SIMULTANEOUS EXPLORATION AND COVERAGE WITH MULTIPLE ROBOTS USING GENERALIZED VORONOI PARTITION.

A Thesis

Submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

by

Vishnu G Nair

(155109ME15P05)

DEPARTMENT OF MECHANICAL ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA

SURATHKAL, MANGALORE -575025

September, 2019

SIMULTANEOUS EXPLORATION AND COVERAGE WITH MULTIPLE ROBOTS USING GENERALIZED VORONOI PARTITION.

A Thesis

Submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

by

Vishnu G Nair

(155109ME15P05)

Under the guidance of

Dr. K.R.Guruprasad
Associate Professor

DEPARTMENT OF MECHANICAL ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA

SURATHKAL, MANGALORE -575025

September, 2019

DECLARATION

I hereby declare that the Research Thesis entitled **'Simultaneous Exploration and Coverage with Multiple Robots using Generalized Voronoi Partition'** which is being submitted to the **National Institute of Technology Karnataka Surathkal**, in partial fulfillment of the requirements for the award of the degree of **Doctor of Philosophy** in the Department of **Mechanical Engineering**, is a **bonafide report of work carried out by me**. The material contained in this Research Thesis has not been submitted to any University or Institution for the award of any degree.

Vishnu G Nair
Registration No.: ME15P05
Dept of Mechanical Engg

Place: NITK, Surathkal

Date:20/09/2019.

CERTIFICATE

This is to certify that the Research Thesis entitled **'Simultaneous Exploration and Coverage with Multiple Robots using Generalized Voronoi Partition'** submitted by **VISHNU G NAIR (Register number:155109ME15P05)** as the record of the research work carried out by him, is *accepted as the Research Thesis submission* in partial fulfilment of the requirements for the award of degree of **Doctor of Philosophy**.

Dr. K.R. Guruprasad
Research Guide
Associate Professor
Dept of Mechanical Engg.
NITK Surathkal 575025.

Chairman - DRPC
(Signature with Date and Seal)

Place: NITK, Surathkal

Date:

ACKNOWLEDGEMENTS

Grateful to God for blessing me with the opportunity, strength and patience to accomplish this task.

It is indeed a great privilege to express my deep sense of gratitude to my supervisor **Dr. K. R. Guruprasad**, Associate Professor, Department of Mechanical Engineering, National Institute of Technology, Surathkal, Karnataka, who suggested me this work. I am obliged to my supervisor for his invaluable encouragement and exemplary guidance throughout this thesis work.

I extend my sincere thanks and regards to the members of the Doctoral committee **Dr. P. Navin Karanth** (Associate Professor, Department of Mechanical Engineering) and **Prof. Ananthanarayana V.S** (Department of Information and Technology) for their guidance, timely suggestions, comments and incredible support throughout the research period.

With deep sense of regard I wish to thank **Prof. Shrikantha S. Rao**, Head of the Department of Mechanical engineering for the support provided for the successful completion of this research work.

The thesis would not have come to a successful completion without the help I received from **Prof. Vinod V Thomas**, Registrar Evaluations, Manipal Academy of Higher Education,Manipal, **Prof. D. Srikanth Rao**, Director Manipal Institute of Technology, Manipal Academy of Higher Education,Manipal and **Prof. Satish Shenoy B**, Head of Department of Aeronautical and Automobile Engineering Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal. I express my deepest gratitude for the support and guidance during my research tenure.

I would like to express my heartfelt thanks to all the staff members in the department of Mechanical Engineering, National Institute of Technology, Surathkal, Karnataka and Department of Aeronautical and Automobile Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal for their moral support and assistance at various stages of my research work.

I thank Nagarakshit, Arjun Sadananda, Minhaz Falaki, Akshar Padman and Sagar Kalburgi for their support in the fulfilment of this research.I am grateful to my fellow research scholars for the support and timely help especially Jeane M D'Souza and Soumya for their frequent motivation and help throughout my research period.

I would also like to thank my friend **Dr. Dileep M. V**, Postdoctoral researcher at Chungnam National university, South Korea for helping me with the proof reading and support he extended during the period of my research.

I wish to pay my utmost honor to my wife, parents, family members and friends. It is with their blessings, warmth and co-operation that this work is crowned with success.

Vishnu G Nair

Place: Surathkal.

Date:20/09/2019

ABSTRACT

The problem of area coverage by mobile robots is very useful in several applications such as vacuum cleaning, lawn mowing, landmine detection and de-mining, planetary exploration, etc. Using multiple robots to cover a specified region is expected to reduce the coverage time, apart from possible robustness to failure of a (or a few) robot(s). In this work we address a multi-robotic area coverage problem. When multiple robots need to cover a given area, the main concern is of avoiding repetitive coverage apart from complete coverage of the given area. Partitioning the area to be covered into cells and allotting one each cell to each of the robots for coverage solves the problem of duplicity, thus avoiding repetitive coverage, in a very simple and elegant manner. However, the spatial partitioning may lead to additional problems leading to either incomplete coverage or coverage overlap near the partition boundary, and possible non-contiguous partitioned cells in the presence of obstacles. Also, the coverage algorithms reported in the literature are either off-line, using complete prior knowledge about the arena, or online, using no a priori knowledge, but there is no provision for using any partial knowledge (of map) when available.

In this thesis we address a problem of coverage path planning for multiple cooperative autonomous mobile robots.

We consider a "partition and cover" approach to the multi-robotic coverage problem due to its inherent advantages of i) independent of the underlying single robot coverage algorithm, ii) reduced memory requirement due to spatial task partitioning, iii) minimal or no communication requirement during performance of the coverage task, and iv) no requirement of special collision avoidance again due the spatial task partitioning. Among the "partition and cover" approaches reported in the literature, we used Voronoi partition based coverage due to its main advantage of possible distributed implementation.

One of the challenges associated with a multi-robot coverage problem is uniform load distribution among the robots. In the context of a "partition and cover" strategy employed in this thesis, this problem boils down to uniform

partitioning assuming that the coverage load is proportional to the are of the coverage. This is a classical problem of equatable partitioning that is addresses in locational optimization or sensor coverage problems. In this work, we provide a very simple solution to this problem by using the concept of the centroidal Voronoi configuration used in the locational optimization/sensor coverage literature. We introduce the concept of deploying "virtual nodes" rather than the robots and partitioning the space based on the "virtual nodes" locations. With this, we avoid unnecessary robot motion (in the sense that motion without performing coverage). We demonstrate with examples that with this approach, the areas of all the cells are approximately same, thus ensuring a uniform coverage load distribution among the individual robots.

We propose Manhattan-VPC, a Manhattan distance based Voronoi Partition coverage algorithm that decomposes a $2D \times 2D$ gridded region completely avoiding partition boundary issues such as coverage gap and coverage overlap, that arise with the use of the standard Voronoi partition. Here, the robot footprint is assumed to be $D \times D$ square. We have established both by formal analysis and simulation and experiments with physical robots, that the proposed Manhattan-VPC provides complete and non-overlapping coverage even in the presence of simple obstacles and completely avoids the partition boundary induced coverage gap and overlap.

We also propose Geodesic-VPC, a Voronoi partition based coverage algorithm using the Geodesic distance in the place of the standard Euclidean distance. With this approach we ensure that the cells that individual robots have to cover are contiguous even in the presence of arbitrary obstacles. However, here, unlike in the case of Manhattan VPC (or the basic VPC), we assume that the map of the environment is available *a priori* to the planner.

We then combine the Manhattan metric over the $2D \times 2D$ grid and Geodesic metric and propose a GM-VPC algorithm. We establish both by formal analysis and simulation experiments that with the GM-VPC algorithm robots provide complete and non-overlapping coverage in the presence of arbitrary known obstacles.

Finally we combine exploration and coverage problems to address a novel SimExCoverage problem. Here, the primary task of the robots is coverage while it uses intermittent exploration to generate partial map that is used by coverage path planner. This approach combines the advantages of both the off-line and online coverage strategies. We first present a single robot SimExCoverage problem and then extend it to a multi-robotic scenario.

While the Manhattan-VPC and SimExCoverage algorithms are suitable for scenarios when map of the area is not available, the Geodesic-VPC and GM-VPC strategies are useful when map of the region is available.

We use a Boustrophedon-like coverage algorithm and the spanning tree based coverage algorithm which represent the approximate cellular decomposition based coverage algorithms and exact cellular decomposition based coverage algorithms reported in the literature as underlying single-robot coverage algorithms for demonstrating the proposed generalized Voronoi partition based coverage strategies and the SimExCoverage algorithms.

Keywords: Multi robot Coverage Path Planning, Partition and Cover, Simultaneous Exploration and Coverage, Voronoi Partitioning.

CONTENTS

List of Figures

CHAPTER 1

INTRODUCTION

Several real-world applications such as autonomous cleaning, lawn moving, land mine detection, spray painting, etc. require a robot to move a coverage tool over the entire area of interest. This problem is referred as area coverage, or complete coverage, or exhaustive coverage. Here, a robot having an associated tool of a given shape, often corresponding to the relevant sensor or actuator, must visit every point within a given bounded work-area, possibly containing obstacles. Since the tool size is typically much smaller than the work-area, the robots task consists of finding and moving along a path that will take the tool over the entire work-area. Thus coverage path planning (CPP) refers to the task of determining a path that passes over all points of an area of interest while avoiding the obstacles. A coverage algorithm or strategy is expected to cover an area completely, with minimal (or no) retraces (or overlap), while avoiding obstacles if any.

Multi-robotic systems have recently been used in several applications due to their ability of performing the assigned task in a more reliable and faster way. Multi robotic systems (MRS) are known for their robustness to failure of a few of the individual robots, apart from reduction in time to complete an assigned task due to load sharing by the individual robots. In a multi-robot complete coverage problem, the goal is to build efficient paths for each of the robots, which jointly cover the whole area. The challenge of using multiple robots to any problems such as complete coverage is, ensuring a cooperation between them in terms of avoiding duplication of the task by two or more robots, or a portion of task being not engaged by any of the robots. For example, in a coverage problem that we address in this work, two (or more) robots should not cover same region, or a region is left uncovered, both due to improper task allocation or coordination between the robots. In this thesis we address a problem of coverage path planning for multiple autonomous mobile robots.

1.1 COVERAGE, EXPLORATION, MAPPING AND LOCALIZATION

In this section we preview a few problems associated with robot path planning related to the CPP problem addressed in this thesis.

Coverage path planning, exploration, localization, and mapping, are a few fundamental problems associated with these applications. *Localization* refers to a problem of estimating the pose (position and orientation) of the mobile robot. This problem is non-trivial as standard positioning techniques such as odometry or GPS are not very accurate. *Mapping* is a process of obtaining the geometric map of a region of interest in terms of obstacle infested part and the free space. Typically these are stored as occupancy map. Simultaneous Localization and Mapping (SLAM) solves localization, and mapping problems simultaneously.

Exploration using mobile robots is another related problem, where the robots use their onboard sensors to obtain the *map* of an initially unknown area, in terms of the number, location, and shape of the obstacles. The information is typically stored in the form of an occupancy map. The purpose of map is to find the free areas within the region of interest where a mobile robot performing certain task can move freely. Most exploration strategies use a discretized space, and exploration is the process of identifying the nature of each cells, occupied or free, using the onboard sensors. The robots have to plan a path maximizing the information gain and minimize the time to obtain complete map of the environment.

Exploration vs Coverage: The terms exploration and coverage are used in the literature refers to different related problems, in many situations interchangeably. In this thesis, by *exploration* we refer to the problem of gathering information on presence or absence of obstacles, in an area of interest, while a *coverage* refers to the problem of making a coverage tool attached to the robot move through each and every point (or cell in a gridded region). While the purpose of exploration is mapping, coverage problem has many applications such as, vacuum cleaning, land-mine detection, lawn mowing, etc. Both exploration and coverage problems involve path planning. Off-line CPP algorithms use the available prior information of the environment to generate a coverage path (CP), while online CPP algorithms rely on onboard sensors to detect obstacles on the go and avoid them. In an *exploration* problem (Lumelsky et al. 1990*a*, Lee & Recce 1997, Yamauchi 1998, Gonzalez & Latombe 1998, Albers et al. 1999), the robot chooses an optimal point from where the exploration results in maximal information gain in terms of the map of the environment. The robot plans the path accordingly and performs exploration, typically using a long range sensor such as a Light Detection And Ranging (LIDAR), or vision-based sensors, to obtain a complete map of the environment. Though for different purposes, both online CPP algorithms and exploration algorithms detect obstacles (including the region boundary) in an environment. Further, exploration also requires the robot to cover (using typically a long rage sensor) the environment in order to obtain the complete map. Here by "cover" an area, we mean that all the points in the space should have come under its sensor range at some point of time. Because of this, the term "coverage" is used in the context of exploration and mapping problems. However, unlike in an ideal CPP problem, in the case of exploration problem, overlap in "sensor coverage" is acceptable. The purpose of coverage in a CPP problem is to serve (or gather information about) each point in the space, while in an exploration and mapping problem, is to obtain the complete map, not necessarily visiting every point in the region. Note that if a robot uses a small range sensor such as contact sensors then robot has to visit all the cells to successfully explore a region, and an online CPP algorithm is also capable of generating the map of the environment. Thus, in the limiting case (of sensor range), both coverage and exploration algorithms may behave in a similar way.

1.2 DESIRED CHARACTERISTICS OF COVERAGE PATH PLANNING ALGORITHMS

Coverage path planning (CPP) is a process of determining a path that passes over all the points in an area of interest. The main aim is to provide complete coverage with minimum or no retraces thereby saving the time for completion and energy. Apart from avoiding obstacles, a few desirable features of CPP algorithms are as follows:

- "Completeness" - Coverage must be complete such that the robot path must moves through all the points in the target area covering it completely.
- "Non-overlapping coverage" - Robot should cover the entire area without any overlapping paths.
- Simple motion (e.g., straight lines or circles) are preferred (for simplicity in control).
- An "optimal" path is desired under available conditions.
- Minimal energy usage under battery capacity constraints.

In the context of multi robot coverage the additional desirable features include:

- Uniform load distribution between the robots.
- Minimal communication between the robots.
- Minimal memory requirement.

1.3 THESIS OUTLINE

We provide a detailed review of the literature on single and multi-robot coverage path planning algorithms reported in the literature. We first provided a brief survey of robot coverage algorithms from an application perspective. We provide a survey of work on single-robot coverage path planning algorithms grouping them together as: those based on exact or approximate cellular decomposition, probabilistic strategies, and finally bio-inspired algorithms. We grouped the work on multi-robotic coverage in the literature into those extended from single-robot techniques, those which are inherently multi-robotic in nature, and "partition and cover" approaches. We then discuss the research gap and the motivation for the problem addressed in this thesis, and the contributions of the thesis.

In Chapter 3, we present an optimal deployment strategy to obtain a uniform Voronoi partition for the multi-robot coverage by introducing a concept of virtual nodes is presented. The virtual nodes are deployed into a centroidal Voronoi configuration, which is shown to be an optimal configuration in the context of sensor coverage in the literature. Instead of the robots getting deployed physically, the use of virtual nodes reduces the battery usage as well as coverage time. Further, the use of virtual nodes eliminates the coverage overlap issue since the physical robots move towards their respective Voronoi cells only at the end of partitioning process. With the help of illustrative examples, we have demonstrated that the proposed partitioning scheme provides an optimal partitioning in the sense of uniformly sized Voronoi cells to be covered by robots, leading to a uniform load distribution among the robots. This further reduces the time of completion of the coverage tasks as all the robots are utilized to same extent.

A "partition and cover" strategy for cooperative multi-robot coverage, using Voronoi partitioning scheme based on Manhattan distance metric in a gridded region is discussed in the Chapter 4. The region divided into $2D \times 2D$ grids, where $D \times D$ is the robot (coverage tool) footprint. This gridded region is partitioned using Manhattan distance-based Voronoi partitioning scheme. With the help of illustrative examples, we demonstrate that the proposed partitioning scheme eliminates partition boundary induced incompleteness and overlap in coverage, using existing single robot coverage strategies.

In Chapter 5, Geodesic-VPC, a "partition and cover" multi-robot area coverage strategy, using geodesic distance based Voronoi partitioning scheme, in the presence of obstacles is discussed. Each robot is allotted the task of covering a Geodesic Voronoi cell. Unlike the standard Voronoi cell (based on the Euclidean distance), the geodesic Voronoi cell is a contiguous region in the free space. As each robot covers the corresponding geodesic Voronoi cell, a passive cooperation between the robots is achieved, thus avoiding coverage duplication and without any requirement of extensive communication during the coverage process. Also, as each robot has to cater to a smaller region and does not require the information of the coverage map of other robots, the memory requirement is also greatly reduced. The proposed Geodesic-VPC is demonstrated using two single robot coverage algorithms, namely, boustrophedon coverage algorithm and STC algorithm, representing exact cellular decomposition based coverage algorithms and approximate cellular decomposition based coverage algorithms, respectively.

In Chapter 6 we propose a strategy which combines two generalization of Voronoi partition namely, Geodesic distance based Voronoi partition and Manhattan distance based Voronoi partition to address contiguity of partition in the presence of obstacles and avoid partition boundary induced coverage gap. The region is divided into $2D \times 2D$ grids, where D is the size of the robot footprint. With the help of illustrative examples, we have demonstrated that the proposed Geodesic-Manhattan Voronoi partition-based coverage (GM-VPC),can achieve complete and non overlapping coverage at grid level provided that the underlying single robot coverage path planning algorithm has similar property. We demonstrated using two representative single robot coverage strategies, namely, Boustrophedon coverage and Spanning Tree Coverage, first based on so called exact cellular decomposition, and the second based on approximate cellular decomposition, that the proposed partitioning scheme completely eliminates coverage gaps and coverage overlap.

In Chapter 7 a novel methodology "simultaneous exploration and coverage" for mobile robots, which combines exploration, mapping, and coverage path planning problems is discussed. The CPP generates robot path, while the exploration provides the map required for CPP. We proposed a SimExCoverage algorithm using a frontier based exploration strategy and off-line STC algorithm as a solution to the proposed SimExCoverage problem. The proposed SimExCoverage algorithm was described with an illustrative example. We then adapted the approach to a multi-robotic scenario as the focus of this thesis is on multi-robotic coverage problem.

While demonstrative results are provided in each chapter, we provide a detailed results and discussion in Chapter 8. We provide a graph-level simulation of the proposed multi-robotic coverage and SimExCoverage strategies along with realistic simulations using V-REP within Matlab environment and finally demonstrative experiments with physical **Fire bird V** mobile robots.

Chapter 9 summarizes the problem and key contributions along with a discussion on direction for future work.

CHAPTER 2

LITERATURE SURVEY

The problem of coverage path planning for mobile robots has attracted a large number of researchers as it useful in a variety of applications. In this chapter we review a few representative coverage path planning strategies for mobile robots reported in the literature. As the problem addressed in this thesis is multi-robotic coverage path planning, we focus only on the coverage path planning algorithms in general, and multi-robotic strategies in particular, reported in the literature. We provide a brief preview of work from the literature on other related problems such as robotic exploration, mapping, etc. as and when they are required in the subsequent chapters.

A survey of various coverage algorithms is provided in (Choset 2001) and (Galceran & Carreras 2013). Choset (2001) classifies these algorithms as (a) off-line or on-line, based on the availability of *a priori* information about the area (in terms of a map), (b) heuristic or provably complete, based on completeness of the coverage, and (c) approximate or exact, based on type of cellular decomposition used. Galceran & Carreras (2013) provide a more recent elaborate survey including both single as well as multi-robotic scenarios. Coverage algorithms have been grouped together as classical coverage algorithms based on exact cellular decomposition, those based on Morse decomposition, Landmark based topological coverage algorithms, grid based coverage algorithms, graph based strategies, coverage algorithms in 3D space, optimal coverage, and coverage under uncertainty.

2.1 APPLICATION OF MOBILE ROBOT COVERAGE ALGORITHMS

Coverage path planning for a mobile robots is useful in several applications. Though in general the underlying path planning strategy may have a similar foundation a specific application typically warrants a specific treatment of the problem. Here provide a brief account of work from the literature on coverage path planning that focus on a specific application. One of the main application where substantial work on coverage path planning for mobile robots is carried out in the literature is land mine detection and de-mining. While a single-robot de-mining strategy has been discussed in (Acar et al. 2003), multi-robotic autonomous landmine detection problem is discussed in (Prithviraj et al. 2012, Dasgupta et al. 2015). The strategy used in (Acar et al. 2003) is based on the boustrophedon decomposition based coverage (Choset 2000) along with probabilistic methods. The authors propose a new approach to handle sensor uncertainty that uses geometrical and topological features rather than sensor uncertainty. For scenarios where some *a priori* information about a minefield is available, the authors expedite the de-mining process by introducing a probabilistic method so that a de-mining robot does not have to perform exhaustive coverage models. Arkin et al. (2000, 1993), Cohen et al. (2008) address a problem of coverage path planning for an autonomous robotic lawn mowing application. Coverage strategy for a milling problem is addressed in (Arkin et al. 2000). In these papers, the authors address specific requirements for law mowing and milling problems. Atkar et al. (2005, 2009) provide coverage path planning algorithms for applications such as spray painting. Here the authors used a boustrophedon decomposition based coverage algorithm (Choset 2000) on curved a surface using the concept of geodesic distance (on a curved surface). A problem of vacuum cleaning is addressed in (Doty & Harrison 1993), and robotic cleaning in (Viet et al. 2013, Oh et al. 2004, Jager & Nebel 2002). Hameed (2014) addresses a problem of complete coverage for an agricultural application. Robotic cleaning problem is addressed in (Kabir et al. n.d.). Here the authors use oscillatory motion along with the typical coverage path.

2.2 SINGLE ROBOT COVERAGE

Though the problem addressed in this thesis is of coverage by multiple robots, we provide a brief survey of work of single robot coverage path planning for three reasons. First, the single robot coverage path planning algorithms provide a foundation for the coverage path planning in general. Second, most multi-robot coverage path planning algorithms are either motivated or extended from the single robot path planning algorithms. Third, the multi-robot coverage planning strategy that we use in this work belong to a class of algorithms that we call 'partition' and 'cover' strategies, where a single-robot coverage algorithm is used in a multi-robotic scenario. In the following, we provide a survey of work on single-robot coverage path planning algorithms grouping them together as: those based on exact or approximate cellular decomposition, probabilistic strategies, and finally bio-inspired algorithms.

2.2.1 Exact cellular decomposition based coverage

In these algorithms the region to be covered is decomposed into cells size of which is much larger than that of the robot. Robot covers each cell using simple scanning (to and fro) motions. Choset & Pignon (1997) use term 'Boustrophedon' for such to and fro motions. An adjacency graph of the decomposed cells is used to ensure that all the decomposed cells are covered.

Trapezoidal decomposition (Latombe 1991) or slab decomposition (Preparata & Shamos 1985) are earliest recorded methods of exact cellular decomposition for robotic coverage. Butler et al. (1999) uses a similar decomposition strategy in a rectilinear environment. Here authors use contact sensors for online decomposition and each cell is covered using a to and fro motion.

Choset & Pignon (1997) proposed an exact decomposition scheme based on the concept of Morse decomposition using critical points. This scheme is based on and extension of the trapezoidal decomposition schemes. This decomposition scheme is called 'Boustrophedon decomposition' as the authors use the Boustrophedon path for coverage of a single cell, and critical points are discovered based on such a path. This decomposition scheme is basically a Morse decomposition using a based straight line slice. A description of more general

Morse decomposition technique is provided in (Choset et al. 2000, Acar et al. 2002), though most practical Morse decomposition scheme is the so called Boustrophedon decomposition that uses straight line slicing.

Several work in the literature discuss the problem of finding critical point using the available sensors (Acar & Choset 2000, 2001, 2002, Acar et al. 2002, 2003, 2006). Garcia & de Santos (2004) address the problem related to critical point detection and on-line generation of the adjacency graph using sensors in a practical scenario that arises in unstructured environments. Authors claim that their paper proposes an enhancement of Choset's coverage method.

While theoretically the Boustrophedon decomposition scheme (a spacial case of the Morse decomposition scheme) cannot handle rectangular region/obstacles in general as it may lead to degenerative scenarios while finding the critical points, in (Batsaikhan, Janchiv & Lee 2013, Batsaikhan, Lee, Kim, Kim & Chong 2013) adapts boustrophedon decomposition based coverage scheme for rectangular obstacles.

Acar et al. (2006) address a coverage problem where the sensor range (footprint) is typically larger than the robot size/footprint, while in most coverage path planning algorithms, the sensor/coverage tool/robot footprint typically assumed to be of the size of the robot. The authors use generalized Voronoi Diagrams for path planning to navigate through the obstacles.

Das et al. (2014) integrate SLAM, exploration, and coverage without relying on GPS or magnetometer data. Here, the mapped region is optimally divided into polygons using dynamic programming/greedy cut decomposition approaches, and finally covering each polygon using a to and fro motion (Boustrophedon path).

In (Viet et al. 2014) the robot performs to and fro motion to cover an unvisited region until it reaches a critical point. The robot detects backtracking points based on its accumulated knowledge and determines the best backtracking point, which is used as the starting point of the next boustrophedon motion. It uses A^* search with smoothed path on tiling so as to reach the starting point with the shortest collision-free path. The authors carried out simulations and experiments in real workspaces using cleaning robots to demonstrate the proposed algorithm.

While the work such as in (Choset & Pignon 1997, Butler et al. 1999, 2000, Acar & Choset 2000, 2001, 2002, Acar et al. 2002, 2003, 2006, Garcia & de Santos 2004, Batsaikhan, Lee, Kim, Kim & Chong 2013, Viet et al. 2014) provide sensor based on online coverage algorithms, the work such as in (Choset 2000, Mannadiar & Rekleitis 2010, Xu et al. 2014, Ling & Stentz 2011) provide offline coverage algorithms. These algorithms cash on the availability of the map of the environment and attempt to provide an optimal coverage path by considering them as a Chinese postman problem.

Though most of the algorithms based on exact cellular discussed here provide a truly complete coverage, in general they can not guarantee non-overlapping coverage.

2.2.2 Approximate cellular decomposition based coverage

In the case of approximate cellular decomposition based coverage algorithms, the decomposed cells are of (or based on) the size of the robot (or the coverage tool footprint). As a cell may be partially occupied by the cell, which is typically not covered by the robot, the coverage here is only complete in resolution sense. In other words, unlike the exact cellular decomposition which decomposes the free space, the approximate cellular decomposition scheme decompose the total space including obstacles. The size of the robot footprint is assumed to be square of sides D.

Zelinsky et al. (1993) divide the region into $D \times D$ sized cells and use the concept of path transform to plan the coverage path. Gonzalez et al. (2003) provide a approximate cellular decomposition based coverage algorithm that uses spiral filling paths with backtracking. In (Gonzalez et al. 2005), the authors extend the BSA algorithm presented in (Gonzalez et al. 2003) by letting the robot cover even partially occupied $D \times D$ cells using wall following algorithm thereby providing a truly complete coverage. Unlike in the case typical approximate cellular decomposition based coverage algorithms that use square gridding, authors in (Oh et al. 2004) use triangular gridding for a problem of coverage by cleaning robots. With the utilization of triangular map, the robot can have 12 navigational directions which makes the path shorter and flexible. Choi et al. (2009) propose an online sensor based coverage algorithm for unstructured environments. The authors propose a special map coordinate assignment scheme based on active wall-finding using the history of sensor readings, to reduce the number of turns on the generated coverage path. An efficient path planner links the simple spiral paths using the constrained inverse distance transform, a concept introduced by the authors. Experiments on both simulated and real cleaning robots are carried out to demonstrate the proposed algorithm. In (Mao et al. 2009) authors combine the template-based and the heuristic coverage path planning approaches thereby attempting to generate optimal coverage paths. Authors claim that the algorithm can be implemented with the low-cost hardware such as the ultrasonic sensors, incremental encoders, DC motors, etc. The algorithm presented in (Lee et al. 2011) uses spiral coverage path, focusses on generation of smooth path to reduce acceleration and achieve faster search rather than avoiding overlap. In (Shivashankar et al. 2011) authors propose four strategies (Iterated Wave Front, Greedy-Scan, Delayed Greedy Scan and Closest-First Scan) for generating cost-effective coverage plans in real time for unknown environments using approximate cellular decomposition. Michel & McIsaac (2012) use the concept of distance transform and the coverage strategy is designed to minimize energy consumption. In (Shnaps & Rimon 2016) authors focus on a problem where the robot has limited battery (measured in terms of maximum distance it can travel when fully charged) and may have to come back to starting location, get charged and then continue coverage. Song & Gupta (2018) propose an coverage algorithm that is built upon the concept of an Exploratory Turing Machine (ETM). The algorithm uses a Hierarchical structure. Performance of the propose algorithm is validated by simulations on Player/Stage and actual experiments in a laboratory setting on autonomous vehicles. The authors also provide Guarantee of the completeness of coverage.

The approximate cellular decomposition based coverage algorithms discussed above (except that in (Oh et al. 2004)) use $D \times D$ grids. Such algorithms (including that in (Oh et al. 2004)) result in only resolution complete coverage (even when a complete coverage is guaranteed, which is not the case with all the algorithms discussed above), in the sense that only completely free $D \times D$ cells are covered and partially occupied $D \times D$ cells are left covered. An exception to this is the algorithm presented in (Gonzalez et al. 2005). In addition, none of the algorithms guarantee non-overlapping coverage.

In contrast, a class of coverage algorithms (Gabriely & Rimon 2001, 2003, Ranjitha & Guruprasad 2015a,b, 2016, Cohen et al. 2008) use $2D \times 2D$ gridding there by providing for return path for robots on reaching a dead end while covering. This results in a guaranteed non-overlapping coverage. Gabriely & Rimon (2001, 2003) first proposed a spanning tree based coverage (STC) algorithm on a $2D \times 2D$ gridded region. They have proposed off-line, online, and ant-like implementation of the proposed algorithms. The basic algorithm presented (STC) guaranteed complete coverage at a resolution of $2D \times 2D$ and a non-overlapping coverage. That is, all completely free $2D \times 2D$ cells are visited exactly once by the robot, while partially occupied $2D \times 2D$ cell is left uncovered. Modified STC algorithm known as Competitive-STC or Full-STC covers all fully free $D \times D$ cells. However, the coverage is no more guaranteed to be non-overlapping. Cohen et al. (2008) provides an improved version of the full-STC algorithm.

While the approximate cellular decomposition based coverage algorithms using $D \times D$ gridding can not guarantee non-overlapping coverage even in the absence of obstacles and those using $2D \times 2D$ grid can not provide complete coverage (in presence of obstacles), and in many situations none of them simultaneously guarantee complete (in true sense) and non-overlapping coverage, coverage algorithms presented in (Ranjitha & Guruprasad 2015a,b, 2016) use $2D \times 2D$ gridding and attempt to provide a truly complete coverage (by covering even partially occupied $D \times D$ cells) with reduced overlap.

It may be observed that a few coverage algorithms presented in the literature such as (Zelinsky et al. 1993, Michel & McIsaac 2012) are off-line in nature, all other (Gabriely & Rimon 2001, 2003, Gonzalez et al. 2003, 2005, Choi et al. 2009, Mao et al. 2009, Shivashankar et al. 2011, Lee et al. 2011, Shnaps & Rimon 2016, Song & Gupta 2018) can be implemented online. Note that any online algorithm can be implemented offline. In addition Gabriely & Rimon (2001, 2003) present ant-like versions of algorithms along with the off-line and online versions.

2.2.3 Other variations based on spatial decomposition

In (Lumelsky et al. 1990*b*, Hert et al. 1996) authors present a coverage algorithm that uses a decomposition of space where cells are fixed in width but the top and bottom (or the ceiling and floor) can have any shape. These schemes have been referred to as *semi approximate* scheme by Choset (2001) in his review paper. There are several algorithms that do not use an explicit decomposition of the space such as (Hameed 2014). Though the work presented in (Viet et al. 2013) uses Boustrophedon decomposition implicitly, it does not use decomposition explicitly, instead keeps track of backtracking points (which are actually the critical points).

2.2.4 Probabilistic coverage algorithms

In this class of algorithms in place of a structured or provably complete logic, probabilistic methods are utilized.

Healey et al. (1995) provide and analyze a random coverage strategy for "pick up and carry away" type unexploded ordnance clearance scenarios. The authors demonstrate that the number and locations of mine disposal areas can expedite the de-mining process. Gage (1995) characterizes random strategies by comparing them to complete or structured strategies. The author shows that the random strategies become as effective as complete coverage when a large number of robots are used or the accuracy of the detector degrades. In (Acar et al. 2003), the authors discussed the coverage path planning for de-mining applications. For scenarios where some a priori information about a minefield is available, authors expedite the de-mining process by introducing a probabilistic method so that a de-mining robot does not have to perform exhaustive coverage. Authors show that the use of complete approaches enables the creation of a filter to reject bad sensor readings, which is necessary for successful deployment of robots. A new approach to handle sensor uncertainty that uses geometrical and topological features rather than sensor uncertainty models has also been proposed by the authors in this work.

2.2.5 Bio inspired coverage algorithms

Another class of coverage algorithms that are presented in the literature may be grouped together as bio-inspired algorithms such as those using Artificial Neural Networks (ANNs), Genetic Algorithms (GA), etc.

Yang & Luo (2004) proposed a coverage path planning based using neural networks. Here the NN is used primarily for representing the map. The path generated is similar to that obtained by Boustrophedon decomposition based coverage algorithms with lower coverage overlap. Authors also extend the work to multi-robotic scenario and demonstrate the same using two robots. Qiu et al. (2006) propose a complete coverage path planning method for mobile robot in uncertain environments. The authors use NN to model environment/map, and 'rolling planning' and Heuristic searching algorithms for planning. Here too, the path generated is similar to that obtained by Boustrophedon decomposition based coverage algorithms as the authors use to and fro motion for coverage. Jimenez et al. (2007) proposed an optimal coverage strategy using the genetic algorithms using motion templates.

2.2.6 Miscellaneous

Bosse et al. (2007) provide a coverage algorithm integrating mapping into coverage (similar to that used in (Rekleitis, New, Rankin & Choset 2008) in a multi-robotic scenario) by moving the robot initially along the known boundary of the region using inward spiral motion. This map is used for coverage using inward spiral motion along with spiral shift using car-like robots with constrained turning radius. This algorithm does not use any spatial decomposition and is neither probabilistic nor guarantees complete and non-overlapping coverage. Hsu et al. (2014) present a coverage path planning algorithm comprising two factors, namely, low working time and high human safety. The optimality in terms of reducing turning time is addressed in the paper. A novel field method is used in this work to avoid collisions with stationary or moving obstacles. A tracking controller is also designed to track the generated optimal path. The proposed methodology works fine in obstacle free scenarios but the presence of obstacles results in path retrace issues.

Battery/energy constraint A few of the work on coverage path planning algorithms address a practically relevant scenario of limit on battery or energy such as those in (Michel & McIsaac 2012, Shnaps & Rimon 2016, Yazici et al. 2014). We have addressed these works in detail earlier and hence do not provide a description here.

There is a vast literature on single robot coverage strategies. We have provided only a representative work from the literature. A more detailed treatment is available in (Choset 2001) and (Galceran & Carreras 2013).

2.3 MULTI ROBOT COVERAGE

Multi robot systems (MRS) constitutes a group of robots working towards performing a common assigned task. MRS has several potential advantages over single-robot systems such as:

- It can achieve better overall performance in terms of the reduced total time required to complete a task or lower energy consumption of the individual robots.
- A MRS is typically robust to failure of a few individual robots.
- MRS can benefit from data fusion, information sharing among the robots, and fault-tolerance because of information redundancy. For example,

multiple robots can localize themselves more efficiently if they exchange information about their position whenever they sense each other.

- A MRS can result in lower cost. Using a number of simple robots can be simpler (to program), cheaper (to build) than using a single powerful robot (that is complex and expensive) to accomplish a task.

- Robots with diverse abilities can be combined together to deal with complex task, and one or several robots may fail without affecting the task completion.

Multi-robot area Coverage Path Planning(CPP) involves visiting every point within a given area by a team of mobile robots. Such tasks are typical to coordinated tasks such as robotic vacuuming, robotic demining or robotic rescue. Here it is sufficient if any one member of the team visits a particular point in the coverage area as repeated visits provides no additional information or value. Revisits are considered as overhead on the task completion.

Several Multi Robotic Coverage (MRC) algorithms have been presented in the literature. In (Galceran & Carreras 2013), an elaborate review on CPP algorithms is presented, including the multi-robotic scenarios. One of the main issues in MRC is the coordination between individual robots to ensure complete and non-repetitive coverage. In this work we group the MRC algorithms reported in the literature into three classes.

1. Coverage by Extended Single-Robot Techniques

2. Inherently Multi-robot coverage methods

3. Partition and Cover methods

In the first kind, a single robot coverage algorithm is extended to a multi-robotic scenario to obtain a MRC algorithm. The second group consists of algorithms designed specifically for multi robot coverage applications. In both these type of coverage algorithms, each robot has to communicate the region it already covered with other robots in the group, or a central server, in order to avoid duplication of task leading to repetitive coverage. Further, each robot (or a central server) has to store the coverage map (in the form of covered and yet to be covered cells) in its memory. This results in increased communication and spatial (memory) overhead. In the third group of algorithms, the area to be covered is decomposed into cells, and a cell or a group of cells is allocated to an individual robot for coverage. Each robot covers the allotted cell(s) using an existing single-robot coverage algorithm. Such a strategy results in passive cooperation, and hence, extensive communication between the robots (or to a central server) can be avoided. Each robot still has to store the coverage map of the allotted cells in its own memory. However, as area of the cell(s) allotted to an individual robot is smaller than that of the entire region, the memory requirement too is on the lower side. Another related problem addressed in the literature is of cooperative sweeping (Kurabayashi et al. 1996, Min & Yin 1998, Ahmadi & Stone 2006), where the robots have to visit each point in space multiple times may be with different frequency for tasks such as surveillance or cleaning.

In the following, we provide a survey of representative work from the literature on MRC algorithms based on the categories as discussed above.

2.3.1 Coverage by extended single robot strategies

As we have seen earlier, a vast literature is available on single robot coverage path planning algorithms/strategies. Many of these strategies may be suitably extended to multi-robotic scenario.

