomposition	Differentiable boundaries (non-rectilinear)	[2,31]	on-line	Capable of generating different cell shapes by utilizing different Morse functions; unable to handle rectilinear environments
Approximate cellular decomposition	Wavefront algorithm	Grid-discretized	[111]	off-line	Minimize secondary visits to grid cells; Modify distance transform to impose different navigation constraints
	Spanning trees	Grid-discretized	[29,57]	on-line	Smoother and shorter coverage path
	Traveling salesman problem	Grid-discretized	[43]	off-line	A heuristic approach that seeks the globally shortest route that visits all the grid cells
Room decomposition	Automatic and Interactive segmentation	Indoor environments	[10]	off-line	Breaks the free space into rooms considering walls and edges
Sensor placement	Polygon partitioning	Non-convex polygonal workspace	[50]	off-line	Decomposes the original non-convex polygon into a set of convex polygons, each covered by one viewpoint
	Randomized sampling	Non-convex workspace	2D [27,39]	off-line	selecting a subset of minimum viewpoints for complete coverage
	Boundary placement	Non-convex workspace	2D [27]	off-line	Reduces the number of viewpoints by considering occlusions and visibility constraints
Multi-robot coverage	Efficient Boustrophedon Multi-Robot Coverage	Unstructured arbitrary-shaped environment	[85]	on-line	Performs task allocation among the robots which operate under the line-of-sight communication restriction
	Multi-domain monitoring using a team of robots	Marine environmental fields	[94]	on-line	Generates multi-domain coverage performed by a team of an aerial vehicle, an autonomous airboat, and an underwater robot
Adaptive coverage	Multi-robot Adaptive Sampling Problem (MASP)	Wide-area exploration with dynamic hotspots	[59]	on-line	Performs wide-area coverage and hotspot sampling using non-myopic path planning.
	Data-driven selective sampling	Marine environmental fields	[65]	on-line	Generates adaptive coverage plan to produce a variable resolution map of the field

Table 2.1: Tabular summary of coverage planning algorithms.

space with a distance equal to the sensor range. Second, it selects additional points to observe the remaining uncovered area with a randomized approach similar to the algorithm described above. The uncovered regions are addressed one by one, differentiating between small and large regions. They select a random point p_u at the border of the uncovered area and place a new viewpoint p_n at a certain distance from p_u to cover as much

area as possible from the uncovered area. Third, an optimization procedure tries to replace two very close points with one point in order to reduce the number of viewpoints. The performance of the boundary placement algorithm is shown in Figure 2.5.

2.1.5 Summary

Table 2.1 summarizes different categories of coverage planning approaches which we discussed. The inspection planning problem in hand for our autonomous underwater robot, requires an off-line coverage path planner to fully inspect an arbitrary shape non-convex 2D field. For our case, the cost of data capturing and post-processing the acquired data is greater than the cost of travelling. We require the robot to stay stationary when triggering the scanner and capture the images required for building a model of the coral reefs on the seafloor. Hence, the sensor placement-based approaches are the most suitable for our survey mission planning. We aim to place as few viewpoints as possible in the environment that generates full coverage of the areas of interest and find the shortest path through the arranged viewpoints by solving a variant of the Traveling Salesman Problem (TSP).

2.2 Autonomous gait switching

In this section, we review the literature on terrain identification and gait adaptation techniques to address our second challenge in a fully autonomous exploration mission on both land and water using an amphibious legged robot.

One of the strengths of legged robots is their ability to traverse a variety of terrains. They often possess a collection of gait patterns to ensure stability and adaptability over a wide range of surfaces such as rocky terrain, a snowy road, a muddy area, a sandy beach, or underwater. A gait is a sequence of periodic leg movements for the purpose of generating locomotion from one place to another. The selection of an appropriate gait and a safe transition between different gait patterns is a critical task for autonomous robots that are designed to navigate over different kinds of challenging terrains.

For a legged robot, each gait pattern is usually tuned for a specific environment and working conditions. For example, different walking behaviors are required for walking on soil or on an icy surface. Similarly, the movement patterns vary when changing speed or performing a turn even in the same environment. The goal of gait transition is to adapt the gait pattern to the environment and working conditions in order to achieve a desired performance such as achieving energy efficiency [64], stability [105], maximum speed, turning [80], or obstacle avoidance [62].

The problem of autonomously adapting gaits to an environment has been explored previously in the literature. There are various challenges in converting among different gaits with flexibility and efficiency. Optimizing gait patterns, smooth transition between gaits, detecting the need to adapt gaits, and determining the perfect time to perform the gait switching have great significance in accomplishing autonomous gait planning for legged robots.

2.2.1 Gait planning and gait transition parameters

A great body of work has discussed the generation of different types of gaits and a mechanically stable locomotion controller to achieve a smooth and continuous transition from one gait to another or change of speed. Santos et al. [90] studied the mechanism for controlling the velocity and generating transition movement patterns for switching gaits inspired by biological concepts of quadruped walking animals such as horses or cats. Weingarten et al. [106] examined the effect of gait parameters on speed and energy efficiency and proposed a gait adaptation system based on the Nelder-Mead algorithm [70]. Kong et al. [53] explored the problem of determining the optimal time to switch gaits within a gait cycle while keeping on moving, considering different phases in a cycle such as the support phase and aerial phase. Although robots are able to perform various gait pattern and safely transition between gaits, detection of the urge to switch gaits is still not necessarily explicit.

2.2.2 Terrain classification

Another challenge in adapting the robot's gait patterns to its environment in real-time is the problem of identifying the surface characterization and selecting the most adequate gait. Therefore, the gait switching problem is closely related to the terrain classification problem which aims to identify the environment in which the robot is functioning. The legged robot utilizes the extracted information from sensor data to assess terrain properties and make a reliable interpretation of its surroundings in order to adapt the walking behavior [64, 103]. In other applications, the robot may require to examine terrain traversability and infer how safe it is to drive over different patches of the environment [79]. Sancho-Pradel et al. [89] presented a survey of the sensing techniques and learning techniques for terrain assessment focusing on its application for autonomous planetary robots.

Sensing techniques

In order for the robots to adjust the gait characteristics to the terrain, they benefit from perception systems to obtain information about the environment for classification purposes. This knowledge can be gathered using various sensors. In this section we review the most common perception methods for a mobile robot to identify its environment, in order to determine the most suitable data sources for our use case.

A common approach to the terrain classification problem is to rely on the visual appearance of the scene to decide on the environment's class. Visual information such as color or texture can provide valuable data for terrain assessment. Manduchi et al. [62] discussed surface classification techniques using color images and presented a Maximum Likelihood strategy using a Mixture of Gaussians to discriminate between various terrains for an off-road vehicle. A semantic segmentation approach was also proposed by Masahiro et al. [76] to categorize every pixel in the image based on gray intensity and image gradients channels, aiming to build a traversability map and identify potential hazards for the robotic exploration of Mars. Otsu et al. [77] argued that vision sensors

are sensitive to environmental conditions such as illumination and hence require a large amount of labeled data. Therefore, some researchers presented co-training approaches to use another source of data besides the color images, such as vibration data [77] or point cloud depth information [103].

There are other researches that aim to classify the terrain based on the LIDAR data. Kragh et al. [54] presents a classification approach for classifying individual points from 3D point clouds acquired using single multi-beam LIDAR scans to identify different terrains in agricultural fields. McDaniel et al. [68] also utilizes a LIDAR scanner for classification and modeling of forested terrain.

Audio signal has also been shown to be a good indicator for material and ground identification. Roy et al. [86] argues that similar to how a blind person might tap his cane on different floor materials, we can tap a boom-mounted microphone on the surface and classify the floor type based on the acoustic signature arising from the contact. A tapping mechanism-based material mapping system for a mobile robot was also proposed in [49]. They identify the material of different objects by recording and processing the sound produced by the tap of a solenoid.

Another promising approach to effectively sense a surface is by considering the inertial information of the robot's body while traversing over different terrains such as in [9, 11, 48, 87]. The idea is that the robot's dynamics highly depend on the terrain's mechanical properties. The physical interactions between the walking robot and the ground cause various acceleration and rotation patterns induced in the robot's structure. Hence, one can probe and analyze the vibration of the system during locomotion in order to identify the surface and adjust the gait selection based on the gathered information. Manjanna et al. [64] proposed a semi-supervised algorithm to obtain a mapping between terrain type and gait parameters, based on the inertial responses of a hexapod robot. Once the terrain type is recognized, they alter the gait cycle frequency, aiming to enhance walking speed and energy efficiency when transitioning between two terrains. Khalili et al. [51] studied inertial measurements to distinguish between various indoor and outdoor ter-

rains for a wheelchair. More interestingly, some work [55,66] has been done to present frameworks for evolving gaits and changing interactions with the terrain in order to make the robot act as an active sensor and improve its terrain discrimination ability.

Terrain classification algorithms can also use actuator data to identify the environment [73]. Similar to how terrain properties affect vibrations, mechanical interactions between the robot's legs and the ground result in varying current consumption for the leg motors. Since legs directly interact with the surface, it is more intuitive to measure the effect on actuators and leg motor currents. Giguere et al. [37] analyzed both actuator data and inertial sensor information to identify the environment of the amphibious Aqua robot [24] for the purpose of autonomously switching from walking gait to swimming gait. They proposed the idea of synchronizing the sensor information with the leg angle and distinguishing the leg angle at which the motor current or acceleration has the largest discrimination and environment types are well separable. Manjanna et al. [66] extended their work and applied a similar approach aiming to enhance walking performance by identifying the terrain type and adjusting the leg cycle frequency.

Tactile feedback has also been considered a good indicator for surface identification in robotics literature [12,99,107]. In contrast with indirect measurements made by vehicle-mounted sensors, tactile sensors come into immediate contact with the surface and measure the terrain properties directly. Therefore, they usually tend to be more precise approach than inertial sensing approaches [32]. Giguere et al. [34,35] proposed the idea of analyzing the tip acceleration patterns induced in a metallic rod dragged along a surface to probe terrain properties. Shill et al. [93] presented a terrain identification approach based on the pressure images generated with pressure sensor arrays that come into direct contact with the terrain surface for a one-legged hopping robot. They believed that as opposed to vibration-based classifiers, their approach is independent of the robot's operating conditions such as gait and speed. Walas [98] mounted a force torque sensor on the robot foot to capture surface properties. They utilized tactile perception in combination with visual and depth perception to adjust gait parameters for a walking robot. Wu

et al. [108] performed surface identification based on the magnitude and distribution of ground reaction forces on the legs of a running robot using an array of capacitive tactile sensors.

Learning techniques

Terrain identification begins with the analysis of raw sensory information. Various tools and data processing methods can be employed for characterizing terrains. This section summarizes a number of the most widely-used learning techniques in the literature for categorizing terrain types based on the captured sensor data.

A well-known machine learning tool for classification that has been used for terrain assessment in research papers is Support Vector Machines (SVM). Bajracharya et al. [6] used a self-supervised method with a linear SVM for traversability classification purposes. A multi-sensor data classification, fusing visual data and vibration signals, using SVM classifier and maximum likelihood estimation is presented in [40]. The authors also exercised Bayesian fusion and meta-classifier fusion techniques to merge the results of the two classifiers for a more accurate terrain classification. In [82], the average values of motion resistance and slippage alongside root mean square and standard deviation of vertical acceleration were combined as input to a proprioceptive SVM classifier. Shi et al. [92] investigated semi-supervised learning approaches to tackle the issue of lack of sufficient labeled data. They proposed a modified Laplacian SVM to utilize unlabeled data for better performance in vibration-based terrain identification. Results presented in [113] suggest SVM is better suited for online terrain classification compared to the other algorithms.

