

Utah State University

DigitalCommons@USU

All Graduate Theses and Dissertations, Fall
2023 to Present

Graduate Studies

5-2024

Decentralized Unknown Building Exploration by Frontier Incentivization and Voronoi Segmentation in a Communication Restricted Domain

Huzeyfe M. Kocabas

Utah State University, huzeyfe.kocabas@usu.edu

Follow this and additional works at: <https://digitalcommons.usu.edu/etd2023>



Part of the [Computer Sciences Commons](#)

Recommended Citation

Kocabas, Huzeyfe M., "Decentralized Unknown Building Exploration by Frontier Incentivization and Voronoi Segmentation in a Communication Restricted Domain" (2024). *All Graduate Theses and Dissertations, Fall 2023 to Present*. 99.

<https://digitalcommons.usu.edu/etd2023/99>

This Dissertation is brought to you for free and open access by the Graduate Studies at DigitalCommons@USU. It has been accepted for inclusion in All Graduate Theses and Dissertations, Fall 2023 to Present by an authorized administrator of DigitalCommons@USU. For more information, please contact digitalcommons@usu.edu.



DECENTRALIZED UNKNOWN BUILDING EXPLORATION BY FRONTIER
INCENTIVIZATION AND VORONOI SEGMENTATION IN A
COMMUNICATION RESTRICTED DOMAIN

by

Huzeyfe M. Kocabas

A dissertation submitted in partial fulfillment
of the requirements for the degree

of

DOCTOR OF PHILOSOPHY

in

Computer Science

Approved:

Mario Harper, Ph.D.
Major Professor

Xiaojun Qi, Ph.D.
Committee Member

Curtis Dyreson, Ph.D.
Committee Member

John Edwards, Ph.D.
Committee Member

Jacob Gunther, Ph.D.
Committee Member

D. Richard Cutler, Ph.D.
Vice Provost of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2023

Copyright © Huzeyfe M. Kocabas 2023

All Rights Reserved

ABSTRACT

Decentralized Unknown Building Exploration By Frontier Incentivization And Voronoi
Segmentation in a Communication Restricted Domain

by

Huzeyfe M. Kocabas, Doctor of Philosophy

Utah State University, 2023

Major Professor: Mario Harper, Ph.D.
Department: Department of Computer Science

Multi-robot exploration in communication-restricted environments presents significant challenges posed by adaptability and complexity persists as unsolved puzzles. While making progress, existing approaches in the decentralized domain often fail due to falsely base-lined assumptions, such as overestimating the feasibility of information broadcasting among agents, assuming the presence of a central oracle for exploration termination, and overlooking the computational overheads associated with path planning algorithms.

This dissertation delves into the advantages of agent distribution within an area, significantly reducing exploration time. It first addresses the city roadway coverage problem using a single agent to minimize travel distance, then explores the integration of multiple agents to enhance efficiency. The proposed approach involves dividing the city into subareas, calculating postman routes, and optimizing these segments. Strategic selection of UAV vehicles further complements the methodology. Simulation results affirm the method's efficacy, demonstrating improved coverage time and enabling informed decision-making by city municipalities, accounting for cost-related considerations.

Extending the area division technique to unknown building exploration, the research evaluates various strategies tailored to building exploration. Comparison between frontier

and unknown target selection strategies is performed, considering factors like area allocation, agent distribution within buildings, robot count, potential agent loss, and environmental density. The introduced unknown goal selection strategy, driven by its greedy nature and superior agent spreading capabilities compared to frontier counterparts, excels in coordinating agents for unknown building exploration.

Additionally, the dissertation addresses the challenge of exploring unknown areas in communication-restricted domains, employing two core techniques: implicit incentivization of frontiers through an agent detection system and area segmentation via Voronoi partitioning. These methods enhance exploration performance by promoting independence and implicit coordination. Simulation results demonstrate the algorithms' effectiveness across diverse communication scenarios.

The proposed techniques have been validated with multiple simulation testing, establishing theoretical performance bounds. The results highlight a substantial performance boost in communication-restricted domains, achieved through our combined approach of Voronoi partitioning and implicit coordination, guiding agents efficiently within the environment.

(87 pages)

PUBLIC ABSTRACT

Decentralized Unknown Building Exploration By Frontier Incentivization And Voronoi
Segmentation in a Communication Restricted Domain

Huzeyfe M. Kocabas

Exploring unknown environments using multiple robots poses a complex challenge, particularly in situations where communication between robots is either impossible or limited. Existing exploration techniques exhibit research gaps due to unrealistic communication assumptions or the computational complexities associated with exploration strategies in unfamiliar domains. In our investigation of multi-robot exploration in unknown areas, we employed various exploration and coordination techniques, evaluating their performance in terms of robustness and efficiency across different levels of environmental complexity.

Our research is centered on optimizing the exploration process through strategic agent distribution. We initially address the challenge of city roadway coverage, aiming to minimize the travel distance of each agent in a scenario involving multiple agents to enhance overall system efficiency. To achieve this, we partition the city into subregions and utilize Voronoi relaxation to optimize the size of postman distances for these subregions. This technique highlights the essential elements of an efficient city exploration.

Expanding our exploration techniques to unknown buildings, we develop strategies tailored to this specific domain. After a careful evaluation of various exploration techniques, we introduce another goal selection strategy, Unknown Closest. This strategy combines the advantages of a greedy approach with the improved dispersal of agents, achieved through the randomization effect of a larger goal set.

We further assess the exploration techniques in environments with restricted communication, presenting upper coordination mechanisms such as frontier incentivization and area

segmentation. These methods enhance exploration performance by promoting independence and implicit coordination among agents. Our simulations demonstrate the successful application of these techniques in various complexity of interiors.

In summary, this dissertation offers solutions for multi-robot exploration in unknown domains, paving the way for more efficient, cost-effective, and adaptable exploration strategies. Our findings have significant implications for various fields, ranging from autonomous city-wide monitoring to the exploration of hazardous interiors, where time-efficient exploration is crucial.

To my wife, kids, and parents, for their love, support, and encouragement. Also, to my late grandpa and grandma, who raised me and taught me the values of life.