The Spanning Tree Coverage (STC)(Gabriely & Rimon 2001) algorithm has been extended to multi-robotic scenario in Hazon & Kaminka (2005), Hazon et al. (2006), Agmon et al. (2009) as multi-robot spanning tree coverage (MSTC) algorithm and (Zheng et al. 2005) as multi-robot forest coverage (MFC) algorithm. In (Agmon et al. 2009) authors create simultaneously multiple spanning trees incrementally (in online version), each starting from starting cell of containing a robot, over the $2D \times 2D$ gridded region such that every $2D \times 2D$ cell is part of exactly one spanning tree edge. Each robot plans a coverage plan though the $D \times D$ sub cells along one of the spanning tree, resulting in complete non-overlapping coverage of all sub cells corresponding to all completely free $2D \times 2D$ cells. The paper also provide a polynomial time tree construction algorithm for off-line coverage. In addition, the (online) solutions proposed in this paper guarantee robustness to failing robots. Senthilkumar & Bharadwaj (2010) extend the MSTC algorithm Agmon et al. (2009) as Simultaneous STC (S-STC), which is further extended as Extended Simultaneous STC (ES-STC) in (Senthilkumar & Bharadwaj 2012). The authors use ant-like robots here, where the robots leave traces which will aid in storing the coverage map and also communication between the robots.

While (Agmon et al. 2009) construct simultaneous spanning trees Zheng et al. (2005) provide a solution by construction a forest (a group of trees). While in Zheng et al. (2005) the authors consider a region of uniform traversability, in (Zheng & Koenig 2007) the algorithm is adapted to a region with non-uniform traversability.

Another single robot coverage strategy, the Boustrophedon decomposition based algorithm Choset (2000) is adapted/extended to a multi-robotic scenario in Kong et al. (2006), Rekleitis, New & Rankin (2008) as multi-robot Boustrophedon coverage algorithm. The concept of Boustrophedon (Morse) decomposition of the space based on the notion of critical points used in the single robot version forms the basis for the multi-robotic versions. In the multi-robotic scenario authors use two robots for exclusively for exploration which find all critical points and and decompose the region into cells, while other robots use these decomposition to perform coverage more or less in a similar manner as the case of the single robot Boustrophedon coverage algorithm. In a similar manner the single robot coverage strategy presented in (Butler et al. 1999) is extended for a multi-robotic system in (Butler et al. 2000).

In (Fazli et al. 2010, 2013, Yazici et al. 2014)) the generalized problem of coverage by a single robot with sensor range typically larger than the robot size addressed in (Acar et al. 2006) is extended to multi-robotic scenario. Yang & Luo (2004) provides a NN based approach for single robot coverage which is also extended into multi-robotic scenario, though the authors limit to two robots while illustrating the multi-robotic coverage strategy. The single robot coverage strategy provided in Viet et al. (2013) has been extended to a multi-robotic scenario in (Viet et al. 2014). Michel & McIsaac (2012) provide a both a single robotic coverage strategy and that extended for a multi-robotic scenario. Jonathan & Dirk (2012) proposed a formation based multi-robotic coverage strategy which may also be seen as extension of the corresponding single robot coverage strategy.

Observe that extension of a single robot coverage strategy into a multi-robotic coverage strategy may not be trivial. The extended multi-robotic coverage strategy may have some component of 'inherently multi-robotic' coverage or the 'partition' and 'cover' approaches.

2.3.2 Inherently multi-robotic strategies

Several multi-robotic coverage algorithms proposed in the literature are inherently multi-robotic in nature, in the sense that they are devised explicitly for multi-robotic systems. Wagner et al. (1999) propose a distributed coverage algorithm for ant robots using evaporating traces. Here the authors use the ability of a group of robots, that communicate by leaving traces, to perform an area coverage task. Batalin & Sukhatme (2006) use a probabilistic approach for multi-robotic coverage. The authors use local dispersion to achieve spreading of the robots and hence cover the given area. In Wilson et al. (2011) use information compression techniques to compress the coverage map of the individual robots and hence reduce the communication overhead. Samuel et al. (2009) proposed a strategy for collaborative coverage using a swarm of miniature robots. Here the authors address issues specific to miniature robots.

2.3.3 Partition and Cover Methodology

In both the classes of coverage algorithms discussed above, fundamentally two strategies are followed to ensure cooperation and avoid coverage overlap. In the first approach, the regions already covered by all the robots are stored in a central location, and the robots continuously communicate with the central information provider ensuring an indirect cooperation. This approach, apart from increased communication overhead and increased spatial (memory) complexity, may not be suitable in situation where a central information provider cannot be used. In fact, the approach is no more distributed. In the second approach, each robot should not only keep track of the region it has already covered, in order to avoid self redundant coverage, but also needs to communicate the covered region to other robots in the team. Though this is a distributed approach, it results in higher communication overhead along with high memory requirement for each robot. In Wilson et al. (2011) authors use information compression to reduce the communication overhead.

A simple and elegant technique to reduce the communication requirement is to use divide and conquer approach. Here, the region to be covered is divided into cells and each robot is allotted a cell or a group of preferably contiguous cells for coverage. This leads to a passive cooperation, requiring no communication between robots while performing coverage. Further, each robot solves a single robot coverage algorithm. In the following we preview work form the literature that follow this 'partition' and 'cover' approach to multi-robotic coverage.

Fontan & Mataric (1998) introduce the concept of territorial multi-robot task allocation for robots performing cleaning or collection tasks. Here the region to be covered is divided into cells (typically fixed rectangular strips) and task of covering a cell is allotted to robots. The territorial decomposition is performed off-line and may not take into account the initial position of the robots. Min & Yin (1998) also follow a similar approach a cooperative sweeping problem. The whole sweep area is segmented into a number of sub-areas with same area size by a task distribution module. The initial task distribution can be done either by a remote station or by a robot before mission execution. Each uncovered sub-area is called a 'task' here and is then assigned to a robot through communication. Authors achieve cooperative between robots so as to avoid collision when the robots move out of their own subarea to help others, through communication. Here the number of sub areas are typically larger than that of the number of robots. Hert & Lumelsky (1998) use polygonal area decomposition scheme for multi robot workspace division, where the number of cells is equal to the number of robots. In (Jager & Nebel 2002) authors divide the region to be cleaned (covered) into polygons and dynamically allocate a contiguous set of polygons to each robot as a task. Ahmadi & Stone (2006) propose a partition and sweep (repeated coverage) algorithm for multiple robots. In the multi-robot Boustrophedon decomposition based coverage algorithm (Kong et al. 2006, Rekleitis, New & Rankin 2008), the area is split into strips and each strip allotted to a robot. Maza & Ollero (2007) proposed a multi-UAV cooperative search strategy using polygonal area decomposition.

While in most *partition and cover* approaches, the region is decomposed into cells by a central computer, and not taking into account the current robot positions, authors in (Guruprasad et al. 2012) propose a Voronoi Partition-based Coverage strategy, where, the robots partition the region to be covered into Voronoi cells, considering their current location as nodes. This is amenable for a completely distributed implementation, as Voronoi cells can be computed in a distributed manner (Bash & Desnoyers 2007, Guruprasad & Dasgupta 2012*b*). In Guruprasad & Dasgupta (2012*a*), Hungerford et al. (2016) authors propose distributed repartitioning scheme to avoid certain problems with the Voronoi partitioning scheme in the presence of obstacles.

(Kapoutsis et al. 2017) propose an off-line partition and cover algorithm, DARP, to divide the area into equal sized regions and allot to each robots to cover. The proposed algorithm provides complete coverage with no path retrace issues. The authors use $2D \times 2D$ gridding and STC Gabriely & Rimon (2001) as the underlying single-robot coverage strategy.

2.3.4 Research gap and Motivation

In multi-robot coverage scenario, the coverage algorithms presented require computaional/memory overhead to ensure non redundant coverage. The partition and cover approaches eliminate the requirement of constant communication and higher memory. Among the partition and cover approaches, VPC has added advantages of having the provision for taking the initial position of the robots into consideration while partitioning and possibility of fully distributed implementation. Each robot has to only store the coverage map within the corresponding Voronoi cell in its memory. However, VPC has several shortcomings. First, the partitioning may result in non uniform load distribution among the robots due to unequal area of Voronoi cells. Second, presence of obstacle may result in non-contiguous Voronoi cells, affecting the coverage performance and also leading to possible collision between robots. Third, the partition boundary itself may lead to incomplete coverage and coverage overlap.

CPP algorithms reported in the literature are typically either off-line or online. The one of the key advantage of off-line algorithms is the absence of sensing and planning phase during coverage process, thus avoiding unnecessary frequent stops and turning, leading to faster coverage, while that of online coverage algorithm is of not requiring a priori map of the environment. There is no provision in the CPP algorithms reported in the literature to accommodate partial information (in form of map) about the environment.

This work attempts to address the shortcoming of VPC algorithms discussed above and also combine the advantages of off-line and online CPP algorithms by combining coverage and exploration problems.

2.4 CONTRIBUTION OF THE THESIS

In this thesis we address a problem of coverage path planning for multiple cooperative autonomous mobile robots.

We consider a "partition and cover" approach to the multi-robotic coverage problem dues to its inherent advantages of i) independent of the underlying single robot coverage algorithm, ii) reduced memory requirement due to spatial task partitioning, iii) minimal or no communication requirement during performance of the coverage task, and iv) no requirement of special collision avoidance again due the spatial task partitioning. Among the ‘partition’ and ‘cover’ approaches reported in the literature, we used Voronoi partition based coverage due to its main advantage of possible distributed implementation.

One of the challenges associated with a multi-robot coverage problem is uniform load distribution among the robots. In the context of a “partition and cover” strategy employed in this thesis, this problem boils down to uniform partitioning assuming that the coverage load is proportional to the are of the coverage. This is a classical problem of equatable partitioning that is addresses in locational optimization or sensor coverage problems. In this work, we provide a very simple solution to this problem by using the concept of the centroidal Voronoi configuration used in the locational optimization/sensor coverage literature. We introduce the concept of deploying “virtual nodes” rather than the robots and partitioning the space based on the “virtual nodes” locations. With this, we avoid unnecessary robot motion (in the sense that motion without performing coverage). We demonstrate with examples that with this approach, the areas of all the cells are approximately same, thus ensuring a uniform coverage load distribution among the individual robots.

We propose Manhattan-VPC, a Manhattan distance based Voronoi Partition coverage algorithm that decomposes a $2D \times 2D$ gridded region completely avoiding partition boundary issues such as coverage gap and coverage overlap, that arise with the use of the standard Voronoi partition. Here, the robot footprint is assumed to be $D \times D$ square. We have established both by formal analysis and simulation and experiments with physical robots, that the proposed Manhattan-VPC provides complete and non-overlapping coverage even in the presence of simple obstacles and completely avoids the partition boundary induced coverage gap and overlap.

We also propose Geodesic-VPC, a Voronoi partition based coverage algorithm using the Geodesic distance in the place of the standard Euclidean distance. With this approach we ensure that the cells that individual robots have to cover are contiguous even in the presence of arbitrary obstacles. However, here, unlike in the case of Manhattan VPC (or the basic VPC), we assume that the map of the environment is available *a priori* to the planner.

We then combine the Manhattan metric over the $2D \times 2D$ grid and Geodesic metric and propose a GM-VPC algorithm. We establish both by formal analysis and simulation experiments that with the GM-VPC algorithm robots provide complete and non-overlapping coverage in the presence of arbitrary known obstacles.

Finally we combine exploration and coverage problems to address a novel SimExCoverage problem. Here, the primary task of the robots is coverage while it uses intermittent exploration to generate partial map that is used by coverage path planner. This approach combines the advantages of both the off-line and online coverage strategies. We first present a single robot SimExCoverage problem and then extend it to a multi-robotic scenario.

While the Manhattan-VPC and SimExCoverage algorithms are suitable for scenarios when map of the area is not available, the Geodesic-VPC and GM-VPC strategies are useful when map of the region is available.

We use a Boustrophedon-like coverage algorithm and the spanning tree based coverage algorithm which represent the approximate cellular decomposition based coverage algorithms and exact cellular decomposition based coverage algorithms reported in the literature as underlying single-robot coverage algorithms for demonstrating the proposed generalized Voronoi partition based coverage strategies and the SimExCoverage algorithms.

CHAPTER 3

CENTROIDAL VORONOI PARTITIONING USING VIRTUAL NODES FOR PARTITION AND COVER APPROACH

As discussed in the previous chapter, one of the problems associated with "partition and cover" approach, in general, and the Voronoi partition based coverage, in particular, is uniformity of partitioning and hence in load distribution among the individual robots performing cooperative coverage. In this chapter we address this problem and provide a Voronoi partitioning strategy based on the concept of centroidal Voronoi configuration.

3.1 VORONOI PARTITIONING

First we preview the concept of Voronoi partition used in this thesis. Voronoi partition named after Georgy Voronoi (Voronoi 1908) , also called Dirichlet tessellation(named after Gustav Lejeune Dirichlet), is a widely used scheme of partitioning a given space based on the concept of "nearness" of points in a set to some finite number of predefined locations in the set (Voronoi 1908, Dirichlet 1850). This concept finds application in many fields such as CAD, image processing (Kosmatopoulos & Christodoulou 1996, Arbelaez & Cohen 2003) and sensor coverage (Cortes et al. 2004). In this we work use the Voronoi decomposition scheme to partition the region to be covered.

By a *partition* of a set X we mean a collection of subsets W_i of X with disjoint interiors such that their union is X itself. Let $Q \subset \mathbb{R}^d$, be a convex polytope. Let $P = \{p_1, p_2, \ldots, p_N\}$, be a finite set of nodes, or generators, or sites, $p_i \in Q$. The *Voronoi partition* generated by P with respect to the Euclidean norm is the collection $\{V_i(P)\}_{i\in\{1,2,\ldots,N\}}$ defined as,

$$V_i(P) = \{q \in Q | \parallel q - p_i \parallel \leq \parallel q - p_j \parallel, \forall p_j \in \mathcal{P}\}$$

The Voronoi cell V_i is the collection of those points which are closest (with respect to the Euclidean metric) to p_i compared to any other point in $\mathcal{P}$. The boundary of each Voronoi partition is the union of a finite number of line segments forming a closed C^0 curve. The boundary of each Voronoi partition is the union of a finite number of line segments forming a closed curve C^0. The intersection of any two Voronoi partitions is either null, a line segment, or a point. Each of the Voronoi cells is a topologically connected non-null set. Basic components of the Voronoi partition are

1. A space which to be partitioned.
2. A set of sites or nodes or generators.
3. A distance measure such as the Euclidean distance.

The Voronoi partition is generalized in a variety of ways (Arbelaez & Cohen 2003, Okabe et al. 2000, Aurenhammer 1991). In this thesis, we use two generalizations of Voronoi partition, namely, that based on Manhattan distance and Geodesic distance. We discuss more about the generalizations used in subsequent chapters.

3.2 PROBLEM SETTING

We consider a problem of N robots cooperatively covering a region of interest Q. The multi-robot coverage strategy is expected to provide a complete and non-overlapping coverage of Q. We use $P = \{p_1, p_2, \ldots, p_N\}$ to represent the configuration of the robots, where p_i represents the position of the ith robot. We use Voronoi partitioning technique to partition Q into Voronoi cells and allot each cell to the corresponding robots for coverage Guruprasad et al. (2012). Here, the main problem addressed is to ensure a more uniform partitioning of Q in terms of the area of the Voronoi cells. Ideally, we expect an equitable partition. That is, area of all the cells is same. But practically, we try to achieve an approximately equitable Voronoi partition. In order to achieve a more uniform partitioning it is important to select the location of nodes based which the Voronoi partition is generated. Further, the nodes should be associated with robots. That is, each node

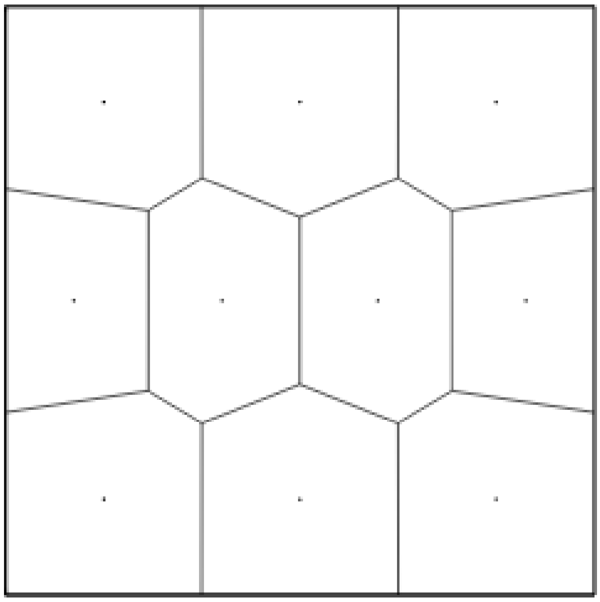


Figure 3.1: A centroidal Voronoi configuration with the corresponding Voronoi partition.

represents a unique robot. Thus problem now reduces to selection and placement of the nodes used for generating the Voronoi partition of Q.

3.2.1 Centroidal Voronoi using Virtual Nodes

In centroidal Voronoi configuration, the nodes which generate the Voronoi cells are located at the centroid of the respective Voronoi cells. This concept was proposed in Du et al. (1999). Figure 3.1 illustrates a centroidal Voronoi configuration. It may be noted that a centroidal Voronoi configuration results in more uniform partitioning. This concept has been used in several locational optimization, facility location Okabe et al. (2000), optimal sensor deployment Cortes et al. (2004) and multi-robot deploymentGuruprasad & Ghose (2011). In these problems, the nodes are made to move toward the respective centroids using a gradient based proportional control law (Lloyd's algorithm) while the Voronoi cells and hence the centroids are recomputed as the nodes move. Eventually, the nodes reach the centroid of the respective Voronoi cells, asymptotically. The formulation and theoretical results can be found in Cortes et al. (2004) and Guruprasad & Ghose (2011, 2013).

In Cortes et al. (2004), Guruprasad & Ghose (2011, 2013) the robots themselves are deployed into the centroidal Voronoi configuration using Lloyd's law. In the context of multi-robotic coverage problem we address in this thesis, such a movement of robot without performing coverage is equivalent to coverage overlap, as once deployed and partitioned, the robots may have to move through these paths while performing coverage. We introduce a concept of *virtual nodes* in this work. Initially the virtual nodes are located at the physical location of the robots. A Voronoi partitioning is generated, based on this initial configuration of the virtual nodes. Once the partitioning is established, the virtual nodes are moved toward the centroids of the respective Voronoi cells, in small steps (Lloyd-like control law). However, the robot do not move physically. During this motion of the virtual nodes, the topology of the initial Voronoi cells changes. So the new centroids corresponding to the present virtual node locations are generated. The process continues until the virtual nodes reaches the centroids of their respective Voronoi cells. Once the virtual nodes reaches the respective centroids, the partitioning so obtained will be the optimal in terms of the coverage effectiveness, implying that the areas of the Voronoi cells are more or less same. Once this final partition is established, the corresponding Voronoi cells are then allotted to the robots of the respective virtual nodes. The strategy followed by each virtual node is given in algorithm below.

Result: Virtual nodes at the centroids,C_{V_i}

while *Compute Voronoi cell* V_i **do**

    Compute centroid C_{V_i} of V_i;

    Move virtual node towards C_{V_i};

    if *position of the virtual node* ,$p_i = C_{V_i}$ **then**

        Go to End;

    else

        Go to while loop;

    end

end

Algorithm 1: Deployment of virtual nodes at centroids

The uniformity in area allotment to each robots can be calculated using

$$\eta = max(A_{ri}/((A(Q)/N)) \tag{3.1}$$

Here,A_{ri} refers to the area covered by i^{th} robot (that is the area of the i^{th} Voronoi cell),$A(Q)$ is the total area to be covered and N is the number of robots in multi-robot system. For optimal partitioning $\eta = 1$. Suboptimal solutions are obtained when $\eta > 1$.

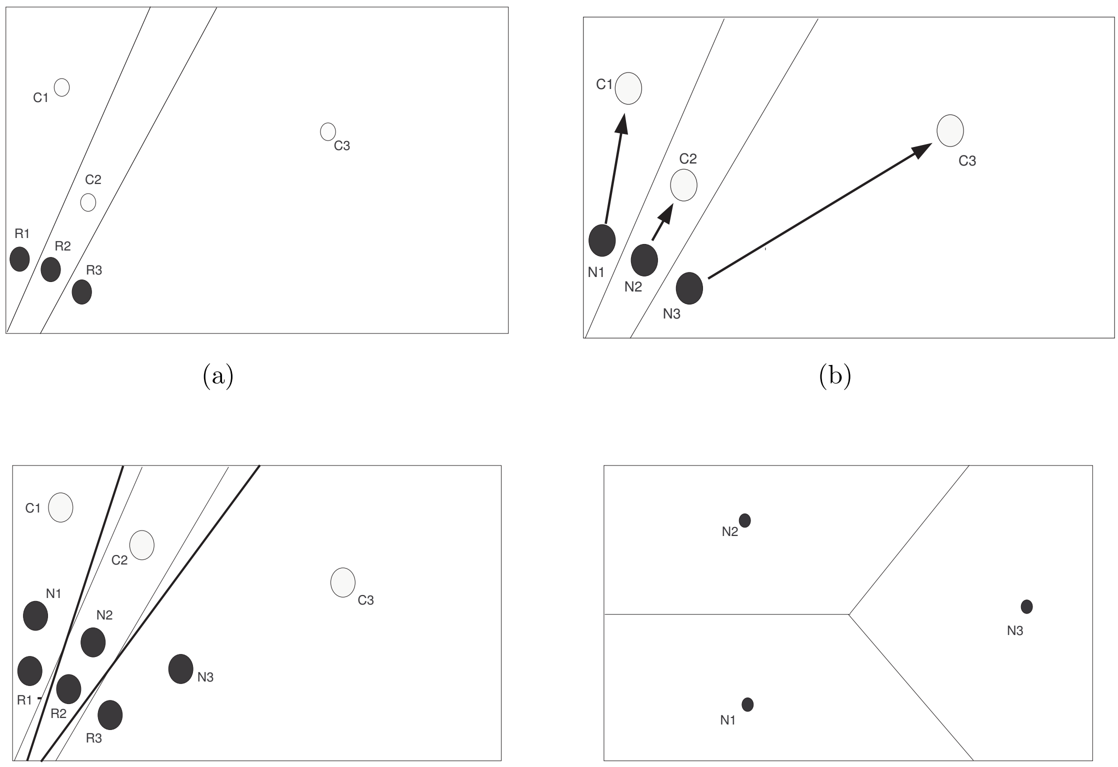


Figure 3.2: Robots R_1,R_2 and R_3 with their corresponding Voronoi cells V_1, V_2 and V_3 and centroids C_1,C_2 and C_3.

Figures 3.2(a)-(d) illustrate the process of deployment of the virtual nodes and the Voronoi partitioning in each stage in a multi-robot system consisting of three robots. The robots are positioned randomly in the workspace and Voronoi partitioning based on the initial positions is generated. This is shown in Figure 3.2(a). The centroids of the corresponding Voronoi cells are computed and the virtual nodes are moved towards the centroids. At each iteration the position of virtual nodes changes so as the Voronoi cell boundaries which results in new centroids. Optimum partitioning occurs when the virtual nodes reaches the corresponding centroid positions 3.2(d). Once the optimum partitioning is generated the actual physical robots move towards their respective Voronoi cells. The proposed Partitioning technique has the following advantages.

1. Since the virtual nodes are moving instead of physical robots, the battery usage can be minimized and can be used for more important tasks like exploration and coverage.

2. The final partition will be optimal in the sense that the area allotted to each robot will be nearly uniform so that the resources available can be utilized to maximum extend.

3. The introduction of the concept of virtual nodes eliminates the coverage overlap issue since the physical robots move towards their respective Voronoi cells only at the end of partitioning process.

3.3 SUMMARY

In this chapter an optimal deployment strategy to obtain a uniform Voronoi partition for a multi-robot coverage algorithm by introducing a concept of virtual nodes is presented. The virtual nodes are deployed into a centroidal Voronoi configuration, which is shown to be an optimal configuration in the context of sensor coverage in the literature. Instead of the robots getting physically deployed, the use of virtual nodes reduces the battery usage as well as coverage time inaddition to the coverage overlap issue since the physical robots move towards their respective Voronoi cells only at the end of partitioning process. With the help of illustrative examples, it is demonstrated that the proposed partitioning scheme provides an optimal uniformly sized Voronoi cells, leading to a uniform load distribution among the robots. This further reduces the time of completion of the coverage tasks as all the robots are utilized to same extent.

CHAPTER 4

MANHATTAN DISTANCE BASED VORONOI PARTITIONING OF A GRIDDED REGION FOR EFFICIENT MULTI-ROBOT COVERAGE.

4.1 INTRODUCTION

As we have discussed earlier, partitioning the area to be covered into cells and allotting one each cell to each of the robots for coverage solves the problem of duplicity, thus avoiding repetitive coverage, in a very simple and elegant manner. Though the *partition and cover* approach solves problems associated with cooperation between robots, and eliminates the on-the-go communication requirement, partitioning itself results in reduced coverage performance in terms of incomplete coverage and coverage overlap at single robot level. In this chapter we propose a Manhattan distance based Voronoi partitioning scheme in a $2D \times 2D$ gridded region to eliminate incomplete coverage and coverage overlap due to presence of cell boundary. Here the robot footprint is assumed to be a square of sides D. Though the proposed approach is not limited to any specific single robot coverage algorithm, we use Boustrophedon-like [1] coverage (Choset 2000) and STC (Gabriely & Rimon 2001) algorithms to demonstrate the coverage performance.

4.2 PROBLEM SETTING

The robot is assumed to have a square footprint of sides D. A robot covers a region while it moves along a path. Typically, whenever possible, robot moves in a straight line. Thus, the swept region will effectively be almost same irrespective

[1] Boustrophedon decomposition based coverageChoset (2000) does not work with rectangular obstacles as the underlying Morse decomposition leads to degenerative conditions. In this work, for the purpose of illustration we use a closely related coverage algorithm known as CC_RButler et al. (2000) for rectilinear environments. Further, we may not follow the exact algorithm, rather use only the fundamental concepts from Butler et al. (2000), Choset (2000), and call it Boustrophedon-like algorithm, as both these (and several other closely related algorithms such as trapezoidal decomposition based coverageLatombe (1991)) use a scanning motion, referred to as Boustrophedon path in Choset (2000).

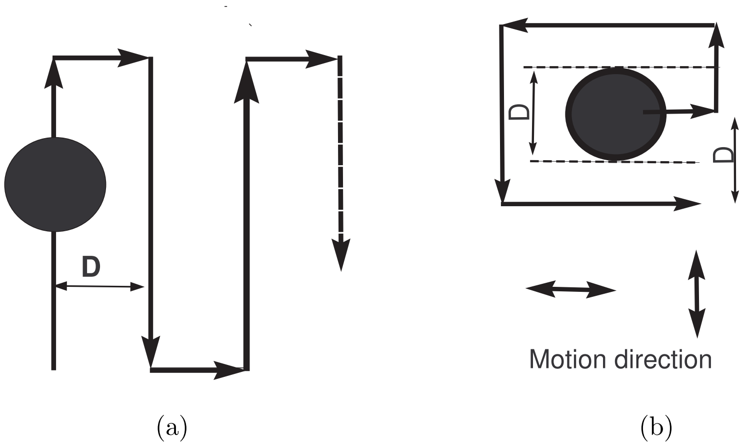


Figure 4.1: Typical robot path during coverage a) back and forth or Boustrophedon path and b) Spiral path. In both cases the robot motion is restricted either in up-down or in right-left directions.

of the exact shape of the coverage tool (circular, line, etc.) as long as it has width D. The size of the footprint is associated with the coverage tool/sensor rather that the physical size of the robot itself, though, typically, the coverage tool size is comparable to that of the robot itself.

Most coverage algorithms use simple back and forth motion, as illustrated in Figure 4.1(a) or spiralling motion as illustrated in Figure 4.1(b), as these directions are most effective while covering a free space up/down and right/left. In certain algorithms, a wall following algorithm may be used (Gabriely & Rimon 2003, Ranjitha & Guruprasad 2015*a*, 2016) to circumnavigate an obstacle in order to achieve a truly complete coverage. As we focus on incomplete or repetitive coverage induced by partition boundary, and not the physical boundary of the region, or the presence of obstacles, we do not consider robot motion apart from in up/down and right/left directions. Note that it is not possible to eliminate the incomplete or repetitive coverage problems arising due to physical boundary of the region and the presence of obstacles in arbitrary situations. In this work, we assume that the robot can have motion only in up/down and right/left directions, as illustrated in Figure 4.1.

Restriction on robot motion direction also leads us to decomposition of the region to be covered into square cells of size $D \times D$. Let us call a $D \times D$ cell as a *sub cell*. A sub cell is covered only if it is completely free of obstacles or completely insider the region to be covered. Thus, we assume that a partially occupied (by obstacle) cell or a cell partially inside the region to be covered remains uncovered. If such cells need to be covered, wall following path needs to be created and it will lead to coverage overlap (Gonzalez et al. 2003, Ranjitha & Guruprasad 2015*a*, 2016). The coverage is said to be complete if all completely free sub cells are visited once by the robot (that is, resolution complete), and non-repetitive, if no free sub cell is visited more than once. This restriction of resolution completeness may be relaxed with the truly complete or exact coverage by using algorithms such as in Ranjitha & Guruprasad (2015*a*,*b*).

The problem addressed in this chapter is to devise a partitioning scheme which will eliminate incomplete or repetitive coverage induced by partition boundary, and use it to devise a multi-robot coverage strategy using a *partition and cover* approach.

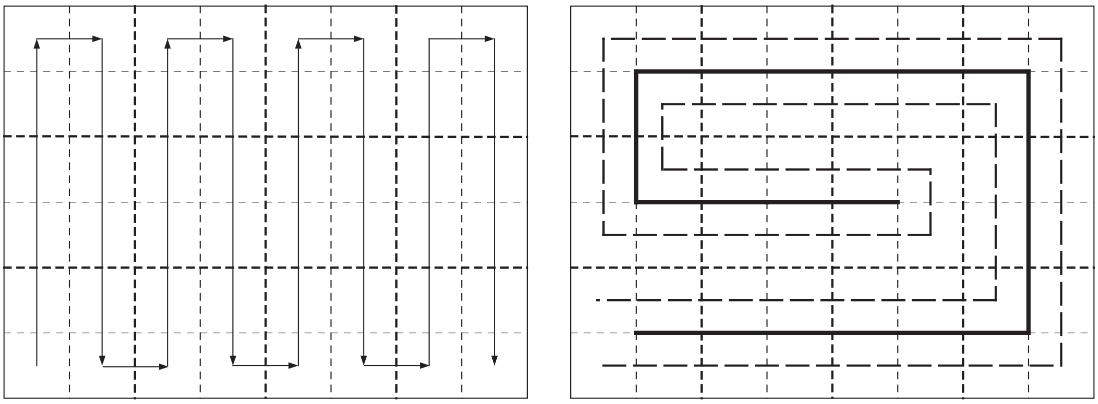
4.3 THE PROPOSED PARTITIONING SCHEME

In this section we develop the partitioning scheme to achieve the objective of eliminating the partition induced incomplete or repetitive coverage. First we shall discuss how the partition affects the performance of a coverage algorithm.

4.3.1 Partition boundary induced issues

Consider a scenario illustrated in Figure 4.2, where, a single robot successfully covers the region completely, without any overlap using either a Boustrophedon-like coverage path (Figure 4.2(a)) or STC algorithm (Figure 4.2(b)). The region shown here can be covered by a single robot completely and without any overlap/retraced path.

Now let us partition the region into two cells and each cell is allotted a robot to accomplish coverage. The main purpose of using multiple (two in this case) robots is to reduce the time required to cover entire area. As each robot has to cover a smaller area now, coverage time is reduced. However, the partition



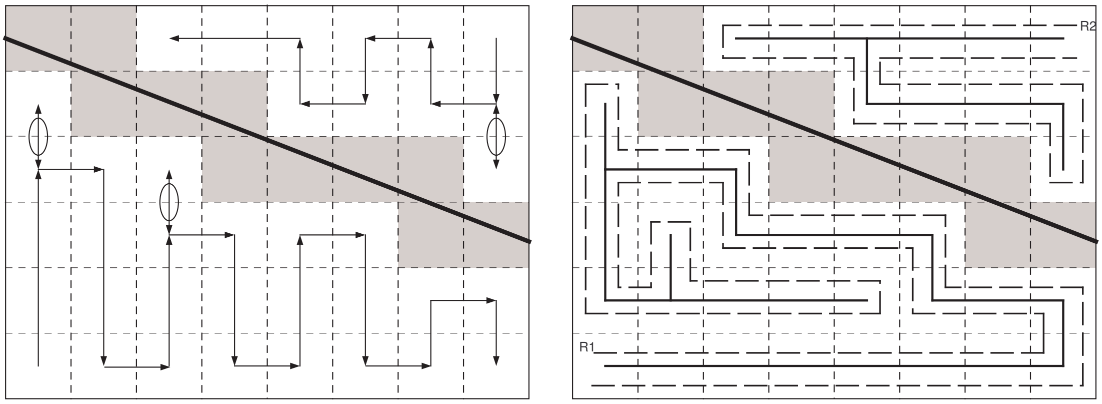
(a) (b)

Figure 4.2: A single robot completely covers the region without any path retrace or coverage overlap with both (a) Boustrophedon-like coverage algorithm and (b) STC algorithm. Robot path is shown with dashed lines while thick lines show spanning tree.

boundary (shown as thick solid line) passes through a few sub cells (shaded) and splits them into two parts. Coverage path of the robots is shown in Figure 4.3(a), with the Boustrophedon-like coverage algorithm, and (b) with the STC algorithm, using solid lines with arrows indicating the direction of motion. The grid lines decomposing the region into $D \times D$ sub cells is shown by dashed lines. The gridding is shown by dashed lines. It can be observed that neither of the robots cover the shaded sub cells, leading to incomplete coverage. Note that, if we force the robots to cover these split (shaded) sub cells, the robot path should be along the partition boundary, and this will lead to coverage overlap. Apart from incomplete coverage, the robot retraces its path at a few instances. These are indicated by circled paths in Figure 4.3(a).

4.3.2 Continuous space vs discrete space

The shaded sub cells are left uncovered by both the robots, as a robot is not allowed to cover part of a sub cell. If instead of decomposing the continuous space, we decompose the gridded space, no sub cells get split. Thus, every sub cell completely belongs to one of the decomposed region. Thus, decomposing in discrete (that is, $D \times D$ gridded) space eliminates the problem of a few sub



(a) (b)

Figure 4.3: A partition boundary (shown as thick solid line) in continuous space leads to coverage gap and/or coverage overlap with (a) Boustrophedon-like coverage path and (b) with STC path. The grids are shown in dashed lines. Robot retraces path in cells circled leading to coverage overlap. Uncovered regions (cells) are shown in grey. Robot path is shown with dashed line while thick lines show spanning tree.

cells not being covered by any of the robots. In Figure 4.4, the same space as shown in Figure 4.3 is decomposed into two regions at the discrete space level. Coverage path of the robots is shown in Figure 4.4 using solid lines with arrows indicating the direction of motion. As expected, as no sub cell is split into parts, each sub cell is covered by exactly one robot. Thus, the problem of incomplete coverage induced by the boundary created by partitioning the continuous space is completely eliminated.

Thus, by partitioning in discrete space, we can eliminate incomplete coverage induced by the partition boundary. However, observe from illustration in Figure 4.4 that the problem of retracing of path (shown with circled robot paths), leading to coverage overlap is still unresolved. Note that, as illustrated in Figure 4.2, if the region is not partitioned, and a single robot covers entire region, then the retracing may be avoided completely. Thus, the retracing of path illustrated in Figure 4.4 is purely induced by the partition boundary.

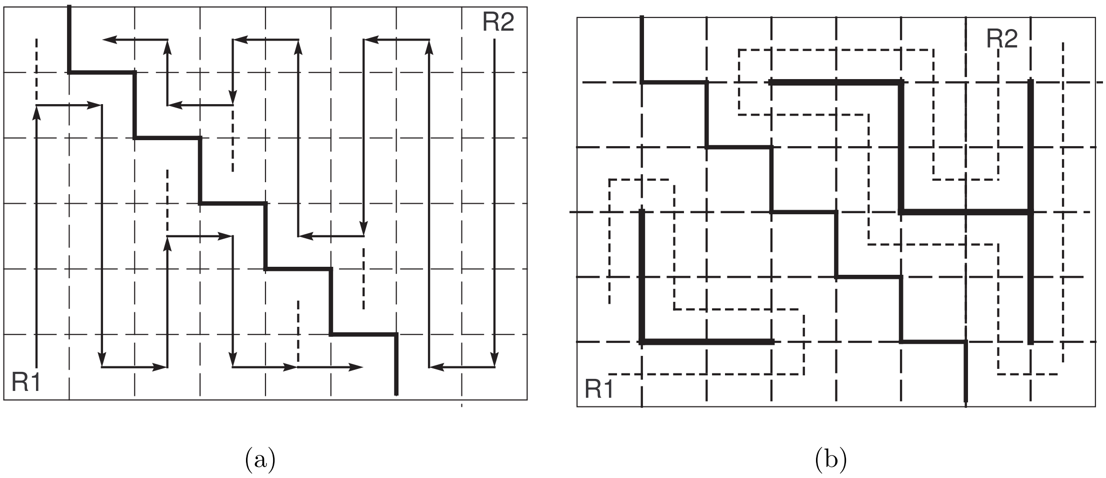
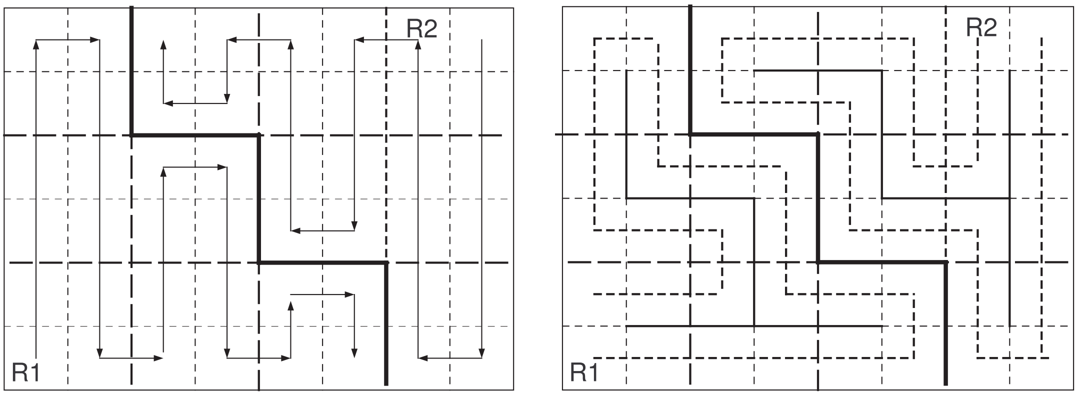


Figure 4.4: A partition in $D \times D$ gridded space (boundary shown as thick solid line) (a) with boustrophedon-like coverage, eliminates the coverage gap, however coverage overlaps may still occur as the robot retraces path (shown with dotted lines) and (b) with STC algorithm results in coverage gap. Robot path is shown with dashed line while thick lines show spanning tree. The grids are shown in dashed lines.