Artificial Neural Networks can provide an accurate and robust solution to the terrain classification problem. Ojeda et al. [74] compared the performance of Neural Networks for 15 different sensor modalities of 5 different terrain types. Convolutional Neural Networks (CNNs) have also been effective for terrain identification due to their excellent feature extraction capabilities. Yan et al. [110] utilized CNN models to derive represen-

tative deep features for a region-based classification problem, followed by a subsequent SVM classifier to assign pixel-level labels. A comparative study of traditional machine learning methods against Deep Neural Networks (DNNs) and CNNs for classifying terrain and estimating wheel slip is provided in [38]. The authors highlight the advantages of deep learning algorithms for being able to perform accurately with raw data without necessarily requiring an expert feature extraction or pre-processing filter.

Among deep learning approaches, Recurrent Neural Networks (RNNs) have been shown to be effective for terrain assessment due to their recurrent structure and ability to process and predict sequence data. Otte et al. [78] presented a visual terrain classification algorithm by generating feature sequences on repeatedly mutated image patches learned with standard RNNs, Long Short Term Memory networks (LSTMs), and Dynamic Context Memory networks (DCMs). A model consisting of CNN architecture that learns deep spatial features, complemented with long-short term memory units that learn complex temporal dynamics was proposed in [96] to categorize surfaces based on vehicle-terrain interaction sound. Bednarek et al. [8] also presented a classification approach based on LSTM architecture that performs on the force and torque signals in the time domain from the interaction of a legged robot foot with the ground. Vulpi et al. [97] argued that self-learned features from deep learning may include temporal information from the data that are not captured by the manually designed features. Hence, they suggested a terrain identification model based on Convolutional Long Short-Term Memory recurrent neural network (C-LSTM) to autonomously search relationships between features and categorize sensory signals as time series.

Besides supervised learning techniques, clustering of terrain-specific sensor data is a common approach to distinguish between different surface types. The advantage of unsupervised learning methods is that labeling of data samples is no longer required. Hence, they can incorporate raw new information as they explore unknown environments and discover new terrains by identifying outliers of a trained model [14]. Giguere et al. [33] introduced a clustering algorithm for time-series sensor measurements that exploits tem-

poral coherence between samples and evaluated the effectiveness of their approach using three different classifiers (linear separator, mixture of Gaussians, and k-Nearest Neighbor).

2.2.3 Summary

In this work, our goal is to enable the amphibious Aqua robot [24] to alter its gait autonomously. The robot is capable of walking on the ground and swimming under the water. We would like the robot to switch between *walk* and *swim* modes when entering and exiting a body of water. We seek the terrain identification approaches to determine the robot's environment and decide on the desired gait.

Among the discussed sensing techniques, we rely on the idea that mechanical interactions between the robot's legs and the surface in the two environments produce different vibrational signatures in the robot's body. Furthermore, we study the motor current for the robot's legs as an indicator of the current environment. Considering that the visual appearance of the scene is highly dependant on the illumination and our experiments are conducted in the evening, the images are poorly lit and usually occluded by lots of air bubbles caused by rapid leg rotations in the water. Hence, we do not consider the robot's camera view a promising indicator of the environment for our problem. In terms of learning technique, we build a Neural Network model to predict the most appropriate gait for our problem.

In the following chapters, we discuss the proposed terrain identification and gait alternation approach and report the performance of our method on real-world data.

Chapter 3

Methodology

This chapter details the proposed approaches to each of our sub-problems, clarifying the components of our methods including the problem definition, the terminology, the assumptions, the steps of the algorithm, and finally the desired outcome. We first describe our solution to the problem of generating the coverage plan for monitoring an area of interest, considering the requirements and specifications of our task. Then moving on to the second problem in hand, we explain our methodology for the gait alternation problem including the choice of sensing space, the prediction model, and details of our network architecture.

3.1 Coverage path planner

Our objective is to generate an efficient coverage plan to image the annotated areas of interest, where coral reefs exist, on the seafloor with an autonomous amphibious robot. We present a coverage algorithm to determine an optimal set of observation points such that the union of their visibility polygon covers the entirety of the target area and generate the coverage path to visit all the observation points. The robot navigates on the generated path and captures data while staying stationary on each of the arranged observation points.

Our algorithm is developed to operate on a planar surface. We assume the surveyor robot navigates at a constant altitude above the surface of interest. It is equipped with a down-looking sensor with a certain FOV able to image the seafloor from the navigating altitude. While a marine robot is navigating underwater at a constant depth, the sensor's FOV varies along the sea surface depending on the height of the target surface. In this work the assumed sensor's FOV is adapted to the imaging requirements for each region and the robot is instructed to keep a certain altitude from the seafloor based on the required FOV: the higher the altitude is, the wider the FOV becomes. In our case, because the sensor is operated at a constant altitude, its visibility disk is approximately constant in each region. Let "survey radius" be the radius of the visibility disk from the specified altitude. In the navigating plane, any protruding region of the seafloor intersecting the robot's horizontal plane is considered to be an obstacle and must be avoided. Figure 3.1 illustrates a sample map of the seafloor where the coral reefs' location is annotated in white, named "free space" hereafter. The gray areas are either obstacles or sandy areas which we are not interested in. Purple polygons indicate different regions of interest (ROI) in which we perform the coverage path planning task.

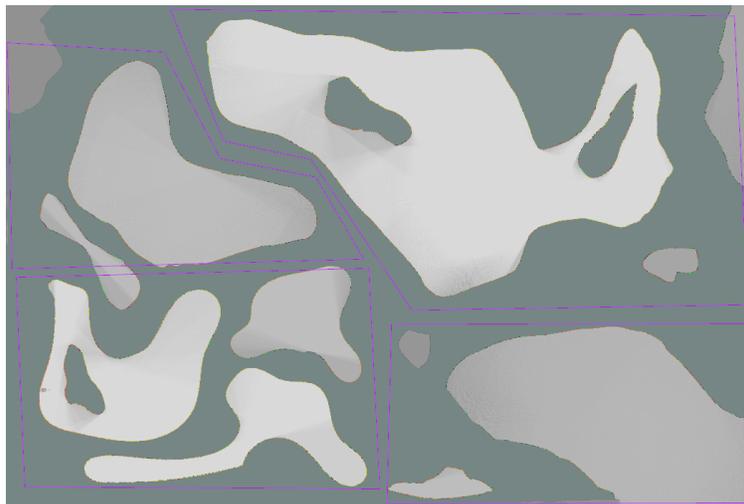


Figure 3.1: A sample map of the seafloor environment. White area is the area of interest to be inspected. The gray area is either obstacles or sandy area which we are not interested in. Purple polygons indicate different regions of interest.

In our task of inspecting the coral reefs, the cost of sensing, including image segmentation and classification, is greater than the cost of moving, and high-quality measurements can not be taken during robot movement. Hence, as described in Section 2.1.4, a sensor placement coverage algorithm is the most suitable to generate a survey plan for the AUV. The proposed coverage planner consists of two decoupled steps. The first is to determine a list of geographical locations or waypoints such that every point in the target area is observable from at least one of the waypoints. The second is to plan a minimum length route that visits all the waypoints while avoiding obstacles. The following sections describe each step of our algorithm.

3.1.1 Viewpoint generation

We aim to place as few viewpoints as possible to cover the entire free space in the region of interest. Obviously, it is unnecessary to place viewpoints closer to obstacles and region boundaries than the survey radius, as in that case, a considerable portion of the visibility disk would become worthless. Furthermore, in the narrow elongated areas where the width of the free space is roughly the same as the perimeter of the visibility disk, it is the most reasonable to place the viewpoints at equal distance from the two sides. Therefore, we suggest first placing the viewpoints near the boundaries and in the narrow spaces and then filling in the remaining area. In order to achieve that, we segment the free space into narrow space and wide space and treat each differently.

We believe that Faigl’s approach to placing sensors at a pre-specified distance from the boundaries [27] has limitations in handling the areas where its width is smaller than four times the survey radius. In such areas, their approach ends up placing viewpoints on two very close parallel lines and results in unnecessary overlapping visibility disks. To overcome this issue, we propose segmenting the map and computing the skeleton of the area to decide on efficient sensor placements.

We assume we are given a Portable Gray Map (PGM) of the environment in advance to generate the covering plan. In order to segment the free space in the ROI, we convolve

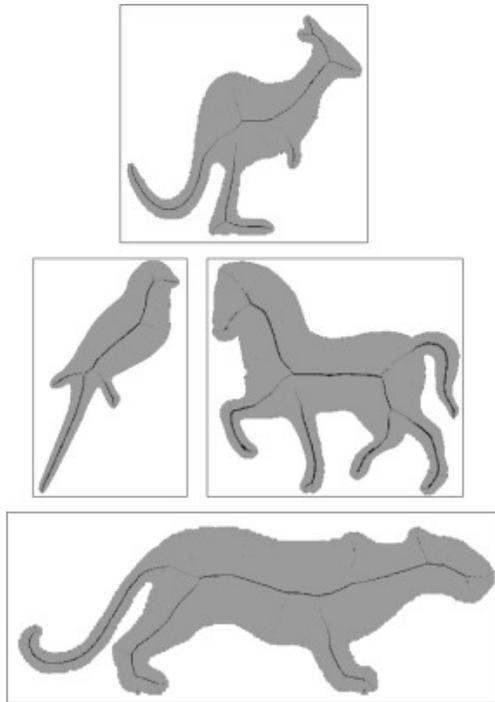


Figure 3.2: An illustration of 2D skeletonization. In each image, the black curve is the morphological skeleton of the gray shape. From [88] with permission of owner.

the original map with a Gaussian filter whose kernel size and kernel standard deviation are proportional to the survey radius, r_s . This expands the boundaries into the free space and then a binary threshold on the free space differentiates between the so-called narrow area and wide area in the ROI. One could think of it as shrinking the free space by a distance roughly the same as the survey radius. To cover the narrow area, we compute the morphological skeleton of the narrow area which appears to be the best for viewpoint placements. The skeleton of an area is a thin one-pixel width version of it that is equidistant to its boundaries. In other words, the boundary of the object is approximated by a continuous curve, known as the skeletal curve of the object [88]. Figure 3.2 provides an illustration of the skeleton of 2D binary shapes. Thinning algorithms extract the skeleton of a shape by iterative erosion and dilation of the original shape while ensuring the connectivity of the curve. We apply the Zhang-Suen thinning algorithm [18, 112] to generate the skeletal curve. The formed curve is at a survey radius distance to the boundaries, and at the mid-line in narrow areas that are roughly the width of two survey radii. Figure 3.3

illustrates the segmentation and skeletonization. The blue polygon indicates the region of interest in which we require full coverage. The white area and bright gray area represent the wide and narrow space respectively and the red curve is the generated skeleton.



Figure 3.3: An illustration of free space segmentation and skeletonization. The blue polygon indicates the region of interest in which we require full coverage. The white area and bright gray area represent the wide and narrow space respectively and the red curve is the generated skeleton.

The primary viewpoints are placed on the skeletal curve at a constant distance d from each other, where $d = 2\alpha r_s$ and $0.7 < \alpha < 1$. The constant α controls the amount of overlap of visibility disks which ensures a full coverage, and facilitates image registration to build a model of the environment. Once the viewpoints are placed at the boundary of the free space, the remainder of the area is covered with additional viewpoints arranged on a square grid in the free space. We allow tweaking the viewpoints at a distance of half of the survey radius, in case a grid point is too close to an obstacle and hence inaccessible.

The last step in viewpoint generation is a post-processing algorithm in order to optimize coverage in the region of interest. We build a binary coverage image that indicates the uncovered regions in the free space. We find contours in the coverage image corresponding to the gaps in coverage and resolve the gaps iteratively. Depending on the area of the uncovered contour, we rearrange the viewpoints in three different ways to

locally optimize coverage. First, for larger gaps, we approximate the bounding rectangle of the contour and apply Simplicial Homology Global Optimization (SHGO) to find the optimal position for an additional viewpoint within the bounding rectangle. The SHGO technique [25] determines the global minimum to the loss function which is defined as the total area in the scan that was previously seen or is occluded. The solution to the optimization problem is the location where a new observation point results in maximum additional coverage. A viewpoint is then placed at this optimal location. Second, for smaller gaps, we select the set of viewpoints whose visibility disks surround the contour. We improve the coverage in that area by rearranging the selected neighboring viewpoints. These viewpoints are moved toward the skeleton or the centroid of the gap depending on the shape of the contour. Thus, the coverage in the gap is improved while reducing the intersection between selected disks with their neighbors. Third, reasonably small gaps whose area is below a certain threshold are disregarded. The threshold is dependent on the gap tolerance. Obviously, since the area is covered with the same size circles that represent the observable surface from each viewpoint, there is a trade-off between the amount of gap and overlap of the disks: the lower the gap tolerance is, the higher the doubly scanned area becomes.