ACKNOWLEDGMENTS

I would like to express my heartfelt gratitude to all committee members for dedicating their time and effort to reviewing my dissertation and for their valuable feedback. Special thanks are due to my advisor, Mario Harper, for accepting me into his lab and providing me with unwavering guidance, support, and motivational talks. I am deeply grateful to Xiaojun Qi for her invaluable guidance and kind directions, which helped me find my way during challenging and difficult times. I also want to extend my appreciation to my helpful colleague, Christopher Allred, for being an excellent labmate and a true friend.

I am immensely thankful for the support provided by the Turkish Ministry of National Education under the law of 1416 - YLSY. Without this support, my dreams would not have become a reality.

Huzeyfe M. Kocabas

CONTENTS

	Page
ABSTRACT	iii
PUBLIC ABSTRACT	v
ACKNOWLEDGMENTS	viii
LIST OF TABLES	xi
LIST OF FIGURES	xii
ACRONYMS	xv
1 INTRODUCTION	1
1.1 Key Contributions	3
2 CITY ROADWAY COVERAGE	4
2.1 Motivation and Earlier Work of Roadway Coverage	4
2.1.1 UAV Deployment Challenges	4
2.1.2 Voronoi Methods for Multi-Drone Coverage	5
2.2 Method	5
2.2.1 City Graph Representation and Postman Problem	5
2.2.2 Divide and Relax the City Network into Sub-Regions	7
2.2.3 City and Drone Selection	11
2.2.4 Convergence Time of Regions	12
2.3 Simulation Results	13
2.3.1 Coverage Time Across Cities	13
2.3.2 Cost Analysis and Recommendations	14
2.4 Conclusions	16
3 UNKNOWN BUILDING EXPLORATION	17
3.1 Partitioning of Unknown Environments	17
3.2 Current Approaches in Autonomous Building Exploration	17
3.3 Method	19
3.3.1 Simulation Design	19
3.3.2 Exploration Strategies	21
3.4 Results	24
3.4.1 Comparison of Exploration Strategies	24
3.4.2 Voronoi Search Time Improvement	25
3.4.3 Unknown Strategies Performance	26
3.4.4 Distribution Effect For Timing	27
3.4.5 Exploration Speed in Relation to Robot Team Size	27
3.4.6 Resilience to Density of the Environment	28

3.4.7	Resilience to Robot Failure Scenarios	29
3.5	Conclusions	31
4	AGENT COORDINATION IN COMMUNICATION RESTRICTED DOMAINS	33
4.1	Earlier Coordination of Multiple Robots in Unknown Environments	34
4.2	Method	35
4.2.1	Communication Scenarios	35
4.2.2	Frontier Incentivization Algorithm	37
4.2.3	Simulation Setup	39
4.3	Results	42
4.3.1	Oracle Based Monitoring Scenario	42
4.3.2	NLOS (Non-Line of Sight) Scenario	44
4.3.3	LOS (Line of Sight) Scenario	45
4.3.4	Agent Team Size and Map Complexity Impacts	47
5	Conclusion and Discussion	50
	REFERENCES	53
	APPENDICES	61
A	Simulators Development	62
A.1	City Coverage Simulator	62
A.2	Unknown Building Exploratio Simulator (UBES)	65
	CURRICULUM VITAE	69

LIST OF TABLES

Table	Page
2.1 Comparison of current modern drone types	11
2.2 Comparison of city and drone types	14
3.1 Exploration Strategies and Descriptions	22
3.2 Comparison of strategies with the average quality ratio for a map area of 100x100	25
3.3 Exploration Time and Replan Count Metrics	26
4.1 Exploration performance in Oracle Scenario	42
4.2 Exploration performance in Non-Line of Sight (NLOS)	44
4.3 Exploration performance in Line of Sight (LOS)	45
A.1 Configuration Settings	68

LIST OF FIGURES

Figure	Page
2.1 The Postman algorithm demonstrated in a sub-region of North Logan, Utah, USA. This route contains 116 total road traversals. Although the total length of all roads in this sub-region is 17.76km, the postman route requires a minimum of 24.1km due to inevitable backtracking. Red arrows illustrates this from step 93 to 97, two inevitable returns happened during steps 95 and 97.	6
2.2 Voronoi relaxation in 3 cities (Logan, Tallahassee, and Madrid). The first column shows the base city network with driveable roads. The second column shows initial sub-regions for twenty agents. The third column is the final sub-region placement after applying Postman Moving Voronoi Coverage (PMVC), all route distances of sub-regions are now similar in length.	9
2.3 Total postman coverage distances in relation to the number of agents. As additional agents are deployed, total length of traversed roadway increases linearly. Dense cities (such as Madrid) are not as impacted as sparse cities with longer average road segments.	10
2.4 Convergence completion time for PMVC technique for Voronoi relaxation in three cities with respect to the number of drones. As the number of agents is increased the computation time increases, and the dispersion of the convergence times also increases.	12
2.5 Completion time for city coverage using various numbers of drones. Due to operating capacity of some drones, map coverage is only possible after deploying more agents. To ensure mission completion and battery stability, drones operate under 70% of their total manufacture stated capacity.	14
2.6 Coverage completion time and cost per agent for the five different drones considered. As drone count is increased, full city coverage time decreases as total cost per km covered increases.	15
3.1 Three phases of building exploration. Agents start from the top left and are assigned initial search goals at the edge of the map to aid in dispersion. After reaching the initial goals, the next goal is assigned based on the strategy in use.	20
3.2 Segmentation of a map among 8 agents. The unknown area is allocated based on Voronoi segments from the initial goal locations (Left - Edge of Map, Center - Center Map, and Right - Equal Spread).	20