4.3.3 Partitioning scheme to avoid path retrace

It can be observed that the robot is forced to retrace the path on entering certain sub cells as there is no free return path for the robot. A path of width $2D$ is required if a return path has to be accommodated. Now if instead of $D \times D$ gridded space, let us consider a $2D \times 2D$ gridded space as illustrated in Figure 4.5. The $2D \times 2D$ grids are shown by thick dashed lines, while thin dashed lines show $D \times D$ gridding. Each $2D \times 2D$ cell is made up of four sub cells. Let us call a $2D \times 2D$ cell as a major cell, a terminology used in Gabriely & Rimon (2001). Now let us partition this $2D \times 2D$ gridded space as illustrated in Figure 4.5. Here, the partition boundary is shown with thick line. Now we can observe that both robots together cover all sub cells without any retrace/overlap. Thus, we can eliminate the partition induced coverage inefficiency completely by partitioning the $2D \times 2D$ gridded region.

In all the scenarios, we have used back and forth (Boustrophedon) path for the purpose of illustration. It is easy to verify that any coverage path (such as spiraling motion) with robot motion restricted to up/down and left/right will lead



(a) (b)

Figure 4.5: A partition in $2D \times 2D$ gridded space (boundary shown as thick solid line) eliminates both the coverage gap and coverage overlap with both (a) Boustrophedon-like coverage and (b) STC algorithms. The grids ($2D \times 2D$) are shown in dashed lines, while solid lines with arrow shows robot path. Thick lines shows spanning tree.

to similar result. Further, we have used a simple scenario for illustrating the problem of incomplete and overlapping coverage with decomposition scheme in the continuous space and $D \times D$ gridded space, and how these partition induced problems can be eliminated using partitioning in a $2D \times 2D$ gridded space, using simple illustrative examples. It may be verified that the arguments presented apply to arbitrary scenarios.

4.3.4 Coverage path and Manhattan distance

As we have mentioned in the problem setting, in most coverage path planning algorithms, the robot moves in either up/down (Y direction) or left/right (X direction), at least in free space. This directional restriction may be relaxed only around the obstacle boundaries or the boundary of the region to be covered (if a truly complete coverage is desirable, at the cost of coverage overlap). In fact, restricting motion along only X or Y directions in free space leads to improved coverage completeness and reduces (or eliminates) coverage overlap.

Now as the robot can move only in either X or Y directions, it makes sense to measure distance between any two points in the space too along X direction and

Y direction. This leads us to a metric known as Manhattan distance. Consider two points $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$. Now the Euclidian distance between these two points is given by,

$$d(P_1, P_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{4.1}$$

The corresponding distance measured using Manhattan metric is

$$d_m(P_1, P_2) = |(x_1 - x_2)| + |(y_1 - y_2)| \tag{4.2}$$

For a robot having to move along either X or Y (not both simultaneously),the distance it needs to travel to reach P_2 starting at P_1 is $d_m(P_1, P_2)$, the Manhattan distance, rather than the Euclidean distance. Thus it makes sense to use Manhattan distance in case of a coverage path planning problem.

4.3.5 Manhattan distance based Voronoi partitioning of $2D \times 2D$ gridded space

Voronoi partitioning (Okabe et al. 2000) has been widely used as an effective spatial partitioning tool in many applications including coverage optimization in multi-agent(robotic) systems (Guruprasad et al. 2012, Guruprasad & Ghose 2011). A standard Voronoi partitioning scheme decomposes a space using the concept of nearness to nodes. Let $I_N = \{1, 2, \ldots, N\}$; $Q \subset \mathbb{R}^2$; and $\mathcal{P} = \{p_1, p_2, \ldots, p_N\}$, $p_i \in Q$, be a set of points in Q called a *node set*. The *Voronoi partition*, generated by $\mathcal{P}$ is the collection $\{V_i(\mathcal{P})\}_{i \in I_N}$ with,

$$V_i(\mathcal{P}) = \{q \in Q | \parallel q - p_i \parallel \leq \parallel q - p_j \parallel, \forall j \in I_N\} \tag{4.3}$$

The Voronoi cell V_i is the collection of those points which are closest to p_i. In the context of the multi-robot coverage problem, N is the number of robots, p_i is the position of the ith robot, Q is the region to be covered.

As we have discussed in previous section, in the context of coverage problem addressed here, the Manhattan distance metric is more suitable than the standard Euclidean metric. From the perspective of robot travel distance, "closeness" of any two points is measured in terms of the Manhattan distance. Thus, it makes sense to replace Euclidean distance in Eqn. (4.3) by the Manhattan distance. Now we have Manhattan distance-based Voronoi partition given by,

$$V_i(\mathcal{P}) = \{q \in Q | d_m(q, p_i) \leq d_m(q, p_j), \forall j \in I_N\} \tag{4.4}$$

4.3.6 Partitioning a gridded space

Note that the Manhattan-distance based Voronoi partitioning scheme given in Eqn. (4.4) partitions the continuous space Q. Now instead if we chose Q_D to be the $D \times D$ gridded (discretized) region Q, to be collection of all gridded cells in it, that is,

$$Q_D = \{c, \text{a gridded cell} | c \subset Q\} \tag{4.5}$$

Now we may partition Q_d into Manhattan distance based Voronoi partition as,

$$V_{Dmi}(\mathcal{P}) = \{c_i \in Q_d | d_m(c_i, p_i) \leq d_m(c_i, p_j), \forall j \in I_N\} \tag{4.6}$$

Here, c_i is the centroid of the gridded cell c. Thus V_{di} is collection of all gridded cells, whose centroid is closest to p_i, in Manhattan sense.

4.4 THE PROPOSED "PARTITION AND COVER" STRATEGY

As every point within a Voronoi cell is closest to the corresponding robot, it makes sense to allot each robot to cover the Voronoi cell associated with it (Guruprasad et al. 2012). In this chapter, in place of the standard Voronoi partition, we use (a generalized) Voronoi partition of the $2D \times 2D$ gridded space, Q_{2D}, using Manhattan distance, d_m. We denote such a partition by V_{2Dm}, and the ith generalized Voronoi cell as V_{2Dmi}.

Let $\mathcal{P} = \{p_1, p_2, \ldots, p_N\}$, $p_i \in Q$, the location of N robots. A Voronoi partition of Q_{2D} is created using the Manhattan distance and the ith robot covers the region V_{2Dmi}. Individual robot may use any single robot coverage algorithm reported in the literature for covering the corresponding (generalized) Voronoi cell.

4.5 SUMMARY

A "partition and cover" strategy for cooperative multi-robot coverage, using Voronoi partitioning scheme based on Manhattan distance metric in a gridded region is discussed in the chapter. The region divided into $2D \times 2D$ grids, where $D \times D$ is the robot (coverage tool) footprint. This gridded region is partitioned using Manhattan distance-based Voronoi partitioning scheme. With the help of illustrative examples, it has been demonstrated that the proposed partitioning scheme eliminates partition boundary induced incompleteness and overlap in coverage, using existing single robot coverage strategies.

CHAPTER 5

GEODESIC VPC - GEODESIC DISTANCE BASED VORONOI PARTITIONING FOR MULTI ROBOT COVERAGE IN NON CONVEX REGIONS

In this chapter *Geodesic-VPC*, a multi-robot coverage algorithm based on the “partition” and “cover” approach using geodesic distance based Voronoi partition to alleviate the problem of disconnected cells in the presence of obstacles is discussed. This eliminates the need for repartitioning.

5.1 VORONOI PARTITIONING AND COVERAGE PROBLEM

As we have discussed earlier, Voronoi partitioning is widely used in several problems such as facility location/locational optimization, robot path planning, multi-agent/robotic systems, sensor networks, etc. One of the main properties of Voronoi cells is that each Voronoi cell is a topologically connected non-null set.In the context of multi-robot coverage problem addressed here, initial positions of robots are used as nodes, and each robot is assigned the task of covering the corresponding Voronoi cell (Guruprasad et al. 2012).The fact that the Voronoi cells partition an area ensures that the coverage is complete and non-repetitive if each robot covers the corresponding Voronoi cell completely without any coverage overlap.

Note that the partitioning scheme does not distinguish between free space and space occupied by the obstacles. The entire space, which may include obstacles (known *a priori* or not), is partitioned. Thus, a Voronoi cell may contain obstacles within it. An obstacle may split a Voronoi cell into two or more topologically disconnected patches of free space as illustrated in Figure 5.1. Here, the region of interest is partitioned into Voronoi cells based on nodes $R1, R2$, and $R3$ (which are positions of three robots in this situation). The presence of an obstacle splits the V_1, the Voronoi cell corresponding to node/robot $R1$ into two topologically disconnected patches. If the robot $R1$ has to reach a point shaded grey, a portion

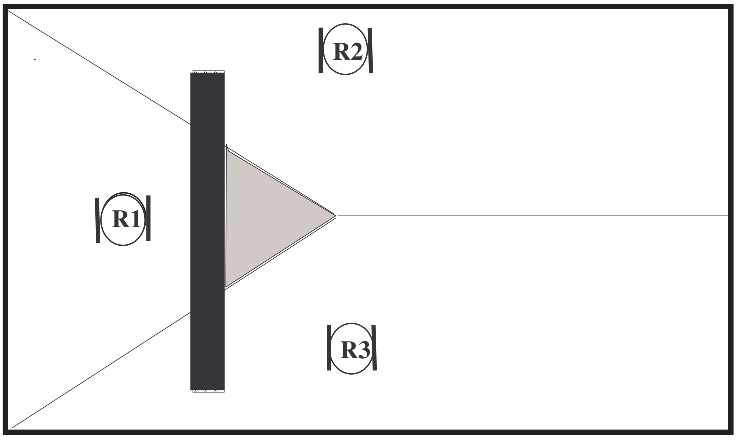


Figure 5.1: The standard Voronoi partitioning with 3 robots (position of robots are used as nodes) in a workspace occupied by an obstacle. The Voronoi cell corresponding to the 'robot R_1' is made up two disconnected patches separated by the obstacle.

of its own Voronoi cell, it has to pass through V_2 or V_3. In the context of a multi-robot coverage problem, $R1$ crossing its Voronoi boundary and moving over the neighboring Voronoi cell, which is being covered by $R2$ (or $R3$), leads to coverage overlap. Further, when robot $R1$ moves in V_2 (or V_3), it has to ensure that it does not collide with the robot $R2$ (or $R3$), unlike in a situation where each robot has to move (cover) only within the corresponding Voronoi cell.

Such situations lead to sub-optimal solutions to the coverage problem and hence increases the time and energy required to complete the task. To avoid coverage overlap, we need to ensure that each robot is assigned a contiguous region to cover. Repartitioning as in Guruprasad & Dasgupta (2012*b*), Hungerford et al. (2016) is one possible solution. A more elegant and easier solution is to incorporate the knowledge of the obstacles into partitioning scheme and ensuring that each cell has a single topologically connected patch of free space. In other words, instead of partitioning the entire region Q, partition only the free space $Q \setminus O$, where O represents the region occupied by obstacles. Geodesic distance based generalization of Voronoi partition has been used in many applications such as in sensor coverage and sensor placement problems Breitenmoser et al. (n.d.), Lee et al. (2014), Fekete et al. (n.d.), Becker et al. (2013), Bhattacharya et al. (2013), Pimenta et al. (2008), Thanou et al. (2013). In Aronov (1989) author discusses algorithm for computing geodesic distance based Voronoi partitions. In this chapter, we propose to use geodesic distance based Voronoi partitioning to overcome the problem of topological disconnected Voronoi cells associated with the standard Euclidean distance.

5.2 GEODESIC-VPC: GEODESIC VORONOI PARTITION BASED MULTI-ROBOT COVERAGE

In this section, we present the proposed multi-robot coverage strategy based on geodesic Voronoi partition. In geodesic distance based generalization of Voronoi partitioning scheme, geodesic distance between points is considered in the place of the Euclidean distance.

The term *geodesic* in its original form comes from *geodesy*, which is the science of measuring the size and shape of Earth. A geodesic in this sense is the shortest route between two points on the Earth's surface. Unlike on a flat surface, on the Earth's surface, shortest route between two points is not a straight line. This generalization of shortest distance between two points (length of the straight line segment on a flat surface) is known as the *geodesic distance*. In general, geodesic distance between any two points is the length of the shortest path between them. In the context of a mobile robot moving on a flat surface containing obstacles, the concept of geodesic distance is still useful. Though the Euclidian distance is still valid on the region of interest, due to the presence of obstacles, a robot may not be able to move between two points along a straight line. In this scenario, the geodesic distance is defined as the shortest path between two points in question that avoids the obstacles. Actual path and hence the geodesic distance depends on the specific path planning algorithm the robot uses. In this chapter, for simplicity, we assume that the obstacles are polygonal in shape, and hence the shortest path between any two points is always a sequence of line segments. Such scenarios are commonly used in the literature in similar situations (Pimenta et al. 2008). The geodesic distance used in this chapter is illustrated in Figure 5.2.

Consider $Q \subset \mathbb{R}^2$, a compact (that is, closed and bounded), not necessarily

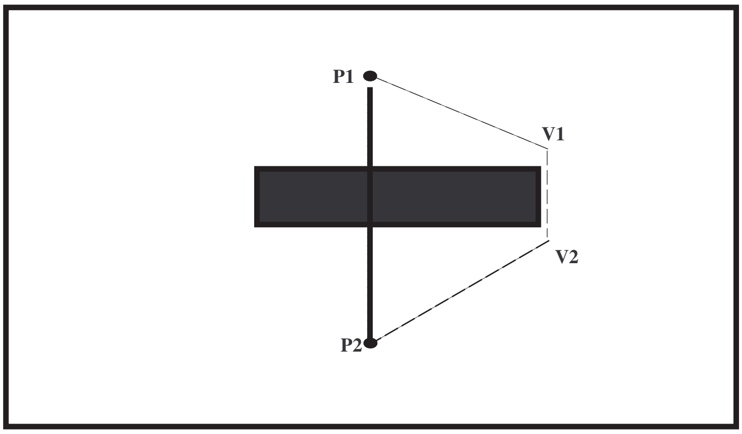


Figure 5.2: Euclidean distance between two points P_1 and P_2 is the straight line joining them, while the geodesic distance is the shortest obstacle free path from P_1 to P_2 , which is made of line segments $P_1 --> v_1 --> v_2 --> P_2$.

a convex set, as the region of interest. Let $O = \bigcup_{i=1}^{m} O_i$, be the region within Q occupied by the m (polygonal) obstacles. Let $d_G(q_1, q_2)$ denote the geodesic distance between points $q_1, q_2 \in Q \setminus O$. Further, let $\mathcal{P} = \{p_1, p_2, \dots, p_N\}$, $p_i \in Q \setminus O$, be the node set. Now a geodesic distance based Voronoi partition of $Q \setminus O$, the free space within Q is given by $\{V_1^G, V_2^G, \dots, V_n^G\}$. Here,

$$V_i^G(\mathcal{P}) = \{q \in Q \setminus O | d_G(q, p_i) \leq d_G(q, p_j), \forall j \in I_N\} \tag{5.1}$$

Observe that, unlike in the case of the standard Voronoi partition which uses the Euclidean distance metric, geodesic distance based Voronoi partition scheme decomposes the free space $Q \setminus O$ rather that the whole of the region Q. Though the standard Voronoi cell is a topologically connected region within Q, as observed in the previous section, it may lead to non-connected region in $Q \setminus O$. Now as the geodesic Voronoi partition scheme decomposes only the obstacle free space within Q, the corresponding cells are always topologically connected. This is illustrated using an example in Figures 5.3 and 5.4. The standard Voronoi partition shown in Figure 5.3 partitions entire region (Q is a rectangular region here), and the Voronoi cell corresponding to node 1 (Shown as $R1$, the 'robot 1') within the obstacle-free region is split into two topologically disconnected patches by the presence of an obstacle. The same region when decomposed using the geodesic distance based

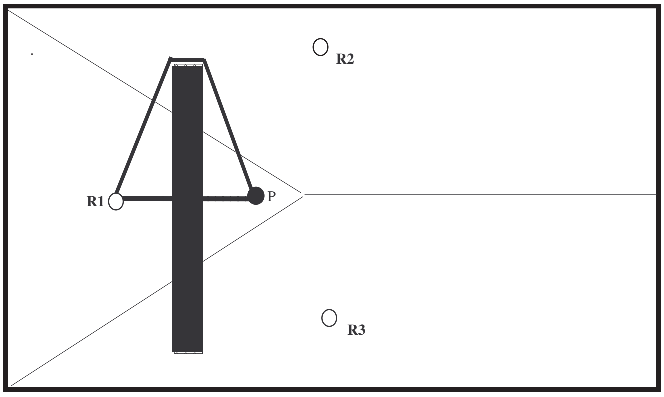


Figure 5.3: Standard Voronoi partition may lead to non-contiguous Voronoi cells. Though the point P is close to R_1 in the Euclidian sense, actual robot path is that avoiding the obstacle (that is, the geodesic path) and covering a larger distance. Though the point P lies in V_1 the Voronoi cell corresponding to R_1, the robot has to pass through V_2 (or V_3), to reach this point.

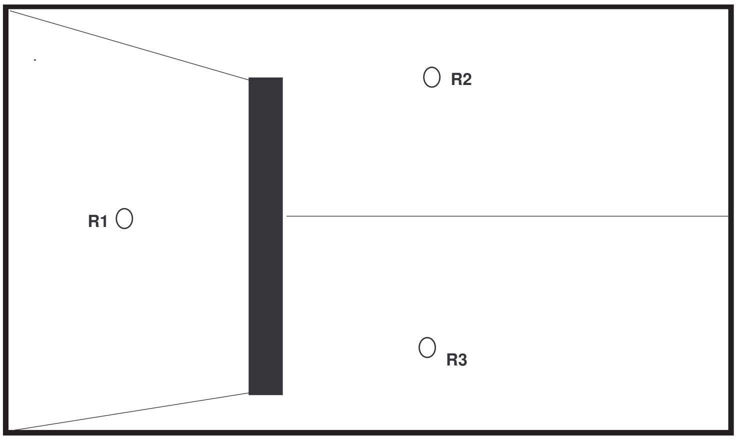


Figure 5.4: Use of geodesic distance ensures that every Voronoi cell is a contiguous.

Voronoi partition results in topologically connected cells, as shown in Figure 5.4.

Once the geodesic distance based Voronoi cells are computed, each robot uses a suitable single robot coverage algorithm to cover the respective Voronoi cell. As in any "partition" and "cover" algorithm, once the partitioning is done, the problem is reduced to a single-robot coverage problem. The Geodesic-VPC is described in Algorithm 2.

In addition to reduced time for completion of the given task (cover the region Q here), multi-robotic systems have another very useful property of being robust to failure of a few individual robots. That is, the given task (cover the region

Geodesic-VPC (Q, I_N, $\mathcal{P}_i(0)$)
Input: Q, I_N, $\mathcal{P}(0)$ //Q: search space, $I_N = \{1, 2, \ldots, N\}$,
$\mathcal{P}(0) = \{p_1(0), p_2(0)...\}$: initial location of each robot $i \in I_N$
Output: void
coverageCompleted ← false;
Broadcast $p_i(0)$ to every $j \in I \setminus \{i\}$;
$V_i^G \leftarrow$ computeGeodesicVoronoiCell($\mathcal{P}_I(0), Q$);
// V_i^G is the Voronoi cell for robot i
// N_i^G is the set of Geodesic Voronoi neighbors of robot i
while *true* **do**
  if *completedCoverage = false* **then**
    Perform motion (action) prescribed by coverage algo (e.g., STC) to cover $V_i^G \setminus O$ //Obstacle free region within V_i^G
    if *coverage algo signals coverage completion* **then**
      coverageCompleted ← true;
      sendMessage("Completed Coverage") to every robot $j \in N_i$
      STOP;
    end
  end
end

Algorithm 2: Geodesic-VPC Algorithm for robot i

Q) may be completed even if a few of the individual robots fail. We may use technique used in Guruprasad et al. (2012) to incorporate this property to the proposed Geodesic-VPC. Each robot broadcasts 'I am alive' at regular intervals. If a robot fails, it will naturally stop sending such as message. Now the neighboring robots may repartition the region and complete the coverage.

Theorem 1 *Given a single-robot on-line coverage algorithm guaranteed to achieve complete non-overlapping coverage, the Geodesic-VPC algorithm achieves complete non-overlapping coverage.*

Proof. By the property of partitioning, $\bigcup_{i=1}^{N}(V_i^G) = Q \setminus O$. As each robot covers the region V_i^G, the corresponding geodesic Voronoi cell completely, the entire region $Q \setminus D$ is covered completely.

Further, as geodesic Voronoi cells have disjoint interiors and each robot covers interior of its own cell, there is no overlapping between robots' covered regions. Now as the single-robot coverage algorithm guarantees no overlapping, no region is covered more than once.

Note that to compute the geodesic distance based Voronoi partition, the information about the obstacles should be available a priori. This information may be either known or may be gathered by robots by performing an exploration. In this chapter, we assume that the map of the region (that is, exact location of obstacles) is known. We are currently working on the scenarios when the robot may start with no knowledge of the map and build the map while covering the region, and updating the Voronoi cells as and when the new information about the obstacles is obtained.

Another aspect of the multi-robot coverage problem is of uniform load distribution amongst the individual robots in terms of coverage time. This problem is common to any multi-robot coverage algorithms, particularly the "partition" and "cover" class of algorithms. The problem requires uniform partitioning of the region, known as equatable partitioning. More uniform coverage can be achieved either by placing the robots more uniformly in the region before starting the coverage, or uniformly placed virtual nodes in the region and partitioning the region based on these virtual nodes. Several strategies such as centroidal Voronoi configuration have been explored in the literature to achieve uniform area partitioning.

5.3 SUMMARY

Geodesic-VPC, a "partition and cover" multi-robot area coverage strategy, using geodesic distance based Voronoi partitioning scheme, in the presence of obstacles is discussed in this chapter. Each robot is allotted the task of covering a Geodesic Voronoi cell. Unlike the standard Voronoi cell (based on the Euclidean distance), the geodesic Voronoi cell is a contiguous region in the free space. As each robot covers the corresponding geodesic Voronoi cell, a passive cooperation between the robots is achieved, thus avoiding coverage duplication and without any requirement of extensive communication during the coverage process. Also, as each robot has to cater to a smaller region and does not require the information of the coverage map of other robots, the memory requirement is also greatly reduced.

CHAPTER 6

GM-VPC: AN ALGORITHM FOR MULTI-ROBOT COVERAGE OF KNOWN SPACES USING GENERALIZED VORONOI PARTITION

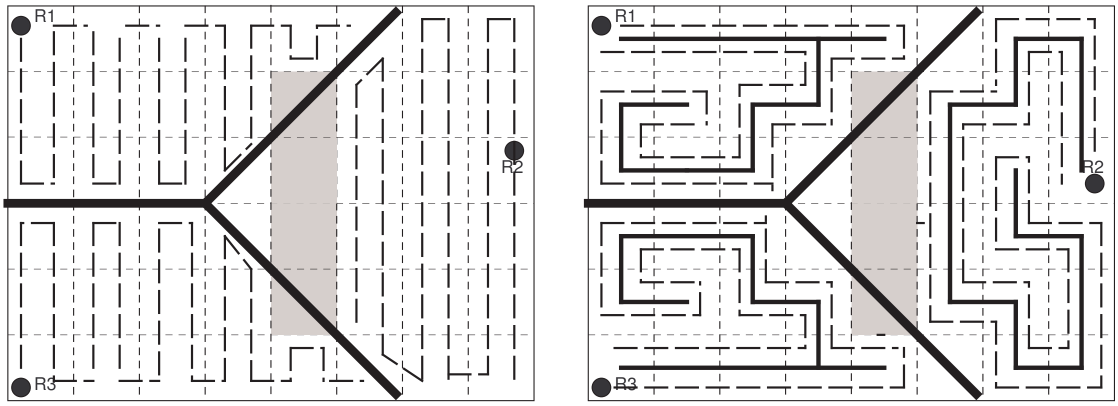
We have observed that the Manhattan distance over the gridded space resolved the partition boundary induced coverage problems, and the Geodesic metric resolved the problem non-contiguous Voronoi cells in the presence of obstacles, associated with the Voronoi partition based 'partition' and 'cover' strategy. In this chapter, we combine these two metrics to completely eliminate both the 'partition boundary induced' and 'obstacle induced' problems associated with the use of standard Voronoi partition for a multi-robot "partition and cover" algorithm.

6.1 THE PARTITIONING SCHEME

In this section we present a generalized Voronoi partitioning scheme that alleviates the problems associated with the standard Voronoi partitioning scheme for a multi-robotic coverage problem, in the presence of obstacles. First, we discuss these problems. We use two representative single robot CPP algorithms for the purpose of illustration, namely, Boustrophedon decomposition based CPP (Choset 2000) and STC (Gabriely & Rimon 2001) algorithms.

6.1.1 Non-contiguous Voronoi cells.

Consider a scenario as illustrated in Figure 6.1. The area of interest Q is a rectangular region with an obstacle (shown with grey cells) at the center. Three robots are required to cover this region. If we use the standard Voronoi partitioning scheme based on the Euclidean distance as shown in the figure, we can observe that a triangular region on the left of the obstacle, which is a portion of V_2, the Voronoi corresponding to the robot $R2$, is topologically disconnected from the rest of the region in V_3 containing $R3$. Thus, either robot $R2$ has to cross V_1 or $V32$ and reach this are to cover, leading to coverage overlap apart



(a) (b)

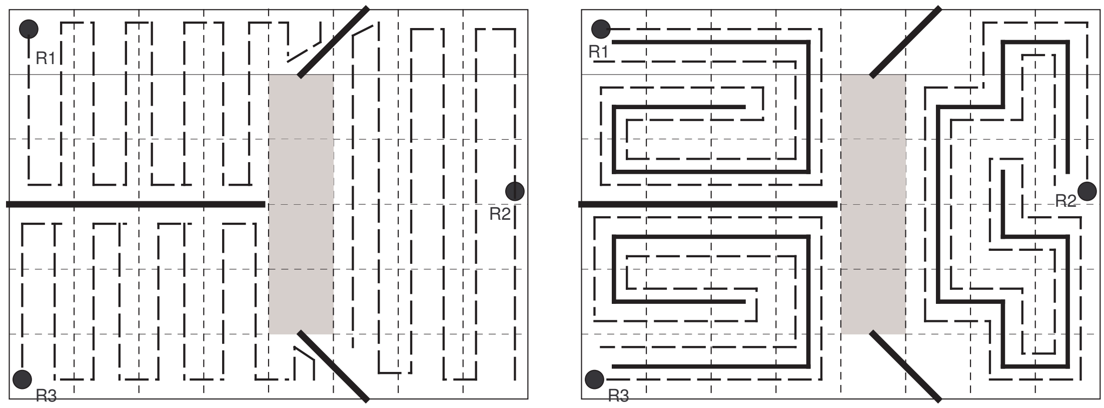
Figure 6.1: Multi-robot coverage path (shown in long dashed lines) using the standard Voronoi partition (dark lines depict the Voronoi cell boundaries) with, (a) Boustrophedon coverage and (b)STC (Spanning tree is shown with solid line) algorithms with three robots shown as R1, R2, and R3. Shaded region in the center is an obstacle. The gridding of the area into $2D \times 2D$ cells is shown by short dashed lines.

from possible collision between robots, or leave this region uncovered. 6.1 (a) and (b) show coverage path (dashed lines) with Boustrophedon algorithm and STC algorithm, respectively. With both the algorithms, unconnected part of V_2 is left uncovered.

This problem of non-contiguous Voronoi cell can be addressed by using geodesic distance in the place of the Euclidean distance. Figure 6.2 shows the same scenario as in Figure 6.1, where Geodesic distance based generalized Voronoi partition is used to divide the region into cells. With this generalization we may observe that all the Voronoi cells are guaranteed to be contiguous region. In fact, this scheme partitions $Q \setminus O$, the obstacle free region within Q, rather than the whole of Q. Coverage path with Boustrophedon and STC algorithms are shown with dashed lines in Figure 6.2 (a) and (b), respectively.

6.1.2 Partition boundary induced coverage gap

Though we can address the problem of non-contiguous cells with the use of geodesic distance, partitioning itself introduces a problem of incomplete coverage. We call this problem as partition-boundary induced incomplete coverage.. This

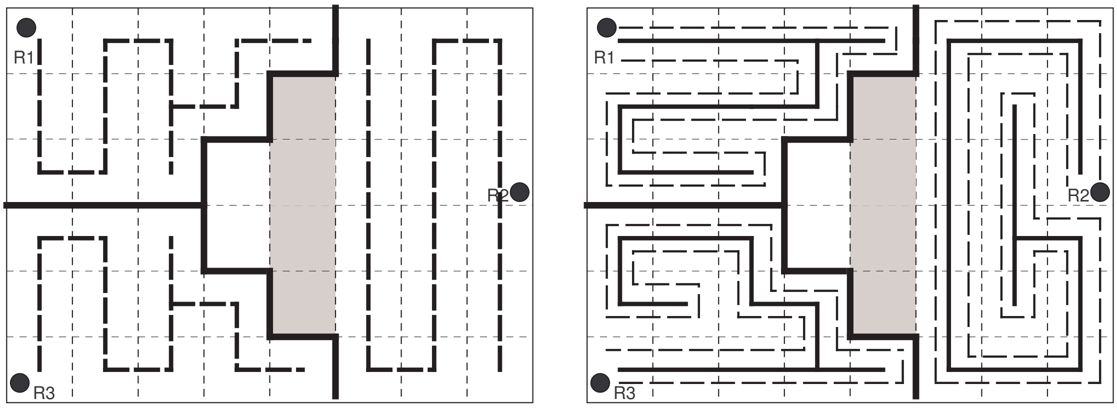


(a) (b)

Figure 6.2: Multi-robot coverage path (shown in long dashed lines) using the Geodesic distance based Voronoi partition (dark lines depict the Voronoi cell boundaries) with (a) Boustrophedon coverage and (b) STC (Spanning tree is shown with solid line) algorithms with three robots shown as R1, R2, and R3. Shaded region in the center is an obstacle. The gridding of the area into $2D \times 2D$ cells is shown by short dashed lines.

can be observed in Figures 6.1 and 6.2. Incompleteness in coverage is more prominent with STC algorithm as it provides only resolution complete coverage path (that is, completeness at gridded space rather than in the continuous space). However, we may observe that the Boustrophedon coverage algorithm results in coverage overlap while it forces a truly complete coverage near the partition boundary. Coverage overlap occurs near partition (or obstacle) boundary because of narrow passage of less than D width available for the robot.

By using Manhattan distance metric in the place of Euclidean distance within a gridded space of size $2D \times 2D$, this problem of partition boundary induced incompleteness/overlap can be completely avoided. Figure 6.3 shows same scenario as in Figures 6.1 and 6.2 with a partitioning of the gridded space into cells using Manhattan distance based Voronoi partitioning scheme. As can be observed from Figure 6.3, the partition boundary is ladder like and always along the horizontal (left/right) or vertical (up/down) directions and aligned with the $2D \times 2D$ grid lines. Each Voronoi cell is a collection of $2D \times 2D$ cells. It can be observed from the Figures 6.3 (a) and (b) that with both Boustrophedon and STC algorithms, robots $R1$ and $R3$ cover the corresponding Voronoi cells completely and without



(a) (b)

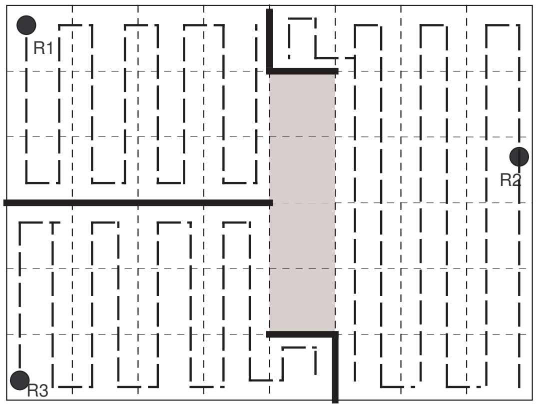
Figure 6.3: Multi-robot coverage path (shown in long dashed lines) using the Manhattan distance based Voronoi partition (dark lines depict the Voronoi cell boundaries) with (a) Boustrophedon coverage and (b) STC(Spanning tree is shown with solid line) algorithms with three robots shown as R1, R2, and R3. Shaded region in the center is an obstacle. The gridding of the area into $2D \times 2D$ cells is shown by short dashed lines.

any overlap/retrace. However, because of the presence of the obstacle, two $2D\times 2D$ cells on the left of the obstacle, which is part of V_2, that corresponding to robot $R2$ are left uncovered as these cells are accessible to $R2$ without crossing Voronoi cells of $R1$ or $R3$. This situation is similar to that illustrated in Figure 6.1 with the standard Voronoi partition.

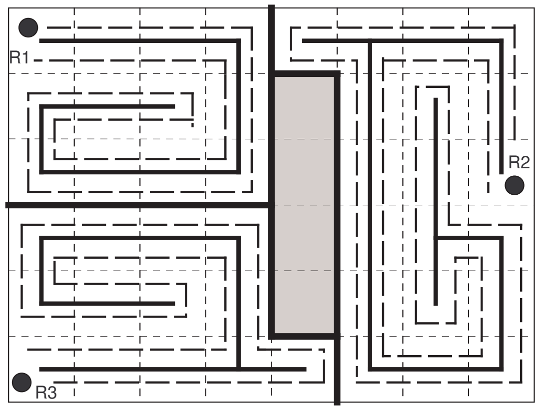
6.1.3 Geodesic-Manhattan distance-based Voronoi partition

Now we present a generalized Voronoi partitioning scheme using a combination of Geodesic distance and Manhattan distance which solves both the problems of non-contiguous Voronoi cells and partition boundary induced coverage gap.

Various distance metrics used here are illustrated in Figure 6.5. Euclidean distance between two points P_1 and P_2, $d(P_1, P_2)$, is the length of straight line between them. Geodesic (or Geodesic-Euclidean) distance, $d_G(P_1, P_2)$, is the length of path $P_1 \rightarrow R_1 \rightarrow R_2 \rightarrow P_2$, the shortest obstacle free path between P_1 and P_2. The Manhattan distance between them, $d_M(P_1, P_2)$, is the length of path $P_1 \rightarrow S \rightarrow P_2$ with restricted horizontal and vertical directions. The



(a)



(b)

Figure 6.4: Multi-robot coverage path (shown in long dashed lines) using the Geodesic Manhattan distance based Voronoi partition (dark lines depict the Voronoi cell boundaries) with (a) Boustrophedon coverage and (b) STC (Spanning tree is shown with solid line) algorithms with three robots shown as R1, R2, and R3. Shaded region in the center is an obstacle. The gridding of the area into $2D \times 2D$ cells is shown by short dashed lines.

Geodesic-Manhattan distance, $d_{GM}(P_1, P_2)$, is the length of path $P_1 \rightarrow T_1 \rightarrow R_1 \rightarrow R_2 \rightarrow T_2 \rightarrow P_2$, the shortest obstacle free path with motion restriction along horizontal and vertical direction. Finally, the Geodesic-Manhattan distance in the $2D \times 2D$ gridded space between P_1 and P_2, $d_{GM}^{2D}(P_1, P_2)$, is the number of cells encountered while moving from cell containing P_1 to that containing P_2 with only sideways or up/down motion is shown with dashed lines along with the cell count. The distance in this case is 9 units.

Let Q_{2D} be the collection of $2D \times 2D$ cells within a region Q, that is,

$$Q_{2D} = \{c, \text{a } 2D \times 2D \text{ cell} | c \subset Q\} \tag{6.1}$$

Now we may partition Q_{2D} into Geodesic-Manhattan distance based Voronoi partition as,

$$V_{GMi}^{2D}(\mathcal{P}) = \left\{c_i \in Q_{2D} | d_{GM}^{2D}(c_i, p_i) \leq d_{GM}^{2D}(c_i, p_j), \forall j \in I_N\right\} \tag{6.2}$$

Here, c_i is the centroid of the gridded cell c. Thus V_{GMi}^{2D} is the collection of all

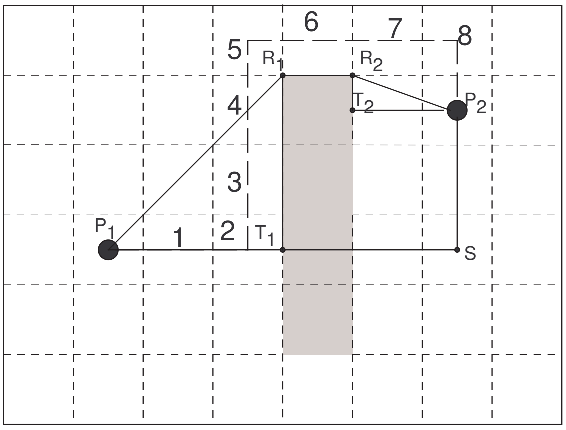


Figure 6.5: Euclidean, Manhattan, Geodesic, and Geodesic-Manhattan distance between two points.

$2D \times 2D$ cells, closest to p_i (that is, by definition to the cell containing p_i, in Geodesic-Manhattan sense.

Remark: In some situations a cell (center of the cell) may be equidistant (in Manhattan sense) from two nodes (robots). Such a cell belongs to Voronoi cell of both robots. That is, the partition boundary is no more a set of measure zero. Such situations may be avoided by using the following approach. Consider two nodes R_1 and R_2 and a cell C equidistant from these two robots.

1. If R_1 is to the left of R_2, then $C \in V_{GM1}^{2D}$
2. If R_1 is to the right of R_2, then $C \in V_{GM2}^{2D}$
3. If R_1 is directly below R_2, then $C \in V_{GM1}^{2D}$
4. If R_1 is directly above R_2, then $C \in V_{GM2}^{2D}$

6.2 DETAILS OF THE GM-VPC ALGORITHM

In this work we use a 'partition and cover' approach to the multi-robot coverage path planning problem, where the region of interest is partitioned into N cells and each of the N robots covers a cell. In order to cover a cell, the robot has to move to every point in the cell. Now as every point within a Voronoi cell is closest to the corresponding robot, it makes sense to allot each robot to cover the

Voronoi cell associated with it as proposed in Guruprasad et al. (2012). In this section, we use a combination of Manhattan and Geodesic metrics in place of the standard Euclidean distance.