3.1.2 Route generation

The last step of any sensor placement-based coverage path planner is to connect the observation points in a way that the traversal cost is minimized. The traversal cost can include the length of the inspection path, the cost of turns, the power consumption, or the time to complete. We pose the problem as a traveling salesman problem (TSP) which is one of the most widely studied problems in combinatorial optimization. We use graph notation and build a complete edge-weighted graph $G = (N, E)$ where N is the set of nodes each representing a viewpoint and $E = \{(i, j) | i \in N, j \in N\}$ is the set of edges. The associated weight of each edge $c(i, j)$ represents the cost to traverse from one viewpoint to another. The problem is then formulated as the problem of finding a Hamiltonian cycle of min-

imum cost in the edge-weighted graph [43]. The resulting cycle is the cheapest way of visiting all the waypoints and returning to the start point.

We aim to minimize the total length of the continuous and collision-free path surveying the region of interest. Therefore, the associated cost to each edge would be the length of the robot path connecting the two nodes. We first compute the trajectory between pairs of nodes and estimate the length of the generated trajectory. Considering that path planning between every pair of nodes is expensive and time-consuming, we only generate the navigational path for neighboring pairs and assign a penalized cost to the remaining pairs. The assigned penalized cost is proportional to the Euclidean distance between the two waypoints on an edge.

The Lin-Kernighan heuristic [58] is generally considered to be one of the most effective methods for generating optimal or near-optimal solutions for the symmetric traveling salesman problem. The Lin-Kernighan algorithm is based on the k -opt heuristic method, which is a tour improvement algorithm. The k -opt operator improves the tour by swapping k of its edges with k new edges if the new edges provide a reduction in the length of the tour. We rely on the Lin-Kernighan algorithm to discover the near-optimal order of waypoints to be visited for an efficient coverage path.

For a graph of n vertices, the tour is optimal if there does not exist an n -opt improving move. However, It has been shown that the complexity would be $O(n^n)$ and it is not an applicable approach. We also know that any k -opt move also covers l -opt moves for $l < k$. Hence, Lin-Kernighan aims to provide a more flexible approach by increasing k as long as improving moves are found to achieve a near-optimal solution.

Having the weighted graph, we generate a greedy nearest neighbor tour as the initial seed for the Lin-Kernighan heuristic algorithm. At each step, we select a node from where we start the tour t_1 , and select either its predecessor or successor t_2 to form an edge to remove. Then we select a node t_3 to connect to t_2 and add an edge that does not belong to the original tour and has a positive gain. We examine connecting either the predecessor or successor of the last node t_4 to shape a more efficient tour. If the new tour is of smaller

cost, we accept the tour and restart the algorithm with the new tour. Otherwise, if the gain is still positive, we search for another node t_5 to add an edge and repeat the steps. The algorithm is based off of 2-opt moves because it is the most efficient local search. Hence, it always performs all the 2-opt moves by exhausting the search over all the possibilities in the selection of the first two edges. The Lin-Kernighan algorithm was first proposed by Lin and Kernighan [58]; later [60] and [42] discussed the approach in more detail for implementation which we used as the basis of our work.

The origin of the TSP was devoted to a complete closed Hamiltonian cycle where the salesman is supposed to visit all the cities and return to the starting point with a minimum total distance. However, in our application, we do not require the robot to drive back to its initial position and restrict the algorithm to make a closed-loop tour. Hence, the solution to our problem is instead the minimum length Hamiltonian path which has a fixed start node n_{robot} , corresponding to the robot's initial position. In order to achieve that, we create two additional imaginary nodes to the graph n_{start} and n_{end} . The cost from n_{end} to all the nodes is zero and the cost from n_{start} to all the nodes is extremely large except from the n_{robot} and n_{end} . Therefore, the path $\{n_{end}, n_{start}, n_{robot}\}$ with zero cost would be the bridge to traverse back to the starting node from any of the nodes in order to close the loop. This reduces the problem to a normal TSP. Once the minimum cost Hamiltonian cycle is generated, removing the imaginary nodes results in the intended optimal Hamiltonian path.

3.2 Autonomous gait switching

During an automated exploration mission, we require the robot to traverse a variety of environments including sandy beach, rocky beach and underwater to be able to reach every point in the target space. Our inspection missions usually start by walking on the land, leads the robot into the body of water, explores the seabed while swimming, and directs the robot back to the seashore. The amphibious legged robot utilizes several modes that

determine the legs' movement pattern when given target trajectories. Each mode, corresponding to a gait pattern, is adjusted for a specific environment or terrain. Therefore, the robot should be capable of transitioning between gaits across a changing environment. In this section, we present an environment identification and gait adaptation approach to enable such autonomous gait transitions.

Our robot, Aqua shown in Figure 3.4, is a hexapod with six independently-controlled leg actuators that are designed for amphibious locomotion [24]. We aim to enable the Aqua robot to switch gaits autonomously. The Aqua robot is usually given inspection tasks that require the robot to go from land to the water and from water back to the land. We should be able to give it a couple of waypoints or tasks in both land and water and be sure that it can switch modes and transition between the two environments safely when needed.



Figure 3.4: The amphibious Aqua robot walking on the shoreline.

The modes relating to this experiment are *walking* and *swimming*. A critical parameter for switching gaits is determining when to perform the transition. It is critical to switch gaits at the right time. An early switch from *walking* to *swimming* when the robot is still navigating on land or shallow water causes destructive damage and may lead to broken legs or over current in the motors.

We propose a method to identify the environment based on vibrations of the robot's body and select the appropriate gait. We present a model which determines on its own whether the Aqua robot should transfer to *swimming* or *walking* mode in real-time while navigating autonomously. With an effective gait planning system, it can safely maneuver on any specified trajectory over land and water.

3.2.1 Robot gaits

Legged robots possess a variety of gaits. A gait is a periodic pattern of movement for locomotion. Gait planning determines how to effectively lift off and place each foot in a sequence in order to move the robot forward. The motion of a hexapod is usually categorized as tripod gait, tetrapod gait, wave gait, and swimming gait depending on the minimum number of legs touching the surface at every moment [53]. The Aqua robot has six arch-shaped legs. It walks in the tripod gait, leaving at least three legs on the ground every second, to ensure stability and optimized speed. At each gait cycle in walking mode, the legs perform a full rotation in two groups of three legs, two on one side and one on the other, to form a stable tripod in contact with the ground while the rotation of the other three legs moves the robot forward. Similarly, a gait cycle in the swimming mode is a periodic rotating motion, where the legs rotate back and forth around some target leg angle.

3.2.2 Sensory data

We rely on the idea that the acceleration of the robot's body is a decent indicator of the environment. We require to distinguish between walking on the ground and walking in shallow water in order to switch from *walking* to *swimming*, and distinguish between swimming in deep water and swimming in shallow water to switch back. Locomotion in each of these environments generates different mechanical interactions between the robot's legs and the surface. These interactions lead to different patterns in the robot's

perturbation, producing different vibrational signatures in the robot's body when traversing in each environment. Hence, the acceleration and rotation of the robot's body are measured and used to estimate the dynamics of the environment and decide on the appropriate gait.

Furthermore, the external resistance against legs' rotation varies in different environments, especially in and out of water. In addition, the varying current consumption for the legs' motors is an indicator of the generated torque. Hence, we also study the motor current for the robot's legs which enables us to assess the dynamics of the surface and determine the current environment.

The Aqua robot is equipped with a 3-axis Inertial Measurement Unit (IMU), measuring a_x, a_y, a_z . We are also tracing the rotation of the robot's body with a 3-rate gyroscope measuring roll ϕ_r , pitch ϕ_p and yaw ϕ_y . The six legs' motor current and the six legs' angle encoder also provide useful information for our task. For the purpose of our particular experiment, the sensor space is reduced to the acceleration measurements along 3 axes and the motor current estimators for six legs. This simplifies the classification dataset to better fit a predictive model. The proposed model is responsible for identifying the mapping relationship in order to detect the environment using the aforementioned sensory data and transition the gaits accordingly.

3.2.3 Deep learning model

We build a model that is capable of predicting whether the robot should switch its gait based on input sensory data. The model is required to determine the appropriate gait for the near future given the sensory data captured in the recent past.

The input to our model is a window of past states. It takes the acceleration along each of the 3 axis, leg motor currents, and the current gait identifier for the last T_1 seconds and predicts what is the gait that the robot will be using in some time in the future. Basically, the output is the mode that the robot will have switched to T_2 seconds in the future. The predictor is a fully-connected neural network with Relu activation functions. By assessing

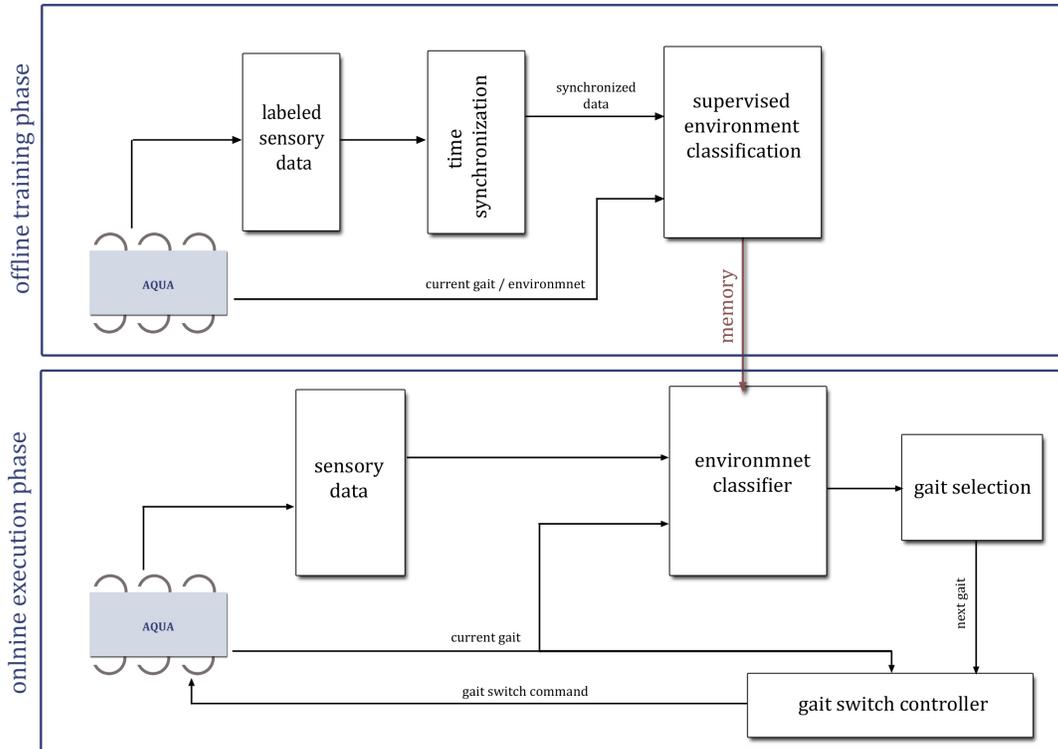


Figure 3.5: Overview flowchart of the gait switching algorithm consisting of two phases: offline supervised training phase and online execution phase.

the patterns in the input data in a recent time interval, and knowing the gait the robot is currently using, the model predicts the appropriate gait for the next steps. This way, the Aqua robot is able to determine the desired mode for the next few seconds and can transition its gaits at the right time to prevent any damage to the robot.