3.3	Choices in goal locations as determined by method. Frontier approaches (Frontier Closest (FC) and Frontier Random (FR)) only consider the yellow frontier cells in determining candidate navigation targets. Unknown approaches (Unknown Closest (UC) and Unknown Random (UR)) typically have larger areas of consideration as any unexplored, non-frontier region is considered.	23
3.4	Comparing the impacts of area allocation and distribution (initial navigation targets) over the exploration speed. Initial distribution scenarios do not greatly impact the speed of full exploration. The strategies that allocate sub-areas have a clear improvement in exploration time compared to their counterparts without segmentation.	27
3.5	Average exploration time related to robot count. As the number of robots increases, the exploration time reduces.	28
3.6	Exploration completion time of the strategies against three different room sizes. A small number (10.0) represents many small rooms, resulting in a densely populated area with many walls. When the room sizes increase, the area between walls becomes more sparse.	29
3.7	Illustration of replan count over time on various exploration strategies. Each point in the upper plot represents the time when replan counts stabilize. . .	30
3.8	(a) - The failure ratio of experiments based on strategies. While the random strategies failed more often, Voronoi Unknown Help Closest (VUHC) performed better than others. (b) - The explored area percentage for the failed experiments for the Unrecoverable failure scenario. Using 4 robots allows exploring more than 70% of the area in most experiments before all robots stop functioning. The failed experiments that used 12 robots even reached more than 95% of the area.	31
4.1	A portion of the simulation interface shows four agents exploring an unknown building. The area gets explored while agents scan their environments with 360-degree LiDAR. We propose a frontier incentivization technique, illustrated in a simple example that two agents cannot see each other, and R3 uses R2's location data to incentivize its frontiers accordingly.	39
4.2	Illustration of the four techniques evaluated in simulation. While robots in (a) and (c) do not utilize frontier incentivization via agent detection, (a) and (b) do not consider using voronoi sub-partitioning. Robots in technique (d) uses both agent detection and voronoi partitioning and (a) only uses frontier-based exploration without any upper parameters.	40
4.3	Three examples of building floor plans were generated using minimum room sizes of 10, 20, and 30, from left to right, respectively.	41

4.4	Average area exploration percentage over time in the Oracle communication scenario. This compares the effects of agent detection and Voronoi partitioning in an unknown area. Once the map is fully explored, the oracle terminates the exploration.	43
4.5	Average area exploration percentage in time for the experiments in NLOS scenarios.	45
4.6	Average area exploration percentage in time for the experiments in LOS scenarios.	46
4.7	Oracle Scenario: Monitoring agents overall in the background to terminate exploration.	48
4.8	NLOS: No type of communication allowed between agents. Local decisions have been taken by agents.	48
4.9	LOS: Limited communication allowed only in line of sight scenarios. Local decisions have been taken by agents.	49
A.1	Left: An OpenStreetMap (OSM) map object of a neighborhood in Logan, Utah as an input to postman route calculation. Right: The output of the postman route is fully calculated, and an instance of the postman moving along the route is illustrated.	64
A.2	Dividing a city into sub-regions using Voronoi partitioning. The centroids are shown with bigger colored dots, and the resulting partitioning is shown in different colors.	65
A.3	Exploration of an unknown building with the size of 50 by 50 meters with four agents, each working on the Frontier Closest algorithm as an exploration strategy. The purple-colored segments of the map represent the unknown areas, the green-colored areas represent the explored known empty areas, and the blue areas represent the explored objects (i.e., walls). The yellow cells on the map are the frontier areas. The red numbers with blue square cells represent the agent locations, and the corresponding white numbers are their goal locations, respectively. The right part of the figure shows the visualization of some values we record, which might be helpful for preliminary analysis.	66

ACRONYMS

GPS	Global Positioning System	2
P2P	Peer-to-Peer	2
UAV	Unmanned Aerial Vehicle	4
ITS	Intelligent Transportation Systems	5
OSMnx	OpenStreetMap and NetworkX	5
PMVC	Postman Moving Voronoi Coverage	xii
OSM	OpenStreetMap	xiv
VUHC	Voronoi Unknown Help Closest	xiii
FC	Frontier Closest	xiii
FR	Frontier Random	xiii
UC	Unknown Closest	xiii
UR	Unknown Random	xiii
VUHR	Voronoi Unknown Help Random	21
VFHC	Voronoi Frontier Help Closest	22
VFHR	Voronoi Frontier Help Random	22
DFC	Decision Frontier Closest	23
LiDAR	Light Detection and Ranging	23
DEGU	Decay Epsilon Greedy Unknown	24
DEGF	Decay Epsilon Greedy Frontier	24
NLOS	Non-Line of Sight	xi
LOS	Line of Sight	xi
UBES	Unknown Building Exploration Simulator	35

CHAPTER 1

INTRODUCTION

Exploration of an unknown area stands as a fundamental challenge in robotics, where systems must often navigate and explore unfamiliar environments efficiently and safely [1,2]. A robot’s ability to effectively explore and understand the operational workspace is critical for navigation and task/mission completion [3,4]. Key tasks for successful exploration include navigating through the environment, obstacle avoidance, and accurate mapping of the surroundings [5]. Uncertainties associated with unexplored environments compound the exploration process (unpredictable map, sensor limitations), emphasizing the need for exploration capabilities that can adapt and perform effectively [5,6].

In recent years, the demand for autonomous exploration has surged across various domains, including search and rescue operations [7–10], city-wide monitoring [11,12], disaster response [13,14], and space exploration [15–17]. Consequently, researchers are developing exploration algorithms to cater to diverse needs, ensuring that robots can explore unknown areas with precision and reliability [18,19].

An exploration algorithm’s primary goal is to complete exploration tasks in the shortest possible time, especially in time-critical applications [20,21]. Achieving this goal necessitates careful sub-goal selection and navigation, which attempts to explore large areas of an unknown map with minimal travel [22,23]. This exploration heavily relies on the robot’s sensor capabilities to perceive the environment and the algorithms that devise strategies to navigate to the optimal location [22,23]. While this task is relatively straightforward for a single robot, it becomes significantly more challenging when multiple robots are involved. In multi-robot exploration, effective robot coordination is crucial to ensure the timely completion of exploration tasks [24,25]. Coordinating these robots is essential to prevent them from exploring the same area, a challenge that persists even in environments where communication between robots is allowed.

Robot deployment in communication restricted environments increases complexity as each platform cannot access communication networks for data sharing or localization via Global Positioning System (GPS) [26, 27]. This challenge arises from the inability to share information among robots, necessitating a heavy reliance on individual sensor capabilities to navigate and explore unknown environments. In these environments, localized communication (Peer-to-Peer (P2P)) between robots may still be possible, but even this may be denied in certain applications (military) [28, 29]. In such cases, robots face the daunting task of exploring unknown areas in teams with little coordination and rely on traditional algorithms for exploration.