Let $\mathcal{P} = \{p_1, p_2, \ldots, p_N\}$, $p_i \in Q \setminus O$, be the location of N robots. A generalized Voronoi partition of Q_{2D}, to be precise of $(Q \setminus O)_{2D}$, is created using the Geodesic-Manhattan distance metric and the ith robot covers the region V^{2D}_{GMi}. Individual robot may use any single robot coverage algorithm reported in the literature for covering the corresponding (generalized) Voronoi cell.

GM-VPC (Q, I_N, $\mathcal{P}_i(0)$)
Input: Q, I_N, $\mathcal{P}(0)$ //Q: search space, $I_N = \{1, 2, \ldots, N\}$,
$\mathcal{P}(0) = \{p_1(0), p_2(0)...\}$: initial location of each robot $i \in I_N$
Output: void
coverageCompleted ← false;
Broadcast $p_i(0)$ to every $j \in I \setminus \{i\}$;
$V_i^G \leftarrow$ computeGMVoronoiCell($\mathcal{P}_I(0), Q, O$);
// V^{2D}_{GMi} is the GM-Voronoi cell for robot i
// N^{2D}_{GMi} is the set of GM-Voronoi neighbors of robot i
while *true* **do**
  if *completedCoverage = false* **then**
    Perform motion (action) prescribed by coverage algo (e.g., STC) to cover $V_i^G \setminus O$ //Obstacle free region within V_i^G
    if *coverage algo signals coverage completion* **then**
      coverageCompleted ← true;
      sendMessage("Completed Coverage") to every robot $j \in N_i$
      STOP;
    end
  end
end

Algorithm 3: GM-VPC Algorithm for robot i

6.3 ANALYSIS OF THE GM-VPC ALGORITHM

In this section we provide a discussion on the properties of the proposed GM-VPC strategy. First we provide a useful definition.

Definition 1 Coverage conductive region: *A region Q possibly containing a set of obstacles O, is said to be* coverage conducive *with D sized robot (coverage* *tool) footprint, if $Q \setminus O$ is a contiguous region made up of union of $2D \times 2D$ sized squares, all with same orientation.*

Note that with a D sized coverage tool, it is not possible for any CPP algorithm to generate a complete coverage path with zero coverage overlap if the region is not "coverage conducive". Algorithms such as Boustrophedon CPP Choset (2000) or BSA Gonzalez et al. (2003) can provide complete coverage at the cost of non-zero coverage overlap. But by design, algorithms such as STC Gabriely & Rimon (2001) result in coverage gap and ensure non-overlapping coverage path.

Theorem 2 *If a single robot CPP algorithm can provide complete and non-overlapping coverage of a given coverage conducive region $Q \setminus O$, then the proposed GM-VPC algorithm using same CPP algorithm as underlying single-robot CP planner, ensures that the robots cooperatively cover the entire $Q \setminus O$ completely without any coverage overlap.*

Proof. If $Q \setminus O$ is coverage conducive, then the GM-Voronoi cells of $Q \setminus O$ too are coverage conducive. That is, each GM-Voronoi cell is made up of contiguous (due to Geodesic distance) set of free $2D \times 2D$ cells (due to Manhattan distance). Thus, the single robot CPP capable of covering a coverage conducive region makes each robot cover the corresponding GM-Voronoi cell completely without any overlap. As the set of GM-Voronoi cells partition $Q \setminus O$, intersection of any to GM-Voronoi cells is either null or made up of line segments (common boundary between neighboring GM-Voronoi cells), and hence coverage path of any two robots do not overlap. Also, union of all GM-VPC being $Q \setminus O$, entire region $Q \setminus O$ is covered. $\square$

Theorem 3 *If a CPP algorithm (such as STC or Boustrophedon) capable of providing optimal coverage of conducive environment (such as ones described above), makes a robot cover $Q \setminus O$ with a specific performance (in terms of % coverage and % overlap), then the proposed GM-VPC using the same CPP algorithm as the underlying single-robot CP planner, makes the robots cooperatively cover $Q \setminus O$ with same performance as the single robot coverage.*

Proof. Let us assume that a single robot covers $x\%$ of $Q \setminus O$ with a $y\%$ overlap. With a CPP algorithm capable of covering a coverage conducive region completely without any overlap, coverage gap and coverage overlap occurs $Q \setminus O$ around the boundaries of the region or obstacles, as these lead to partially occupied $2D \times 2D$ cells (or strips of width $2D$). If we include partially occupied major cells while partitioning using GM-Voronoi partitioning scheme, same major cells which are partially occupied in $Q \setminus O$ remain partially occupied in GM-Voronoi cells. All the completely free major cells in $Q \setminus O$ remain completely free after partitioning (unlike with standard or Geodesic distance based Voronoi partitioning scheme), and belong to one of the GM-Voronoi cells. The CPP algorithm covers all free major cells exactly once and coverage overlap occurs only when partially occupied cells are being covered. As the number and relative position of partially occupied cells remain unchanged before and after partitioning, total amount of overlap (in case algorithms such as BoustrophedonChoset (2000) or Competitive-STCGabriely & Rimon (2003)) and/or coverage gap (in the case of algorithms such as STC (Gabriely & Rimon 2001) or Competitive-STC (Gabriely & Rimon 2003)) with a singe robot covering $Q \setminus O$ or each robot covering the corresponding GM-Voronoi cells.

Significance of these results is that, as we have observed in earlier section, with standard Voronoi partition, Geodesic (Euclidean based) Voronoi partition, and even the Manhattan Voronoi partition based VPC strategies using Boustrophedon and STC algorithms may lead to incomplete coverage (in the case of STC) and coverage overlap (in the case of boustrophedon). However, same single robot algorithms provide complete and non-overlapping coverage by using the proposed GM-VPC strategy.

The observation is not limited to Voronoi partitioning scheme alone. The facts that i) the free space $Q \setminus O$ rather than whole Q itself is partitioned (Geodesic distance) and ii) partitioning of the $2D \times 2D$ gridded region rather than the continuous space is partitioned (Manhattan distance), are the main factors that lead to the nice properties discussed here in this section. To summarize, Voronoi partitioning gives us a way of incorporating the concept of "closeness" while partitioning, use of Geodesic distance gives the contiguous cells, and finally, use of Manhattan distance retains the $2D \times 2D$ grid structure which is very useful in the context of a theoretically provable CPP algorithm.

6.4 SUMMARY

We proposed a strategy which combines two generalization of Voronoi partition namely, Geodesic distance based Voronoi partition and Manhattan distance based Voronoi partition to address contiguity of partition in the presence of obstacles and avoid partition boundary induced coverage gap. The region is divided into $2D \times 2D$ grids, where D is the size of the robot footprint. With the help of illustrative examples, we have demonstrated that the proposed Geodesic-Manhattan Voronoi partition-based coverage (GM-VPC) can achieve complete and non overlapping coverage at grid level provided that the underlying single robot coverage path planning algorithm has similar property.

CHAPTER 7

SIMULTANEOUS EXPLORATION AND COVERAGE-SIMEXCOVERAGE

In this chapter,a novel concept of simultaneous exploration and coverage for mobile robots performing area coverage task is discussed. The primary task of the mobile robot is to completely cover an initially unknown region. The robots perform intermittent exploration during coverage in order to update the map of the environment, which in turn is used to plan coverage path.

7.1 SINGLE ROBOT SIMEXCOVERAGE

Coverage path planning, exploration, localization, and mapping, are a few fundamental problems associated with coverage path planning applications. *Localization* refers to a problem of estimating the pose (position and orientation) of the mobile robot. This problem is non-trivial as standard positioning techniques such as odometry or GPS are not very accurate. *Mapping* is a process of obtaining the geometric map of a region of interest in terms of obstacle infested part and the free space. Typically these are stored as occupancy map. Simultaneous Localization and Mapping (SLAM) solves localization, and mapping problems simultaneously.

There is not a clear cut definition for the terms exploration and coverage in the literature. The definitions given are case specific. In this chapter by exploration we mean gathering information about the nature of workspace, which is discretized as cells, using sensors. In other words, exploration is the process of identifying the nature of each cells, occupied or free, using sensors. The cells outside the sensor range are considered as unknown or un-explored. A cell is marked covered if the robot was physically present on it at any present or past times. Obstacle occupied cells cannot be covered but can only be explored.

Area *coverage* by a autonomous mobile robot is problem which is very useful in many applications such as floor cleaning, land-mine detection, lawn mowing, etc. In this problem, known as coverage path planning (CPP), a robot has to move a coverage tool attached (such as land mine detection sensor, in land mine detection application) through all points in the region to be covered. Apart from this completeness of coverage, it is also desirable to have non-repetitive coverage, that is, the robot should not visit a point more than once.

In an *exploration* problem (Lumelsky et al. 1990*a*, Lee & Recce 1997, Yamauchi 1998, Gonzalez & Latombe 1998, Albers et al. 1999) the robot chooses an optimal point from where the exploration results in maximal information gain in terms of the map of the environment. The robot plans the path accordingly and performs exploration, typically using a long range sensor such as a Light Detection And Ranging (LIDAR), or vision -based sensors, to obtain a complete map of the environment. Though for different purposes, both online coverage path planning algorithms and exploration algorithms detect obstacles (including the region boundary) of an environment. Further, exploration also requires the robot to cover the environment in order to obtain the complete map. Because of this, the term "coverage" is used in exploration and mapping problems. However, the purpose of coverage in a CPP problem is to serve (or gather information about) each point in the space, while in an exploration and mapping problem is to obtain the complete map. In the case of exploration and mapping, the robot need not visit every point in the region.

Typically, the robots carry long range sensors in exploration/mapping applications, while they carry short range sensors in CPP problems. An online CPP algorithm detects obstacles with a local short range sensor and uses this information only to avoid the obstacle, and not to generate a complete map of the environment. A mobile robot may be equipped with several senors which may include both long range and short range obstacle detection sensors. However, CPP algorithms either use only a short range sensor in spite of availability of long range sensor, or may use only short range information from the longer range sensors (such as say, ultrasonic or IR sensors) discarding the long range information, as they need to detect an obstacle only in the next cell.

Further, in most online CPP algorithms, at each instance, only a path from current cell to immediately next cell is generated. That is, a robot has to stop at each cell, sense the environment, and decide on which of the neighboring cell to move next, and then move. However, in the case of an off-line CPP algorithm, entire path is planned at one go, and the robot may follow the path smoothly, and without any unnecessary stops.

The coverage algorithms reported in the literature are either off-line, using complete a priori knowledge about the arena, or online, using no a priori knowledge. In this chapter, we propose a CPP strategy that can utilize available partial knowledge about arena (in terms of map), while updating this knowledge on the go. Toward achieving this objective, we propose a novel simultaneous exploration and coverage (SimExCoverage) problem, where the robots perform exploration to obtain map of the unknown region while covering the explored arena. The primary objective here is to cover the arena, while a map may be obtained at the end as a byproduct. In an exploration and mapping) problem, primary tasks are robot path planning and obtaining the map of the explored region using the onboard sensors. Path planning involves deciding on next point from which exploration is carried out to maximize the information gain in terms of the map/explored area. In SimExCoverage problem proposed here, robot moves along the path planned by the CP planner (coverage strategy). However, the robot still has to plan/decide on the point along this path where exploration is performed.

7.1.1 Problem setting

In this work the off-line version of STC (Gabriely & Rimon 2001) algorithm for the CPP part of the problem is presented. STC algorithm requires the arena to be decomposed into $2D \times 2D$ square cells, where the robot footprint (or the footprint of the coverage tool) is assumed to be $D \times D$. Each $2D \times 2D$ cell is called a major cell, which has four $D \times D$ minor cells. A spanning tree is created as major cells as nodes, while the actual robot path is created through the minor cells.

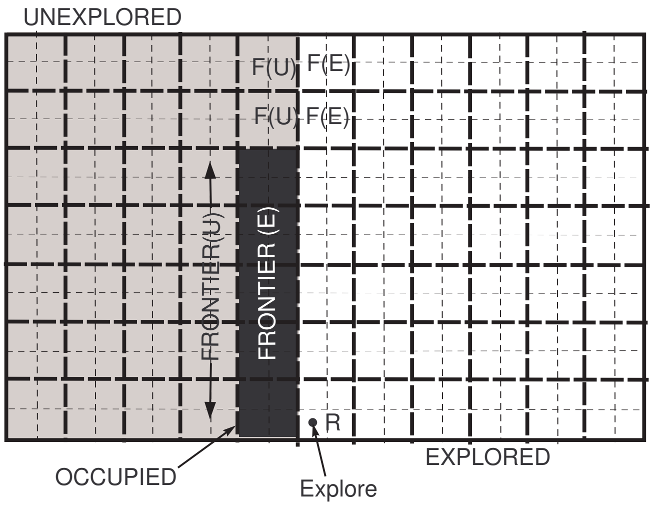


Figure 7.1: The arena is decomposed into 'major cells' of size $2D \times 2D$ (shown with thick long dashed boundary) and each major cell has four $D \times D$ sized minor cells (shown with thin dashed line boundary). Dark cells are 'occupied' (by obstacles) and remaining cells are 'free' (of obstacle). An instance of exploration from divides the region into 'explored' and 'unexplored' regions. Explored region is made up of explored free cells (white cells) explored occupied cells (dark cells). The frontier is set of explored ('FRONTIER(E) or F(E)) and unexplored ('FRONTIER(U) or F(U)) cells on the boundary separating the explored and unexplored regions.

Further, the STC algorithm assumes that a major cell is either 'free' of obstacle, or 'occupied' by the obstacle. Thus, even if a major cell is partially occupied by an obstacle, it is assumed to be completely occupied. For this reason, Choset (Choset 2001) classifies such algorithms as approximate cellular decomposition based CPP algorithms. The coverage is considered complete when the robot passes through all minor cells, and non-repetitive, if no minor cell is visited more than once. Figure 7.1 illustrates the problem setting discussed here. The exploration process divides the major cells into 'known' (or 'explored') and 'unknown' (or 'unexplored') cells. A known cell or explored cell may be 'free' or 'occupied'. In Figure 7.1, explored 'free' cells are unshaded (white) and explored 'occupied' cells are shown with black. The grey shaded cells are unexplored or unknown cells. Frontier cells form the boundary between explored and unexplored regions. The frontier cells may be 'explored frontier' (part of explored region) or 'unexplored frontier' (part of unexplored region) cells. The 'explored' frontier cells are marked 'FRONTIER(E)' or ‘F(E), while the ‘unexplored frontier cells are marked ‘FRONTIER(U)’ or ‘F(U) in the figure.

In their work, Acar et al. (2002), studied sensor data in unstructured environments. It focusses on the rejection of bad sensor data and its application in complete coverage. But this chapter is not related to sensor data quality. In this chapter the sensor data over a small region (most probably some few adjacent cells) is considered to get the details on the occupancy of the cells. Also the sensors considered in this work are not on continuously throughout the coverage process. The sensors are on only during the exploration phase for a small period of time when the robot reaches a frontier cell.

In Shnaps & Rimon (2016), the optimal coverage path generation for battery powered mobile robots is presented. The work focusses on generating coverage paths so as to reduce the battery usage. In this chapter also the battery utilization is minimized by turning on the sensors for a very less time as explained above. Also this work is mostly concentrated on solving coverage and exploration problems simultaneously rather than creating a path which optimizes battery usage alone. Also there can be retrace of the already covered regions which is solved in this research. In this work, we assume that the robot is equipped with a LIDAR sensor for exploration. The CPP algorithm uses the map provided by the exploration algorithm, and hence does not make use of any sensors. We assume that the robot is exactly localized, as the main focus of this work is to propose and demonstrate a simultaneous exploration and coverage problem. SLAM or other localization techniques may be used in practice.

7.1.2 Proposed SimExCoverage algorithm

In the proposed SimExCoverage problem, we combine the exploration and mapping and the CPP problems, the primary task being area coverage. Figure 7.2 illustrates the SimExCoverage problem. Exploration generates the map, which the CPP algorithm uses to generate the coverage path.

The robot moves along the coverage path. While performing coverage, at

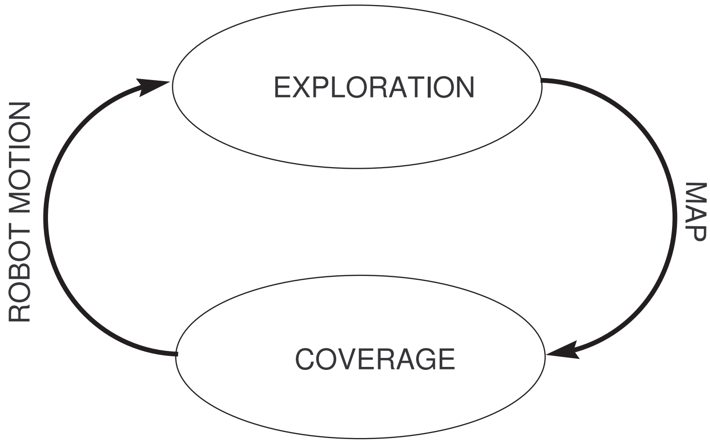


Figure 7.2: SimExCoverage problem combines Exploration and Coverage (CPP) problems. Exploration provides map for coverage path planning, while the CPP provides the path for robot motion. While the robot is moving along the coverage path, at exploration is carried out at a certain locations.

certain points along the coverage path, robot performs exploration. Unlike in an exploration problem, exploration strategy in SimExCoverage does not perform path planning. Localization is very important for any path planning algorithm. In this work, we assume exact localization is available as the focus of the work is on combining exploration and coverage problems to generate complete and non-repetitive coverage path. In practical applications, techniques such as SLAM may be used to obtain more precise localization of the robots. In fact, exploration and mapping are part of the SimExCoverage problem addressed here and hence incorporating SLAM algorithms is not very difficult.

The proposed SimExCoverage using off-line STC CPP algorithm is provided in Algorithm 4.

7.1.3 Exploration

Before the beginning of the first exploration, all (major) cells are considered 'unknown' or 'unexplored', while the 'explored' and 'frontier' lists are empty. Each exploration involves following steps. Steps in an exploration instance is shown as Algorithm 5

SimExCoverage-STC

1 Explore.

2 Generate a Spanning Tree (ST) over the graph formed by explored uncovered 'free' major cells.

3 Generate a CP through minor cells circumnavigating the ST edges through minor cells.

4 When the robot reaches a minor cell in which the corresponding major cell has a neighbor in unknown cell list, and the minor cell shares an exploration frontier with that unknown cell GOTO 1.

5 If starting minor cell is reached - STOP.

Algorithm 4: The proposed SimExCoverage algorithm using off-line STC CPP algorithm

Explore

1 SCAN 360^o using sensor (LIDAR).

2 Identify obstacle/free space.

3 Identify the cells containing obstacle and update the obstacle cell list.

4 Update free cell list with the free cells identified.

5 Update the unknown list by removing both obstacle cells and free cells from it.

6 Update the frontier cell list.

Algorithm 5: Exploration algorithm used in the proposed SimExCoverage algorithm shown in Algorithm 4.

7.1.4 Coverage

Using off-line version of STC algorithm (Gabriely & Rimon 2001) for coverage path planning, a minimal spanning tree (MST) is created over the graph formed by major cells within the explored region, using the list of 'occupied' and 'free' cells. We use Kruskal's algorithm for generating a MST. Robot path through minor cells circumnavigating the spanning tree edges is then created. The robot path is created such that the robot moves on the right side of the spanning tree edges. Now as the robot moves along the CP, whenever it is on

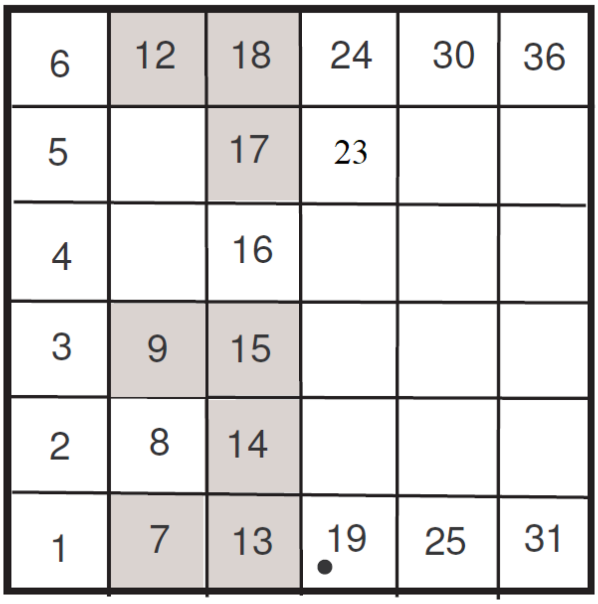


Figure 7.3: The arena to be covered is decomposed into cells of size $2D \times 2D$. Cell numbering format is as shown. Occupied cells ($7, 9, 12, 13, 14, 15, 17$, and 18) are shaded. Robot starts at a minor cell in the major cell no. 19.
.

a frontier (known) cell and encounters a frontier (unknown) cell as its neighbor, the robot switches to 'explore' mode. A MST is created when a newly 'explored' region is added, and is appended to the existing ST, and the CP is continued from the current cell.

Now we shall describe the SimExCoverage-STC with an illustrative example. Consider a scenario as shown in Figure 7.3. We consider a region which is decomposed into 6×6 major cells. The cells $7, 9, 12, 13, 14, 15, 17$, and 18 are occupied (by obstacle) and rest of the cells are free. Initially the robot is located in a minor cell of the major cell no. 19.

As the first step in the Algorithm 4, the robot performs an exploration from cell no. 19. The result of this exploration is shown in Figure 7.4 (a). Explored 'free' cells are shown with white. Explored 'occupied' cells are $13, 14, 15, 17$, and 18. The remaining region (cells) are unexplored. Now a ST is created in this explored region, shown by red lines passing through major nodes (red dot). The corresponding CP is created as the robot moves along this path (shown in blue line), as shown in Figure 7.4 (b). When the robot reaches a minor cell in the major cell 23, which is a explored 'frontier' cell, it shares boundary with the unexplored 'frontier' cell 16, it performs second exploration. With the second exploration, few more cells are are explored (shown in white), and a ST is created through the newly added 'explored' cells. This is illustrated in Figure 7.4 (c). Now when the robot reaches a minor cell in the major cell 4 (see Figure 7.4 (c)), an explored 'frontier' cell, robot performs exploration as this cell shares a boundary with unexplored 'frontier' cell 3. The ST is created in this newly explored region (cells $1, 2$, and 3). The robot CP is created as the robot moves on the right side of the ST edges. While the robot reaches a minor cell in the major cell 2 from major cell 1, it explores and finds that cell 8 is free, and a ST edge is added to this cell. Finally, the robot reaches the starting major cell (19) as shown in Figure 7.4 (d). The next minor cell from this position is the same as the starting minor cell. Thus, the robot terminates SimExCoverage-STC. As we observe here at this instance, entire region is explored and coverage is complete and without any coverage overlap/repetitive coverage.

Note that unlike the online STC algorithm (or any online CPP algorithm), here the robot does not use its sensors, which apart from requiring energy and time, also requires the robot (or the sensor alone) to scan all the neighbors (270^o) at each minor cell. During coverage the robot has to simply follow the CP (as in off-line CPP), and only look for frontier cells, where the mode is switched to exploration. Also, unlike an off-line CPP algorithm, SimExCoverage-STC does not require a priori knowledge of the map of the arena. Apart from the complete and non-repetitive coverage, SimExCoverage-STC generates the map of the environment which may be used for any other purposes, including repeated area coverage. Any repeated area coverage can now be performed completely off-line.

7.1.5 Properties of SimExCoverage-STC algorithm

This section provides an informal discussion on some of the expected properties of the proposed off-line STC based SimExCoverage algorithm.

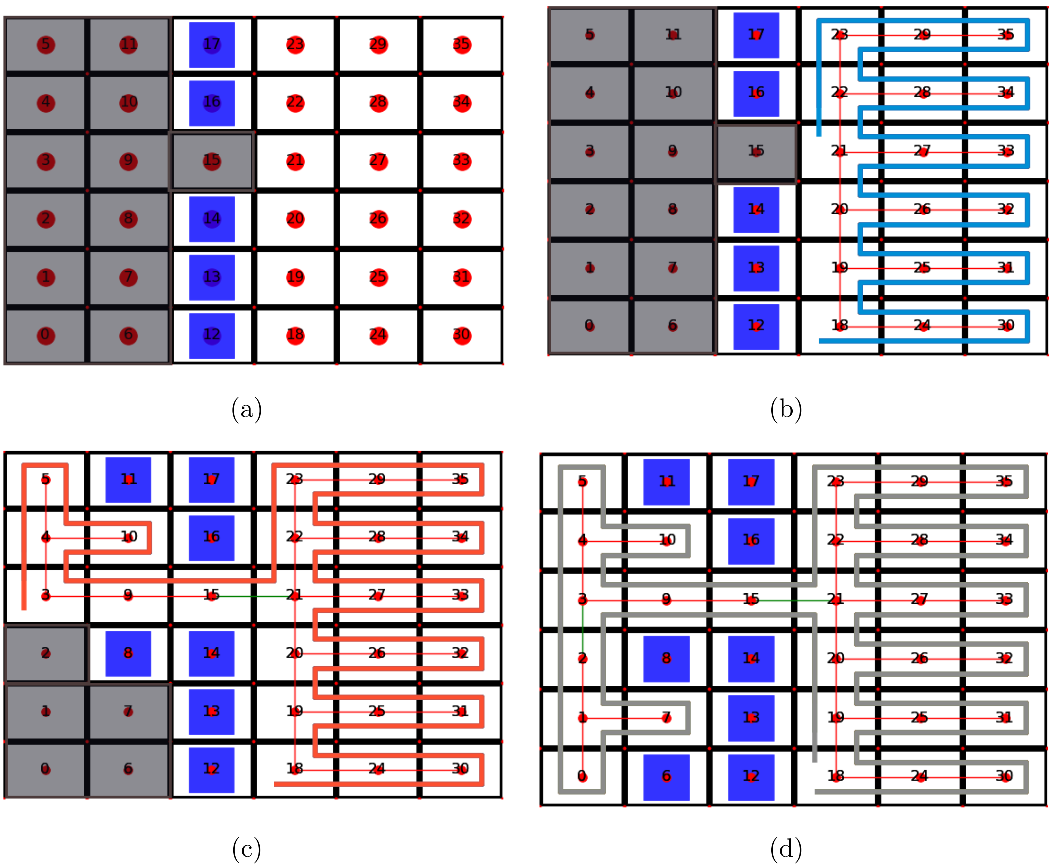


Figure 7.4: Coverage path and exploration steps. (a) First exploration, (b) Second exploration instance after complete coverage of free explored cells, (c) third exploration instance, and (d) the last exploration and successful completion of the coverage. Explored free (major) cells are unshaded, unexplored cells are shaded with gray, and explored 'occupied' cells are shaded with blue (dark gray/black without color). The nodes corresponding to major cells are shown with (red) dots. The ST edges are (red) lines passing through nodes corresponding to the free major cells. Coverage path (at graph-level, not actual robot path) passes through minor nodes/cells (not shown in Figure for clarity) around the ST edges is shown with blue lines.

Completeness and non-repetitiveness

Any SimExCoverage algorithm has an exploration strategy and an off-line CPP algorithm. The completeness and non-repetitiveness of coverage depends on the underlying exploration and CPP algorithms. In the case the proposed SimExCoverage-STC, we use a typical frontier-based exploration strategy with off-line STC CPP algorithm.

Property 1: Completeness of exploration

1a The region to be covered/explored Q has an obstacle free topologically connected sub region $Q \setminus O$.

1b Exploration is complete if $Q \setminus O$ is covered

- Explored and unexplored region is connected by the frontier cells.

1c There exists least one adjacent, unoccupied frontier unexplored-explored cell pair, called *exploration window*, as long as current unexplored region does contain a subset of $Q \setminus O$.

1d Coverage path with off-line STC algorithm within the current unexplored region will encounter the *exploration window*.

1e If no *exploration widows* exist in the frontier cells, then it implies that the exploration is complete.

Property 2: Completeness and non-repetitiveness of coverage

2a Within initially explored region, a spanning tree is formed within this region.

2b On successive exploration, when a new previously unexplored region is found/explored, a local spanning tree is created within this newly explored region. These two trees have exactly one common node which happens to be the point of exploration. Thus when these two trees are combined, the resulting graph is still a tree and in fact spans the new region which is the updated total explored region.

2c once the entire region Q is explored, the tree created by appending several trees form a spanning tree within $Q \setminus O$.

2d Now by the property of STCGabriely & Rimon (2001), the coverage path is guaranteed to be complete in entire by the property of STCGabriely & Rimon (2001) in $Q \setminus O$ and non-repetitive.

By the Properties 1 and 2, we can conclude that the proposed SimExCoverage-STC algorithm provides complete and non-repetitive coverage of the obstacle free region $Q \setminus O$ within the area of interest Q.

7.1.6 Comparative time-to-complete/battery economy of coverage

Mobile robots depend on onboard battery for the energy requirement for motion, sensors, and processing. Time for completion of the coverage is one of the prominent deciding factor along with the number of turning motion required which determines the energy consumed by the motors. Another important component which determines the energy consumed for the battery is number of sensing operation.

The total length of the coverage path is same both with online STC and the proposed SimExCoverage-STC algorithms as both provide non-repetitive paths. Number of turns in the path also is comparable. However, the fundamental advantage of the proposed SimExCoverage-STC algorithm lies in obstacle detection. In online STC, in each minor cell the robot has to use its sensors to find if the neighboring cells contain obstacles. Apart from energy used by battery, typically, the robot has to turn 270^o in each cell to find whether three of its neighboring major cells contain obstacles. Thus, battery is consumed in each cell for turning (motor) and sensing (sensor and processor).

In contrast, in SimExCoverage-STC algorithm, very few instances of exploration are used to obtain the occupancy map. This drastically reduces the battery consumption by motors, sensors, and the processors. Within the explored region, once the entire STC is created, the robot has only move along the prescribed path without intermittent stoppage and sensing until next exploration point/cell is reached. Thus, reduced instances sensing for obstacle with the proposed SimExCoverage-STC reduces battery consumption and time-to-complete the coverage task. This observation is equally valid for SimExCoverage with any other coverage algorithm.

Note that unlike the online STC algorithm (or any online CPP algorithm), here the robot does not use its sensors, which apart from requiring energy and time, also requires the robot (or the sensor alone) to scan all the neighbors (270^o) at each minor cell. During coverage the robot has to simply follow the CP (as in off-line CPP), and only look for frontier cells, where the mode is switched to exploration. Also, unlike an off-line CPP algorithm, SimExCoverage does not require a priori knowledge of the map of the arena. Apart from the complete and non-repetitive coverage, SimExCoverage generates the map of the environment which may be used for any other purposes, including repeated area coverage. Any repeated area coverage can now be performed completely off-line.

The proposed methodology can be a promising candidate for complete coverage as compared with its online counterparts. In most of the online coverage algorithms available in literature, the robots are deployed at some points and the robot will use its sensors continuously to find the obstacles and generates the path accordingly. But in the proposed methodology the sensors are on only during the exploration phase for small periods of time when the robot reaches a frontier cell. It can save a lot of battery power .In addition the coverage time will be less since before starting each phases, the robot have the map of its explored region. So the proposed methodology can be viewed as a combined online-offline coverage method. The switching between exploration and coverage phases occur when the robot reaches a frontier cell (the known unoccupied cell next to an unknown cell).

7.2 MULTI ROBOT SIMEXCOVERAGE

In this section the implementation of SimEx coverage in multirobotic system is discussed.The robots partition the workspace using Manhattan distance based

Voronoi partitioning, discussed earlier and then the SimEx coverage algorithm is implemented by each one of them.

7.2.1 Multi-robot Coverage

The coverage algorithms reported in the literature are either off-line, using complete a priori knowledge about the arena, or online, using no a priori knowledge, though SLAM algorithms may be used as an aid to both off-line and online CPP algorithms to improve robot localization. Song and Gupta Song & Gupta (2018) present an online Boustrophedon-like (that is, to-and-fro motions) single-robot coverage algorithm, based on approximate cellular decomposition, incorporating exploration into the algorithm. An exploratory turing machine (ETM) generates a coverage path online using multiscale adaptive potential surfaces. In this work, the limited exploration is use to avoid local minima that may be encountered in algorithms such as the online version Boustrophedon decomposition based CPP algorithmsRekleitis, New & Rankin (2008). In Minhaj et al. (Accepted), we proposed a single robot simultaneous exploration and coverage (SimExCoverage) problem, which combines problems of single robot coverage and exploration for mapping.

In this section, we adapt the basic ideas presented in Minhaj et al. (Accepted) to a multi-robotic scenario and present a novel problem of Multi-Robot Simultaneous Exploration and Coverage (MR-SimExCoverage) which combines the advantages of both online and off-line coverage algorithms. We illustrate the proposed MR-SimExCoverage using Voronoi Partition based Coverage using Manhattan distance (Manhattan-VPC), a “partition and cover” strategy as underlying multi-robot coverage algorithm Vishnu & Guruprasad (Accepted) and single robot frontier based exploration strategy for mapping. We show that the proposed algorithm to solve the MR-SimExCoverage problem guarantees to provide a complete and non-overlapping coverage. Simulation results using Matlab/V-rep are provided to demonstrate the proposed algorithm.

7.2.2 Problem Statement

We consider a region $Q \subset \mathbb{R}^2$, topologically connected, known in terms of its boundaries, as area to be covered by N mobile robots. The region Q may contain $O = \{O_1, O_2, \ldots, O_m\}$ a set of obstacle not known a priori to the robots in terms of their number, size, shape or location within Q. The coverage tool attached to the robot has a footprint which is a square of size D. The robot is also equipped with obstacle detection sensor whose range is assumed to be sufficiently large compared to the region . The problem addressed in this work is to find a strategy to efficiently use the sensor and cover $Q \backslash O$ using N robots completely, without any coverage overlap. We use an approximate cellular decomposition scheme (Gabriely & Rimon (2001)), and the region is said to be covered if all the free cells are visited by the robot.

7.2.3 MR-SimExCoverage Problem

In this section, we discuss the proposed multi-robot simultaneous exploration and coverage problem. First we discuss the situation motivating this work.

Online vs Off-line CPP

In an online CPP algorithm, robot has to *sense* for obstacles and *plan* for the next move at each cell, before *act*ing to move to the next planned cell. In an off-line CPP algorithm, complete robot path is planned before the start of coverage. During coverage, robot has to only move according to the generated path. The *plan*ning phase and *act*ing phase are decoupled. This has an advantage of completing the coverage faster compared to its online counterpart as intermittent sensing and planning is absent, in spite of the fact that both online and off-line algorithms using same logic (such as online STC and off-line STC Gabriely & Rimon (2001)).

Online CPP algorithms have several disadvantages compared to their off-line counterpart.

- First problem is associated with the sensing. Consider for example online

STC Gabriely & Rimon (2001) algorithm. Here, in each cell, robot has to scan (at most) three neighboring prospective cells for presence of obstacles. This scanning may require the robot physically turn by about 270^0 (at most). This process of sensing and physical turning (if required, depending on how the sensor is mounted on the robot), results in battery consumption (for sensing, processing, and turning the robot when required) and also time delay.

- After sensing, the robot has to plan (decide) the next cell to which it has to move. As in each cell the robot needs to sense and plan its next move, the robot can move only in steps stopping at each cell leading to slow coverage.
- This stop and move (even when the final path segment is straight line) consumes additional battery for acceleration and stopping (breaking) leads to lose of power.

These problems do not arise in an off-line CPP algorithm as entire path generated off-line is available to the robot before it begins to move. Once it starts moving to cover the region, it can move continuously without stopping in between, and also no sensing is required, leading to much lesser battery consumption and faster completion of coverage. The scenario is similar in any online CPP algorithm compared to its off-line counterpart. For example, in online version of Boustrophedon decomposition based coverage Choset (2000), robot has to sense its environment continuously looking for the critical points.

Thus, whenever it is possible, it is more convenient to use an off-line CPP algorithm.However, off-line CPP algorithms require prior knowledge of the map of the environment. Now consider a more common situation, where the robot is equipped with fairly large range sensor to detect obstacles, rather than a short range proximity sensor typically assumed in CPP algorithms. The online algorithms still use information about occupancy of only neighboring grids and discard the rest of information the sensor provides, simply because there is no provision in the online CPP algorithm for using this information about occupancy map of the farther cells to plan the next series of moves.

Combining Exploration and Coverage

In this work, we combine the problem of multi-robotic coverage and exploration and propose a novel multi-robot simultaneous exploration and coverage (MR-SimExCoverage) problem to take advantage of both online and off-line CPP algorithms for the coverage of arbitrary unknown regions. The proposed MR-SimExCoverage problem is illustrated in Figure 7.5. Here, the robots perform exploration and update the map, which in turn is used for off-line CPP. While the path is generated for coverage, exploration is performed at instances which maximize the information gain in terms of map building. The sequence of coverage and exploration continues until the entire region is completely covered. The primary task here is coverage, while exploration is used as an aid. A map of the area is also generated as a byproduct. Here, the path planning is required only based on coverage task. Exploration is carried whenever robot, which is following the CP generated reaches a point (cell) where exploration leads to maximum information gain (such as on reaching a frontier cell).

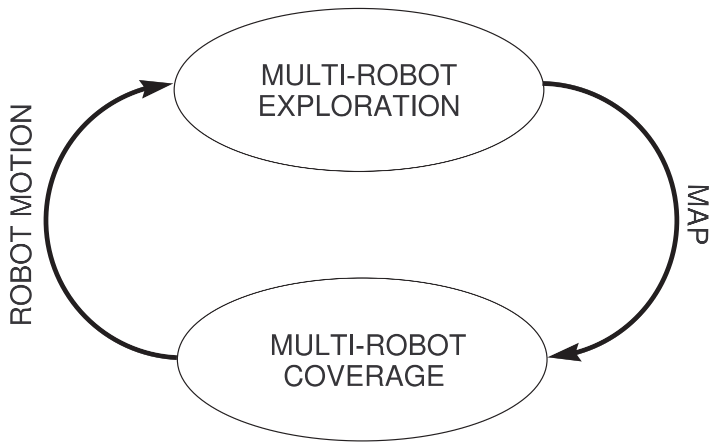


Figure 7.5: Block diagram illustrating the MR-SimExCoverage problem.

There are several possible ways to solve the underlying multi-robot CPP problem and multi-robot exploration problems in order to solve the MR-SimExCoverage problem discussed here. Any combination of algorithms may be used in principle to solve the proposed MR-SimExCoverage problem.