The proposed approach consists of two phases. In the offline supervised training phase, the model learns to identify the environment, which the robot is approaching, and the appropriate gait related to environment class based on the labeled sensory data. The training labeled data is gathered when navigating in each of the two environments while manually controlling the gait selection. The classifier parameters are stored in memory for use in real-time autonomous gait switching. During the online execution phase, sensory data is recorded and classified as one of the labeled environments and its corresponding

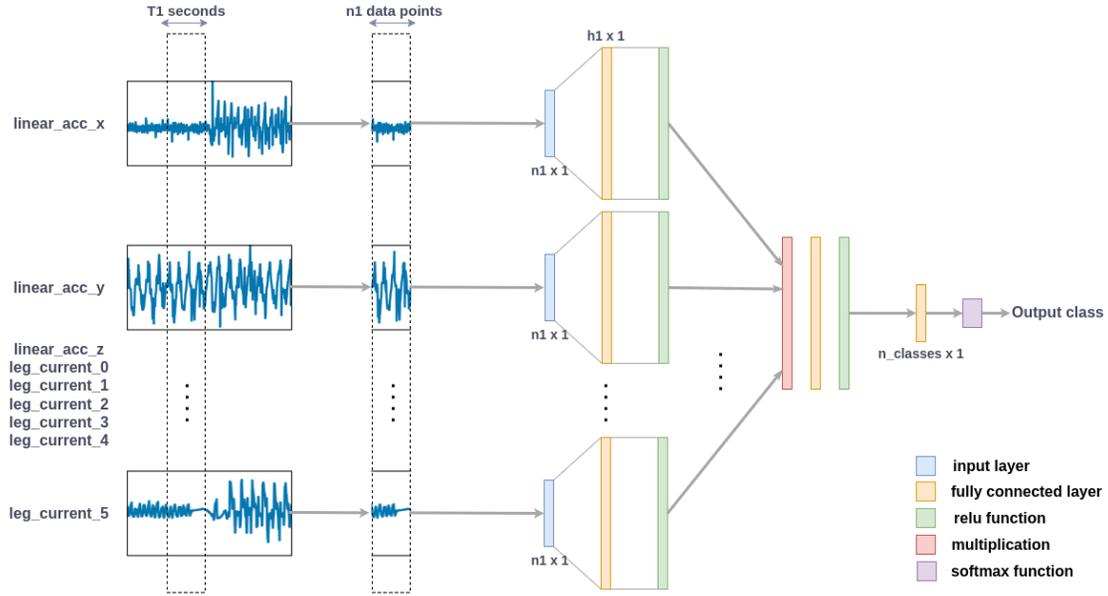


Figure 3.6: The gait selection Neural Network architecture consisting of fully connected layers and Relu activation function. The input to the model is a block of sensory data for the past T_1 seconds, predicting the gait in the next T_2 seconds.

gait. The selected gait is fed back into the robot to adapt the robot’s behavior in real-time. Figure 3.5 depicts the flowchart of the algorithm.

At each timestamp during the online execution phase, a block of sensory data from the last T_1 interval is formed. The block of sensory data for the past T_1 seconds, corresponding to n_1 data-points is fed into the classifier. The Neural Network’s architecture is shown in Figure 3.6. We train the model on the real-world dataset in 400 epochs. We consider cross entropy loss as our evaluation criterion and apply Adam optimizer to update the network parameters. The model identifies the environment that the robot is approaching by analyzing the sequence of recent past data. Once we have identified the terrain, the model determines the necessity to adapt gaits and signals the robot to switch accordingly if required. Hence, we ensure the adaptability of the robot’s gait to its environment in real-time. In our experiments, the training and testing data contain gait switches in similar environmental conditions and locations. An advantageous future direction would be to

attempt generalizing the model in order to make it more robust to the environment's conditions, such as steeper water and rougher wave conditions.

Chapter 4

Results and Discussion

In this chapter, we describe our experimental testbed and report the results from our analysis to show a performance comparison of our method to that in the existing literature. Similar to previous chapters, we start with presenting the details about the coverage planning problem and continue with the gait switching problem.

4.1 Coverage path planner

To validate our proposed algorithm in Section 3.1, and more importantly, as the first step toward deployment on real robot systems, we test our methodology in simulation. We are interested in assessing if the proposed coverage planning approach can quickly generate a coverage plan, the sensing configurations and the inspection path, for a mobile robot in complex and arbitrary shape environments.

The coverage planner algorithm is implemented in Python and runs on Robot Operating System (ROS). The algorithm is validated on a Gazebo simulator by Clearpath Robotics Inc¹. Our focus is on offline coverage planner where the map of the environment is already known in advance. In our simulator, the user defines the region of interest (ROI) by drawing a polygon within which we perform full coverage over the free

¹Clearpath Robotics Inc. <https://clearpathrobotics.com/>

space. Therefore, the area of interest can be any arbitrary shape area with holes as can be seen in Figure 3.1. The user may also set the desired visibility constraints or the survey radius for each ROI depending on the altitude of the robot's navigating plane. Then, our ROS package is responsible for computing the observation points and the coverage path to visit all the points in order to fully inspect the annotated region.

We use the ROS action server and client interface to communicate with the rest of the stack. Hence, we are able to provide periodic feedback during execution or preempt the callback if necessary. Upon receiving an action goal, defining the region of interest and the survey radius, we first determine survey point placement to fully cover the survey area for the given parameters. We start by segmenting the free space into so-called wide space and narrow space in order to treat each differently as discussed in 3.1.1. The skeleton of the narrow space is generated using the implementation of the Zhang-Suen thinning algorithm [18] in OpenCV. Figure 4.1 illustrates a map of the environment where the blue polygon is the user defined region of interest, the white area is the wide space, and the light gray area is the narrow space whose skeletal curve is marked in red. The image clearly shows how a region will be regarded differently depending on the survey radius. The image on the left is the segmentation and skeletonization outcome when the survey radius is set to 3 meters, while the survey radius for the image on the right is set to 6 meters.

The observation points are first placed on the skeletal curve which is proposed to be the most efficient place to capture as much area as possible near the obstacles and in narrow elongated areas. The points are placed at a constant distance from each other to allow some degree of overlap for image registration purposes once we are required to build a model of the environment. The distance is proportional to the survey radius and is configurable from the globally viewable ROS parameter server. Then the wide area is filled with observation points placed on a square grid, allowing a displacement by half a survey radius for inaccessible viewpoints. The points which are close to obstacles where the robot's footprint will not fit are called inaccessible points. Two instances of the

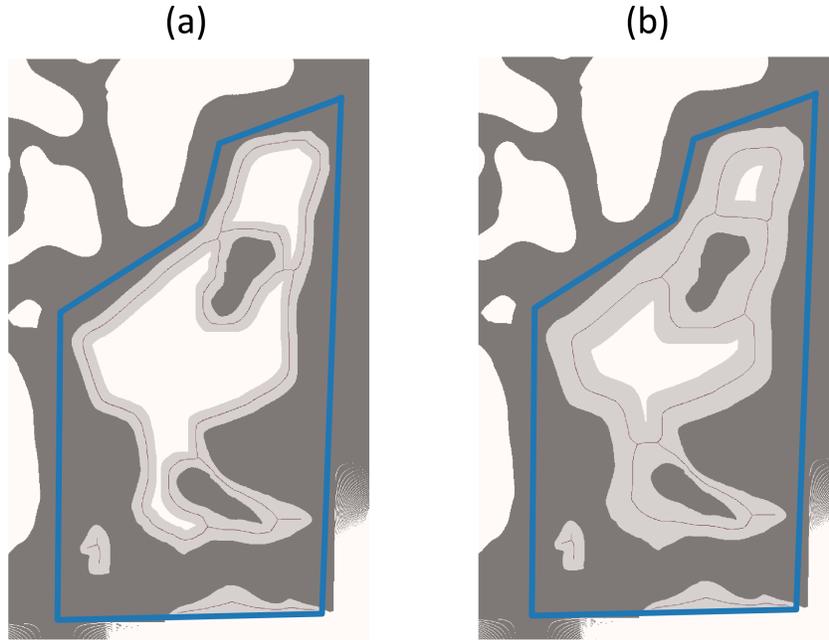


Figure 4.1: The effect of survey radius on segmentation and skeletonization of the free space. Survey radius is (a) 3 meters and (b) 6 meters in this experiments. In each map, the blue polygon is the user defined region of interest, the white area is the wide space, the light gray area is the narrow space whose skeletal curve is marked in red

generated viewpoints for two different regions of interest are provided in Figure 4.2. At each row, (a) is the free space segments and skeleton for the desired ROI, represented with a blue polygon. Point placements on the skeleton and on the relaxed grid are marked with blue and red dots respectively in (b). Image (c) shows the coverage disks of the generated viewpoints.

The optimization procedure relies on *findContours* function of OpenCV to detect any gaps in the coverage plan that are not negligible. We base our decision about the significance of the gap on the area of detected contours. Figure 4.3 demonstrates how rearranging the viewpoints surrounding the gap enhances the coverage in the intended areas. Gray circles are the coverage disks with a survey radius of 4 meters and (a) shows the coverage before optimization. In (b), (c) and (d) , each representing one step in iterative optimization, the bounding rectangle of each detected contour is shown in red. The big

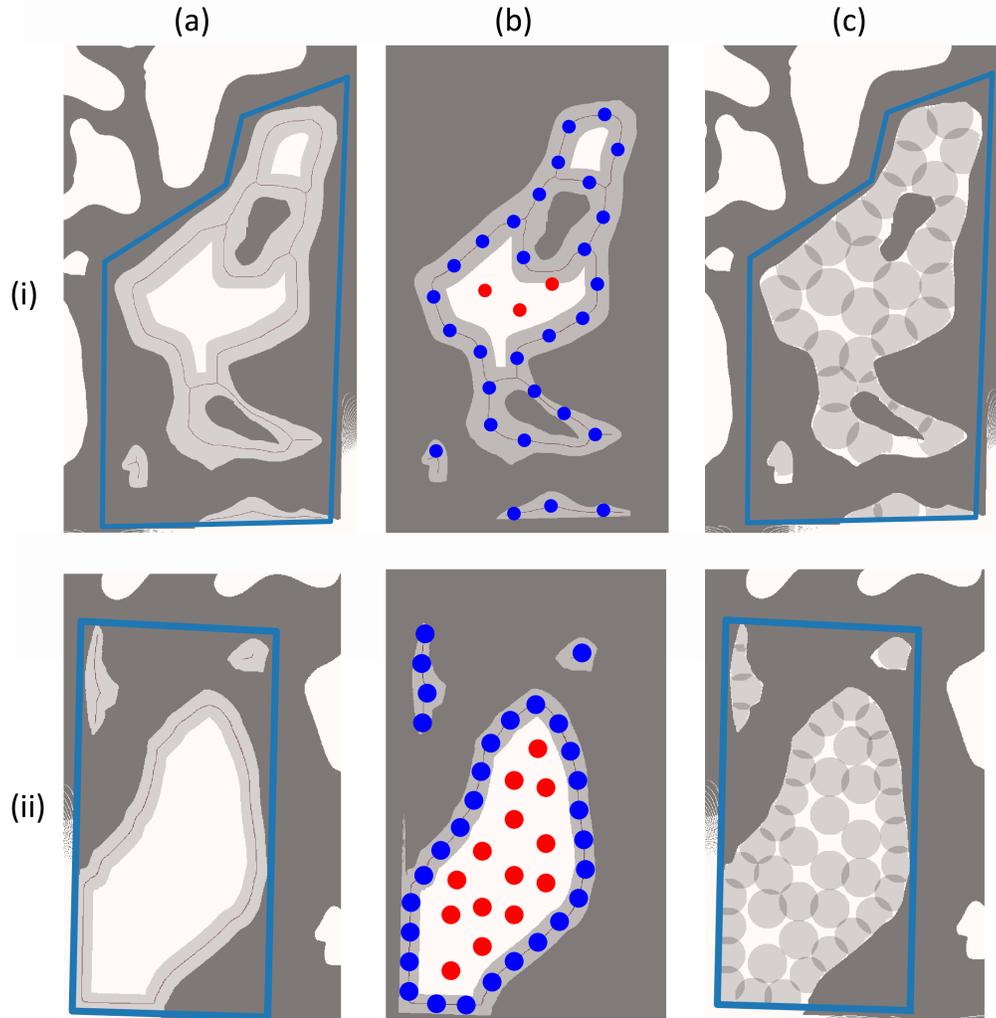


Figure 4.2: Observation point placements on environment segments before optimisation. (i) and (ii) illustrate the results for two different regions of interest. At each run, (a) represents the segmentation and skeletonization of the environment. (b) marks the point placement on the curve and in wide space with blue and red dots, respectively. Finally, (c) shows the coverage disks.

red dots are where additional viewpoints are required. The small red and pink dots indicate the rearranged viewpoints which are displaced from the pink dot position to the red dot position in order to cover the missing area. The rightmost image, (e) is the disk placements after optimizing the coverage plan. It has been observed that the optimization procedure significantly improves the total coverage while avoiding an unreasonable

increase in the number of viewpoints, and at the same time reduces the total overlap of coverage disks by eliminating redundant viewpoints.