The most common unknown area exploration algorithms are based on a closest Frontier exploration strategy [30], depending on the specific choice in the algorithm, the frontier selection can be modified to improve performance [31]. However, most of these approaches are either tested in specific environments [32, 33], assume the capability of data sharing between agents [34], or with a central map (oracle) [35]. However, these approaches are not applicable in fully communication-restricted domains where robots cannot share any information.

To address these challenges, task allocation strategies such as Voronoi partitioning of the exploration area can be utilized to assign independent sub-tasks to each robot [32, 36, 37]. The subregions of the partitioned area are assigned to robots at the beginning phase of exploration. After the task allocation, robots explore their assigned area without any communication or explicit coordination. This approach can be used in a wide range of applications when communication is fully restricted, and robots fully rely on their own sensor capabilities to explore unknown areas.

We can partially relax the zero-communication condition to improve coordination locally, without explicit sharing of information among agents. Robots can utilize signal strength to estimate other robots' locations in the operation domain. Utilizing this information, an implicit coordination mechanism can be developed which chooses frontiers

differently. Combining the Voronoi partitioning and agent detection mechanism, robots improve exploring individual subregions and better coordinates in their assigned subdomains.

This document details the technology developed for multi-robot coverage, exploration, and coordination in uncertain and unknown maps. In Chapter 2, we demonstrate how to cover a city in a timely manner using multiple robots by subdividing the city road network into subregions and assigning a similar load of work to each agent. Chapter 3 details a similar area division technique applied in unknown buildings and measures its effects on exploration performance. Also, the potential loss of agents during exploration and its impact on the speed of exploration can be observed. Chapter 4 explains the coordination using Voronoi partitioning and agent detection mechanisms under various communication restrictions.

1.1 Key Contributions

The major contributions of this work are listed:

- A novel approach to cover a city using multiple robots subdividing the city network and assigning a similar load of work to each agent. (Chapter 2)
- Demonstration of the benefits of segmentation of unknown building and different distribution techniques' performances under different complexity of building maps (Chapter 3)
- Investigating the performance effects of exploration strategies under the two agent loss scenarios (full loss and recoverable) (Chapter 3)
- Coordinating multiple agents in various levels of communication restricted domains using Voronoi partitioning and agent detection mechanism (Chapter 4)

CHAPTER 2

CITY ROADWAY COVERAGE

2.1 Motivation and Earlier Work of Roadway Coverage

In recent years, Unmanned Aerial Vehicle (UAV) systems have been used in various key tasks related to smart cities such as package delivery, wireless access points, traffic monitoring, firefighting, and agriculture [38–40]. UAV applications in smart cities often include many drone platforms operating simultaneously to complete missions and tasks. While safety and security is crucial to improve adoption in future smart cities, operating quality and stability of drone deployments in urban areas has often slowed their use.

2.1.1 UAV Deployment Challenges

While UAVs have the potential to significantly improve future smart cities, there are implementation challenges. Some are policy-based, such as UAV flight regulation; others are technical (software and hardware) limitations such as battery, payload, and intelligent autonomy.

While developments in rotor-type UAV technology constantly push the envelope of capabilities, significant hurdles remain in exploring faster, longer-ranged systems. In particular, typical rotor-type UAV systems operate at a range and speed, often directly tied to battery capacity, that is generally only 30-45 minutes; thus limiting UAV-based ITS operations. Conversely, fixed-wing UAV systems have larger payloads and longer operation times and ranges. However, due to holonomic constraints that limit sudden changes in direction, fixed-wing UAV systems are not well suited for road network monitoring.

Full roadway coverage at city-scale within reasonable times requires a drone asset with high velocity and maneuverability. Many current applications of municipal UAV surveillance, such as the Flying Police Eye [41], struggle to perform as the operating velocity of

a UAV is unable to keep up with moving vehicles. Other drone surveillance approaches, such as high-velocity UAVs at high altitudes, benefit from a wide field of view and lack of building obstruction in flight, but cannot capture detailed information to support a high degree of Intelligent Transportation Systems (ITS) observability. These systems are also challenging to maintain and deploy, thus difficult to justify their high operating costs.

An ideal solution to rapid city full-roadway observation is to simultaneously deploy many UAVs to operate in different parts of the city. Current approaches require a human supervisor, per policy in many countries, to determine boundaries for individual or swarms of UAVs. Human direct supervision is still essentially required to guide, control, and patrol using UAVs as they add to existing ITS burdens. Thus, autonomously determining the number of agents, their deployment positions, and assigning routes/tasks that lie within agent capabilities is crucial.

2.1.2 Voronoi Methods for Multi-Drone Coverage

Many of the modern approaches regarding the area division problem in multi-drone context [42, 43] rely upon Lloyd’s algorithm [44] or the Voronoi partitioning [45]. Although these approaches are suitable to apply to the two-dimensional euclidean distance plane, dividing the city streets as euclidean spaces does not give an optimum solution for multi-drone deployment as roadways are rarely evenly distributed homogeneously, a requirement for many voronoi relaxation techniques [46]. While applying a Voronoi space partitioning onto a city network gives equally divided areas, the street lengths in these areas would vary significantly.

2.2 Method

2.2.1 City Graph Representation and Postman Problem

We created a simulation environment that demonstrates multi-drone systems coverage for city map with various network structures. During the creation of this environment, we accessed city networks using OpenStreetMap and NetworkX (OSMnx), a Python pack-

age that leverages geospatial data from OpenStreetMap [47] to analyze street networks. OpenStreetMap encodes information valuable for drone-based city street coverage such as intersection positions, road segment lengths, and geographic and locally projected coordinate systems. Street network analysis using NetworkX (a tool designed for complex graph structures) was also leveraged.