Multi-robot coverage strategy and multi-robot exploration strategies can be independent as long as path planing is decided by the underlying MR-CPP algorithm and exploration strategy provides necessary map to the MR-CPP algorithm. Figure 7.6 shows MR-SimExCoverage by three robots, while Figure 7.7 show the problem being solved by the ith robot. Each robot plans a CP based on the consolidated map and the coverage information (such as cells covered) obtained from other robots and performs exploration as it moves along the planned CP. The exploration information may also be shared with the neighboring robots to update the map.

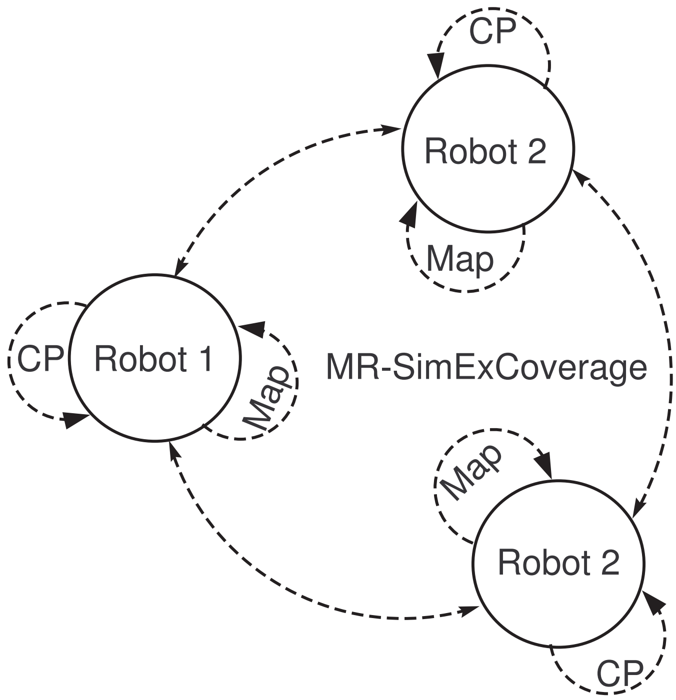


Figure 7.6: Block diagram illustrating a typical distributed MR-SimExCoverage by three robots.

The Proposed MR-SimExCoverage Algorithm

In this section we present an algorithm to solve the MR-SimExCoverage problem. We use a "divide and conquer" approach, where we use a "partition and cover" and "partition and explore" strategies converting a multi-robot SimExCoverage problem into several single robot SimExCoverage problems. We

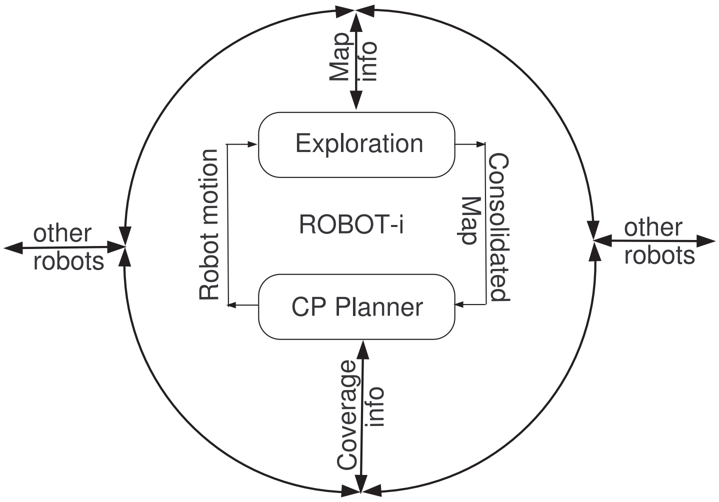


Figure 7.7: The ith robot performing the MR-SimExCoverage in a distributed manner.

use Voronoi Partition based on Manhattan distance to decompose the region to be covered $Q \setminus O$ into generalized (Manhattan) Voronoi cells (referred to as Manhattan Voronoi cell for brevity) with the initial position of the robots as nodes, and then let each robot solve a single robot SimExCoverage problem within the corresponding Manhattan Voronoi cell. . Figure 7.8 shows the SR-SimExCoverage being performed by each robot within the corresponding Manhattan Voronoi cell. Unlike in the general situation as illustrated in Figures 7.6 and 7.7, robots do not need explicit communication once the Q is partitioned. However, note that the partitioning itself depends on the robots' position.

7.2.4 Illustrative Example

In this section, we describe the process involved in the MR-SimExCoverage using STC as underlying CPP algorithm. In this example the exploration, spanning tree edge creation and the robot path (through the sub nodes) are hand drawn for illustration purpose. They do not replace the simulation results presented in later section.

The scenario considered is shown in Figure 7.9. We consider two robots as it essentially captures the multi-robotic scenario and at the same time easy to

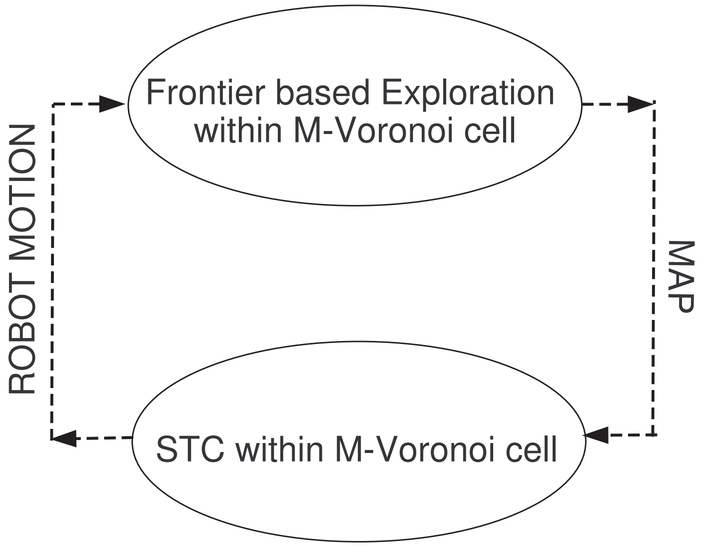


Figure 7.8: Each robot performs a single robot SimExCoverage within the corresponding Manhattan Voronoi cell.

comprehend. The region is divided into 5×5 major cells of size $2D \times 2D$. Each major cell has four sub cells of size $D \times D$. Region is partitioned into Manhattan Voronoi cells (boundary is shown as thick line). We have only two cells containing obstacles. This makes it simple to explain the process.

Various stages of exploration, spanning tree construction, and robot coverage path are shown in Figure 7.11. Figure 7.10 illustrates process of exploration (first instance in this case). The cells which are visible (accessible) to the senor are explored, while the cells on which the obstacle shadow (shown in grey) is cast (partially or fully) are not explored. Both robots perform the first instance of exploration from the cell they are initially located as illustrated in Figure 7.11 (a). Explored (free) cells are shown in white while unexplored cells are shown with grey. Both the occupied cells are explored/detected in the first instance itself. After exploration, spanning tree shown as solid line passing through the major nodes, that is, center of major cells is created through explored free cells. In this stage the "exploration window" cell pairs for robot $R1$ are $\{(d_5, d_4), (c_5, c_4)\}$ and for robot $R2$ are $\{(c_1, c_2), (b_1, b_2), (d_2, c_2)\}$. CP is created on right side of the spanning tree edge though the sub nodes.

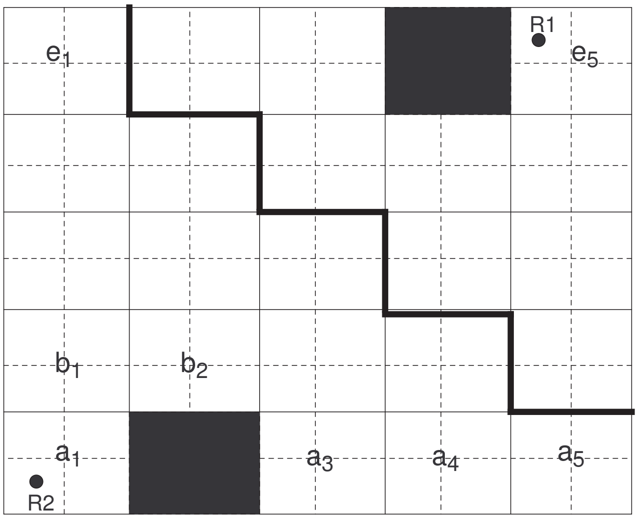


Figure 7.9: A scenario with two robots $R1$ and $R2$. The region is gridded into $2D \times 2D$ major cells (shown in solid grid lines) and each major cell is divided into $D \times D$ (size of robot/coverage tool footprint) sub cells (shown with dashed grid lines). Region is partitioned into Manhattan Voronoi cells (boundary is shown as thick line). Cells are numbered as $a_1, a_2, \ldots, e_5$ to aid the description.

As the robot $R1$ reaches a sub cell corresponding to major node d_5, it encounters an exploration window (d_5, d_4). It performs the second instance of exploration here and updates the explored cells (shown with white cells) as shown in Figure 7.11 (b). Note that the exploration window (d_5, d_4) is no more an exploration window as both these cells are now explored. Exploration window list is updated along the frontier cells. A spanning tree is created through newly explored cells As the robot $R2$ is yet to encounter the exploration window, it does not perform any exploration.

Robot $R1$ on reaching a sub cell in node d_4 it encounters the exploration window (d_4, c_4) and performs the third instance of exploration. This is shown in Figure 7.11 (c). With these three instances of exploration the robot $R1$ successfully explored entire region V_{2DM1}, the corresponding Manhattan Voronoi cell.

Robot $R2$ encounters the exploration window (d_2, c_2) when it reaches a sub cell in major node d_2, it performs second instance of exploration and creates spanning tree over the newly explored (major) cells. With these two exploration instances, the robot $R2$ completely explores V_{2DM2}, the corresponding

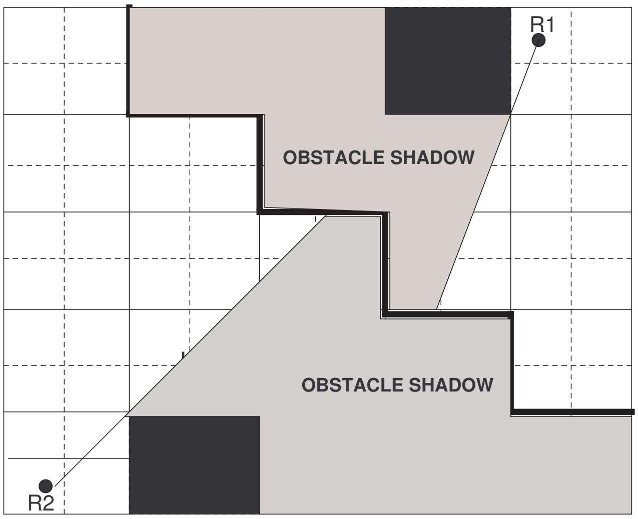


Figure 7.10: The obstacle shadow obstructs exploration sensor's field of view, creating an obstacle shadow shown in grey. The cells which are fully or partially covered by the obstacle shadow are not explored. Rest of the cells shown with white (free) and black (occupied) are explored.

Manhattan Voronoi cell. Note that after each exploration, the newly created spanning tree branches out of the existing spanning tree. The concatenated successive spanning trees form a spanning tree within each Manhattan Voronoi cells. Also, the coverage path generated by each of the robots pass through every sub cell corresponding to free major cells, completely covering corresponding Manhattan Voronoi cells. This is shown in Figure 7.12. As each robot completely explores the corresponding Voronoi cells, entire region is explored. Also, as each robot covers the corresponding Manhattan Voronoi cell completely without any overlap, the entire region is covered completely without any overlap.

7.2.5 Analysis of the MR-SimExCoverage-STC Algorithm

In this section we provide a discussion on some of the properties of the MR-SimExCoverage problem presented and the proposed algorithm using Manhattan-VPC with off-line STC as underlying single robot coverage and frontier based exploration algorithms. We observe that with the proposed algorithm, each robot solves a single-robot SimExCoverage problem within the corresponding Manhattan Voronoi cell using off-line STC as underlying CPP algorithm and an frontier

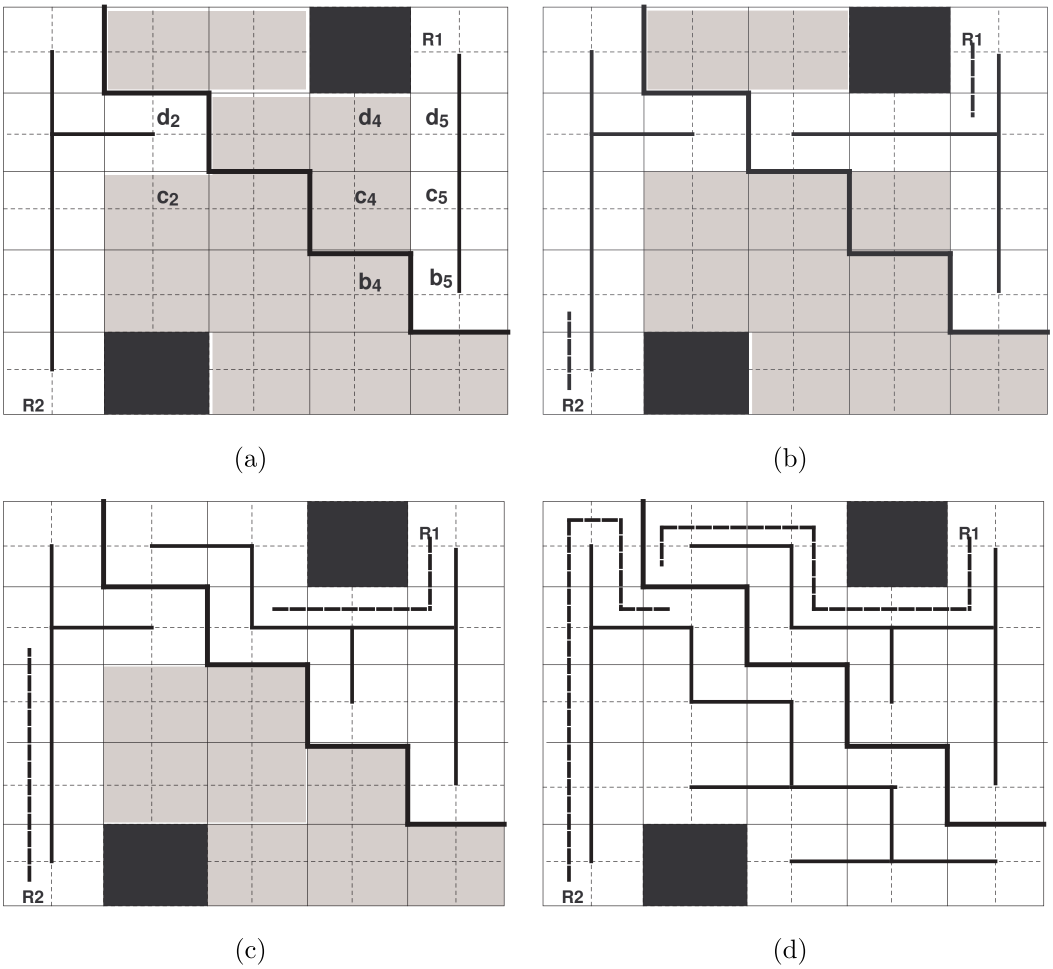


Figure 7.11: Various stages of exploration, spanning tree construction, and robot coverage path.(a)Both robots perform the first instance of exploration from the cell they are initially located.Explored (free) cells are shown in white while unexplored cells are shown with grey.After exploration, spanning tree shown as solid line passing through the major nodes, that is, center of major cells is created through explored free cells. In this stage the "exploration window" cell pairs for robot $R1$ are $\{(d_5, d_4), (c_5, c_4)\}$ and for robot $R2$ are $\{(c_1, c_2), (b_1, b_2), (d_2, c_2)\}$. CP is created on right side of the spanning tree edge though the sub nodes.(b)Second instance of exploration and updating of the explored cells (shown with white cells). As the robot $R2$ is yet to encounter the exploration window, it does not perform any exploration.(c) Robot $R1$ on reaching a sub cell in node d_4 it encounters the exploration window (d_4, c_4) and performs the third instance of exploration.(d) Fully explored workspace with the generated spanning tree for coverage.

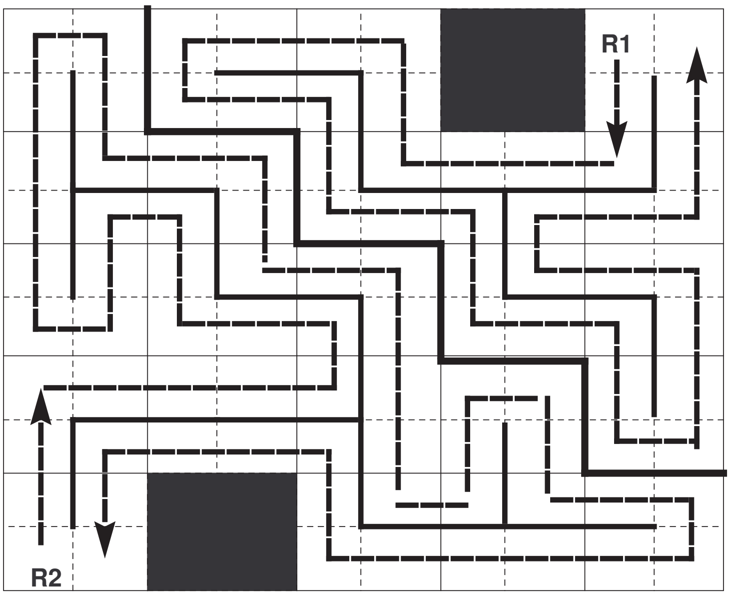


Figure 7.12: Scenario of completed MR-SimEx Coverage.The concatenated successive spanning trees form a spanning tree within each Manhattan Voronoi cells. Also, the coverage path generated by each of the robots pass through every sub cell corresponding to free major cells, completely covering corresponding Manhattan Voronoi cells

based exploration strategy. We call this single robot strategy as Single Robot-SimExCoverage-STC algorithm. One of the important assumptions we make is that each of the Manhattan-Voronoi cells is topologically connected/contiguous regions in space.

7.2.6 Completeness and Non Overlapping Coverage

First we provide a few results to establish the completeness and non-overlapping nature of the coverage with the proposed MR-SimExCoverage-STC algorithm.

Lemma 1 *With the proposed SR-SimExCoverage-STC algorithm the robot successfully explores the corresponding Manhattan Voronoi cell.*

Proof. The robot path passes through all sub-cells of the explored region by the completeness property of the off-line STC algorithm (Gabriely & Rimon (2001)). Hence, the robot CP created by the STC algorithm within the currently explored region encounters an exploration window if and only if the unexplored region contains free space.There will be atleast one exploration window if the unexplored region contains accessible free cells.If exploration window exists then the CP will pass through the F(E) cell.Whenever an exploration window is encountered, the exploration process adds a newly explored region (containing at least some free cells) to the previously explored region. Old and newly explored regions form a contiguous region connected by the exploration window through which the newly explored region was discovered. □

Lemma 2 *With the proposed SR-SimExCoverage-STC algorithm the robot successfully covers the corresponding Manhattan Voronoi cell completely.*

Proof. The tree created by the off-line algorithm spans all the major cells within the currently explored region by the property of the off-line STC algorithm (Gabriely & Rimon (2001)). The tree created in the newly explored region after an instance of exploration spans all the major cells within the newly explored region again by the property of the off-line STC algorithm (Gabriely & Rimon (2001)). These two trees are connected only at the major cell corresponding to the frontier(E) cell (first cell in the exploration window pair) from where the exploration was performed. It is easy to see that the concatenated graph is still a tree and spans all the major cells in the union of old and newly explored regions. In fact the newly created spanning tree branches out of the original spanning tree. Now as the SR-SimExCoverage guarantees complete exploration within the corresponding Manhattan Voronoi cell (by Lemma 1), a tree (which happens to be a spanning tree) is created over all the free major nodes within the explored region, which is whole of the free space within the corresponding Manhattan Voronoi cell. Now by the property of the off-line STC algorithm, the CP passes through all the sub cells corresponding to free major cells in the corresponding Manhattan Voronoi cell[1]. □

Lemma 3 *With the proposed SR-SimExCoverage-STC algorithm the robot*

[1]Note that the STC algorithm guarantees only resolution (major cell) level completeness.

successfully covers the corresponding Manhattan Voronoi cell without any coverage overlap.

Proof. Follows from the property of the off-line STC algorithm (Gabriely & Rimon (2001)). □

Proposition 1 *With the proposed SR-SimExCoverage-STC each robot explores and provides complete and non overlapping coverage of the corresponding Manhattan Voronoi cell.*

The proof follows from Lemmas 1, 2, and 3.

Theorem 4 *The proposed MR-SimExCoverage-STC guarantees complete and non-overlapping coverage of the $Q \setminus O$.*

Proof. Union of free regions within all the Manhattan Voronoi cell is $Q \setminus O$. As free region within each Manhattan Voronoi cell is guaranteed to be covered completely by individual robots (Proposition 1), entire free region $Q \setminus O$ is covered completely. Each robot providing a non-overlapping coverage (again by Proposition 1), and also, intersection between any two Manhattan Voronoi cell (in terms of major cells) being null, guarantees that the CP is non overlapping. □

7.2.7 Reduced Energy and Time-to-complete Coverage

Another important result, which we only discuss informally in this section is about reduction in time and battery used to complete the coverage with the proposed MR-SimExCoverage as compared to an online coverage algorithm. As we discussed earlier, a typical online coverage algorithm needs to use its sensors continuously (such as looking for critical points in online version of Boustrophedon decomposition based CPP (Choset 2000) or at every cell (such as looking for free neighboring cells in online STC (Gabriely & Rimon 2001). With the proposed simultaneous exploration and coverage, only a smaller number of exploration instances are sufficient to provide complete map to the CP planner, which generates CP using off-line version of the corresponding coverage algorithm. While fewer instances of exploration reduces energy requirement for sensing (including physical turning of robots to scan, if required) substantially, off-line CP planning reduces time required for coverage by ensuring that the robots move continuously without stopping (and may be turning for sensing) and waiting for sensing and planning at each cell. The simulation results are presented in results and discussions chapter.

7.3 SUMMARY

In this chapter a novel methodology “Simultaneous Exploration and Coverage” for mobile robots, which combines exploration, mapping, and coverage path planning problems is discussed. The CPP generates robot path, while the exploration provides the map required for CPP. We proposed a SimExCoverage algorithm using a frontier based exploration strategy and off-line STC algorithm as a solution to the proposed SimExCoverage problem.

The proposed SimExCoverage algorithm was described with an illustrative example. Simulation results at graph-level demonstrated that the proposed algorithm successfully covers an arena, not known to the robots a priori, completely without any overlap. Further, the algorithm was implemented on a Turtlebot within ROS-Gazebo environment, and the simulation results show that the robot path generated passes through all the major cells without any repetitive coverage.The simulation results are presented in ”results and discussions” chapter.

The algorithm proposed combines exploration and coverage .The chapter does not studies the optimality of the solution obtained, rather it combines online and offline coverage methods to get an efficient coverage simultaneously with exploration. The obstacle details (maps) obtained after each exploration phases are exact representation of the scenario so that a coverage path can be generated based on it. The coverage path so generated will be an optimal path (based on the algorithm used) for that map slice. The next map slice is obtained in next exploration phase and so on. If we merge all these map slices together we will get the complete map of the scenario. Also since optimal coverage paths are generated for each map slices, the merging of which gives an optimal solution.

CHAPTER 8

RESULTS AND DISCUSSIONS

In this chapter the simulation results obtained for each of the chapters discussed are presented.

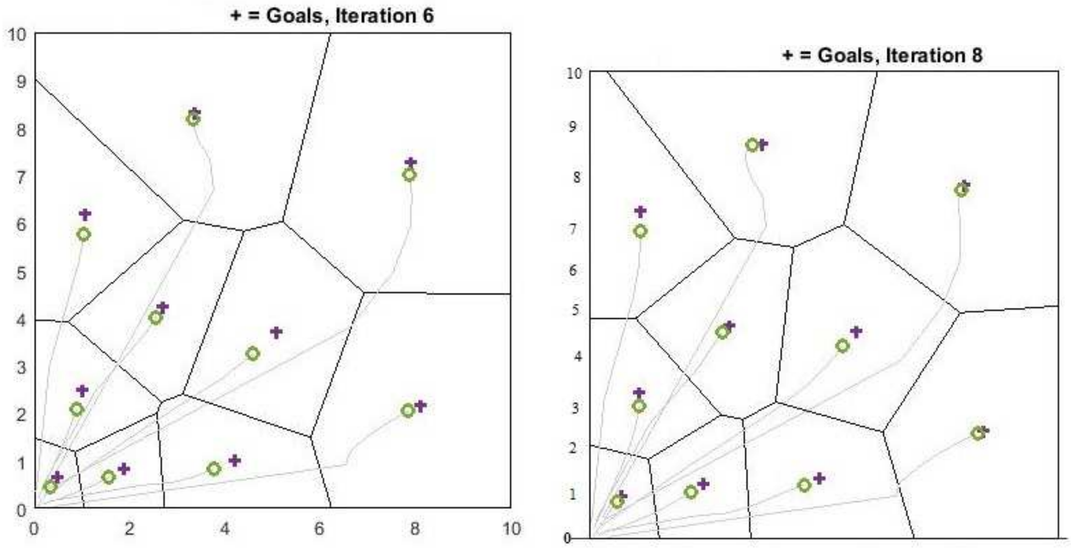
8.1 CENTROIDAL VORONOI PARTITION USING VIRTUAL NODES

In this section we provide simulation results to demonstrate the proposed deployment and optimal partitioning strategy for centroidal Voronoi partitioning using virtual nodes. The simulation is carried out in Matlab environment with ten robot system. Total of 100 iterations are done. Figures 8.1 to 8.3 shows the scenario at the end of various iterations.

In all the figures, small circles, o, indicates the centroids of the Voronoi cell and + indicate the current position of the virtual nodes. Note that the initial location of the virtual nodes are identical to that of the physical robots. During the deployment process only the virtual nodes move, while the physical robots are stationary. It can be observed that from an initial non-uniform partition as shown in figure 8.1(a), the deployment of the virtual nodes into centroidal Voronoi configuration (that is, the virtual nodes are at or sufficiently close to the respective centroids) leads to a more uniform Voronoi partitioning.

8.2 MANHATTAN DISTANCE BASED VORONOI PARTITIONING

In this section we provide simulation results to demonstrate the proposed "partition" and "cover" strategy using Manhattan distance based Voronoi partitioning. We considered Boustrophedon coverage algorithm (Choset 2001), a coverage algorithm based on exact cellular decomposition and Spanning Tree Coverage (STC) (Gabriely & Rimon 2001), a coverage algorithm based on approximate cellular decomposition, as representative single robot coverage



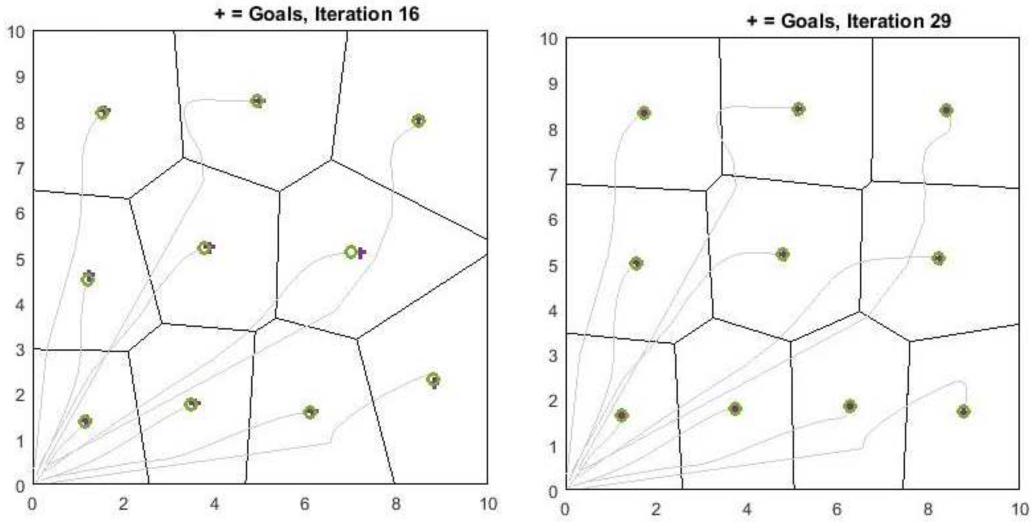
(a) (b)

Figure 8.1: A 10 robot system implemented in Matlab.(a)The scenario after 6 iterations. The o and the + sign represents virtual nodes and centroids (goals) respectively.(b)The scenario after 8 iterations.

strategies. The simulations are carried out at graph level in Matlab as well as in V-Rep simulator. Experiments on physical robots (Fire Bird) are also performed.

Figure 8.4(a) shows boustrophedon-like coverage path using a standard Voronoi partition. We can observe incomplete coverage as some of the cells through which a partition boundary passes through are not covered leading to incomplete coverage. At the same time we may also observe that a few cells adjacent to uncovered cells are covered twice (as there are no return path), as shown by dashed lines, leading to coverage overlap. Figure 8.5(a) shows STC path again using standard Voronoi partition. Here too, we can observe incomplete coverage as some of the cells through which a partition boundary passes through are not covered leading to there is no coverage overlap in incomplete coverage. Unlike in the case of boustrophedon-like coverage, there is no coverage in this case due to the property of STC algorithm. However, number of sub cells uncovered along the diagonal partition boundary is more than that in the case of boustrophedon coverage.

Figure 8.4(b) shows coverage path using boustrophedon coverage algorithm



(a) (b)

Figure 8.2: The scenario after(a)16 iterations and (b)29 iterations.

with Manhattan distance-based Voronoi partitioning of $D \times D$ gridded space. We can observe that there are no uncovered cells. However, a few cells along the partition boundary are covered twice (shown with dashed lines) leading to coverage overlap.

Figure 8.5(b) shows STC path again using Manhattan distance-based Voronoi partitioning of $D \times D$ gridded space. Here too, we can observe incomplete coverage as some of the cells through which a partition boundary passes through are not covered leading to there is no coverage overlap in incomplete coverage. As observed before, STC algorithm does not lead to coverage overlap. Figures 8.4(c) and 8.5(c) show coverage path using proposed Manhattan distance-based Voronoi partitioning of the $2D \times 2D$ gridded space, with Boustrophedon-like coverage and STC algorithms, respectively. As it can be observed from these results, the coverage is complete and without any overlap in both cases. That is, a complete, non-overlapping coverage is achieved irrespective the single-robot coverage algorithm used.

Next we present results of simulation experiments with the actual robot motion using both Boustrophedon-like coverage and STC algorithms in V-Rep simulator. We used a robot model known as DR12 robot available within the

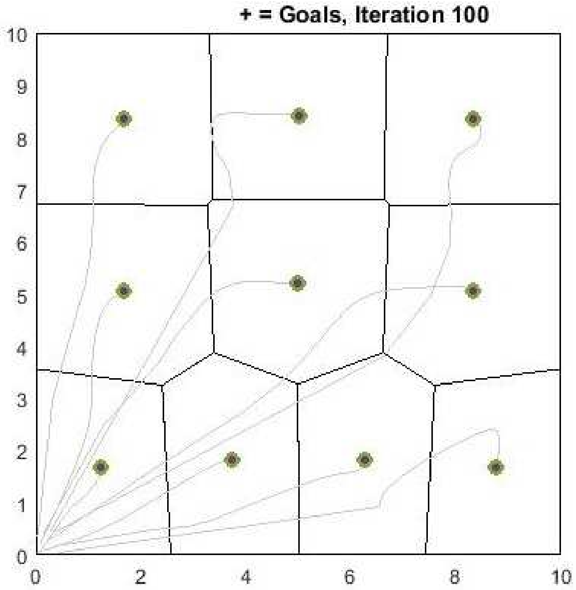
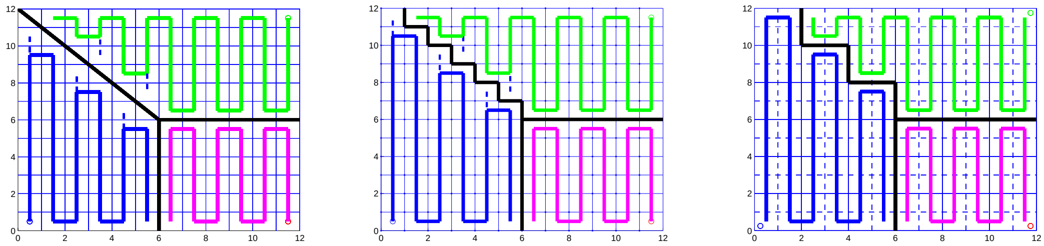


Figure 8.3: . The final scenario after 100 iterations. The $'o'$ and the '$+'$ sign represents virtual nodes and centroids respectively are at the same positions. The partitioning obtained is uniform with respect to the area allotted to each robots.

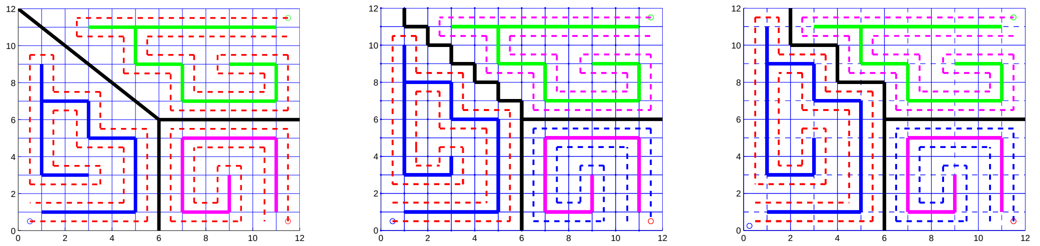
simulation environment to demonstrate the proposed methodology. Figures 8.6(a) and (b) show the actual (simulated) robot path with Boustrophedon-like coverage and STC algorithms, respectively, as the underlying single-robot coverage algorithms. The deviation of the robot path from the graph-level path through the centers of sub cells is due to imperfect robot motion because of wheel skidding, turning errors, etc. In spite of these imperfections, it may be observed that the robot moves through all sub ($D \times D$) cells providing a complete coverage without any repetitive coverage, irrespective of the single robot CPP algorithms used.

Finally we present results of experiments with Fire Bird V, a Atmega2560 based robotic research platform designed by ERTS Lab, CSE, IIT Bombay and manufactured by Nex Robotics Pvt Ltd. The differential wheeled robot is equipped with 3 white line sensors, 5 Sharp GP2D12 IR range sensor, 8 analog IR proximity sensors, 8 analog directional light intensity sensors, and 2 position encoders. The robot has wireless ZigBee communication, USB communication, Wired RS232 (serial) communication , and simplex infrared communication capabilities.



(a) (b) (c)

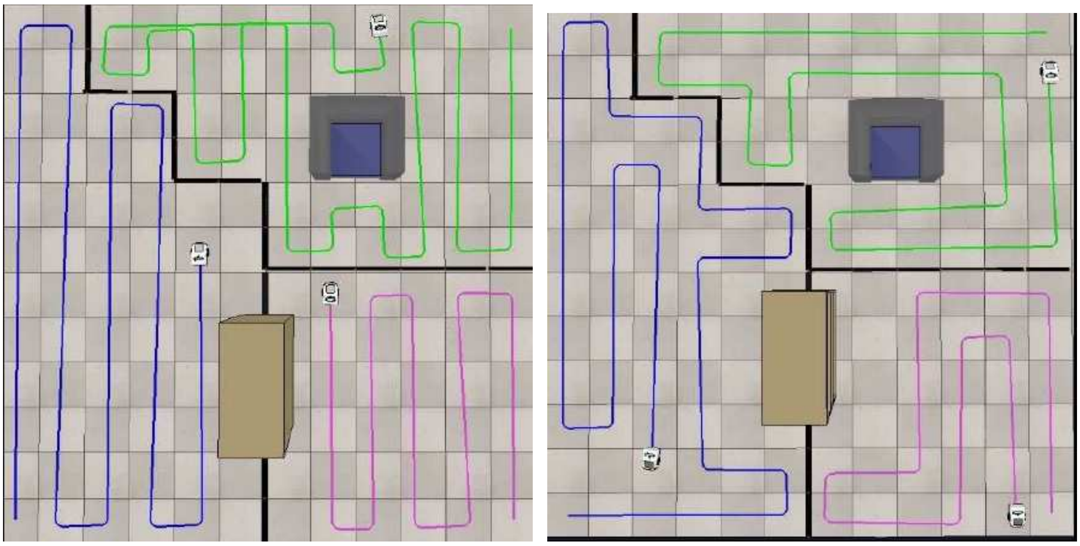
Figure 8.4: Coverage by three robots using boustrophedon-like coverage algorithm using (a) standard Voronoi partition, Manhattan distance based Voronoi partition using (b) $D \times D$, and (c) $2D \times 2D$.



(a) (b) (c)

Figure 8.5: Coverage with STC algorithm using (a) standard Voronoi partition, Manhattan distance based Voronoi partition using (b) $D \times D$, and (c) $2D \times 2D$.