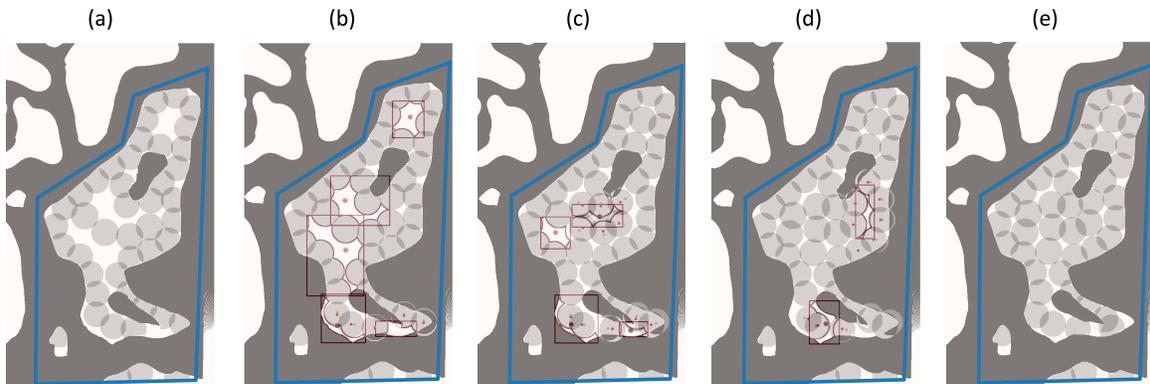


Figure 4.3: Coverage enhancements in missing areas with optimisation procedure.(a) is the generated coverage before optimisation. (b), (c) and (d) represent iterative coverage enhancement steps. (e) shows coverage density map after optimisation.

Once the ROS action result for placing observation points is received, a subsequent ROS action server is responsible for connecting the desired waypoints in a minimum-length tour. We rely on the Lin-Kernighan algorithm and perform k -opt moves in the edge-weighted graph of the waypoints, in order to heuristically solve an instance of Traveling Salesman Problem as described in Section 3.1.2. We do not restrict the path to return to the initial position. In case of multiple regions of interest, we encourage visiting all the waypoints within an ROI before traversing to the next, by penalizing inter ROI travel costs.

To demonstrate the capabilities of the proposed algorithm, we evaluated the algorithm within 4 environments and 20 regions of interest for the set of visibility disk radii 2, 4, 6, 10 meters. Our implementation handles range constraints and sight constraints by considering the occluded areas from each of the observation points. Figure 4.4 shows the coverage plan for three survey missions in ROS visualization tool, rviz. At each run, various visibility constraints are applied to each ROI, represented by purple polygons. The images on the left, displaying coverage disks in gray, serve as a coverage density map

showing how many times has every point on the workspace been covered. The images on the left provide the complete inspection path through all the waypoints, marked with green dots, for full coverage of the target surface. The robot starts from point S at the bottom left corner of the image.

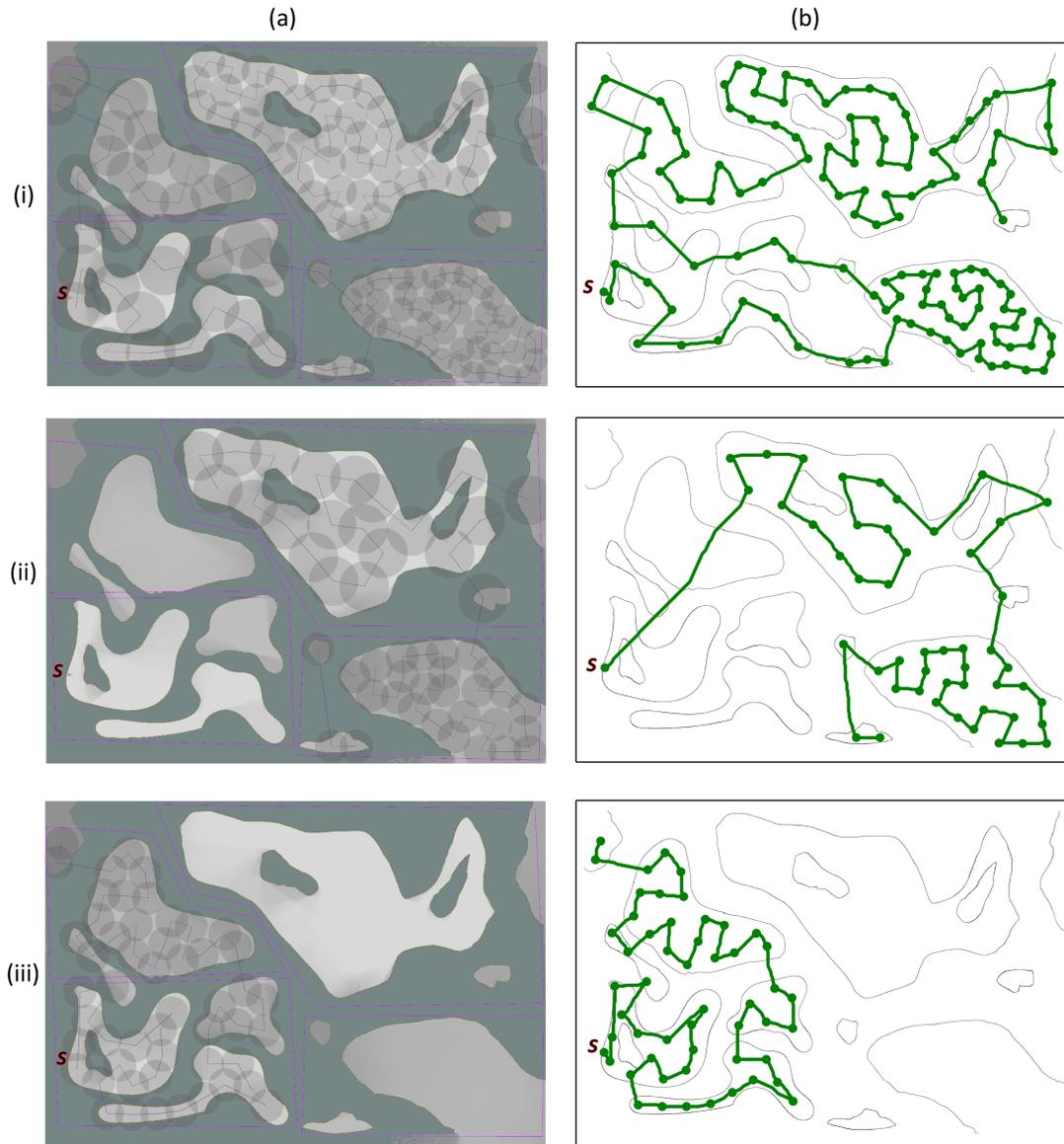


Figure 4.4: Final inspection path of the environment for several regions of interest. Each row represents one trial run. (a) depicts coverage density with gray coverage disks and the visiting order. (b) shows the coverage path for the robot to follow in order to fully inspect the area.

We examined two qualities of the coverage planner solution. We computed the total covered area from the generated viewpoint configuration and expressed the coverage ratio as the percentage of observed area to the total free area. The most efficient survey plan is when we cover each point in the environment exactly once. Hence, we also examine the amount of unnecessary coverage overlapping and define overlap ratio as the percentage of total repeatedly covered area to the total free area. Table 4.1 summarizes the average of coverage ratio and overlap ratio over 20 runs for 4 different visibility ranges for our algorithm. The results confirm that smaller survey radii lead to a higher coverage ratio at the expense of small-scale overlap. There is an unavoidable trade-off between overlapping area and missing area when covering a region with same-size circles; the smaller the overlap is, the larger the gaps become. This threshold is adjustable with the user’s preference from ROS parameter server. We compare our algorithm with the boundary placement algorithm presented by Faigl et al [27] described in Section 2.1.4. Their approach, having similar conditions and constraints to us, tries to place a number of waypoints at which the mobile robot performs scanning while staying stationary. Looking at the scores of the two approaches (Table 4.1), our algorithm outperforms the boundary placement algorithm. This is because the boundary placement algorithm does not take narrow areas into account and ends up having larger overlaps by placing the viewpoints too close to each other.

algorithm	visibility range(meters)	coverage ratio	overlap ratio
Our algorithm	2	93.6%	19.2%
	4	92.3%	14.8%
	6	92.7%	13.0%
	10	85.5%	9.5%
Boundary placement [27]	2	86.4%	22.6%
	6	80.3%	18.6%

Table 4.1: Quantitative performance of the coverage planner algorithm. Our algorithm outperforms the boundary placement algorithm by improving coverage while reducing the repeatedly scanned area at the same time.

We also performed time tracking and obtained traces for each scope of the algorithm during the run-time in order to investigate where the most time is spent in the program. We used the Scalopus system [100] to view traces in Chrome Tracing API. An example is provided in Figure 4.5 for coverage planning for two regions of interest yielding 153 observation points which indicates that the duration of point generation and route generation pieces are 5.49s and 15.65s respectively. These include the time it takes to interpret the action client’s request, repeatedly build and publish the action feedback messages, publish the result and terminate the connection. Exploring the traces for several runs, we figured out that skeletonization in point generation and path planning between pairs of nodes in route generation are the most time-consuming parts of the algorithm.



Figure 4.5: Computational time of the algorithm for an experiment with two ROIs and 153 observation points. The horizontal axis is the run-time and each color block represents a block of the algorithm. Skeletonization in point generation and path planning between pairs of nodes in route generation are the most time consuming parts of the algorithm.

Based on the experimental results, the presented algorithm appears to perform very well for any arbitrary shape environment with holes. The number of viewpoints is at an acceptable level and suitable locations while achieving more than 90% coverage in the target area. The results indicate that our algorithm performs better compared to the previous similar sensor placement-based approaches. When the sensors’ visibility area is round-shaped and we require covering the space with same size disks, it is acknowledged that some coverage overlap is inevitable; and even necessary for image post-processing steps. The algorithm is shown to be able to completely inspect the area and reduce the coverage overlap at the same time. The idea of placing the viewpoints on the skeletal curve close to boundaries and in corridors has also benefited route generation to reduce the length of the inspection tour.

4.2 Autonomous gait switching

In order to validate the effectiveness of the proposed gait controller algorithm, we performed experiments on real-world data to evaluate whether our technique can successfully determine the right time to switch the robot’s gaits. We carried out several experiments with the Aqua robot and collected sensory information while the robot navigates on different terrains and switches gaits. In this section, we present the gait transition results in order to demonstrate the capability of our algorithm.

The Aqua robot operates on Robot Operating System (ROS) and the proposed algorithm is implemented in python. The gait selection model is trained and tested on the gathered real-world data sources on a ROS server. Due to the restrictions on social gatherings during the Covid-19 pandemic, we were unable to test the proposed model on the robot in the field. However, we have validated the efficacy of our algorithm on simulation using real-world data from the robot running in the field.