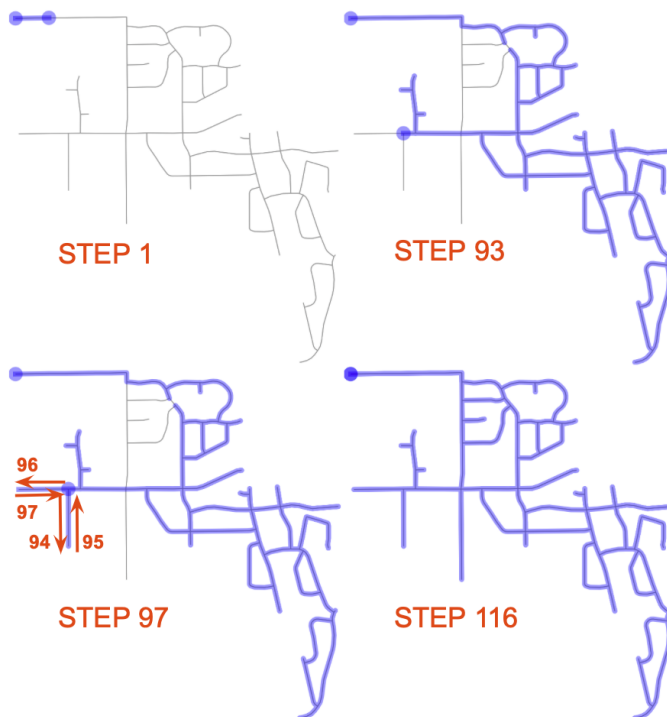


Fig. 2.1: The Postman algorithm demonstrated in a sub-region of North Logan, Utah, USA. This route contains 116 total road traversals. Although the total length of all roads in this sub-region is 17.76km, the postman route requires a minimum of 24.1km due to inevitable backtracking. Red arrows illustrates this from step 93 to 97, two inevitable returns happened during steps 95 and 97.

After accessing the street data through [OSMnx](#), we investigated how an agent (e.g. drone) would cover a road map with minimized cost, in distance or energy during its traversal. We choose to employ Postman Coverage (also known as Chinese Postman Problem, Postman Tour, or Route Inspection Problem) created specifically to determine the shortest path through an undirected graph. The objective of the Postman Coverage algorithm we employ is to optimize a route starting and stopping at a deployment point with the fewest

number of "re-visits" of road segments. Streets were further limited to only those drivable. The optimized route that a drone would follow (shown in Figure 2.1) demonstrates how a typical route would be computed.

Calculating a postman coverage route for a network, such as small city neighborhood can be easily computed, however, as the network structure becomes more complex, more time to calculate the postman route is required. This is because the complexity of the Postman algorithm for undirected graphs is $O(n^3)$ [48]. Optimization then quickly becomes computationally expensive due to the algorithm's innate complexity.

When considering hardware limitations (as of the time of writing) of rotor-type drones, maximum flight times (30-45 minutes) and flight speeds (16-21 meters per second) are common. With these range and speed limitations, covering a street network with a single drone is beyond the capability's of any one platform. Multi-agent systems are then needed to resolve limitations, this however depends on an algorithm that can intelligently assign sub-regions and optimal routes from the full road-graph which honors platform capabilities.

2.2.2 Divide and Relax the City Network into Sub-Regions

To use multiple drones for a given city, each drone's coverage task should be independent of another to eliminate unnecessary overlap and collision risks. Additionally, the coverage distances (based on a postman algorithm) of each sub-area is constrained at roughly the same size based on the number of drones.

The standard postman algorithm is not well suited for application in dense and large cities, due to the computational burdens stemming from searching through many roads, and required development of an algorithm that divides the city into manageable sub-regions that an agent can realistically cover. The developed algorithm called **PMVC**, is shown in Algorithm 1.

These sub-regions are generated using the Voronoi graph method [46], having initial centroids placed on the vertices of the road graph. Initial placement distributes the centroids uniformly as this leads to a better initial seeding. All nodes and the edges (street and their start/end points) on the graph are assigned to the closest centroid. Streets are often divided

Algorithm 1: PMVC Algorithm

Input: OSM Map Object, Coordinate Values of Drones
Output: Divided Voronoi Map for Similar Traversal Time on Drones
Data: Following OSM City Maps: *Logan, Tallahassee, Madrid*

```

1  $C \leftarrow$  number of drones
2  $G \leftarrow$  city graph
3  $g \leftarrow$  sub-region of a city
4  $C(l) \leftarrow$  Selection of Drone Coordinates( $G, C$ )
5  $C(g) \leftarrow$  sub-region for a drone,  $\forall g \in G$ 
6 for  $10 \leq C \leq 50$  do
7    $List[C(g)] = CalcPostmanDistances(G, C(l))$ 
8    $M \leftarrow$  average(List[C(g)])
9    $L = 0.3 * M$ 
10  while  $C(g) < M - L$  or  $C(g) > M + L$  do
11    if  $C(g) < M - L$  then
12      if  $C(g) < L$  then
13        /* random placement in the biggest sub-region */
14         $C(l) \leftarrow$  MoveInBiggestSubregion
15       $C(l) \leftarrow$  Move Towards Neighbor
16    else if  $C(g) > M + L$  then
17       $C(l) \leftarrow$  Move Away Neighbor
18    else
19       $C(l) \leftarrow$  Keep Stable
20     $CalcPostmanDistances(G, C(l))$ 
     $CheckPotentialLoops(C(l), C(l_{prev}))$ 

```

into segments allowing a single road to partially belong to one voronoi region, improving overall completion times.

After creating the initial Voronoi sub-regions of a city, we calculate the postman route and the coverage distances for each sub-region similarly as applied in [49]. PMVC finds the shortest closed path (or circuit) for a given graph (e.g. city neighborhood) such that it visits every edge (e.g. street) on that graph.

Uniform initial seeding of voronoi centroids often produces postman coverage distances that are significantly varied such that each agent is not guaranteed to have the same roadway length. Initial regions are often too large for a drone to be able to cover within flight times. PMVC algorithm reduces the difference between the largest and smallest route