Localization is one of the fundamental requirement in any path planning algorithm such as that addressed in this work. Localization may be achieved by the use of odometric sensors/dead reckoning. However, it is well known that it is prone to errors and accuracy also depends on the type of surface on which the robot moves. In this work, as the algorithm is based on $2D \times 2D$ grids and the robots move along the center of sub nodes (which form $D \times D$ grids), we use grids printed with solid black lines. The robot uses its line following senors to follow the grid (as the path is always along the grids). In reality, the robots can use the onboard proximity sensors and plan the path (STC or Boustrophedon-like) and move along the path (that is, grid lines) using the onboard line following sensors. However, we provide the planned path to the robot and let it follow it using line following algorithm, as the propose of this experiment is only to demonstrate the proposed "partition" and "cover" approach rather that the actual single robot CP



(a) (b)

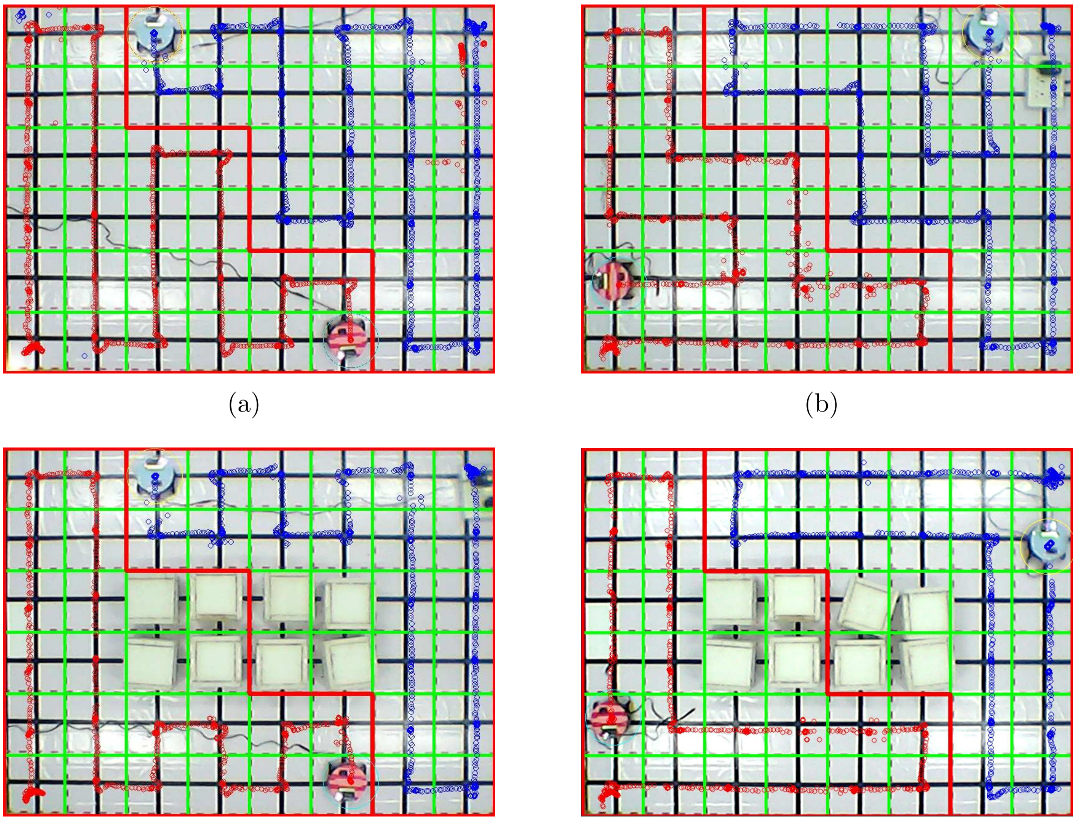
Figure 8.6: Robot path using (a) Boustrophedon-like coverage and (b) STC algorithm, as the single robot coverage algorithm in V-Rep simulator.

Planning algorithm.

Figures 8.7(a)-(d) show the coverage path with two fire bird robots using the proposed Manhattan-VPC.Figures 8.7(a) and (b) show the coverage path (green and red circles being traced) in a obstacle-free environment, using the Boustrophedon-like coverage and STC algorithms, respectively. 8.7(c) and (d) show the coverage path (green and red circles being traced) in the same environment, but containing obstacles, using the Boustrophedon-like coverage and STC algorithms, respectively. In all situations, it may be observed that the robots cover all the free sub cells.

8.3 GEODESIC DISTANCE BASED VORONOI PARTITIONING FOR MULTI ROBOT COVERAGE

In this section we provide illustrative examples to demonstrate the proposed geodesic-VPC strategy using two single-robot coverage algorithms reported in the literature, namely, spanning tree-based coverage (STC)Gabriely & Rimon (2001) and Boustrophedon coverageChoset (2000) algorithms. While the former represents the algorithms based on approximate cellular decomposition, the latter

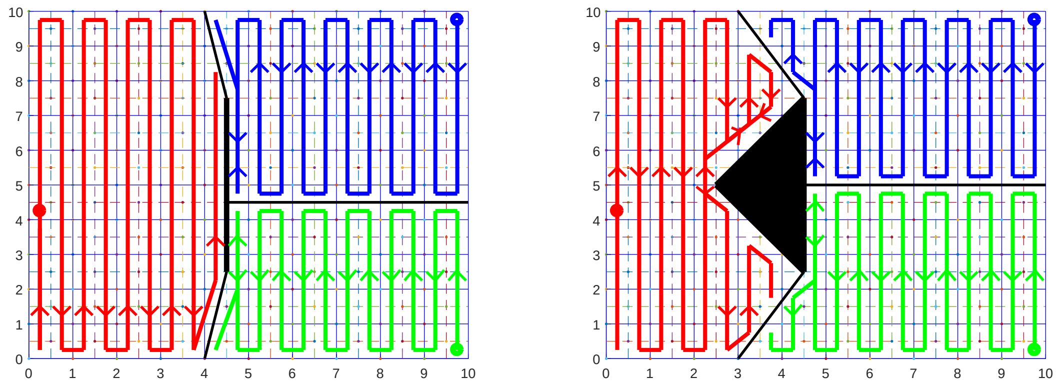


(a) (b) (c) (d)

Figure 8.7: Coverage path followed by robots using the proposed Manhattan-VPC. Obstacle free environment: (a) Boustrphedon-like coverage and (b) STC algorithm. Environment with obstacle: (c) Boustrphedon-like coverage and (d) STC algorithm.

represents that based on exact cellular decompositionChoset (2000). These two algorithms are used here only for the purpose of illustration/demonstration. Any single robot coverage algorithm may be used for the proposed geodesic-VPC. In fact the processes of *partition*ing and *coverage* are decoupled. The overall coverage performance (in terms of completeness and overlap) depends on the single-robot coverage algorithm. However, Geodesic-VPC strategy ensures that there is no duplication of coverage between any two robots and also each robot needs to cover a contiguous region. We have used Matlab for geodesic Voronoi partitioning and graph-level simulation of single-robot coverage algorithms. By graph-level simulation we mean, that only the path generated by the path-planning algorithm over the graph formed by the grids is shown rather than actual path followed by the robot. We also carried out simulation experiments using a realistic robot model within V-rep simulation environment.

First, we present results of graph-level simulation experiments. Figure 8.8 shows coverage path generated by Boustrophedon coverage algorithm in two scenarios, a) with a line obstacle, and b) with a triangular obstacle. In both situations, each geodesic Voronoi cell is a contiguous (topologically connected) region, and corresponding robot covers the cell. Robot paths are shown with directed lines, where the head direction shows the direction of robot motion. Being a exact cellular decomposition based method, each robot covers the corresponding geodesic Voronoi cell completely. However, by the nature of the Boustrophedon coverage algorithm, the coverage path is repetitive near the boundaries of the cell and the obstacle. This is shown by two sided arrows. Figure 8.9 shows coverage path generated using STC algorithm again in two



(a) (b)

Figure 8.8: Coverage path generated by Boustrophedon coverage algorithm using geodesic Voronoi partitioning of the workspace in presence of obstacle in two scenarios, (a) with a line obstacle and (b) with a triangular obstacle. Arrow marks show the direction of the robot motion. Boustrophedon coverage algorithm though provides complete coverage, it leads to coverage overlap (indicated by arrow marks in both directions) at several instances.

scenarios, a) with a line obstacle, and b) with a triangular obstacle. In STC, we divide the region of interest into square grids called minor cells of size $D \times D$, which is the size of the coverage tool footprint. Four minor cells are combined to form a major cell/grid of size $2D \times 2D$. In each of the geodesic Voronoi cells, each of completely free major grid is covered and there are no repetitive paths.

However, the STC algorithm does not create path through partially occupies major grids, and leading to incomplete coverage near the partition/obstacle boundaries. With both single robot coverage algorithms, the geodesic Voronoi

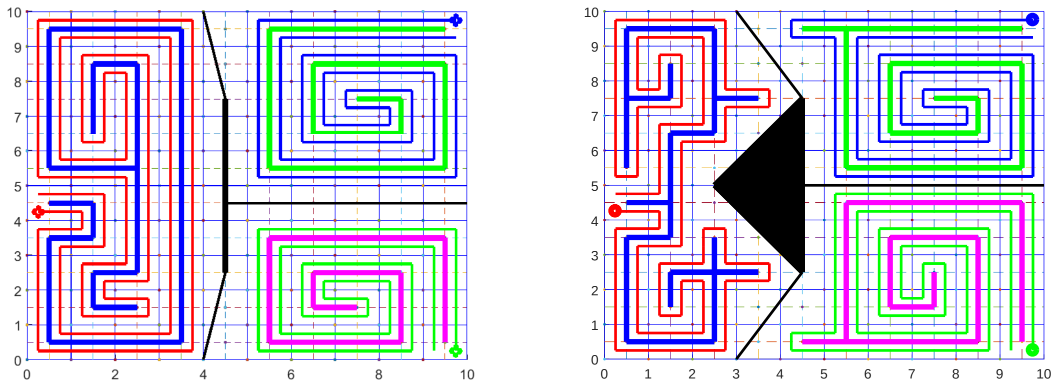


Figure 8.9: Coverage path generated by STC algorithm using geodesic Voronoi partitioning of the workspace in presence of obstacle using three robots in two scenarios, (a) with a line obstacle and (b) with a triangular obstacle. The robot path is shown by thinner lines around the spanning tree created (Shown with thick line) within each of the Geodesic distance based Voronoi cells). STC provides non-overlapping coverage within each Geodesic Voronoi cell, with coverage gaps near the partition/obstacle boundaries.

partitioning provides a contiguous region for coverage by individual robots and also, there is no overlap between the coverage area of any two robots. Boustrophedon and STC algorithms are used here only for demonstration purpose. Any single robot coverage algorithms such as that given in Ranjitha & Guruprasad (2015*b*), which provides a complete coverage of each cell without unnecessary coverage overlap can be used.

Now we present the results of robot motion simulation with Geodesic-VPC using both Boustrophedon coverage and STC algorithms in V-Rep simulator. We used a robot model known as DR12 robot, a differential wheeled robot with bumper/contact sensor, available within the simulation environment to demonstrate the proposed Geodesic-VPC. An environment used for simulation is shown in Figure 8.24. Figures 8.11 (a) and (b) show the robot paths generated using Geodesic-VPC with Boustrophedon coverage and STC algorithms, respectively, as the underlying single-robot coverage algorithms. The robot path can be seen to similar to that obtained using the graph-level simulation. Actual robot path deviates slightly from the graph-level path due to motion constraints (such as turning radius) and errors due to wheel skidding/imperfect localization. Both simulation results presented in this section are demonstrative as the focus of this work is on the multi-robotic "partition" and "cover" strategy rather than on the underlying single-robot CPP algorithm.

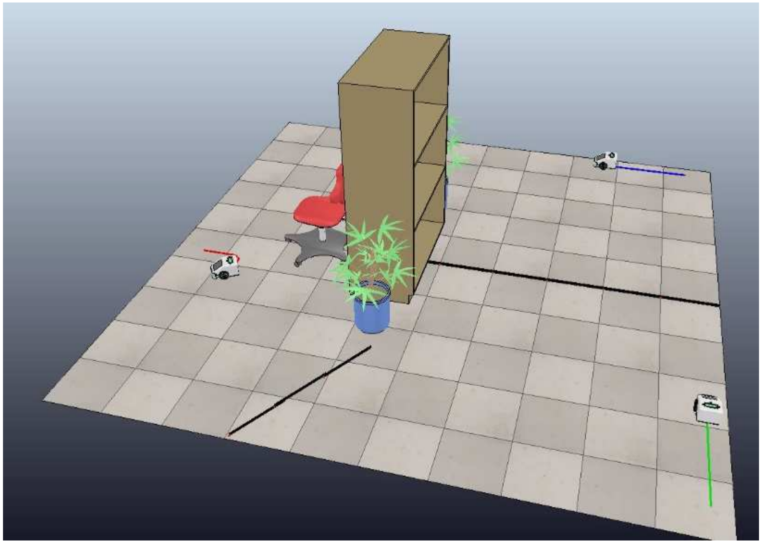


Figure 8.10: A scenario used for simulation in V-Rep simulator environment. Black lines show the geodesic Voronoi cell boundaries.

8.4 GEODESIC MANHATTAN-VPC

In this section we present results of simulation experiments carried out in V-Rep realistic simulation environment and also on physical robots to demonstrate the proposed GM-VPC algorithm and compare it with VPC strategies based on the Euclidean distance, Manhattan distance (Manhattan-VPC), and geodesic distance (Geodesic-VPC).

8.4.1 V-Rep simulation results

Figure 8.12 shows a scenario modeled in V-Rep. DR12 robots, available in V-rep simulator is used for the simulation. Also we have used Boustrophedon-like

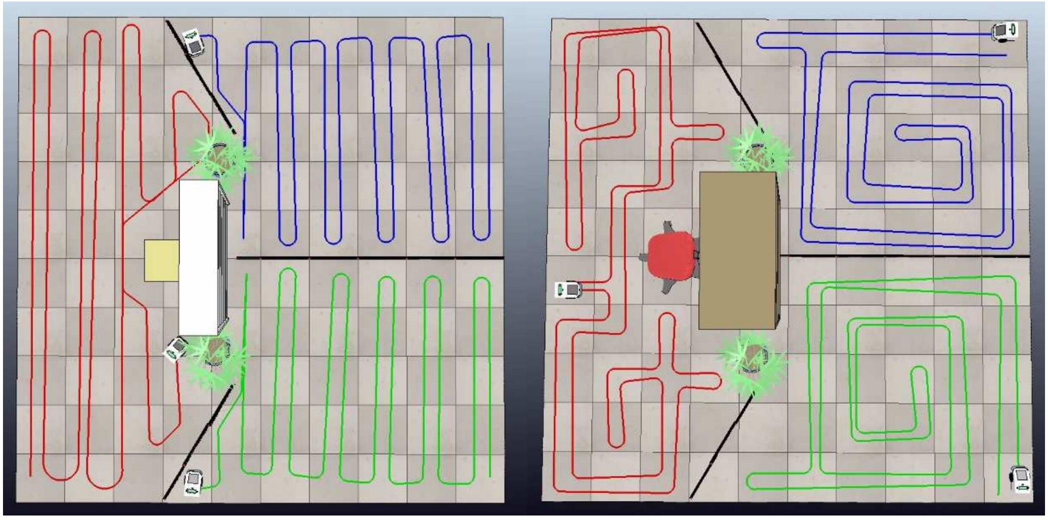


Figure 8.11: Robot path with Geodesic-VPC using the (a) Boustrophedon coverage algorithm, (b) the STC algorithm, as the single robot coverage algorithm in V-Rep simulator.

CPP algorithm and STC Gabriely & Rimon (2001) as underlying single robot CPP algorithms for the purpose of demonstration.

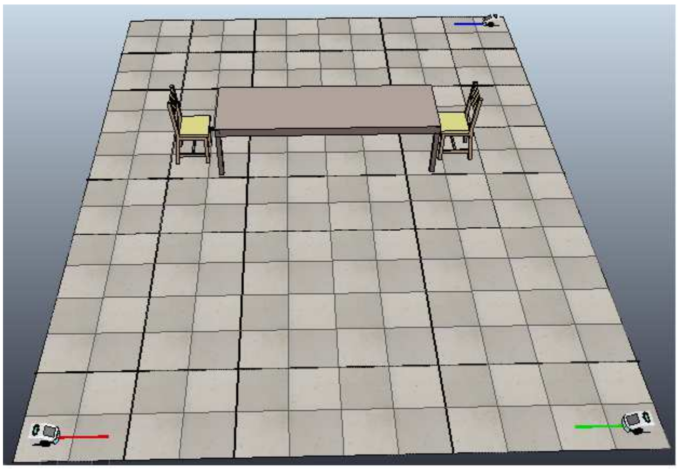
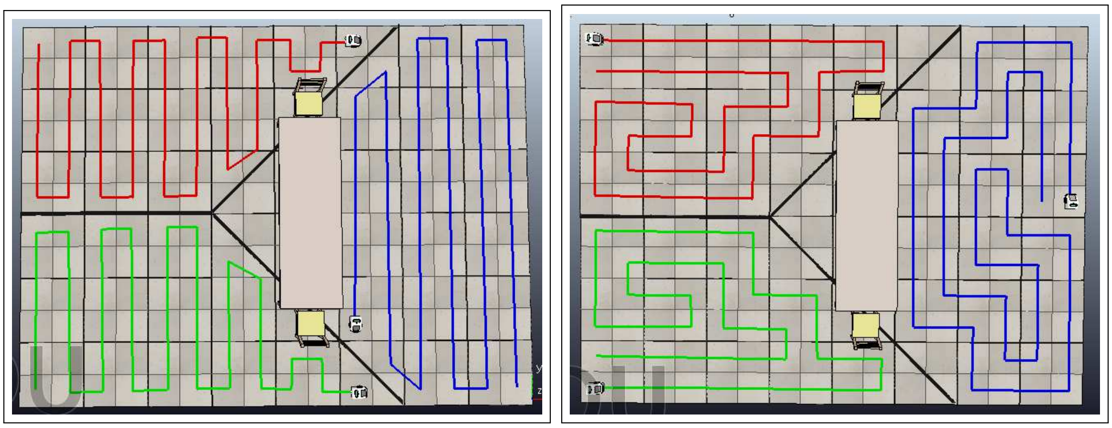


Figure 8.12: A scenario generated in V-rep simulation environment. Three DR12 robots are used for the simulation.

Figures 8.13(a) and (b) show coverage using single robot Boustrophedon-like and STC CPP algorithms, respectively, both using the standard Voronoi partitioning scheme. Both algorithms do not cover a (triangular) portion of V_2 (corresponding to the robot R2, on the right side), which is disconnected from the rest of V_2 where the robot is initially located. Further, the robots do not cover a few cells around the partition boundary with the STC algorithm, but it ensures no coverage overlap. With a careful observation we can note that the robot (with square footprint) coverage area with the Boustrophedon-like coverage path has overlap near the partition boundary, even though there is no retrace or intersection of the path.

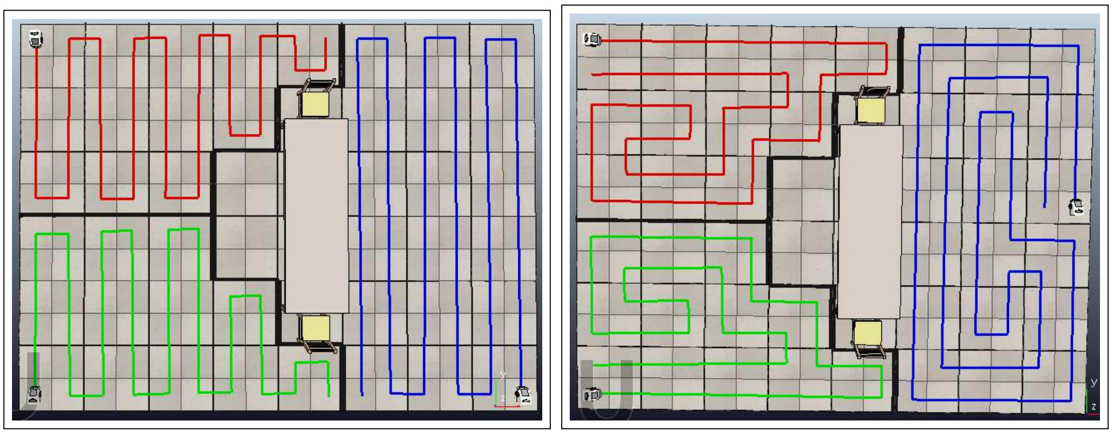


(a) (b)

Figure 8.13: Robot coverage path using (a) Boustrophedon-like and (b) STC CPP algorithm with Voronoi partitioning of the workspace using the standard Euclidean distance.

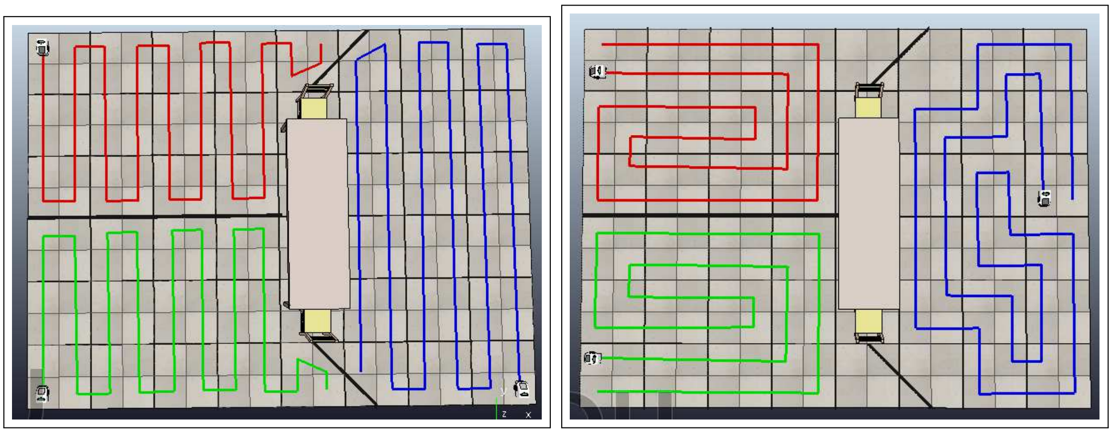
Figures 8.14(a) and (b) show coverage using single robot Boustrophedon-like and STC CPP algorithms, respectively, both using the Manhattan distance based Voronoi partitioning scheme. Both algorithms do not cover a (rectangular) portion of V_{M2}^{2D} (corresponding to the robot R2, on the right side), which is disconnected from the rest of cell where the robot is initially located. However, with the use of Manhattan distance, the coverage gap (in the case of STC) and coverage overlap (in the case of Boustrophedon-like coverage) is completely avoided.

Figures 8.15(a) and (b) show coverage using single robot Boustrophedon-like and STC CPP algorithms, respectively, both using the geodesic distance based Voronoi partitioning scheme. With geodesic partitioning each of the Voronoi cell is a contiguous region. However, it can be observed that the partition boundary cuts through the $2D \times 2D$ cells, which leads to coverage gap (Figure 8.15(a)) with the STC algorithm and coverage overlap (Figure 8.15(b))with Boustrophedon-like coverage algorithm.



(a) (b)

Figure 8.14: Robot coverage path using (a) Boustrophedon-like and (b) STC CPP algorithm with Voronoi partitioning of the workspace using the Manhattan distance.



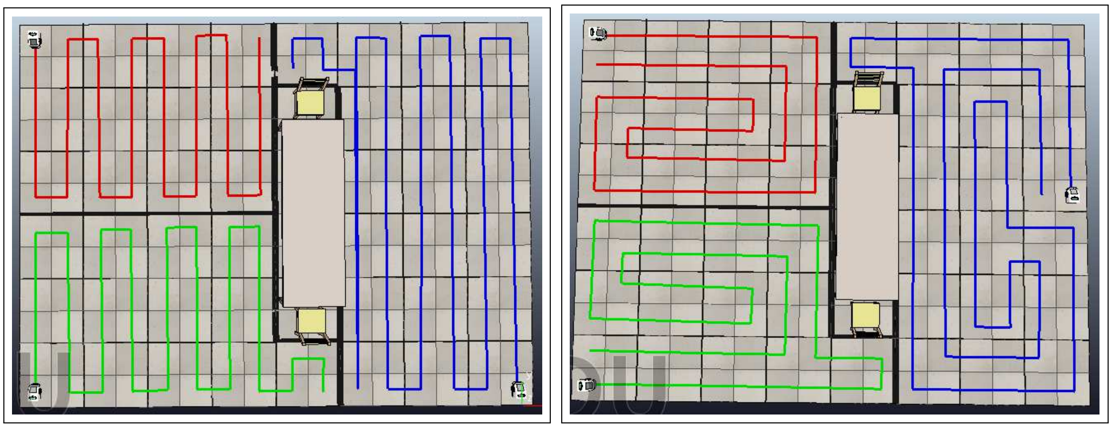
(a) (b)

Figure 8.15: Robot coverage path using (a) Boustrophedon-like and (b) STC CPP algorithm with Voronoi partitioning of the workspace using the geodesic distance.

Finally we show the simulation results with the proposed GM-VPC strategy in Figures 8.16(a) and (b), again using both Boustrophedon-like coverage and STC algorithms. As expected, with both the single robot algorithms, robot covers entire region completely and without any retrace/coverage overlap.

8.4.2 Experiments using Fire Bird V robots

Now we present results of experiments conducted using Fire Bird V robots. Fire Bird V is Atmega 2560 based robotic research platform designed by ERTS Lab, CSE, IIT Bombay and manufactured by Nex Robotics Pvt Ltd. The



(a) (b)

Figure 8.16: Robot coverage path using (a) Boustrophedon-like and (b) STC CPP algorithm with Voronoi partitioning of the workspace using the proposed Geodesic-Manhattan distance.

differential wheeled robot is equipped with 3 white line sensors, 5 Sharp GP2D12 IR range sensor, 8 analog IR proximity sensors, 8 analog directional light intensity sensors, and 2 position encoders. The robot has wireless ZigBee communication, USB communication, Wired RS232 (serial) communication , and simplex infrared communication capabilities.The robot has following dimensions: Diameter: 16cm Height: 10cm Weight: 1300gms.

Localization is one of the fundamental requirement in any path planning algorithm or even execution of a planned path on a mobile robot. In lab environments localization may be achieved by the use of odometric sensors/dead reckoning, use of overhead cameras, or motion capture systems. Each of these methods has its own disadvantages. While the motion capture system are typically expensive the odometry and overhead camera systems are economical options. Dead reckoning method is prone to errors and accuracy also depends on the type of surface on which the robot moves. With both overhead camera and motion capture system an external computer is required and the localization problem is solved outside the robot itself. In this work, as the algorithm is based on $2D \times 2D$ grids and the robots move through the sub nodes (which form a $D \times D$ gridded environment), we printed $D \times D$ grids with solid black lines. The robot uses its line following senors to follow the grid (as the path is always along the grids). These guide lines serve as directional guides (replacing the orientational localization) and the grid point where two perpendicular guide line meet serve as relative positional guides (replacing the positional absolute localization). Thus, by using only printed grid lines as guides, robot can execute a planned path, that is move along the planned path in the gridded space. Figure 8.17 shows a photograph of a Fire Bird robot in a printed gridded environment.

Robots can use the onboard proximity sensors to detect obstacles and plan the path (using STC or Boustrophedon-like coverage algorithms) and move along the path (that is, grid lines) using the onboard line following sensors. However, in this section we provide the planned path to the robot and let it follow it using line following algorithm, as the propose of this experiment is only to demonstrate the proposed GM-VPC strategy rather the planning process using the single robot CPP algorithms.

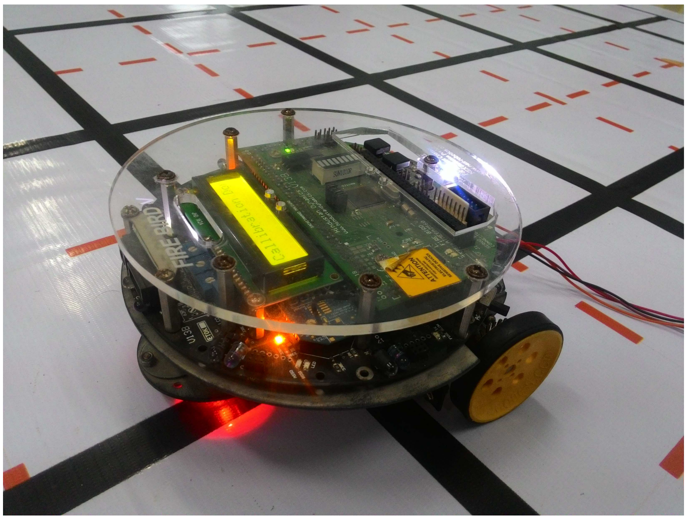


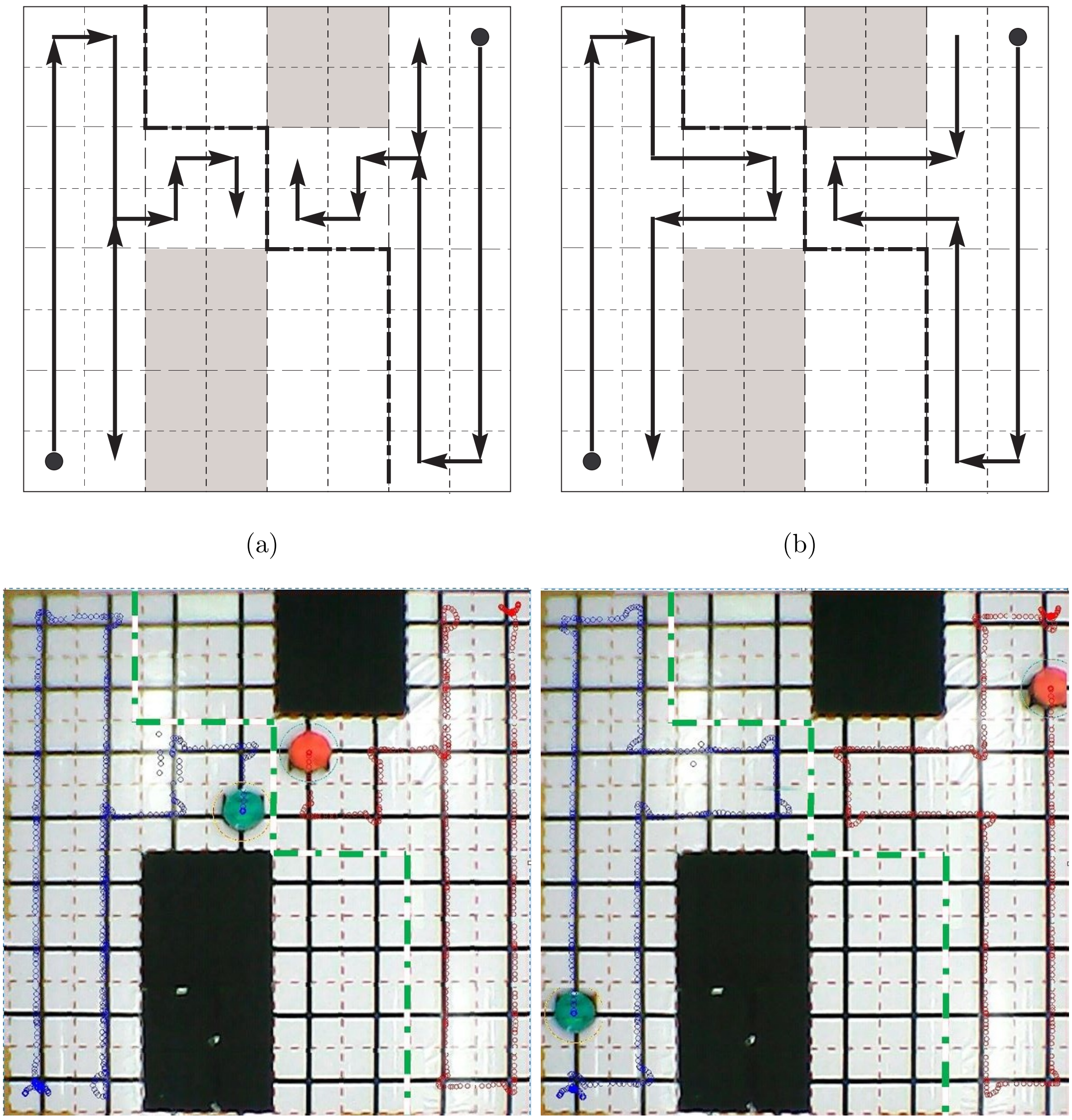
Figure 8.17: Photograph of a Fire Bird V robot in the gridded environment. Dashed (red) lines show the $D \times D$ sub cells, dark black gridded lines pass through the sub nodes along which the robot needs to move.

Figures 8.18 show the coverage using the Manhattan-VPC strategy. Figures 8.18(a) and (b) show the coverage path generated using Buoutrophedon-like coverage and STC algorithms and Figures 8.18(c) and (d) show the corresponding path followed by the robots. Long-short dashed line indicates the partition boundary. A few cells are not covered as these are unreachable to the respective robots. Also robot path (shown with double sided arrows in (a)) with Boustrophedon-like coverage leads to coverage overlap. Robot path is traced by small circles (shown with red and blue circles) in all the experiments.

Figures 8.19 show the coverage using the Geodesic-VPC strategy. Figures 8.19(a) and (b) show the coverage path generated using Buoutrophedon-like coverage and STC algorithms and Figures 8.19(c) and (d) show the corresponding path followed by the robots. With STC algorithms several cells through which the partition boundary (shown with long-short dashed line) pass through are left uncovered. Though Boustrophedon-like coverage algorithm covers these cells completely, they result in coverage overlap. As in the case of M-VPC, these coverage gap/overlap are induced by the partition boundary.

Figures 8.20 show the coverage using the Geodesic-VPC strategy. Figures 8.20(a) and (b) show the coverage path generated using Buoutrophedon-like coverage and STC algorithms and Figures 8.19(c) and (d) show the corresponding path followed by the robots. Unlike with the M-VPC or G-VPC strategies, GM-VPC strategy results in complete coverage. Boustrophedon-like path results in coverage overlap. However, this coverage overlap is not induced by the partition boundary, but due to the nature of the Boustrophedon-like coverage algorithm itself.

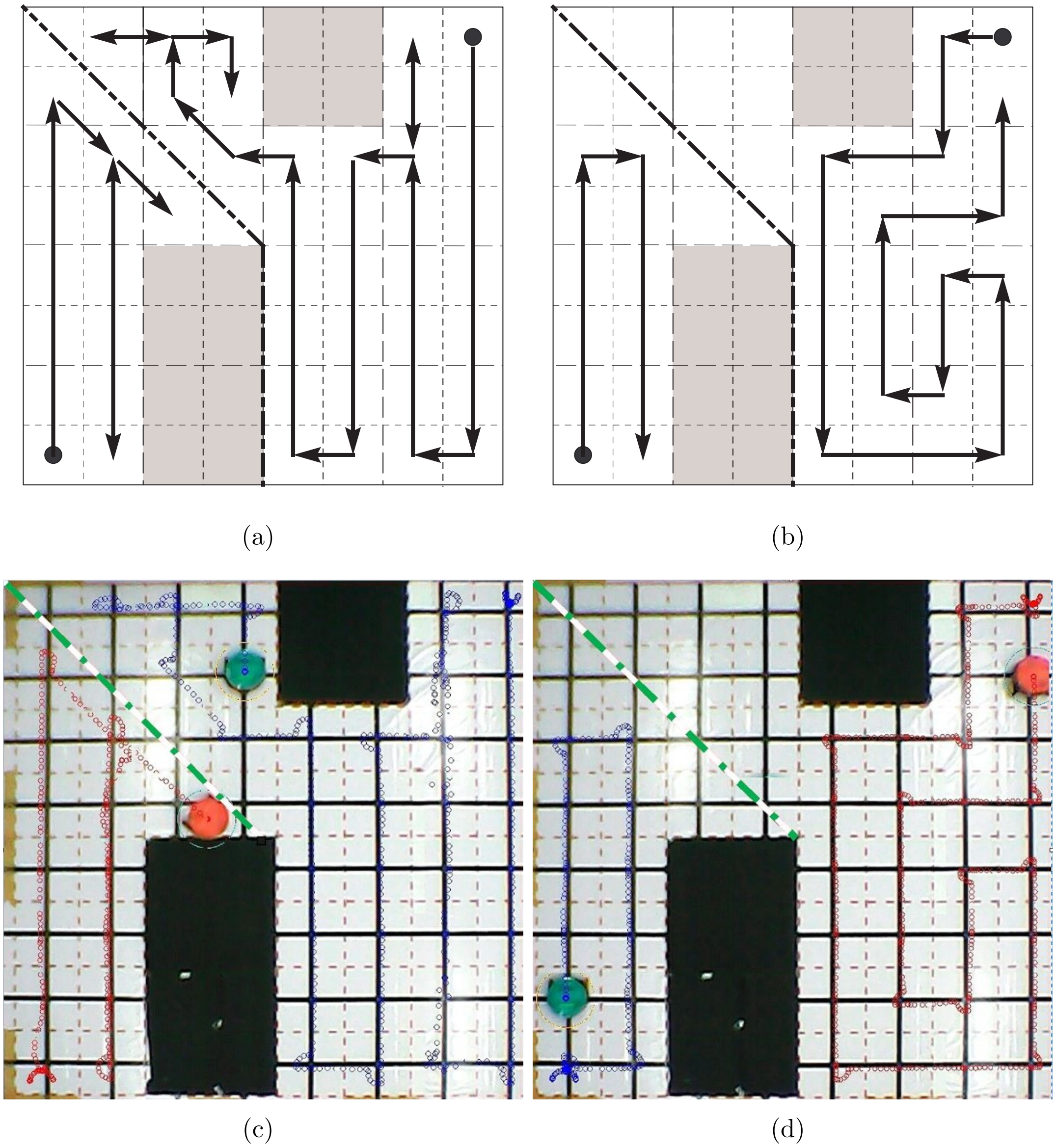
The results presented here endorse our claim that the proposed GM-VPC strategy provides complete and non-overlapping coverage of a coverage conducive region. Note that the incomplete coverage with 'resolution complete' algorithms such as STC, or coverage overlap with 'exact' algorithms such as Boustrophedon-like coverage occur only around the obstacle or partion/region boundaries. With the introduction of the geodesic-Manhattan distance for partitioning the region, partition boundaries no longer cause such problems. A complete coverage of a region can be achieved even when the region is not "coverage conducive", by using algorithms which attempt to provide complete coverage at the cost of unavoidable overlap such as Gabriely & Rimon (2003), Ranjitha & Guruprasad (2016, 2015*a*). Even with these single robot algorithms, the proposed GM-VPC



(a) (b)

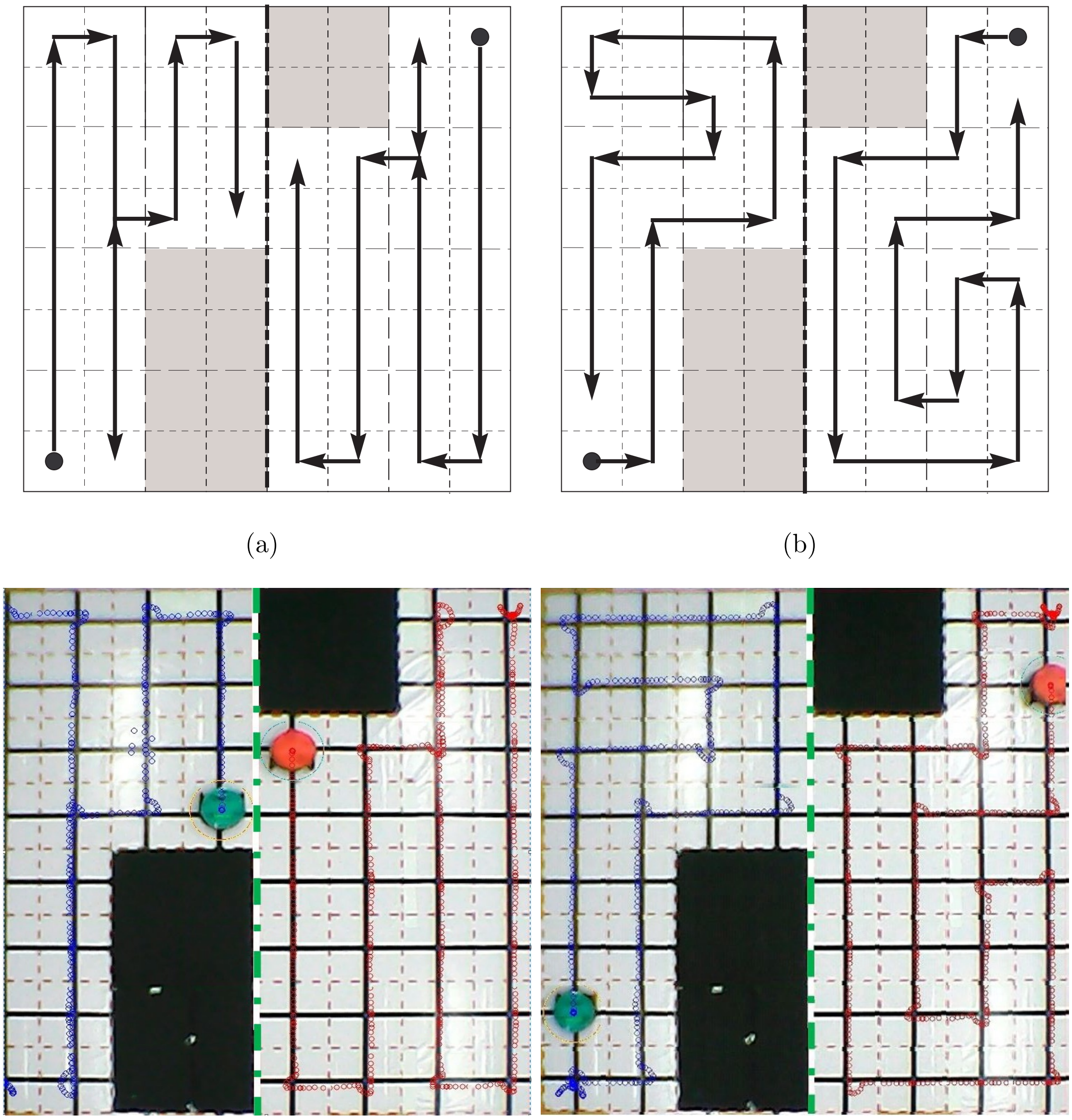
(c) (d)

Figure 8.18: Robot coverage with Manhattan-Voronoi partitioning. The generated path is shown in (a) using Boustrophedon-like coverage and (b) using STC algorithms. The corresponding robot path in the gridded environment are shown in (c) and (d). The small circles being traced are the robot paths and the robot is shown in last sub-cell at the end of coverage. Long-short dashed line indicates the partition boundary. A few cells are not covered as these are unreachable to the respective robots. Also robot path (shown with double sided arrows in (a)) with Boustrophedon-like coverage leads to coverage overlap.