4.2.1 Data collection

We have collected about 5 to 6 hours of data during the field trials conducted in Quebec, Ontario and Barbados on sandy and rocky beaches. In the data collection phase, the Aqua robot is teleoperated and receives the signal for mode change from a human operator. We have collected about 40 trial runs for entries and exits. At each run, the Aqua robot walks on the beach for a few seconds, approaches the water while in walking mode, and switches to *swim* mode when the water is deep enough. The robot navigates around the water for a short while and transitions back to *walk* mode when it has hit the shore. Figure 4.6 demonstrates two of the trial runs: 4.6a the Aqua robot transitions from the ground to the lake on a rocky beach and 4.6b transitions from the lake to the ground on a sandy beach.

The collected ROS bag files contain the linear accelerations, leg motor currents, forward and downward view images along with the manual mode change information. A



(a)



(b)

Figure 4.6: Gait transition experiments in the field. (a) transition from *swim* to *walk* when approaching to the shore on a rocky beach. (b) transition from *walk* to *swim* when entering the lake on a sandy beach.

rosvbag or bag is a file format in ROS for storing ROS message data. Images are stored at 10 FPS and IMU data is published at 50Hz. We applied a time synchronization between data nodes and did data segmentation to create a reliable dataset for gait adaptation experiments.

Furthermore, we have collected about an hour and a half of inopportune data, aiming to capture the data patterns in abnormal conditions when the Aqua robot switches gaits too early or too late. For example, the Aqua is in *walk* mode while in deep water when it

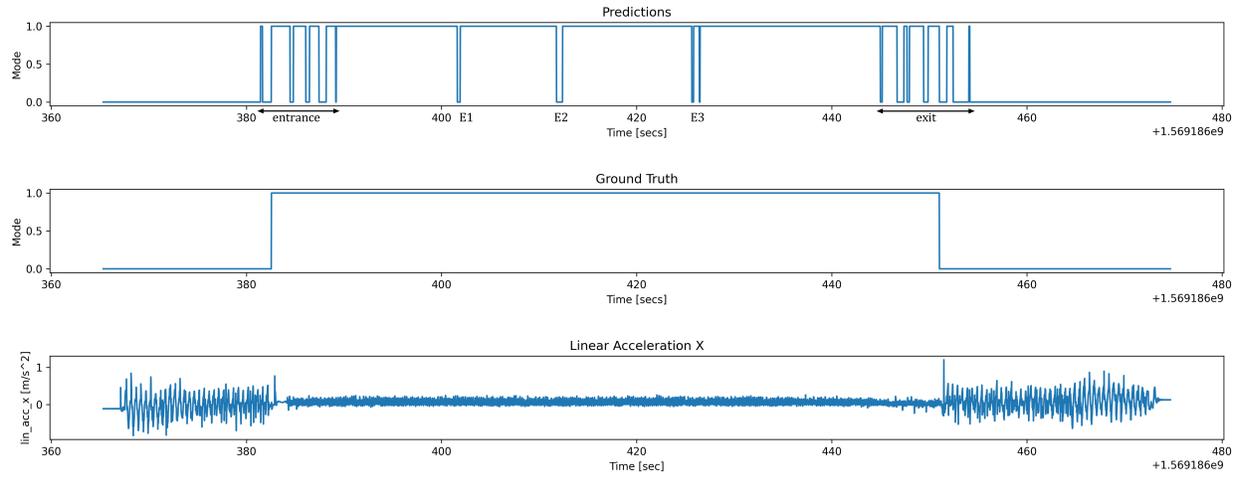
should be swimming. The ROS bag data files are accompanied by CSV files containing the timestamps in which the robot has switched gaits and the timestamps in which it should have switched gaits instead. The interval between the two timestamps would have been the right time to transition. We can benefit from the inopportune data sources to study the data patterns in this time interval and detect faults in gait selection. For example, we can learn how the peak in leg motor currents would change if the robot is struggling in *swim* mode although it has reached the shore. We believe this is a promising future direction to utilize the inopportune data, in order to recognize an improper gait selection and determine the necessity of a gait transition.

4.2.2 Evaluation

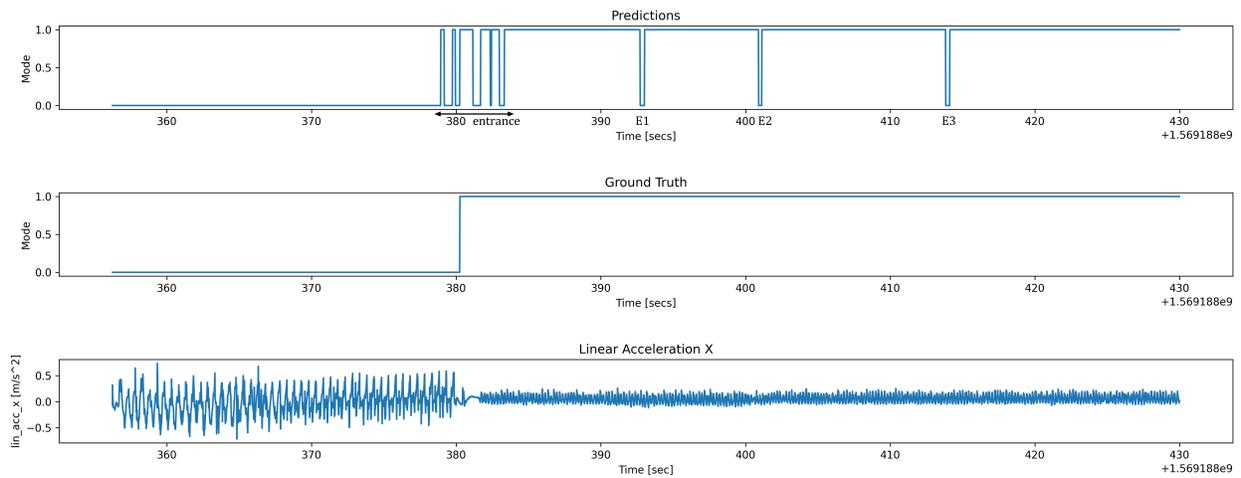
We built a fully-connected neural network and trained on the gathered dataset as described in Section 3.2.3. The model takes a window of the sensory data points, acceleration, and leg currents, during the last T_1 seconds and outputs the desired gait in T_2 seconds in the future. There are multiple choices for T_1 , T_2 , and for smoothing the gait mode signal. We evaluate different choices for the parameters of the model using accuracy on the validation set as the benchmarking metric.

Figure 4.7 shows the gait selection results on a model for which $T_1 = 10$ and $T_2 = 5$ with low-pass filtering applied to the mode predictions. In each sub-figure from top to bottom, the line graphs visualize the algorithm-selected gait, the actual gait, and linear acceleration versus time for one trial run. Sample 4.7a starts with the Aqua walking on the beach, entering the water, swimming around, and switching back to walking when exiting the lake. Sample 4.7b demonstrates one entrance to the water, alternating gait from walking to swimming. The actual gait is the mode that was manually controlled by a human user.

The graphs indicate that a switch signal has occurred several times in the entrance and exit intervals. We argue that repetition of the switch command in those intervals is not because of a missclassification error. The validation experiments were conducted by



(a)



(b)

Figure 4.7: Results of gait adaptation to environment with multiple transitions (two entry and one exit). The line graphs visualize the algorithm-selected gait, the actual gait and linear acceleration versus time for one trial run.

playing back the recorded ROS bag files from field trials to replicate a sample entry and exit run. Although the switch command has been delivered several times once the robot has reached the shallow water area, the robot was not changing gaits. Hence, the ground truth mode was consistently fed to the classifier as the current gait pattern and made the algorithm signal for changing gaits multiple times.

The gait adaptation algorithm performs efficiently well, and the switch decision was made at the correct time intervals when the robot is close to the seashore. We observe that there is a small classification error such as in time slots E_1, E_2, E_3 in Figure 4.7. Nevertheless, these misclassifications are a minor part of the data and most of the data is correctly identified. The proposed algorithm achieved success and the results prove its ability to distinguish between environments and switch gaits accordingly.

Chapter 5

Conclusion and Future Work

5.1 Summary

In this thesis we addressed two of the many challenges in automated outdoor environment exploration, particularly aiming for marine environment monitoring using an amphibious autonomous robot. First, we discussed coverage planning to generate an efficient inspection path. Second, we studied autonomous gait switching to ensure walking behavior adaptability for a fully autonomous coverage mission on both land and water.

We explored outdoor coverage path planning algorithms and presented a novel sensor placement-based coverage planner algorithm that outperforms a previous method. We demonstrated the process of generating a route that can be used to survey a region of interest with a complex shape. Our algorithm notably reduces the number of viewpoints, where the robot performs scanning, by considering occlusions, redundant areas, and undesirable repeated scans. The proposed coverage planner differentiates between narrow and wide areas and estimates the skeletal curve of the area of interest in order to determine the best set of sensor placements which ensures complete coverage while minimizing the overall route length. We verified the effectiveness of our algorithm through simulation.

Having in mind the necessity of locomotion in both ground and underwater environments for a fully autonomous robotic mission, we examined gait adaptation algorithms for legged robots. We aimed to enable the amphibious Aqua robot to identify the surface and adapt its leg movement patterns to its environment. Our algorithm relies on inertial measurements and leg actuator feedback to decide on the gait switching command. We performed validation on real-world data collected with the Aqua robot walking and swimming on sandy and rocky beaches.

In Chapter 1, we presented the problem description and the motivation behind our research. Our ultimate goal is to protect marine life and conserve coral reefs from destruction which necessitates frequent reef inspection missions. Automating seabed monitoring arises different challenges. Our research tackled coverage planning and gait adaptation problems described in Section 1.1. Chapter 2 provided an overview of previous research on coverage planning algorithms and gait adaptation techniques in Sections 2.1 and 2.2 respectively. Details of our proposed methods were explained in Chapter 3. Section 3.1 described our two-staged approach that intends to generate a coverage mission plan to capture data from a two-dimensional spatial field. We obtained the best set of sensing locations to fully observe the desired area and solved a Traveling Salesman Problem to determine the most efficient overall robot trajectory. In Section 3.2 we proposed a real-time terrain identification approach to distinguish between ground and water and switch the robot's gait accordingly. Chapter 4 reported implementation details, our experimental testbed, and evaluation findings.

5.2 Future directions

Here, we briefly discuss some extensions and additional research directions that we believe are promising.

We presented an offline coverage planner that creates the survey plan for an already known environment. Currently, if an unexpected obstacle is encountered along the tra-

jectory, the system updates the environment's map with the location and shape of the obstacle, annotates the already covered areas on the map, and then generates a new complementary survey plan to resume inspection of the remaining area. To make this approach more robust to dynamically changing environments with moving obstacles, the planner could be extended to a Next Best View (NBW) planner where the best sensing locations are determined during the survey execution. The idea is that at each step, the most informative sensing location with the highest information gain is selected locally instead of planning all the viewpoints prior to commencing the mission. Another direction to expand this research would be to deploy multiple data-sampling robots for environment exploration. The research question that we would like to answer in the future is how the team of robots can collaborate to achieve effective coverage with minimal communication between the robots.

The gait alternation algorithm currently identifies the robot's environment based on IMU measurements and leg motor currents. In order to further improve the autonomous gait switching results, we would like to experiment with other sources of data discussed in Section 2.2.2 and combine several sets of sensory information such as body rotation, tactile information, or visual appearance. Considering the fact that the input data to our model is time series, we would like to experiment with Recurrent Neural Networks (RNN) that are known to be successful in sequential data analysis. An RNN retains a memory of the its recent past steps through temporal feedback loops. In the future, we would also examine clustering techniques for terrain classification. This could enable us to discover new environments as they appear; for instance when the robot faces oysters or muddy beaches that it has never experienced before. We believe this should be beneficial because a more reliable and extensive identification could protect the robot from false gait selection and destructive damage to its legs.