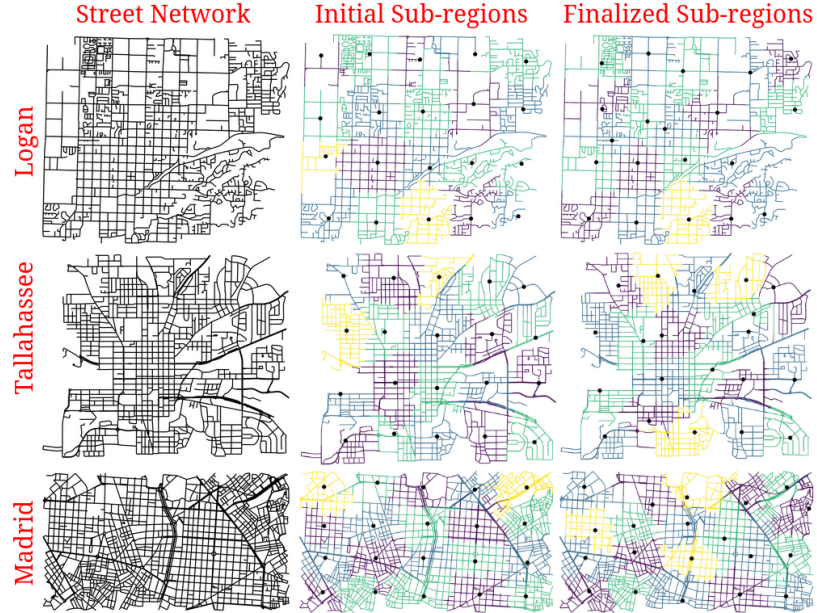


Fig. 2.2: Voronoi relaxation in 3 cities (Logan, Tallahassee, and Madrid). The first column shows the base city network with driveable roads. The second column shows initial sub-regions for twenty agents. The third column is the final sub-region placement after applying PMVC, all route distances of sub-regions are now similar in length.

lengths of each sub-region to ensure all agents are able to cover their areas (shown in Figure 2.2). It is computationally challenging to ensure identical route distances for all drone regions, thus, a tolerance threshold was added based on the average route length of sub-regions.

The Voronoi relaxation operates on a principle of attraction or repulsion from the most or least valuable neighbor respectively, and each decision is based upon a calculation of a weight factor. The sub-regions, which have a smaller postman coverage area than average - tolerance (shown in Line 11 of Algorithm 1), move towards the most valuable neighbor by calculating a weight factor as shown in equation 2.1. While the area of the neighbor is positively correlated with the weight factor, the distance has the opposite effect. Due to the city network complexity, in certain situations, some of the sub-regions' postman distances are even smaller than the tolerance value. In these circumstances, the centroid is re-located under the sub-region that has the biggest postman distance.

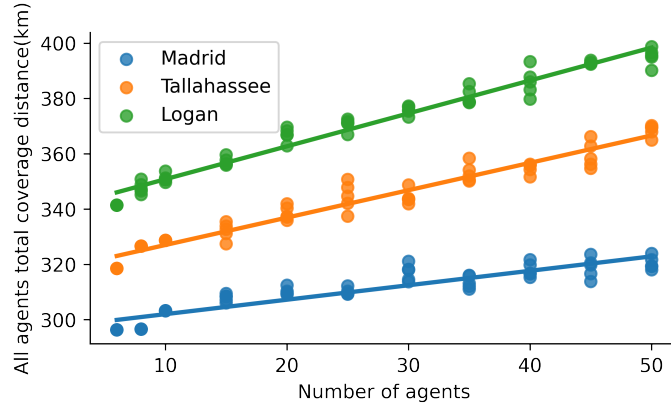


Fig. 2.3: Total postman coverage distances in relation to the number of agents. As additional agents are deployed, total length of traversed roadway increases linearly. Dense cities (such as Madrid) are not as impacted as sparse cities with longer average road segments.

$$weight_i = \frac{NeighborArea_i}{NeighborDistance_i} \quad (2.1)$$

Conversely, the sub-regions, which have a larger postman coverage area than average + tolerance (shown in Line 15 of Algorithm 1), move away from the least valuable neighbor. To locate the least valuable neighbor, the algorithm calculates weight factors with each of its neighbors using the equation 2.2. Both area and distance of a neighbor have a negative correlation with calculated weight factors. In other words, having a small area as well as proximity would lead to calculating a high weight factor, which means the least valuable neighbor to move away.

$$weight_i = \frac{1}{NeighborArea_i * NeighborDistance_i} \quad (2.2)$$

Finally, the sub-regions, which their postman distances are already in the range of (average - tolerance) and (average + tolerance) kept stable. Once every sub-regions' total postman distance is within tolerance, the algorithm outputs the finalized sub-regions as shown in the right column of Figure 2.2.

2.2.3 City and Drone Selection

We selected three different cities to test our simulation environment and observe general performance. The cities were limited to having the same summed road lengths (250 km) to observe the Voronoi relaxation in different road structure scenarios.

Cities were specifically selected based on specific road characteristics such as various geometric shapes, highways in cities, the density of the streets, and the population of the city. The first city (Logan, UT), is a comparatively small town with sparse roads and a simple road system (measured by average road-to-road connections). A second city (Tallahassee, FL) was chosen for its' mix of multiple highway roads and densely populated areas. The third city (Madrid, Spain) was picked being one of the most populated cities in Europe.

Table 2.1: Comparison of current modern drone types

Drone Type	Transmission Range (km)	Operational Range (km)	Flight Time (min)	Max Speed (m/s)	Price (\$)
A	9.0	48.0	40	20	2500
B	15.0	57.9	46	21	2049
C	12.0	35.3	31	19	999
D	12.0	32.6	34	16	759
E	10.0	29.7	31	16	449

To cover the 250 kilometers of roadway, we considered a range of 10 to 50 drones for all three cities. Within this range, it is possible to find drones can cover potential sub-regions. As stated earlier, each computed coverage route is inevitably significantly larger than the theoretical minimum (25km each for 10 drones) due to necessary revisits. As the Voronoi relaxation is a time-consuming process, the step size difference between the number of drones picked was 5.

In each of the three cities, increasing the number of drones enhances the overall postman distance covered by all drones (see Figure 2.3). Using 10 drones to cover Madrid's 250km portion will result in a total route distance of 300km, which is unavoidable as revisiting streets becomes increasingly required. However, covering the same region with 50 drones

will result in a total distance of roughly 320km, implying that increasing the number of drones will increase re-visits, affecting the overall length of the full coverage route.

Several drones were chosen (Table 2.1) based on commercial availability in similar application spaces. Their costs, flight speed, flight time, and operation range are listed. Rather than mentioning the specific company or product names, we focus on the drone characteristics. Any drones with those characteristics (flight time, speed, cost) could be used to reproduce the results we listed here. We note the price difference among the selected drones is affected by the hardware, sensors, and software associated as well. All considered drones can be used for city coverage as they are equipped with long-range transmission and can cover large amounts of area in terms of flight time and flight speeds.