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

Figure 8.19: Robot coverage with Geodesic-Voronoi partitioning. The generated path is shown in (a) using Boustrophedon-like coverage and (b) using STC algorithms. The corresponding robot path in the gridded environment are shown in (c) and (d). The small circles being traced are the robot paths and the robot is shown in last sub-cell at the end of coverage. Long-short dashed line indicates the partition boundary. While the STC coverage leaves all the cells through which the partition boundary passes uncovered, Boustrophedon-like coverage algorithm results in coverage overlap.



(a)

(b)

(c)

(d)

Figure 8.20: Robot coverage with the proposed GM-VPC strategy and (a) using Boustrophedon-like coverage and (b) STC algorithms in the gridded environment. Long-short dashed line indicates the partition boundary.

strategy completely eliminates coverage overlap around the partition boundary. However, even Boustrophedon decomposition scheme cannot completely avoid partition boundary induced coverage overlap. It only reduces it as the width of each cell with Boustrophedon decomposition is typically larger that with the trapezoidal decomposition. Partition induced coverage overlap (in Boustrophedon-like schemes) can be completely avoided only if the cell width is integral multiple of $2D$. This is exactly what GM-VPC scheme achieves.

8.5 SINGLE ROBOT SIMEX COVERAGE

In this section we provide results of simulation experiments to demonstrate the proposed single robot SimExCoverage algorithm using STC.

8.5.1 Graph-level simulation

First we provide result of a graph-level simulation. In graph-level simulation actual robot motion has not been considered. We generate spanning tree over the graph formed by the free major cells and the coverage path over the graph formed by the free minor cells.

We considered an arena which is decomposed into 16 major cells as shown in Figure 8.21. Obstacles split the arena into two topologically disconnected regions. If the robot starts from the cell as shown, it can not reach and hence cover the region marked 'unreachable'. The problem here is to cover the 'reachable' region from the starting cell completely without any overlap. Further, the algorithm should stop when the coverage is complete. This region is not known to the robot a priori, except for the boundary of the region.

Figures 8.22 and 8.23 show snapshots of the SimExCoverage process at each exploration step. The first exploration from the starting (minor) cell provides a partial map of the environment as shown in Figure 8.22 (a). The ST created through explored 'free' major cells is also shown in the figure. The algorithm creates CP through minor cells on the right side of the ST edges. When the robot reaches a minor cell in explored frontier cell, which has a neighboring unexplored

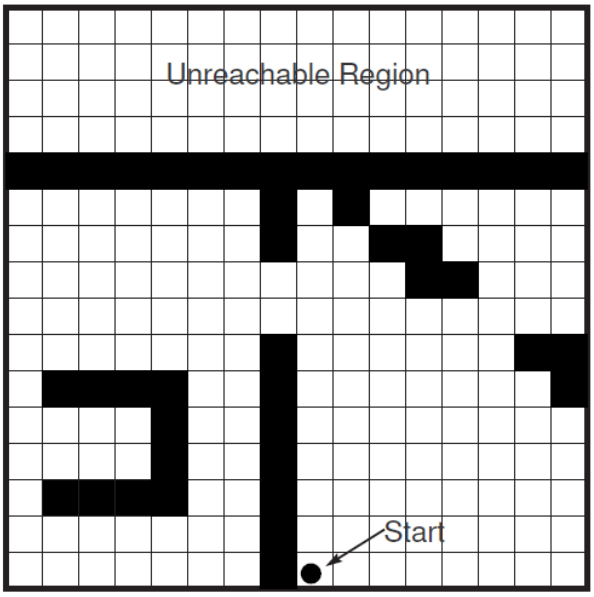
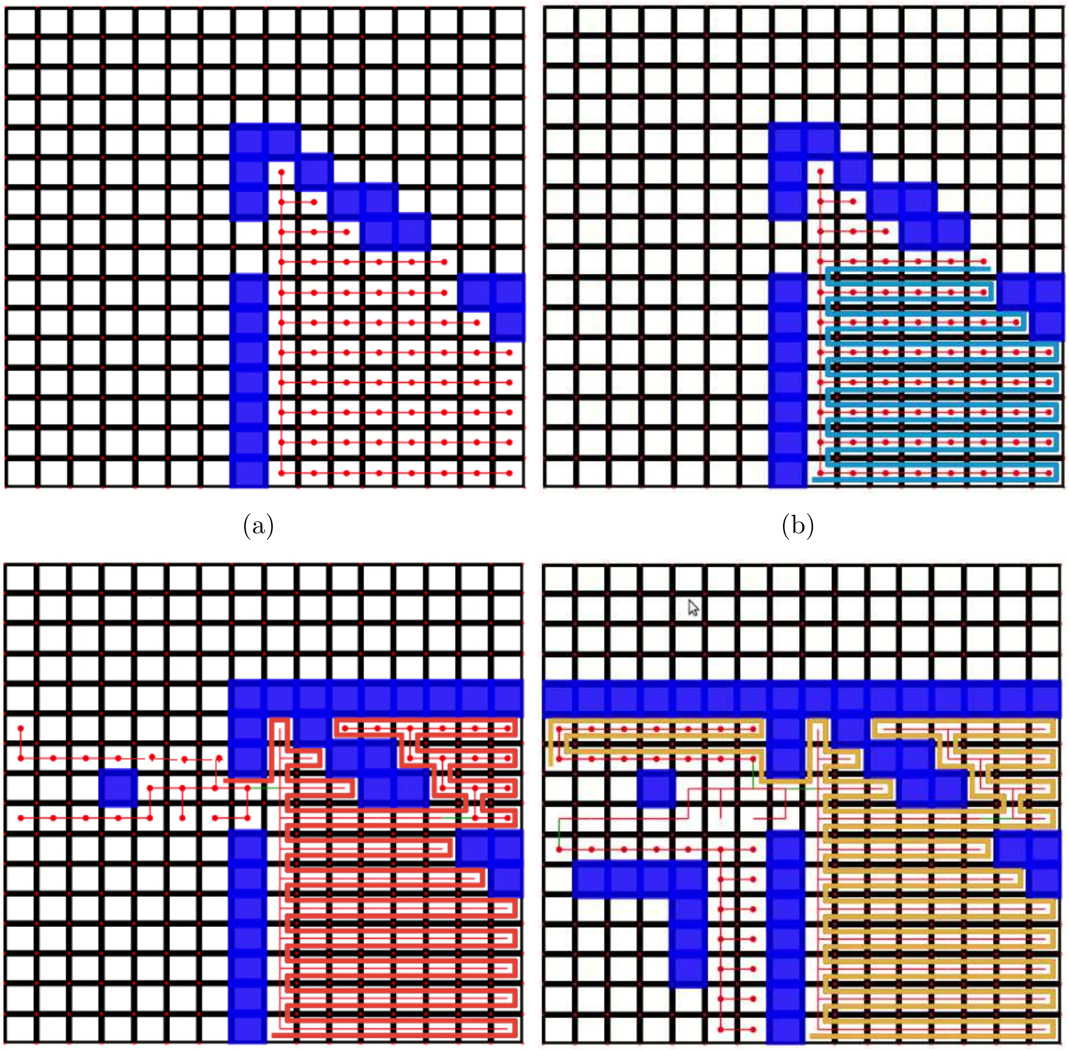


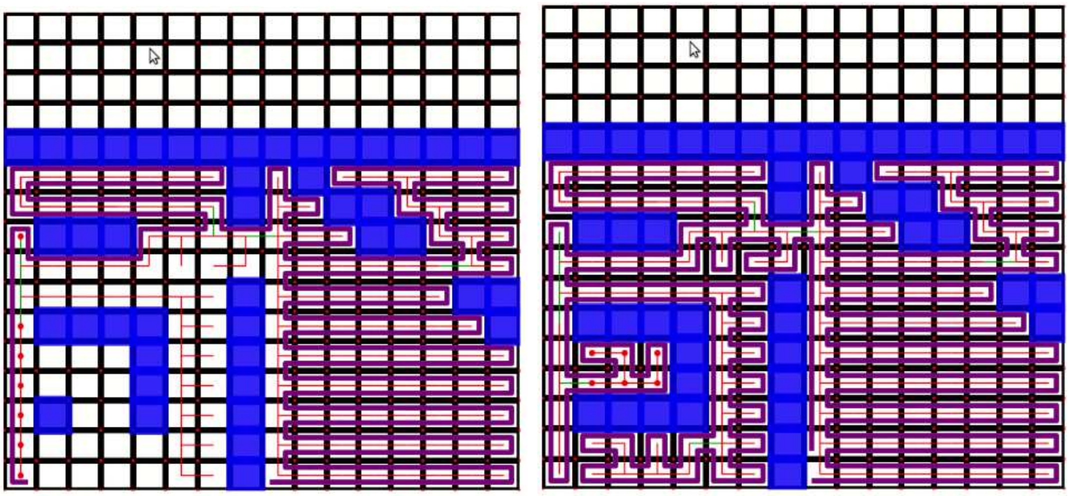
Figure 8.21: An arena to be covered is divided into 16×16 major cells. Occupied cells are shown black. Note that the region marked 'unreachable' can not be reached by the robot from its starting location as shown. The problem of SimExCoverage is to explore and cover the reachable region.

frontier cell as shown in Figure 8.22 (b), the second exploration is performed. Note that the robot has not finished covering the region explored by the first exploration instance, over which ST has been created. After creating ST in the newly explored region and merging it with already created ST, robot generates CP and starts moving along the generated CP. The third instance of exploration is performed as shown in Figure 8.22 (c), when the robot reaches an 'explored' frontier cell adjacent to an 'unexplored' frontier cell. At this instance, the robot has completed covering the region explored in the second instance, but the region explored at the first exploration instance is not yet covered completely. The appended ST through newly explored area is also shown in the figure. The process of exploration, creation of ST, and coverage continue as shown in Figures 8.22 (d) and Figure 8.23 (a) until the coverage is complete as shown in Figure 8.23 (b). Note that robot comes back to the region it explored first after covering all other newly explored region, and continues covering it. The termination condition (the next CP leading to the starting minor cell) ensures that the algorithm stops when the coverage is complete. As observed from the CP shown in Figure 8.23 (b), the proposed



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

Figure 8.22: SimExCoverage of a square region with 256 grids (major cells) shown in Figure 8.21 (a) ST created after the first exploration. (b) Robot performs second exploration while covering already explored region (c) ST created after 3rd exploration (d) 4th exploration and corresponding ST and CP. Blue (grey) shaded cells are explored occupied cells. Thin lines with major nodes (shown with thick dot) are used to show the spanning tree and the robot path through sub cells are shown on either side of the spanning tree edges.



(a) (b)

Figure 8.23: SimExCoverage of a square region with 256 grids - continuation of Figure 8.22: (a) Scenario after 5th exploration. b) Complete non-repetitive coverage achieved after the last exploration. Thin lines with major nodes (shown with thick dot) are used to show the spanning tree and the robot path through sub cells are shown on either side of the spanning tree edges.

SimExCoverage-STC algorithm provides complete and non-repetitive coverage. Further, the complete map (in terms of occupied and free cells) is also obtained as a byproduct.

8.5.2 Simulation with V-rep/Matlab

The proposed SimExCoverage-STC algorithm was simulated in Vrep simulation environment. The various stages of the implementation is given below.Figure 8.24 shows a scene created in V-Rep simulator for simulating SimExCoverage.The exploration sensor range is also shown.Figure 8.25 shows the snapshots of various stages of robot coverage using SimExCoverage with STC in V-Rep simulator. The exploration sensor is on only during the exploration phase when the robot reaches a frontier cell.Final scenario is shown in 8.26.

8.5.3 Simulation using a Turtlebot in ROS/Gazebo

The proposed SimExCoverage-STC has been implemented on a Turtlebot mobile robot platform using LIDAR sensors, within ROS-Gazebo environment.

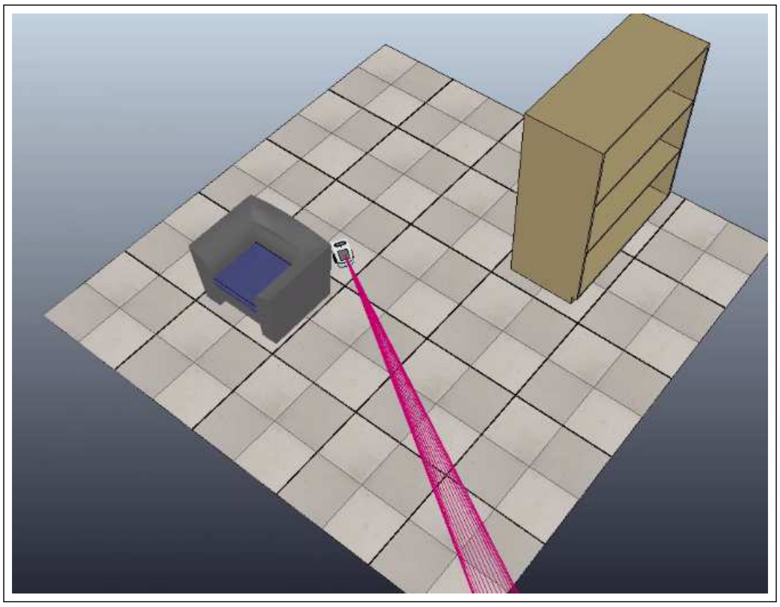


Figure 8.24: The scene created in Vrep simulator for simulating SimEx Coverage.The exploration sensor is also shown.
.

A simple scenario as shown in Figure 8.27 is considered for the purpose of demonstration of SimExCoverage-STC on a Turtlebot.

Figure 8.27 shows the first exploration of the region. The laser scan is shown with blue region, which indicates the explored region. The shadows of obstacle, shown as white region is not explored with the current robot location. With this information, occupancy map of the major cells in the explored region is obtained.

Figure 8.28 shows actual robot path (not the graph-level planned path) at different time instances. The robot performed three explorations and completed the coverage. Note that the entire major cell which is partially occupied is considered to be 'occupied' and is not covered as this is the requirement of the STC CPP algorithm used here.

The program that implements the SimExCoverage on the Turtlebot in ROS-Gazebo environment can in principle be used on a physical Turtlebot. As we have used LIDAR as exploration sensor, the physical Turtlebot too should be equipped with LIDAR. Instead of LIDAR, vision-based mapping too can be used. The exploration/mapping part of the program needs to be modified to use onboard camera. As the STC algorithm provides only a resolution complete

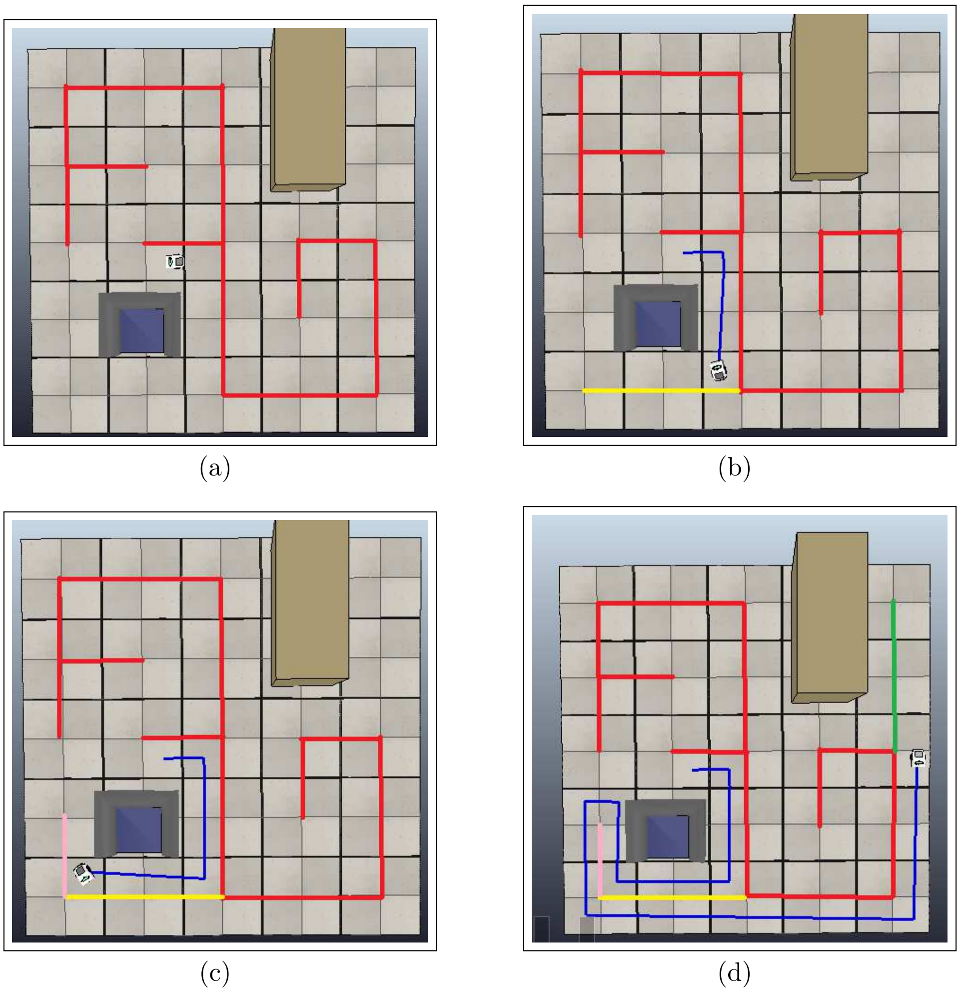


Figure 8.25: Snapshots of various stages of robot coverage using SimExCoverage with STC in Vrep simulator. The exploration sensor is on only during the exploration phase when the robot reaches a frontier cell.(a)The tree generated after first exploration.(b) The tree generated after second exploration (yellow).It is merged with the previous tree.(c)The tree generated after third exploration (shown in light pink color). (d) Final exploration and the corresponding tree(green).

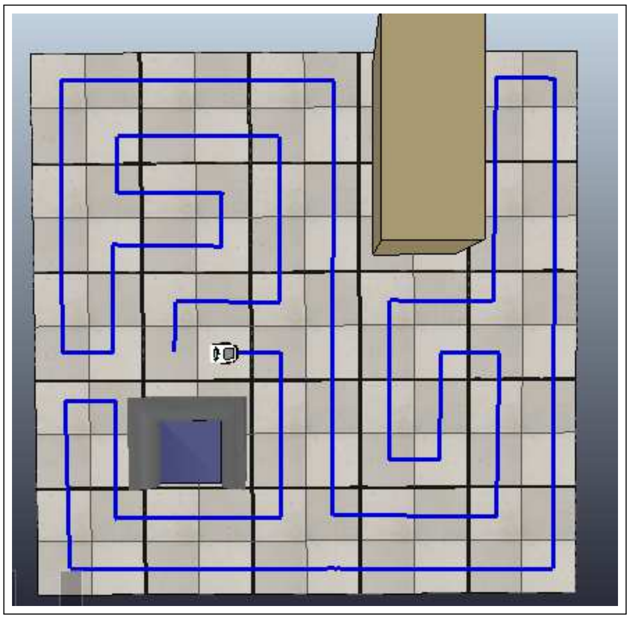


Figure 8.26: Final covered workspace. The blue (grey) thick line shows the robot path. Spanning tree edges are not shown for clarity.

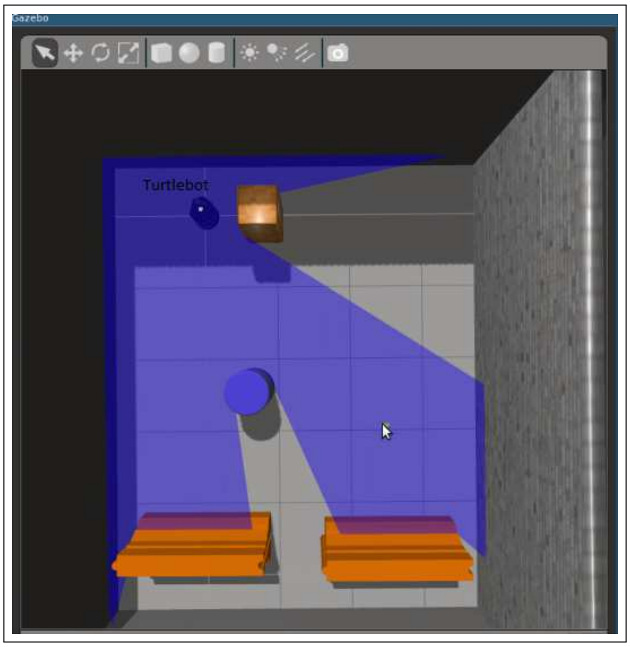
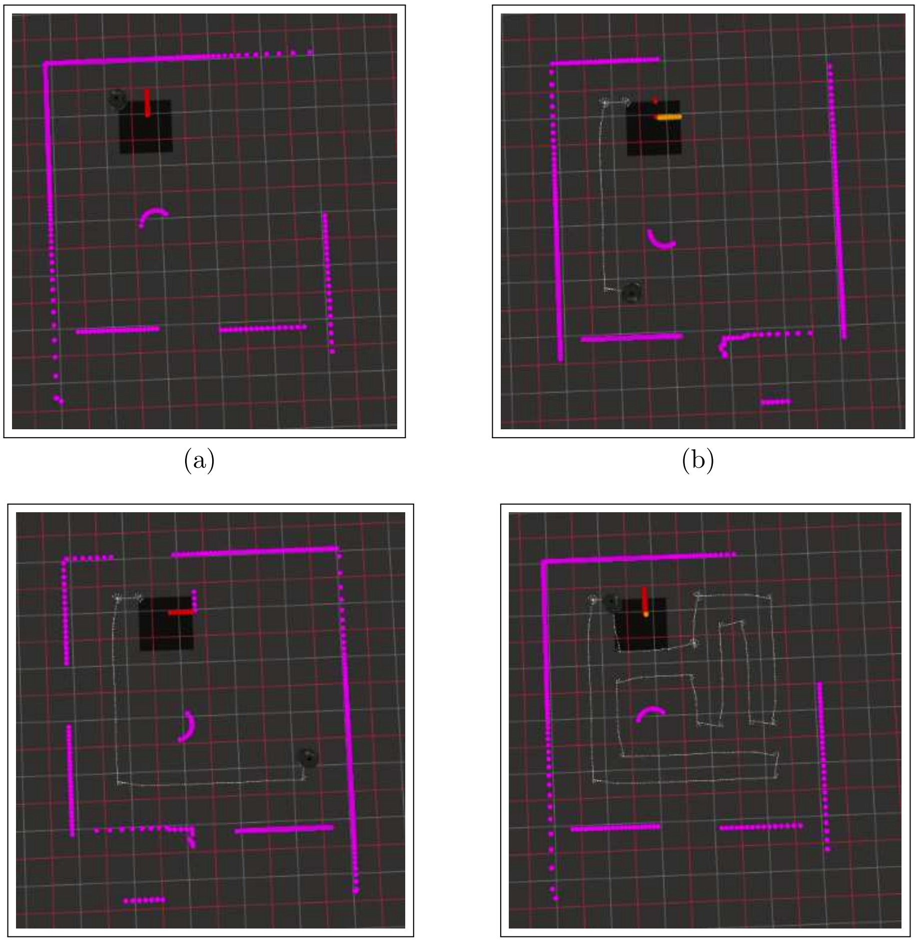


Figure 8.27: An environment within ROS-Gazebo containing obstacles. The robot is located at the top left corner as shown. Figure also shows the first exploration process. Blue (gray without color)s region shows the explored region while white region shows the shadows due to obstacles and hence unexplored region.



(a) (b)

(c) (d)

Figure 8.28: Snapshots of robot coverage using SimExCoverage. Robot successfully covers the free region using SimExCoverage algorithm using three exploration. Red grids show major cells, while gray/white grids show minor cells.

coverage, the SimExCoverage using STC can also guarantee only a resolution complete coverage. In the place of STC, STC based algorithms such as Gabriely & Rimon (2003), Ranjitha & Guruprasad (2015*a*, 2016), which attempt cover even partially occupied cells improving the coverage efficiency, at the cost of some coverage overlap may be used with minor modifications to the proposed SimExCoverage-STC algorithm. Any other CPP algorithm such as Boustrophedon algorithmChoset (2000) can also be used as underlying CPP algorithm with suitable modification to proposed STC based SimExCoverage algorithm.

8.6 MR-SIMEX COVERAGE

In this section we present results of simulation experiments carried out to demonstrate the proposed MR-SimExCoverage-STC algorithm solving the proposed MR-SimExCoverage problem. We have implemented the proposed MR-SimExCoverage-STC algorithm to generate the robot path at the graph level using Matlab. Here by graph-level path, we mean, that the planned path through the sub-nodes. These results are used primarily to illustrate the proposed MR-SimExCoverage problem and also the MR-SimExCoverage-STC algorithm.The path generated may be used for coverage by robots. We also provide results of a more realistic simulation carried out in V-rep/Matlab environment, to demonstrate the proposed MR-SimExCoverage problem and the proposed STC based algorithm.

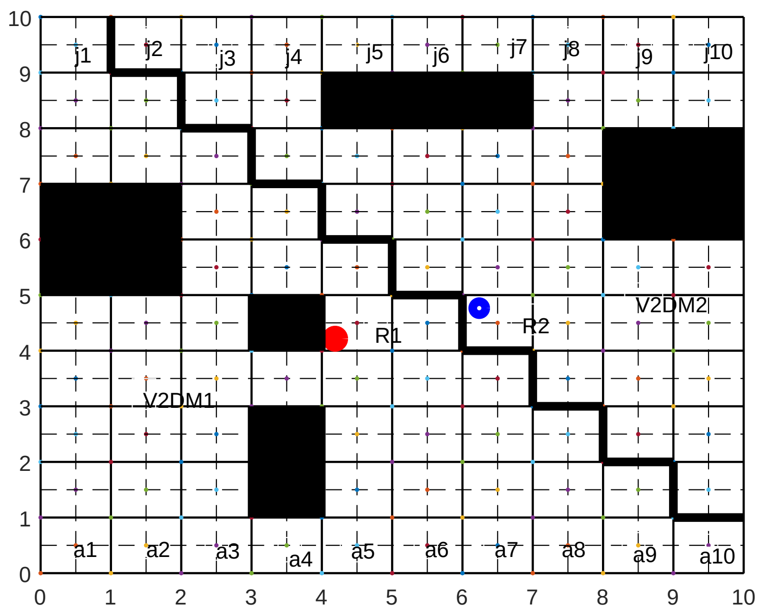
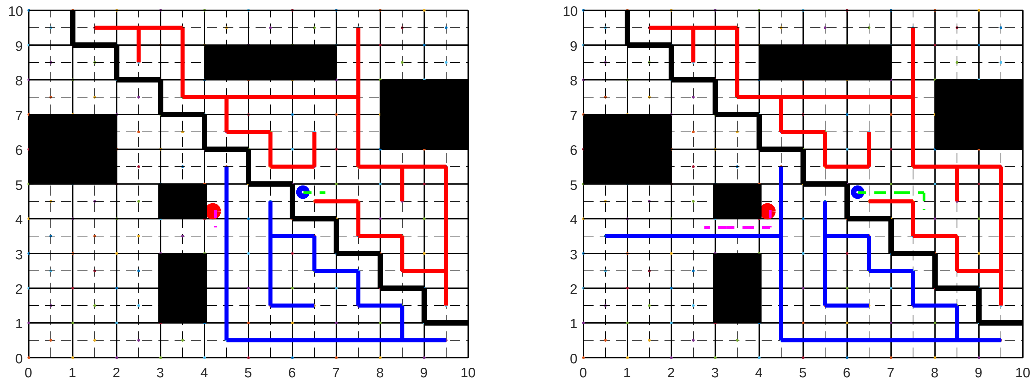


Figure 8.29: Initial scene with two robots R1 and R2, shown as discs (red and blue in color respectively) with Manhattan distance based Voronoi partitioned gridded workspace.The partition boundary is shown with black (dark) step like lines which divides the workspace into two Voronoi cells V_{2DM1} and V_{2DM2} respectively.Solid blue (grey in b/w) thin lines show $2D \times 2D$ gridding (major cells) and dashed blue (grey in b/w) thin lines show $D \times D$ grids (sub cells). Cell numbering scheme $a1, \ldots, a10, \ldots j1, \ldots, j10$ is also shown for the purpose of aiding the explanations.

First we present results of graph level simulation in Matlab and illustrate the proposed MR-SimExCoverage process using the MR-SimExCoverage-STC algorithm. The initial positions of the robots in the workspace, based on which a Manhattan distance based Voronoi partitioning is created, are given in Figure 8.29, along with the partitioning. We have numbered the cells for the purpose of aiding the description.Now we describe the working of the MR-SimExCoverage-STC in steps, using the snapshots of simulation results.



(a) (b)

Figure 8.30: Spanning tree generated after exploration phases shown with thick solid lines through the major nodes (center of major cells).(a)After first exploration phase (b) After second exploration.The red and green dotted lines through the sub nodes (center of $D \times D$ sub cells) represents the red and blue robot coverage paths respectively

The scenario after the first instance of exploration is shown in Figure 8.30(a). The robot R1 (shown as red disc in cell e5) explores the corresponding Manhattan Voronoi cell V_2DM1 from its initial location and generates spanning tree over the major nodes within the explored region. The spanning tree through the major nodes at the center of major cells is shown by thick (blue in color) lines. Similarly, the robot R2 (shown as blue disc in cell e7) performs first exploration within V_DM2 and constructs a spanning tree over the explored major nodes (shown in thick red lines). Note that the major cells to which no spanning tree edges are created correspond to unexplored region. Figures 8.30-8.33 shows various instances of exploration by either of the robots, the process of appending spanning tree edges and the CP generated. Both robots start moving on the CP generated on the

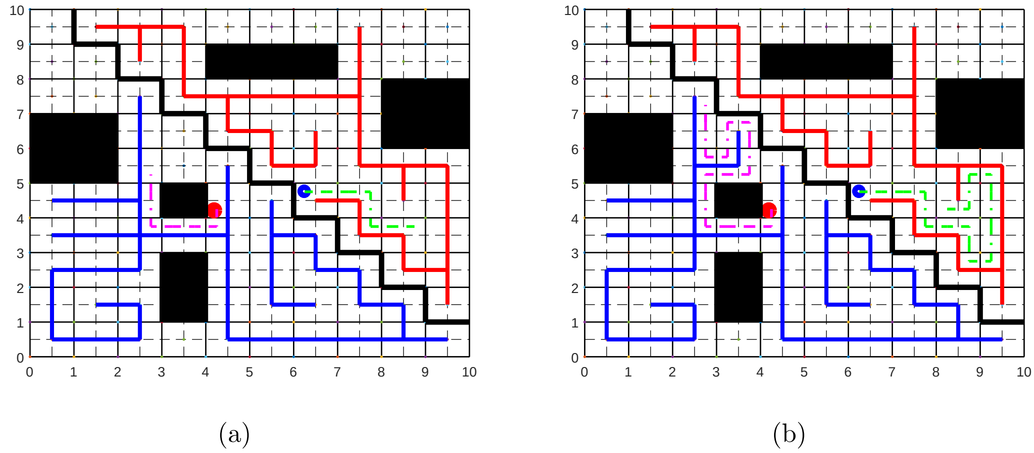
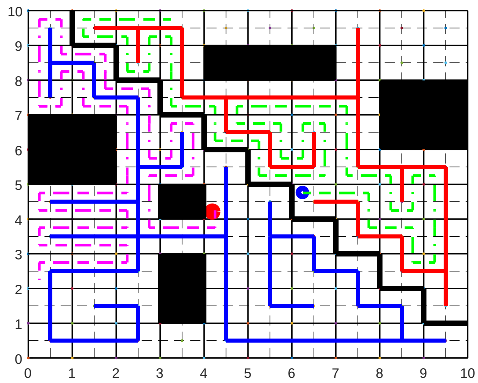


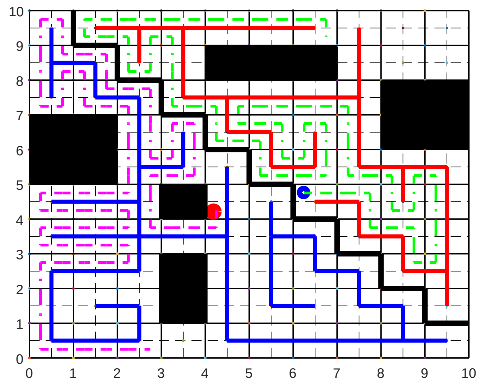
Figure 8.31: Spanning tree generated after exploration phases shown with thick solid lines through the major nodes (center of major cells).(a)After third exploration phase (b) After fourth exploration

right side of the spanning tree edges (shown by dashed lines) to cover the explored region. When robot R1 reaches cell $d5$, a frontier cell/exploration window, it performs exploration to add a new region into explored region within V_2DM1. The CP planner creates spanning tree over the major cells in the newly explored region, as shown in Figure 8.30(b). The existing tree and the new tree is connected at the cell d5 (which was a frontier cell after the first exploration and also an exploration window from where the second exploration is performed (by robot R1). Note that the combined graph is still a spanning tree within the total explored region.

The robot enters a frontier cell (exploration window) d3 while following the CP along the spanning tree edges, and performs third instance of exploration. Robot R1 performs a total of 6 instances of exploration at the cells $e5$ (Figure 8.30(a)), $d5$(Figure 8.30(b)), $d3$(Figure 8.31(a)), $f3$(Figure 8.31(b)), $h3$(Figure 8.32(a)), and $a3$(Figure 8.33(a)), to completely explore V_2DM1, which has 48 free major cells. Similarly, the robot R2 performs 3 exploration instances at cells $e7$(Figure 8.30(a)), $j4$(Figure 8.32(b)), and $i8$ (Figure 8.33(b)), to completely explore V_2DM2, which has 38 free major cells. Now as compared to obstacle detection using sensors at every free major cell, that is, 86 instances of sensing with online STC algorithm (only coverage), MR-SimExCoverage requires only 9 exploartion/sensing (6 by Robot R1 and 3 by robot R2) to cover the region. This

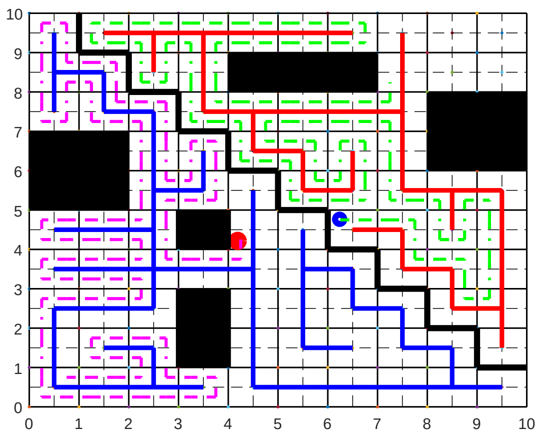


(a)

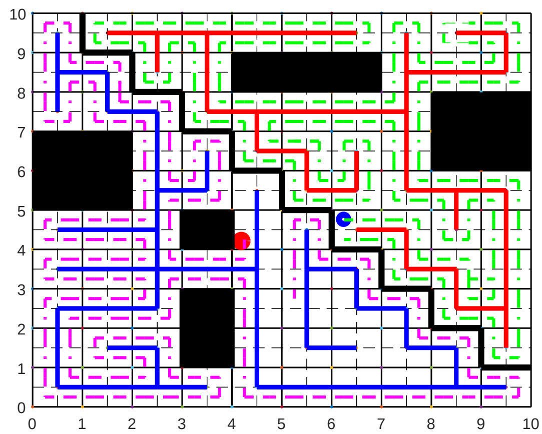


(b)

Figure 8.32: Spanning tree generated after exploration phases shown with thick solid lines through the major nodes (center of major cells).(a) After fifth exploration phase (b) After sixth exploration



(a)



(b)

Figure 8.33: Spanning tree generated after exploration phases shown with thick solid lines through the major nodes (center of major cells).(a)After seventh exploration phase (b) After eighth exploration

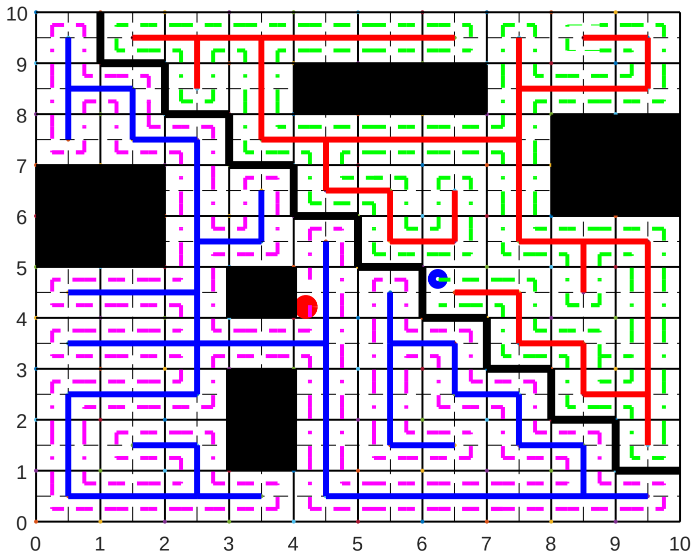


Figure 8.34: Final simultaneously explored and covered workspace.The blue and red lines represents the final spanning tree generated after all exploration phases by red and blue robots respectively. The dashed lines through the sub nodes (center of $D \times D$ sub cells) represents the corresponding CP.