Bibliography

- [1] ACAR, E. U., AND CHOSET, H. Critical point sensing in unknown environments. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)* (2000), vol. 4, IEEE, pp. 3803–3810.
- [2] ACAR, E. U., CHOSET, H., RIZZI, A. A., ATKAR, P. N., AND HULL, D. Morse decompositions for coverage tasks. *The international journal of robotics research* 21, 4 (2002), 331–344.
- [3] ACAR, E. U., CHOSET, H., ZHANG, Y., AND SCHERVISH, M. Path planning for robotic demining: Robust sensor-based coverage of unstructured environments and probabilistic methods. *The International journal of robotics research* 22, 7-8 (2003), 441–466.
- [4] ARAIN, M. A., CIRILLO, M., BENNETTS, V. H., SCHAFFERNICHT, E., TRINCAVELLI, M., AND LILIENTHAL, A. J. Efficient measurement planning for remote gas sensing with mobile robots. In *2015 IEEE International Conference on Robotics and Automation (ICRA)* (2015), IEEE, pp. 3428–3434.
- [5] ARKIN, E. M., FEKETE, S. P., AND MITCHELL, J. S. Approximation algorithms for lawn mowing and milling. *Computational Geometry* 17, 1-2 (2000), 25–50.

- [6] BAJRACHARYA, M., TANG, B., HOWARD, A., TURMON, M., AND MATTHIES, L. Learning long-range terrain classification for autonomous navigation. In *2008 IEEE International Conference on Robotics and Automation* (2008), IEEE, pp. 4018–4024.
- [7] BARRAQUAND, J., AND LATOMBE, J.-C. Robot motion planning: A distributed representation approach. *The International Journal of Robotics Research* 10, 6 (1991), 628–649.
- [8] BEDNAREK, J., BEDNAREK, M., WELLHAUSEN, L., HUTTER, M., AND WALAS, K. What am i touching? learning to classify terrain via haptic sensing. In *2019 International Conference on Robotics and Automation (ICRA)* (2019), IEEE, pp. 7187–7193.
- [9] BERMUDEZ, F. L. G., JULIAN, R. C., HALDANE, D. W., ABBEEL, P., AND FEARING, R. S. Performance analysis and terrain classification for a legged robot over rough terrain. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2012), IEEE, pp. 513–519.
- [10] BORMANN, R., JORDAN, F., LI, W., HAMPP, J., AND HÄGELE, M. Room segmentation: Survey, implementation, and analysis. In *2016 IEEE international conference on robotics and automation (ICRA)* (2016), IEEE, pp. 1019–1026.
- [11] BROOKS, C., IAGNEMMA, K., AND DUBOWSKY, S. Vibration-based terrain analysis for mobile robots. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation* (2005), IEEE, pp. 3415–3420.
- [12] BROOKS, C., IAGNEMMA, K., AND DUBOWSKY, S. Vibration-based terrain analysis for mobile robots. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation* (2005), IEEE, pp. 3415–3420.
- [13] BROWN, B. Coral bleaching: causes and consequences. *Coral reefs* 16, 1 (1997), S129–S138.

- [14] BURGARD, W., ET AL. Novelty detection and online learning for vibration-based terrain classification. *Intelligent Autonomous Systems 10: IAS-10* (2008), 16.
- [15] CANNY, J. F., AND LIN, M. C. An opportunistic global path planner. *Algorithmica* 10, 2 (1993), 102–120.
- [16] CARLSSON, S., JONSSON, H., AND NILSSON, B. J. Finding the shortest watchman route in a simple polygon. *Discrete & Computational Geometry* 22, 3 (1999), 377–402.
- [17] CHAKRABARTY, K., IYENGAR, S. S., QI, H., AND CHO, E. Grid coverage for surveillance and target location in distributed sensor networks. *IEEE transactions on computers* 51, 12 (2002), 1448–1453.
- [18] CHEN, W., SUI, L., XU, Z., AND LANG, Y. Improved zhang-suen thinning algorithm in binary line drawing applications. In *2012 International Conference on Systems and Informatics (ICSAI2012)* (2012), IEEE, pp. 1947–1950.
- [19] CHOSET, H. Coverage of known spaces: The boustrophedon cellular decomposition. *Autonomous Robots* 9, 3 (2000), 247–253.
- [20] CHOSET, H. Coverage for robotics—a survey of recent results. *Annals of mathematics and artificial intelligence* 31, 1 (2001), 113–126.
- [21] COOK, W. Concorde tsp solver. <https://www.math.uwaterloo.ca/tsp/concorde/index.html>, 2003.
- [22] CURRENT, J. R., AND SCHILLING, D. A. The covering salesman problem. *Transportation science* 23, 3 (1989), 208–213.
- [23] DHILLON, S. S., AND CHAKRABARTY, K. Sensor placement for effective coverage and surveillance in distributed sensor networks. In *2003 IEEE Wireless Communications and Networking, 2003. WCNC 2003.* (2003), vol. 3, IEEE, pp. 1609–1614.

- [24] DUDEK, G., GIGUERE, P., PRAHACS, C., SAUNDERSON, S., SATTAR, J., TORRES-MENDEZ, L.-A., JENKIN, M., GERMAN, A., HOGUE, A., RIPSAN, A., ET AL. Aqua: An amphibious autonomous robot. *Computer* 40, 1 (2007), 46–53.
- [25] ENDRES, S. C., SANDROCK, C., AND FOCKE, W. W. A simplicial homology algorithm for lipschitz optimisation. *Journal of Global Optimization* 72, 2 (2018), 181–217.
- [26] FABRIZI, E., AND SAFFIOTTI, A. Augmenting topology-based maps with geometric information. *Robotics and Autonomous Systems* 40, 2-3 (2002), 91–97.
- [27] FAIGL, J., KULICH, M., AND PŘEUČIL, L. A sensor placement algorithm for a mobile robot inspection planning. *Journal of Intelligent & Robotic Systems* 62, 3 (2011), 329–353.
- [28] FANG, C., AND ANSTEE, S. Coverage path planning for harbour seabed surveys using an autonomous underwater vehicle. In *OCEANS'10 IEEE SYDNEY* (2010), IEEE, pp. 1–8.
- [29] GABRIELY, Y., AND RIMON, E. An on-line coverage algorithm of grid environments by a mobile robot. In *2002 IEEE Conference on Robotics & Automation, Washington DC* (2002).
- [30] GALCERAN, E., AND CARRERAS, M. Efficient seabed coverage path planning for asvs and auvs. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2012), IEEE, pp. 88–93.
- [31] GALCERAN, E., AND CARRERAS, M. A survey on coverage path planning for robotics. *Robotics and Autonomous systems* 61, 12 (2013), 1258–1276.
- [32] GIGUERE, P. Unsupervised learning for mobile robot terrain classification. *PhD thesis* (2010).
- [33] GIGUERE, P., AND DUDEK, G. Clustering sensor data for autonomous terrain identification using time-dependency. *Autonomous Robot*, 26 (2009), 171–186.

- [34] GIGUERE, P., AND DUDEK, G. Surface identification using simple contact dynamics for mobile robots. In *2009 IEEE International Conference on Robotics and Automation* (2009), IEEE, pp. 3301–3306.
- [35] GIGUERE, P., AND DUDEK, G. A simple tactile probe for surface identification by mobile robots. *IEEE Transactions on Robotics* 27, 3 (2011), 534–544.
- [36] GIGUERE, P., DUDEK, G., PRAHACS, C., PLAMONDON, N., AND TURGEON, K. Unsupervised learning of terrain appearance for automated coral reef exploration. In *2009 Canadian Conference on Computer and Robot Vision* (2009), IEEE, pp. 268–275.
- [37] GIGUERE, P., DUDEK, G., SAUNDERSON, S., AND PRAHACS, C. Environment identification for a running robot using inertial and actuator cues. In *Robotics: Science and Systems* (2006).
- [38] GONZALEZ, R., AND IAGNEMMA, K. Deepterranechanics: Terrain classification and slip estimation for ground robots via deep learning. *arXiv preprint arXiv:1806.07379* (2018).
- [39] GONZÁLEZ-BANOS, H. A randomized art-gallery algorithm for sensor placement. In *Proceedings of the seventeenth annual symposium on Computational geometry* (2001), pp. 232–240.
- [40] HALATCI, I., BROOKS, C. A., AND IAGNEMMA, K. Terrain classification and classifier fusion for planetary exploration rovers. In *2007 IEEE aerospace conference* (2007), IEEE, pp. 1–11.
- [41] HAMEED, I. A. Intelligent coverage path planning for agricultural robots and autonomous machines on three-dimensional terrain. *Journal of Intelligent & Robotic Systems* 74, 3 (2014), 965–983.
- [42] HELSGAUN, K. An effective implementation of the lin–kernighan traveling salesman heuristic. *European journal of operational research* 126, 1 (2000), 106–130.

- [43] HELSGAUN, K. General k-opt submoves for the lin-kernighan tsp heuristic. *Mathematical Programming Computation* 1, 2 (2009), 119–163.
- [44] HERT, S., TIWARI, S., AND LUMELSKY, V. A terrain-covering algorithm for an auv. In *Underwater Robots*. Springer, 1996, pp. 17–45.
- [45] HOEGH-GULDBERG, O. Climate change, coral bleaching and the future of the world’s coral reefs. *Marine and freshwater research* 50, 8 (1999), 839–866.
- [46] HOEGH-GULDBERG, O., MUMBY, P. J., HOOTEN, A. J., STENECK, R. S., GREENFIELD, P., GOMEZ, E., HARVELL, C. D., SALE, P. F., EDWARDS, A. J., CALDEIRA, K., ET AL. Coral reefs under rapid climate change and ocean acidification. *science* 318, 5857 (2007), 1737–1742.
- [47] HOFNER, C., AND SCHMIDT, G. Path planning and guidance techniques for an autonomous mobile cleaning robot. *Robotics and autonomous systems* 14, 2-3 (1995), 199–212.
- [48] IAGNEMMA, K. D., AND DUBOWSKY, S. Terrain estimation for high-speed rough-terrain autonomous vehicle navigation. In *Unmanned Ground Vehicle Technology IV* (2002), vol. 4715, International Society for Optics and Photonics, pp. 256–266.
- [49] KANNAN, S. S., JO, W., PARASURAMAN, R., AND MIN, B.-C. Material mapping in unknown environments using tapping sound. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2020), pp. 4855–4861.
- [50] KAZAZAKIS, G. D., AND ARGYROS, A. A. Fast positioning of limited-visibility guards for the inspection of 2d workspaces. In *IEEE/RSJ International Conference on Intelligent Robots and Systems* (2002), vol. 3, IEEE, pp. 2843–2848.
- [51] KHALILI, M., MCCONKEY, K. T., TA, K., WU, L. C., VAN DER LOOS, H. M., AND BORISOFF, J. F. Development of a learning-based terrain classification framework

- for pushrim-activated power-assisted wheelchairs. In *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)* (2020), IEEE, pp. 4762–4765.
- [52] KHAN, A., NOREEN, I., AND HABIB, Z. On complete coverage path planning algorithms for non-holonomic mobile robots: Survey and challenges. *Journal of Information Science & Engineering* 33, 1 (2017).
- [53] KONG, Z., XU, M., WANG, X., ZHOU, Y., AND ZHANG, S. Gait planning and gait transition of amphihex-i. In *2014 IEEE/ASME International Conference on Advanced Intelligent Mechatronics* (2014), IEEE, pp. 66–71.
- [54] KRAGH, M., JØRGENSEN, R. N., AND PEDERSEN, H. Object detection and terrain classification in agricultural fields using 3d lidar data. In *International conference on computer vision systems* (2015), Springer, pp. 188–197.
- [55] LARSON, A. C., VOYLES, R. M., BAE, J., AND GODZDANKER, R. Evolving gaits for increased discriminability in terrain classification. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2007), IEEE, pp. 3691–3696.
- [56] LATOMBE, J.-C. *Robot motion planning*, vol. 124. Springer Science & Business Media, 2012.
- [57] LEE, T.-K., BAEK, S.-H., CHOI, Y.-H., AND OH, S.-Y. Smooth coverage path planning and control of mobile robots based on high-resolution grid map representation. *Robotics and Autonomous Systems* 59, 10 (2011), 801–812.
- [58] LIN, S., AND KERNIGHAN, B. W. An effective heuristic algorithm for the traveling-salesman problem. *Operations research* 21, 2 (1973), 498–516.
- [59] LOW, K. H., DOLAN, J. M., AND KHOSLA, P. Adaptive multi-robot wide-area exploration and mapping. In *Proceedings of the 7th International Joint Conference on Autonomous agents and Multiagent systems-Volume 1* (2008), pp. 23–30.