2.2.4 Convergence Time of Regions

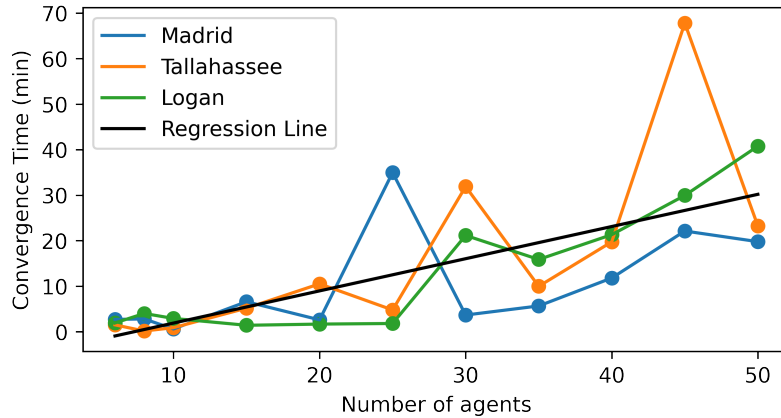


Fig. 2.4: Convergence completion time for [PMVC](#) technique for Voronoi relaxation in three cities with respect to the number of drones. As the number of agents is increased the computation time increases, and the dispersion of the convergence times also increases.

Finding the converged graph using [PMVC](#) algorithm takes time to compute because of the Voronoi relaxation. This is due to the postman problem must be solved in each iteration of the Voronoi relaxation. In addition, using different sizes of drone setups also brings another burden to the algorithm. The relation of using various numbers of drones and its effect over time to complete relaxation of the Voronoi sub-regions in three cities is

demonstrated in Figure 2.4.

Increasing the number of agents has a direct correlation with the convergence time to relax Voronoi regions. The regression line illustrates that the convergence time has an increasing trend overall for all cities. When there are 20 or fewer drones on the map, the algorithm converges and outputs the final sub-regions much faster compared to having more drones. Even though there are highs and lows for a certain number of drones, starting from using 25 or more drones, the convergence time is getting doubled for all three cities.

2.3 Simulation Results

Using the PMVC algorithm, 45 experiments were run. Various numbers of drone agents were virtually deployed (from 10 to 50) for each city, average data is plotted from these experiments Figure 2.5.

2.3.1 Coverage Time Across Cities

To assess the capabilities of the various drone types, road coverage completion time is calculated based on the traversal area of each sub-graph for each individual drone. The velocity and total length of roadway in the sub-graph is utilized to compute coverage time as shown in Equation 2.3 where N_{agents} is the number of unique drones deployed, G_{area_i} is the PMVC determined road-length for sub-region i , and $v * 0.7$ is the tolerance weighted velocity.

$$time_c = \frac{(\frac{1}{N_{agents}} \sum G_{area_i})}{(v * 0.7)} \quad (2.3)$$

The completion time of city-wide road coverage with different characteristic drones is illustrated in Figure 2.5. While most maps, having a 250 km total road length, can be covered with 10 drones within 40 minutes, the same map can be covered with 50 drones in less than 10 minutes. Although all chosen drone platforms are capable of covering a given city area with 50 units, a significant time difference in complete coverage exists. This is largely due to the top-speed difference between systems. This time difference may be crucial

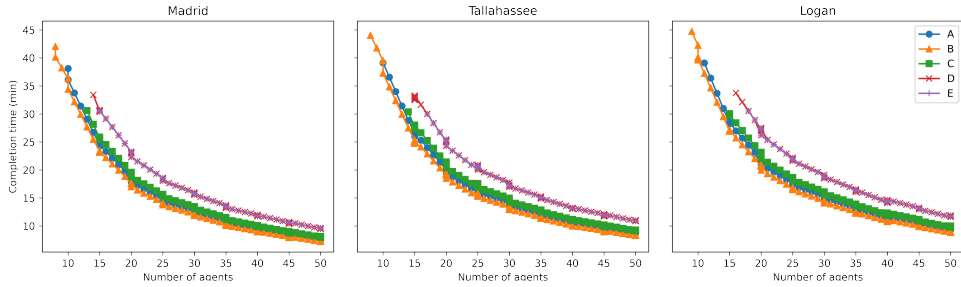


Fig. 2.5: Completion time for city coverage using various numbers of drones. Due to operating capacity of some drones, map coverage is only possible after deploying more agents. To ensure mission completion and battery stability, drones operate under 70% of their total manufacture stated capacity.

in some circumstances and should be weighed by adopters.

Not all drones are capable of covering the sub-regions if the drone count is too low, causing PMVC to fail to find suitable full-coverage solutions. The drones that are less capable (less flight time, slower speed) require more agents. Table 2.2 demonstrates this relationship by stating the required minimum number of drones of each type to cover the selected three cities. Further, this table denotes changes in coverage time as agent count is increased.

Table 2.2: Comparison of city and drone types

Drone Type	Madrid			Tallahassee			Logan		
	Min Agents	Time (min)	Cost (\$)	Min Agents	Time (min)	Cost (\$)	Min Agents	Time (min)	Cost (\$)
A	10	38.11	25000	10	39.13	25000	11	39.10	27500
B	8	42.04	16392	8	44.03	16392	9	44.73	18441
C	13	30.60	12987	14	30.37	13986	15	30.04	14985
D	14	33.40	10626	15	33.27	11385	16	33.73	12144
E	15	30.70	6735	17	29.99	7633	18	30.51	8082

2.3.2 Cost Analysis and Recommendations

It was observed that deploying more inexpensive drones(e.g. D, E) might be more cost-efficient than deploying fewer expensive drones(e.g. A, B). In the case of Tallahassee,

potentially deploying ten relatively expensive Type A drones results in more than three times as much expense. The cost of running 20 type E drones is substantially lower while getting better coverage. Costs per kilometer of coverage (shown in Figure 2.6) similarly shows that less expensive drones (C, D, and E) is likely a superior option over A or B drones. We note that these costs do not include installation of deployment platforms. Deployment of Type-A drones is rarely in the interest of the city as they cover areas slower, and cost significantly more.