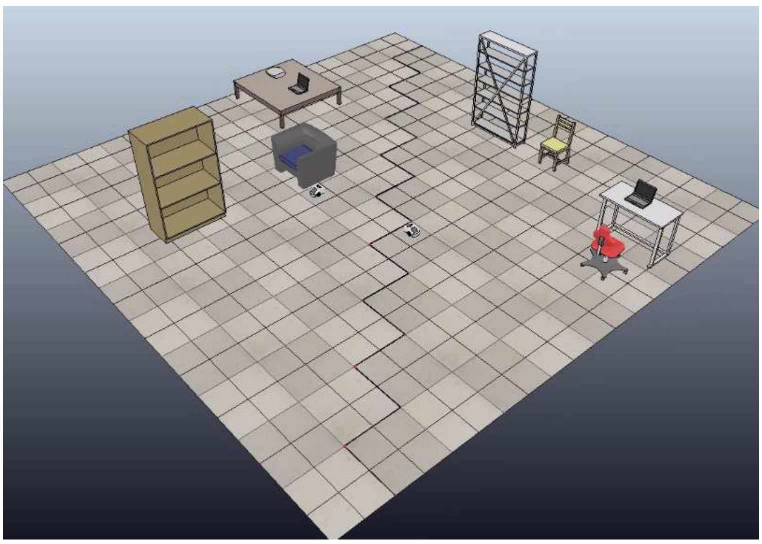


Figure 8.35: A scene generated in V-rep simulator with two robots. The major cells are shown with alternate grey and white cells along with four sub cells embedded in each of the major cells. Dark zig-zag lines show the boundary of Manhattan Voronoi cells.

amounts to the usage of only 10% of battery usage for sensors (a rough estimate assuming same sensor being used in both online STC and exploration process, and neglecting robot turning that may be required in the case of obstacle detection in online STC) with the proposed MR-SimExCoverage-STC compared to that being used when an online coverage (STC, to be specific) algorithm is used. A similar argument may be extended to any combination of coverage and exploration strategies used to solve the proposed MR-SimExCoverage problem, though exact numbers depend on the scene, and the details of the underlying algorithms.

Figure 8.34 shows the complete CP by both the robots. As expected, it can be observed that every sub cell corresponding to free major cell is covered exactly once by either of the robots. That is, the proposed MR-SimExCoverage-STC provided a complete non-overlapping coverage. Also, the map of the region in terms of free and occupied cells is available at the end of exploration with MRSimEx Coverage process.

Now we present results of more realistic simulation of MR-SimExCoverage using V-rep Rohmer et al. (2013) simulation environment. We used a robot model known as DR12 robot, a differential wheeled robot with an exploration sensor, available within the simulation environment to demonstrate the proposed MR-SimEx Coverage.Initial scene is shown in Figure 8.35. Figure 8.36 shows different stages of exploration and robot CP. It can be observed that the entire workspace, which is not known to the robots a priori, is explored and covered without any retrace or coverage gaps, demonstrating the theoretical claims of completeness and non-overlapping of coverage. We have used only two robots for clarity in presentation. The algorithm being distributed in nature, any number of robots can be used, and a similar performance is expected.

8.7 SUMMARY

The work showcases an attempt to combine exploration and mapping. It is to demonstrate that simultaneous exploration and coverage is possible with any single robot coverage algorithm. Also the exploration phase needed is very less

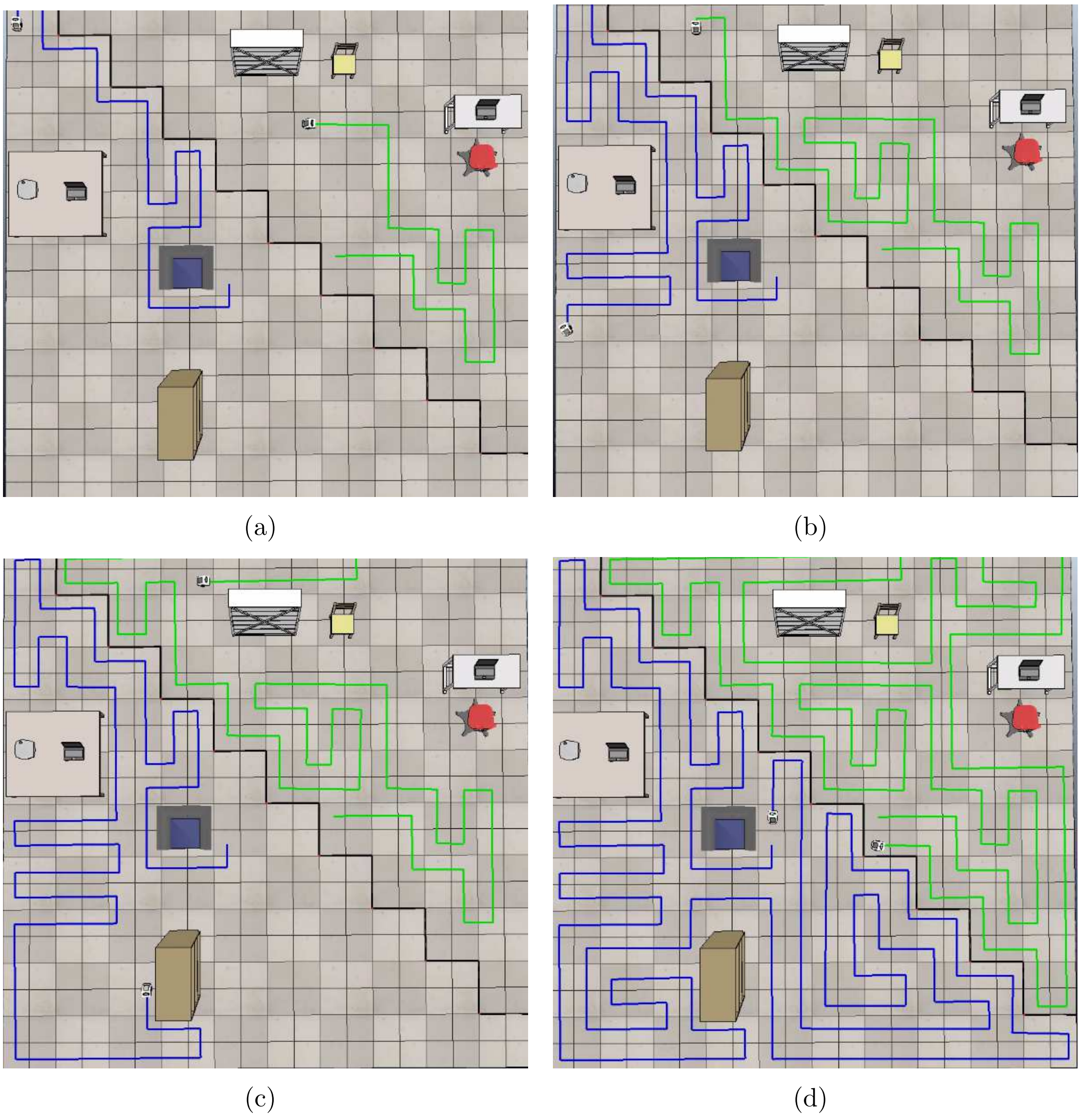


Figure 8.36: Snapshots of various stages of coverage with DR12 robot in V-rep simulation environment. Robot path (a) after the first instance of exploration, (b) after the fifth instance of exploration, (c) after the sixth instance of exploration, and (d) at the end of MR-SimExCoverage,the robots reaches the starting sub cell. Only sub cells are shown and we do not show spanning tree edges for clarity. Dark (colored) lines passing through the center of sub cells shows the robot path.

as compared to classical online methods. So the battery power needed will be definitely low. The path length depends on the type of coverage algorithm used. Any algorithms can be used and it does not change the essence of online-offline coverage methodology proposed in this chapter.

CHAPTER 9

CONCLUSION AND SCOPE FOR FUTURE WORK

In this chapter we summarize the contribution of the thesis and discuss scope for possible future work.

9.1 CONCLUSIONS

In this thesis we addressed a problem of coverage path planning for multiple cooperative autonomous mobile robots.

We considered a "partition and cover" approach to the multi-robotic coverage problem dues to its inherent advantages of i) independent of the underlying single robot coverage algorithm, ii) reduced memory requirement due to spatial task partitioning, iii) minimal or no communication requirement during performance of the coverage task, and iv) no requirement of special collision avoidance again due the spatial task partitioning. Among the "partition and cover" approaches reported in the literature, we used Voronoi partition based coverage due to its main advantage of possible distributed implementation.

One of the challenges associated with a multi-robot coverage problem is uniform load distribution among the robots. In the context of a "partition and cover" strategy employed in this thesis, this problem boils down to uniform partitioning assuming that the coverage load is proportional to the are of the coverage. This is a classical problem of equatable partitioning that is addresses in locational optimization or sensor coverage problems. In this work, we provided a very simple solution to this problem by using the concept of the centroidal Voronoi configuration used in the locational optimization/sensor coverage literature. We introduced the concept of deploying "virtual nodes" rather than the robots and partitioning the space based on the 'virtual nodes' locations. With this, we avoid unnecessary robot motion (in the sense that motion without performing coverage). We demonstrated with examples that with this approach, the areas of all the cells are approximately same, thus ensuring a uniform coverage load distribution among the individual robots.

We proposed Manhattan-VPC, a Manhattan distance based Voronoi Partition coverage algorithm that decomposes a $2D \times 2D$ gridded region completely avoiding partition boundary issues such as coverage gap and coverage overlap, that arise with the use of the standard Voronoi partition. Here, the robot footprint is assumed to be $D \times D$ square. We have established both by formal analysis and simulation and experiments with physical robots, that the proposed Manhattan-VPC provides complete and non-overlapping coverage even in the presence of simple obstacles and completely avoids the partition boundary induced coverage gap and overlap.

We also proposed Geodesic-VPC, a Voronoi partition based coverage algorithm using the Geodesic distance in the place of the standard Euclidean distance. With this approach we ensure that the cells that individual robots have to cover are contiguous even in the presence of arbitrary obstacles. However, here, unlike in the case of Manhattan VPC (or the basic VPC), we assume that the map of the environment is available *a priori* to the planner.

We then combined the Manhattan metric over the $2D \times 2D$ grid and Geodesic metric and propose a GM-VPC algorithm. We establish both by formal analysis and simulation experiments that with the GM-VPC algorithm robots provide complete and non-overlapping coverage in the presence of arbitrary known obstacles.

Finally we combined exploration and coverage problems to address a novel SimExCoverage problem. Here, the primary task of the robots is coverage while it uses intermittent exploration to generate partial map that is used by coverage path planner. This approach combines the advantages of both the off-line and online coverage strategies. We first present a single robot SimExCoverage problem and then extend it to a multi-robotic scenario. While the Manhattan-VPC and SimExCoverage algorithms are suitable for scenarios when map of the area is not available, the Geodesic-VPC and GM-VPC strategies are useful when map of the region is available.

We used a Boustrophedon-like coverage algorithm and the spanning tree based coverage algorithm which represent the approximate cellular decomposition based coverage algorithms and the exact cellular decomposition based coverage algorithms reported in the literature as underlying single-robot coverage algorithms for demonstrating the proposed generalized Voronoi partition based coverage strategies and the SimExCoverage algorithms.

9.2 SCOPE FOR FUTURE WORK

The experiments we presented with physical robots in this work are only for demonstrative purpose. It is expected that complete physical implementations throw new theoretical challenges and give rise to more interesting research problems. For example certain assumptions such as availability of localization, perfectness of the sensors, etc., will be challenged and new strategies may be required to handle imperfect localization and sensor.

Further, each sub problem such as gridding, obstacle detection, partitioning, executing the planned motion, etc. for a physical robot with given constraints such as limit on the sensor range, sensor/localization noise, limit on battery, type of drive used, etc. give rise to academically interesting and practically very useful research problems.

Another very useful work is to combine the GM-VPC, which is an off-line strategy, Manhattan-VPC, and MR-SimExCoverage, which are online in nature to handle partial information. While in Manhattan-VPC, partitioning is easier, in GM-VPC partitioning requires complete map. If no obstacle is assumed, when map is not available, GM-VP and Manhattan-VP are identical. As and when the obstacles are encountered, the GM-Voronoi cells may be recomputed. Such a scenario leads to several new challenges such as a robot which was originally assigned a task of covering a Voronoi cell, now changes. A part of its old cell now may belong to a different Voronoi cell, and a new part may be added to its cell.These are only a few of the numerous possibilities for extending the coverage strategies presented in this thesis.

Bibliography

Acar, E. U. & Choset, H. (2000), "Critical point sensing in unknown environments.", *in* '*Proceedings, IEEE International Conference on Robotics and Automation.*', Vol. 4, pp. 3803–3810.

Acar, E. U. & Choset, H. (2001), "Robust sensor-based coverage of unstructured environments.", *in* '*Proceedings, IEEE/RSJ International Conference on Intelligent Robots and Systems.*', Vol. 1, pp. 61–68.

Acar, E. U. & Choset, H. (2002), "Exploiting critical points to reduce positioning error for sensor based navigation.", *in* '*Proceedings, IEEE International Conference on Robotics and Automation.*', Vol. 4, pp. 3831–3837.

Acar, E. U., Choset, H. & Lee, J. Y. (2006), "Sensor based coverage with extended range detectors.", *IEEE Transactions on Robotics.* **22**(1), 189–198.

Acar, E. U., Choset, H., Rizzi, A. A., Atkar, P. N. & Hull, D. (2002), "Morse decompositions for coverage tasks.", *The International Journal of Robotics Research.* **21**(4), 331–344.

Acar, E. U., Choset, H., Zhang, Y. & Schervish, M. (2003), "Path planning for robotic demining: Robust sensor based coverage of unstructured environments and probabilistic methods.", *International Journal of Robotics Research.* **22**(8), 441 – 466.

Agmon, N., Hazon, N. & Kaminka, G. A. (2009), "The giving tree: Constructing trees for efficient offline and online multi-robot coverage.", *Annals of Math and Artificial Intelligence.* **52**(2-4), 43–168.

Ahmadi, M. & Stone, P. (2006), "A multi robot system for continuous area sweeping tasks.", *in* '*Proceedings, IEEE International Conference on Robotics and Automation.*', pp. 1724–1729.

Albers, S., Kursawe, K. & Schuierer, S. (1999), "Exploring unknown environments with obstacles.", *in* '*Proceedings, Symposium on Discrete Algorithms.*'.

Arbelaez, P. A. & Cohen, L. D. (2003), "Generalized voronoi tessellations for vector valued image segmentation.".

Arkin, E. M., Fekete, S. P. & Mitchell, J. S. B. (1993), "the lawnmower problem.", *in* '*Proceedings, Canadian Conference on Computational Geometry*', pp. 461–466.

Arkin, E. M., Fekete, S. P. & Mitchell, S. B. J. (2000), "Approximation algorithms for lawn mowing and milling.", *Computational Geometry.* **17**(1), 25 – 50.

Aronov, B. (1989), "On the geodesic voronoi diagram of point sites in a simple polygon.", *Algorithmica.* **4**(1), 109–140.

Atkar, P. N., Conner, D. C., Greenfield, A., Choset, H. & Rizzi, A. A. (2009), "Hierarchical segmentation of piecewise pseudoextruded surfaces for uniform coverage.", *IEEE Transactions on Automation Science and Engineering.* **6**(1), 107–120.

Atkar, P. N., Greenfield, A., Conner, D. C., Choset, H. & Rizzi, A. A. (2005), "Uniform coverage of automotive surface patches.", *The International Journal of Robotics Research.* **24**(11), 883–898.

Aurenhammer, F. (1991), "Voronoi diagrams–a survey of a fundamental geometric data structure.", *ACM Comput. Survey.* **23**(3), 345–405.

Bash, B. A. & Desnoyers, P. J. (2007), "Exact distributed voronoi cell computation in sensor networks.", *in* '*Proceedings, Sixth IEEE/ACM Conference On Information Processing in Sensor Networks.*', pp. 236–243.

Batalin, M. A. & Sukhatme, G. S. (2006), "Spreading out: A local approach to multi-robot coverage.", *in* '*Distributed Autonouos Robotic Systems.*', pp. 373–382.

Batsaikhan, D., Janchiv, A. & Lee, S. (2013), *"Sensor Based Incremental Boustrophedon Decomposition for Coverage Path Planning of a Mobile Robot."*, *Springer Berlin Heidelberg.*, pp. 621–628.

Batsaikhan, D., Lee, S., Kim, D., Kim, J. H. & Chong, N. Y. (2013), “Scan matching online cell decomposition for coverage path planning in an unknown environment.”, *International Journal of Precision Engineering and Manufacturing.* **14**(9), 1551–1558.

Becker, A., Fekete, S. P., Kroller, A., Lee, S. K., McLurkin, J. & Schmidt, C. (2013), “Triangulating unknown environments using robot swarms.”, *in* ‘*Proceedings, 29th Annual ACM Symposium on Computational Geometry.*’, pp. 345–346.

Bhattacharya, S., Michael, N. & Kumar, V. (2013), “Distributed coverage and exploration in unknown non-convex environments.”, *in* ‘*Distributed Autonomous Robotic Systems.*’, Springer, pp. 61–75.

Bosse, M., NouraniVatani, N. & Roberts, J. (2007), “Coverage algorithms for an under-actuated car-like vehicle in an uncertain environment.”, *in* ‘*Proceedings, IEEE International Conference on Robotics and Automation.*’, pp. 698–703.

Breitenmoser, A., Schwager, M., Metzger, J. C., Siegwart, R. & Rus, D. (n.d.), “Voronoi coverage of non convex environments with a group of networked robots.”, *in* ‘*Proceedings, IEEE International Conference on Robotics and Automation.*’.

Butler, Z. J., Rizzi, A. A. & Hollis, R. L. (1999), “Contact sensor-based coverage of rectilinear environments.”, *in* ‘*Proceedings, IEEE International Symposium on Intelligent Control Intelligent Systems and Semiotics.*’, pp. 266–271.

Butler, Z. J., Rizzi, A. A. & Hollis, R. L. (2000), “Complete distributed coverage of rectilinear environments.”.

Choi, Y. H., Lee, T. K., Baek, S. H. & Oh, S. Y. (2009), “Online complete coverage path planning for mobile robots based on linked spiral paths using constrained inverse distance transform.”, *in* ‘*Proceedings, IEEE/RSJ International Conference on Intelligent Robots and Systems.*’, pp. 5788–5793.

Choset, H. (2000), "Coverage of known spaces: The boustrophedon cellular decomposition.", *Autonomous Robots.* **9**(3), 247–253.

Choset, H. (2001), "Coverage for robotics- A survey of recent results.", *Annals of mathematics and artificial intelligence.* **31**(1-4), 113–126.

Choset, H., Acar, E., Rizzi, A. A. & Luntz, J. (2000), "Exact cellular decompositions in terms of critical points of morse functions.", *in* '*Proceedings, IEEE International Conference on Robotics and Automation.*', Vol. 3, pp. 2270–2277 vol.3.

Choset, H. & Pignon, P. (1997), "Coverage path planning: The boustrophedon decomposition.", *in* '*Proceedings, International Conference on Field and Service Robotics.*'.

Cohen, M. W., Sirotin, I. & Rave, E. (2008), "Lawn mowing system for known areas.", *in* '*Proceedings, International Conference on Computational Intelligence for Modelling Control Automation.*', pp. 539–544.

Cortes, J., Martinez, S., Karatas, T. & Bullo, F. (2004), "Coverage control for mobile sensing networks.", *IEEE Transactions on Robotics and Automation.* **20**(2), 243–255.

Das, A., Diu, M., Mathew, N., Scharfenberger, C., Servos, J., Wong, A., Zelek, J. S., Clausi, D. A. & Waslander, S. L. (2014), "Mapping, planning, and sample detection strategies for autonomous exploration.", *Journal of Field Robotics.* **31**(1).

Dasgupta, P., Baca, J., Guruprasad, K. R., Melendez, A. M. & Jumadinova, J. (2015), "The COMRADE system for multirobot autonomous landmine detection in postconflict regions.", *Journal of Robotics.* .

Dirichlet, G. (1850), "ber die reduktion der positiven quadratischen formen mit drei unbestimmten ganzen zahlen.", *Journal fr die Reine und Angewandte Mathematik.* **40**, 209–227.

Doty, K. L. & Harrison, R. R. (1993), "Sweep strategies for a sensory-driven, behavior-based vacuum cleaning agent.", *in* '*AAAI Fall Symposium Series.*', pp. 1–6.

Du, Q., Faber, V. & Gunzburger, M. (1999), "centroidal voronoi tessellations: Applications and algorithms.", *SIAM Review.* **41**(4), 637–676.

Fazli, P., Davoodi, A. & Mackworth, A. K. (2013), "Multi robot repeated area coverage.", *Autonomous Robots.* **34**(4), 251–276.

Fazli, P., Davoodi, A., Pasquier, P. & Mackworth, A. K. (2010), "Complete and robust cooperative robot area coverage with limited range.", *in* '*Proceedings, IEEE/RSJ International Conference on Intelligent Robots and Systems.*', pp. 5577–5582.

Fekete, S. P., Kamphans, T., Kroller, A., Mitchell, J. & Schmidt, C. (n.d.), "Exploring and triangulating a region by a swarm of robots.", *in* '*Proceedings, APPROX, LNCS.*'.

Fontan, M. S. & Mataric, M. J. (1998), "Territorial multi-robot task division.", *IEEE Transactions on Robotics and Automation.* **14**(5), 815–822.

Gabriely, Y. & Rimon, E. (2001), "Spanning tree based coverage of continuous areas by a mobile robot.", *Annals of Math and Artificial Intelligence.* **31**, 77–98.

Gabriely, Y. & Rimon, E. (2003), "Competitive on-line coverage of grid environments by a mobile robot.", *Computational Geometry.* **24**(3), 197–224.

Gage, D. (1995), "Many robot MCM search systems.", *in* '*Proceedings, Autonomous Vehicles in Mine Countermeasures Symposium.*', pp. 55–63.

Galceran, E. & Carreras, M. (2013), "A survey on coverage path planning for robotics.", *Robotics and Autonomous Systems.* **61**(12), 1258–1276.

Garcia, E. & de Santos, P. G. (2004), "Mobile robot navigation with complete coverage of unstructured environments.", *Robotics and Autonomous Systems.* **46**(4), 195 – 204.

Gonzalez, E., Alarcon, M., Aristizabal, P. & Parra, C. (2003), "BSA: A coverage algorithm.", *in* '*Proceedings, IEEE/RSJ International Conference on Intelligent Robots and Systems.*', Vol. 2, pp. 1679–1684.

Gonzalez, E., Alvarez, O., Diaz, Y., Parra, C. & Bustacara, C. (2005), "BSA: A complete coverage algorithm.", *in* '*Proceedings, IEEE International Conference on Robotics and Automation.*', pp. 2040–2044.

Gonzalez, H. H. & Latombe, J. C. (1998), "Planning robot motions for range image acquisition and automatic 3d model construction.", *in* '*Proceedings, AAAI Fall Symposium.*'.

Guruprasad, K. & Dasgupta, P. (2012*a*), "Distributed spatial partitioning of an initially unknown region for a multi-robot coverage application.", *in* '*Autonomous Robots and Multirobot System workshop.*'.

Guruprasad, K. R. & Dasgupta, P. (2012*b*), "Distributed voronoi partitioning for multi-robot systems with limited range sensors.", *in* '*Proccedings, IEEE/RSJ International Conference on Intelligent Robots and Systems.*', pp. 3546–3552.

Guruprasad, K. R. & Ghose, D. (2011), "Automated multi agent search using centroidal Voronoi configuration.", *IEEE Transactions on Automation Science and Engineering.* **8**(2), 420–423.

Guruprasad, K. R. & Ghose, D. (2013), "Performance of a class of multi robot deploy and search strategies based on centroidal Voronoi configurations.", *International Journal of Systems Science.* **44**(4), 680–699.

Guruprasad, K. R., Wilson, Z. & Dasgupta, P. (2012), "Complete coverage of an initially unknown environment by multiple robots using voronoi partition.", *in* '*Proceedings, International Conference on Advances in Control and Optimization of Dynamical Systems.*'.

Hameed, I.A. (2014), "Intelligent coverage path planning for agricultural robots and autonomous machines on three-dimensional terrain.", *Journal of Intelligent and Robotic Systems.* **74**(3), 965–983.

Hazon, N. & Kaminka, G. A. (2005), "Redundancy efficiency and robustness in multi-robot coverage.", *in* '*Proceedings, IEEE International Conference on Robotics and Automation.*', pp. 735–741.

Hazon, N., Mieli, F. & Kaminka, G. A. (2006), "Towards robust on-line multi-robot coverage.", *in* '*Proceedings, IEEE International Conference on Robotics and Automation.*', pp. 1710–1715.

Healey, A. J., McMillan, S., Jenkins, D. & McGhee, R. B. (1995), "BUGS: basic UXO gathering system.", *in* '*Proceedings, Autonomous Vehicles in Mine Countermeasures Symposium.*', pp. 8–32.

Hert, S. & Lumelsky, V. (1998), "Polygon area decomposition for multiplerobot workspace division.", *International Journal of Computational Geometry and Applications.* **8**, 437–466.

Hert, S., Tiwari, S. & Lumelsky, V. (1996), "A terrain-covering algorithm for an auv.", *Autonomous Robots.* **3**, 91–119.

Hsu, P. M., Lin, C. L. & Yang, M. Y. (2014), "On the complete coverage path planning for mobile robots.", *Journal of Intelligent and Robotic Systems.* **74**(3), 945–963.

Hungerford, K., Dasgupta, P. & Guruprasad, K. R. (2016), "A repartitioning algorithm to guarantee complete, non overlapping planar coverage with multiple robots.", pp. 33–48.

Jager, M. & Nebel, B. (2002), "Dynamic decentralized area partitioning for cooperating cleaning robots.", *in* '*Proceedings, IEEE International Conference on Robotics and Automation.*', Vol. 4, pp. 3577–3582 vol.4.

Jimenez, P. A., Shirinzadeh, B., Nicholson, A. & Alici, G. (2007), "Optimal area covering using genetic algorithms.", *in* '*Proceedings, IEEE/ASME international conference on advanced intelligent mechatronics.*', pp. 1–5.

Jonathan, R. & Dirk, A. (2012), "Multi robot coverage to locate fixed targets using formation structures.", *CoRR* **1202.2261**.

Kabir, A. M., Kaipa, K. N., Marvel, J. & Gupta, S. K. (n.d.).

Kapoutsis, A., Chatzichristofis, S. & Kosmatopoulos, E. (2017), "DARP: Divide areas algorithm for optimal multi-robot coverage path planning.", *Journal of Intelligent and Robotic Systems.* **86**(3), 663–680.

Kong, C. S., Peng, N. A. & Rekleitis, I. (2006), "Distributed coverage with multi-robot system.", *in* '*Proceedings, IEEE International Conference on Robotics and Automation.*', pp. 2423–2429.

Kosmatopoulos, E. B. & Christodoulou, M. A. (1996), "Convergence properties of a class of learning vector quantization algorithms.", *IEEE Transactions on Image Processing.* **5**(2), 361–368.

Kurabayashi, D., Ota, J., Arai, T. & Yoshida, E. (1996), "Cooperative sweeping by multiple mobile robots.", *in* '*Proceedings, IEEE International Conference on Robotics and Automation.*', Vol. 2, pp. 1744–1749.

Latombe, J. C. (1991), *"Robot motion planning."*, *Kluwer academic publishers.*

Lee, D. & Recce, M. (1997), "Quantitative evaluation of the exploration strategies of a mobile robot.", *International Journal of Robotics Research.* **16**(4), 413–447.

Lee, S. K., Fekete, S. P. & McLurkin, J. (2014), "Geodesic topological voronoi tessellations in triangulated environments with multi-robot systems.", *in* '*Proceedings, IEEE International Conference on Robotics and Automation.*'.

Lee, T., Baek, S., Choi, Y. & Oh, S. (2011), "Smooth coverage path planning and control of mobile robots based on high-resolution grid map representation.", *Robotics and Autonomous Systems.* **59**(10), 801 – 812.

Ling, X. & Stentz, A. (2011), "An efficient algorithm for environmental coverage with multiple robots.", *in* '*Proceedings, IEEE International Conference on Robotics and Automation.*', pp. 4950–4955.

Lumelsky, V. J., Mukhopadhyay, S. & Sun, K. (1990*a*), "Dynamic path planning in sensor based terrain acquisition.", *IEEE Transactions on Robotics and Automation.* **6**(4), 462–472.

Lumelsky, V., Mukhopadhyay, S. & Sun, K. (1990*b*), "Dynamic path planning in sensor-based terrain acquisition.", *IEEE Transactions on Robotics Automation.* **6**(4), 462–472.

Mannadiar, R. & Rekleitis, I. (2010), "Optimal coverage of a known arbitrary environment.", *in* '*Proceedings, IEEE International Conference on Robotics and Automation.*', pp. 5525–5530.

Mao, Y., Dou, L., Chen, J., Fang, H., Zhang, H. & Cao, H. (2009), "Combined complete coverage path planning for autonomous mobile robot in indoor environment.", *in* '*Proceedings, seventh Asian Control Conference.*', pp. 1468–1473.

Maza, I. & Ollero, A. (2007), "Multiple uav cooperative searching operation using polygon area decomposition and efficient coverage algorithms.", *in* '*Proceedings, Distributed Autonomous Robotic Systems*', Springer Japan, Tokyo, pp. 221–230.

Michel, D. & McIsaac, K. (2012), "New path planning scheme for complete coverage of mapped areas by single and multiple robots.", *in* '*Proceedings, IEEE International Conference on Mechatronics and Automation.*', pp. 1233–1240.

Min, T. W. & Yin, H. K. (1998), "A decentralized approach for cooperative sweeping by multiple mobile robots.", *in* '*Proceedings,IEEE/RSJ International Conference on Intelligent Robots and Systems.*', Vol. 1, pp. 380–385.

Minhaj, F., Akshar, P., Guruprasad, K. R. & Vishnu, G. N. (Accepted), "SimExCoverage: Simultaneous exploration and coverage by a mobile a robot.", *in* '*Proceedings, XV Control Instrumentation System Conference-CISCON.*'.

Oh, J. S., Choi, Y. H., Park, J. B. & Zheng, Y. F. (2004), "Complete coverage navigation of cleaning robots using triangular cell based map.", *IEEE Transactions on Industrial Electronics.* **51**(3), 718–726.

Okabe, A., Boots, B., Sugihara, K. & Chiu, S. (2000), *"Spatial Tessellations. Concepts and Applications of Voronoi Diagrams."*.

Pimenta, L. C. A., Kumar, V., Mesquita, R. C. & Pereira, G. A. S. (2008), "Sensing and coverage for a network of heterogeneous robots.", *in* '*Poceedings, IEEE Conference on Decision and Control.*', pp. 3947–3952.

Preparata, F. P. & Shamos, M. I. (1985), *"Computational Geometry: An Introduction."*, *Springer-Verlag.*

Prithviraj, D., Melendez, A. M. & Guruprasad, K. R. (2012), "Multi-robot terrain coverage and task allocation for autonomous detection of landmines.", Vol. 8359, pp. 8359 – 8359 – 14.

Qiu, X., Song, J., Zhang, X. & Liu, S. (2006), "A complete coverage path planning method for mobile robot in uncertain environments.", *in* '*Proceedings, sixth World Congress on Intelligent Control and Automation.*', Vol. 2, pp. 8892–8896.

Ranjitha, T. D. & Guruprasad, K. R. (2015*a*), "STCTC: A spanning tree-based competitive and truly complete coverage algorithm for mobile robots.", *in* '*Proceedings, Advances in Robotics International Conference of Robotics Society of India.*'.

Ranjitha, T. D. & Guruprasad, K. R. (2015*b*), "Truly complete competitive robot coverage path planning using approximate cellular decomposition.", *in* '*Proceedings, International Conference of Robotics Society of India Advances in Robotics.*'.

Ranjitha, T. D. & Guruprasad, K. R. (2016), "Pseudo spanning tree-based complete and competitive robot coverage using virtual nodes.", *in* '*IFACPapersOnLine: Proc of 4th IFAC Conference on Advances in Control and Optimization of Dynamical Systems.*'.

Rekleitis, I., New, A. P., Rankin, E. S. & Choset, H. (2008), “Efficient boustrophedon multi-robot coverage: an algorithmic approach.”, *Annals of Mathematics and Artificial Intelligence.* **52**, 109–142.

Rekleitis, I., New, A. & Rankin, E. (2008), “Efficient boustrophedon multi robot coverage: an algorithmic approach.”, *Annals of Mathematics and Artificial Intelligence.* **52**.

Rohmer, E., Singh, S. P. N. & Freese, M. (2013), “V-REP: A versatile and scalable robot simulation framework.”, *in* ‘*Proceedings,IEEE/RSJ International Conference on Intelligent Robots and Systems.*’, pp. 1321–1326.

Samuel, R., Nikolaus, C. & Alcherio, M. (2009), “Collaborative coverage using a swarm of networked miniature robots.”, *Robotics and Autonomous Systems.* **57**(5), 517 – 525.

Senthilkumar, K. & Bharadwaj, K. (2010), “Multi robot terrain coverage by constructing multiple spanning trees simultaneously.”, *International Journal of Robotics and Automation.* **25**.

Senthilkumar, K. S. & Bharadwaj, K. K. (2012), “Multi robot exploration and terrain coverage in an unknown environment.”, *Robotics and Autonomous Systems.* **60**, 123–132.

Shivashankar, V., Jain, R., Kuter, U. & Nau, D. (2011), “Real time planning for covering an initially unknown spatial environment.”, *in* ‘*Proceedings, International Florida Artificial Intelligence Research Society Conference.*’, pp. 63–68.

Shnaps, I. & Rimon, E. (2016), “Online coverage of planar environments by a battery powered autonomous mobile robot.”, *IEEE Transactions on Automation Science and Engineering.* **13**(2), 425–436.

Song, J. & Gupta, S. (2018), “$\varepsilon^\star$: An online coverage path planning algorithm.”, *IEEE Transactions on Robotics.* **34**(2), 526–533.

Thanou, M., Stergiopoulos, Y. & Tzes, A. (2013), "Distributed coverage using geodesic metric for non-convex environments.", *in* '*Proceedings, IEEE International Conference on Robotics and Automation.*', pp. 933–938.

Viet, H. H., Choi, S. & Chung, T. (2014), "BoB: an online coverage approach for multi robot systems.", *Applied Intelligence.* **42**.

Viet, H. H., Dang, V.-H., Laskar, M. N. U. & Chung, T. (2013), "BA*: an online complete coverage algorithm for cleaning robots.", *Applied Intelligence.* **39**(2), 217–235.

Vishnu, G. N. & Guruprasad, K. R. (Accepted), "Manhattan distance based voronoi partitioning for efficient multi-robot coverage.", *in* '*Proceedings, XV Control Instrumentation System Conference-CISCON.*'.

Voronoi, G. (1908), "Nouvelles applications des paramtres continus la thorie des formes quadratiques deuxime mmoire recherches sur les parallllodres primitifs.", *Journal fr die reine und angewandte Mathematik.* **134**, 198–287.

Wagner, I. A., Lindenbaum, M. & Bruckstein, A. M. (1999), "Distributed covering by ant robots using evaporating traces.", *IEEE Transactions on Robotics and Automation.* **15**(5), 918–933.

Wilson, Z., Whipple, T. & Dasgupta, P. (2011), "Multi robot coverage with dynamic coverage information compression.", *in* '*Proceedings,International Conference on Informatics in Control Automation and Robotics.*', pp. 236–241.

Xu, A., Viriyasuthee, C. & Rekleitis, I. (2014), "Efficient complete coverage of a known arbitrary environment with applications to aerial operations.", *Autonomus Robots.* **36**(4), 365–381.

Yamauchi, B. (1998), "Frontier based exploration using multiple robots.", *in* '*Proceedings, Second International Conference on Autonomous Agents.*', pp. 47–53.

Yang, S. X. & Luo, C. (2004), "A neural network approach to complete coverage path planning.", *IEEE Transactions on Systems, Man, and Cybernetics.* **34**(1), 718–724.

Yazici, A., Kirlik, G., Parlaktuna, O. & Sipahioglu, A. (2014), "A dynamic path planning approach for multirobot sensor-based coverage considering energy constraints.", *IEEE Transactions on Cybernetics.* **44**(3), 305–314.

Zelinsky, A., Jarvis, R. A., Byrne, J. C. & Yuta, S. (1993), "Planning paths of complete coverage of an unstructured environment by a mobile robot.", *in* '*Proceedings, International Conference on Advanced Robotics.*', pp. 533–538.

Zheng, X., Jain, S., Koenig, S. & Kempe, D. (2005), "Multi robot forest coverage.", *in* '*Proceedings, IEEE/RSJ InternationalConference on Intelligent Robots and Systems.*', pp. 3852–3857.

Zheng, X. & Koenig, S. (2007), "Robot coverage of terrain with non uniform traversability.", *in* '*Proceedings, IEEE/RSJ InternationalConference on Intelligent Robots and Systems.*', pp. 3757–3764.