- [60] MAHÉO, A. Implementing lin-kernighan in python. <https://arthur.maheo.net/implementing-lin-kernighan-in-python/>, 2018.
- [61] MANDERSON, T., HIGUERA, J. C. G., CHENG, R., AND DUDEK, G. 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) (2018)*, IEEE, pp. 1885–1891.
- [62] MANDUCHI, R., CASTANO, A., TALUKDER, A., AND MATTHIES, L. Obstacle detection and terrain classification for autonomous off-road navigation. *Autonomous robots* 18, 1 (2005), 81–102.
- [63] MANJANNA, S. Multi-robot planning strategies for non-myopic spatial sampling (2021). *PhD thesis* (2021).
- [64] MANJANNA, S., AND DUDEK, G. Autonomous gait selection for energy efficient walking. In *2015 IEEE International Conference on Robotics and Automation (ICRA) (2015)*, IEEE, pp. 5155–5162.
- [65] MANJANNA, S., AND DUDEK, G. Data-driven selective sampling for marine vehicles using multi-scale paths. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2017)*, IEEE, pp. 6111–6117.
- [66] MANJANNA, S., DUDEK, G., AND GIGUERE, P. Using gait change for terrain sensing by robots. In *2013 International Conference on Computer and Robot Vision (2013)*, IEEE, pp. 16–22.
- [67] MANJANNA, S., KAKODKAR, N., MEGHJANI, M., AND DUDEK, G. Efficient terrain driven coral coverage using gaussian processes for mosaic synthesis. In *2016 13th Conference on Computer and Robot Vision (CRV) (2016)*, IEEE, pp. 448–455.

- [68] MCDANIEL, M. W., NISHIHATA, T., BROOKS, C. A., SALESSES, P., AND IAGNEMMA, K. Terrain classification and identification of tree stems using ground-based lidar. *Journal of Field Robotics* 29, 6 (2012), 891–910.
- [69] MORAVEC, H., AND ELFES, A. High resolution maps from wide angle sonar. In *Proceedings. 1985 IEEE international conference on robotics and automation* (1985), vol. 2, IEEE, pp. 116–121.
- [70] NELDER, J. A., AND MEAD, R. A simplex method for function minimization. *The computer journal* 7, 4 (1965), 308–313.
- [71] NIELSEN, L. D., SUNG, I., AND NIELSEN, P. Convex decomposition for a coverage path planning for autonomous vehicles: Interior extension of edges. *Sensors* 19, 19 (2019), 4165.
- [72] NIETO-GRANDA, C., ROGERS, J. G., TREVOR, A. J., AND CHRISTENSEN, H. I. Semantic map partitioning in indoor environments using regional analysis. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2010), IEEE, pp. 1451–1456.
- [73] OJEDA, L., BORENSTEIN, J., AND WITUS, G. Terrain trafficability characterization with a mobile robot. In *Unmanned ground vehicle technology VII* (2005), vol. 5804, SPIE, pp. 235–243.
- [74] OJEDA, L., BORENSTEIN, J., WITUS, G., AND KARLSEN, R. Terrain characterization and classification with a mobile robot. *Journal of field robotics* 23, 2 (2006), 103–122.
- [75] OKSANEN, T., AND VISALA, A. Coverage path planning algorithms for agricultural field machines. *Journal of field robotics* 26, 8 (2009), 651–668.
- [76] ONO, M., FUCHS, T. J., STEFFY, A., MAIMONE, M., AND YEN, J. Risk-aware planetary rover operation: Autonomous terrain classification and path planning. In *2015 IEEE aerospace conference* (2015), IEEE, pp. 1–10.

- [77] OTSU, K., ONO, M., FUCHS, T. J., BALDWIN, I., AND KUBOTA, T. Autonomous terrain classification with co-and self-training approach. *IEEE Robotics and Automation Letters* 1, 2 (2016), 814–819.
- [78] OTTE, S., LAIBLE, S., HANTEN, R., LIWICKI, M., AND ZELL, A. Robust visual terrain classification with recurrent neural networks. *Proceedings; Presses Universitaires de Louvain: Bruges, Belgium* (2015), 451–456.
- [79] PAPADAKIS, P. Terrain traversability analysis methods for unmanned ground vehicles: A survey. *Engineering Applications of Artificial Intelligence* 26, 4 (2013), 1373–1385.
- [80] PRATIHAR, D. K., DEB, K., AND GHOSH, A. Optimal turning gait of a six-legged robot using a ga-fuzzy approach. *AI EDAM* 14, 3 (2000), 207–219.
- [81] RAZALI, N. M., GERAGHTY, J., ET AL. Genetic algorithm performance with different selection strategies in solving tsp. In *Proceedings of the world congress on engineering* (2011), vol. 2, International Association of Engineers Hong Kong, pp. 1–6.
- [82] REINA, G., MILELLA, A., AND GALATI, R. Terrain assessment for precision agriculture using vehicle dynamic modelling. *Biosystems engineering* 162 (2017), 124–139.
- [83] REKLEITIS, I., DUDEK, G., AND MILIOS, E. Multi-robot collaboration for robust exploration. *Annals of Mathematics and Artificial Intelligence* 31, 1 (2001), 7–40.
- [84] REKLEITIS, I., LEE-SHUE, V., NEW, A. P., AND CHOSET, H. Limited communication, multi-robot team based coverage. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004* (2004), vol. 4, pp. 3462–3468 Vol.4.
- [85] REKLEITIS, I., NEW, A. P., RANKIN, E. S., AND CHOSET, H. Efficient boustrophedon multi-robot coverage: an algorithmic approach. *Annals of Mathematics and Artificial Intelligence* 52, 2 (2008), 109–142.

- [86] ROY, N., DUDEK, G., AND FREEDMAN, P. Surface sensing and classification for efficient mobile robot navigation. In *Proceedings of IEEE International Conference on Robotics and Automation* (1996), vol. 2, pp. 1224–1228 vol.2.
- [87] SADHUKHAN, D., AND MOORE, C. A. On-line terrain estimation using internal sensors. In *Florida Conf. on Recent Advances in Robotics* (2003), Citeseer, pp. 195–199.
- [88] SAHA, P. K., BORGEFORS, G., AND DI BAJA, G. S. Skeletonization and its applications—a review. *Skeletonization* (2017), 3–42.
- [89] SANCHO-PRADEL, D. L., AND GAO, Y. A survey on terrain assessment techniques for autonomous operation of planetary robots. *JBIS-Journal of the British Interplanetary Society* 63, 5-6 (2010), 206–217.
- [90] SANTOS, C. P., AND MATOS, V. Gait transition and modulation in a quadruped robot: A brainstem-like modulation approach. *Robotics and Autonomous Systems* 59, 9 (2011), 620–634.
- [91] SHERMER, T. C. Recent results in art galleries (geometry). *Proceedings of the IEEE* 80, 9 (1992), 1384–1399.
- [92] SHI, W., LI, Z., LV, W., WU, Y., CHANG, J., AND LI, X. Laplacian support vector machine for vibration-based robotic terrain classification. *Electronics* 9, 3 (2020), 513.
- [93] SHILL, J. J., COLLINS, E. G., COYLE, E., AND CLARK, J. Terrain identification on a one-legged hopping robot using high-resolution pressure images. In *2014 IEEE International Conference on Robotics and Automation (ICRA)* (2014), IEEE, pp. 4723–4728.
- [94] SHKURTI, F., XU, A., MEGHJANI, M., HIGUERA, J. C. G., GIRDHAR, Y., GIGUERE, P., DEY, B. B., LI, J., KALMBACH, A., PRAHACS, C., ET AL. Multi-domain monitoring of marine environments using a heterogeneous robot team. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2012), IEEE, pp. 1747–1753.

- [95] UĞUR, A., AND AYDIN, D. An interactive simulation and analysis software for solving tsp using ant colony optimization algorithms. *Advances in Engineering software* 40, 5 (2009), 341–349.
- [96] VALADA, A., AND BURGARD, W. Deep spatiotemporal models for robust proprioceptive terrain classification. *The International Journal of Robotics Research* 36, 13-14 (2017), 1521–1539.
- [97] VULPI, F., MILELLA, A., MARANI, R., AND REINA, G. Recurrent and convolutional neural networks for deep terrain classification by autonomous robots. *Journal of Terramechanics* 96 (2021), 119–131.
- [98] WALAS, K. Terrain classification and negotiation with a walking robot. *Journal of Intelligent & Robotic Systems* 78, 3 (2015), 401–423.
- [99] WALTER, W. G. An electro-mechanical animal. *dialectica* (1950), 206–213.
- [100] WANDERS, I. Scalopus. <https://github.com/iwanders/scalopus>, 2019.
- [101] WANG, P. *View planning with combined view and travel cost*. PhD thesis, School of Engineering Science-Simon Fraser University, 2007.
- [102] WANG, P., KRISHNAMURTI, R., AND GUPTA, K. View planning problem with combined view and traveling cost. In *Proceedings 2007 IEEE International Conference on Robotics and Automation* (2007), IEEE, pp. 711–716.
- [103] WANG, T.-H., SUNG, D.-G., AND LIN, P.-C. Terrain classification, navigation, and gait selection in a leg-wheel transformable robot by using environmental rgbd information. In *2018 International Automatic Control Conference (CACCS)* (2018), IEEE, pp. 1–1.
- [104] WEI-PANG, C., AND NTAFOSS, S. The zookeeper route problem. *Information Sciences* 63, 3 (1992), 245–259.

- [105] WEINGARTEN, J. D., BUEHLER, M., GROFF, R. E., AND KODITSCHKEK, D. Gait generation and optimization for legged robots. *2003 IEEE International Conference on Robotics and Automation (ICRA)* (2003).
- [106] WEINGARTEN, J. D., LOPES, G. A., BUEHLER, M., GROFF, R. E., AND KODITSCHKEK, D. E. Automated gait adaptation for legged robots. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004* (2004), vol. 3, IEEE, pp. 2153–2158.
- [107] WEISS, C., FECHNER, N., STARK, M., AND ZELL, A. Comparison of different approaches to vibration-based terrain classification. In *EMCR* (2007).
- [108] WU, X. A., HUH, T. M., MUKHERJEE, R., AND CUTKOSKY, M. Integrated ground reaction force sensing and terrain classification for small legged robots. *IEEE Robotics and Automation Letters* 1, 2 (2016), 1125–1132.
- [109] XIE, J., CARRILLO, L. R. G., AND JIN, L. An integrated traveling salesman and coverage path planning problem for unmanned aircraft systems. *IEEE control systems letters* 3, 1 (2018), 67–72.
- [110] YAN, Y., RANGARAJAN, A., AND RANKA, S. An efficient deep representation based framework for large-scale terrain classification. In *2018 24th International Conference on Pattern Recognition (ICPR)* (2018), IEEE, pp. 940–945.
- [111] ZELINSKY, A., JARVIS, R. A., BYRNE, J., YUTA, S., ET AL. Planning paths of complete coverage of an unstructured environment by a mobile robot. In *Proceedings of international conference on advanced robotics* (1993), vol. 13, Citeseer, pp. 533–538.
- [112] ZHANG, T. Y., AND SUEN, C. Y. A fast parallel algorithm for thinning digital patterns. *Communications of the ACM* 27, 3 (1984), 236–239.

- [113] ZHAO, K., DONG, M., AND GU, L. A new terrain classification framework using proprioceptive sensors for mobile robots. *Mathematical Problems in Engineering* 2017 (2017).
- [114] ZHU, D., TIAN, C., SUN, B., AND LUO, C. Complete coverage path planning of autonomous underwater vehicle based on gbnn algorithm. *Journal of Intelligent & Robotic Systems* 94, 1 (2019), 237–249.