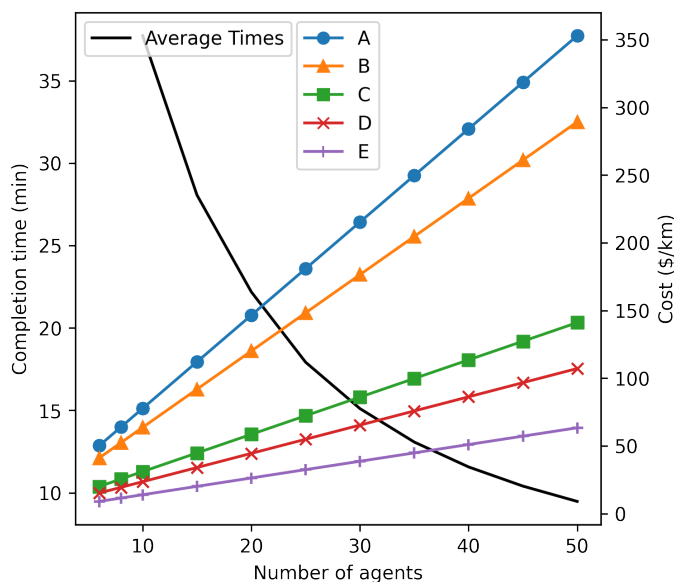


Fig. 2.6: Coverage completion time and cost per agent for the five different drones considered. As drone count is increased, full city coverage time decreases as total cost per km covered increases.

While results favor the use of many lower capacity systems, drones such as A and B are capable of higher payloads which may be more suitable for smart city applications. Some of these payloads may include sensitive cameras, additional thermal sensors, and communication relay devices rendering these platforms more suitable than the affordable drones. This work does not assess the applicability of these additional sensors, rather, [PMVC](#) provides potential for cities to test their own drone platforms and unique road systems in our environment.

2.4 Conclusions

In this chapter, we highlighted pivotal insights for optimizing drone deployments in urban environments. Deploying multiple, cost-effective drones is essential for efficient city-wide coverage, significantly impacting coverage time and costs. However, we mainly focused optimizing roadway coverage in well-defined urban environments with known travel conditions.

It's important to acknowledge that there are scenarios where the benefits of the [PMVC](#) algorithm may not apply as seamlessly. In situations where observations are occluded or domains are unknown, such as in interior building environments, much of the traditional road network-based advantages of [PMVC](#) might be lost.

Interior exploration presents a distinct set of challenges and opportunities. When navigating in the bounded interior space such as a building or facility, robots face complexities like communication constraints, unpredictable objects/structures, and the need to make real-time decisions about next potential region to travel. Addressing these challenges and leveraging the capabilities of using multiple robots for interior exploration is a promising area for future research. While [PMVC](#) has proven effective for outdoor roadway coverage, developing new methods that account for uncertainty in unknown indoor environments is essential.

CHAPTER 3

UNKNOWN BUILDING EXPLORATION

3.1 Partitioning of Unknown Environments

Building on the foundations established in the first chapter on urban roadway coverage optimization using Voronoi Relaxation, our research endeavors expand into uncharted territories: interior building explorations. While our prior work shed light on the intricacies of optimizing multiple agent deployments in well-defined urban environments, the focus now shifts to the complexities of unknown and occluded spaces within buildings. This paradigm shift introduces a new set of challenges and opportunities, demanding innovative approaches that can navigate dynamic obstacles, adapt to unpredictable environments, and make real-time decisions within confined spaces.

In this second chapter, we explore the utilization of Voronoi partitioning, a technique we previously employed for outdoor roadway coverage, into interior building exploration strategies. Our aim is to pioneer methods that seamlessly integrate Voronoi divisioning into unknown area explorations, bridging the gap between structured outdoor environments and intricate indoor spaces. By adapting and enhancing Voronoi-based algorithms, we pave the way for more efficient, adaptive, and versatile robot explorations within diverse, complex indoor settings.

3.2 Current Approaches in Autonomous Building Exploration

Unknown area exploration, which mainly focuses on generating maps of unknown environments using single or multiple robots, has been studied over the years due to its applicability in critical applications. One of the earlier and commonly used techniques in the field is frontier-based explorations, first presented in [30]. The frontier method has been extensively evaluated in the field [50–52] due to its ease of implementation and functionality in

most situations. Later, frontier detections started to be adapted into 3D environments [53], and used with aerial systems [54, 55]. Further studies investigated frontiers that considered information gain [56] or goal selection techniques from a set of frontiers [57]. While frontier-based approaches are considered simple as they choose the closest available frontier, they carry potential drawbacks that might lead to oscillatory behavior, where robots repeatedly navigate between advantageous frontiers. Additionally, when applied in multi-agent scenarios, the strategy can induce leader-follower dynamics [58].

Sampling-based approaches overcome this issue since they randomly or strategically produce multiple goal locations and assess the anticipated information gain that could result in exploring each of those goals. Due to its nature of assessing the potential gain of multiple goal locations, it is accepted as a more suitable approach for complex structures [59] compared to frontier-based approaches. However, this approach is computationally expensive due to the calculation of multiple locations' information gain [60].

Later efforts studied the exploration of unknown areas taking advantage of either frontier-based, sampling-based, or hybrid approaches. These include random walk [61], greedy-based frontier explorations [32, 62], auction-based approaches [58, 63, 64], and segmentation of unknown areas [65–67]. Among those studies, current multi-robot unknown environment exploration approaches aim to minimize repeated coverage while improving coordination and increasing exploration speed [31, 68, 69]. Few methods are wholly dedicated to supporting unknown area exploration for complex interior structures and are more suitable in general applications [70, 71].

Both types of exploration algorithms (frontier and sampling-based) are commonly designed to be greedy in nature. These methods choose from the closest frontiers [30, 53, 72, 73] or the sample that gives the highest information gain [74–76] is picked to reduce the travel distance but increase exploration efficiency for robots. However, the field has been lacking in illustrating how computationally efficient strategies based on FC will perform in complex environments, and how they are affected by the initial distribution of robots in the environment. Partitioning the search space to each agent is also critically important [37, 77].

Even though it has potential drawbacks, frontier-based exploration is still commonly used in the field not only because of its simplicity [55] but also its theoretical understandability and applicability. This chapter focuses on the importance of various factors in designing a modified frontier-based exploration strategy (UC) with minimal computational requirements that perform well in complex environments. Investigating the impact of the initial distribution of robots in the environment and how the environment’s segmentation could affect the strategies’ performance were also tested in our experiments. While testing their performance differences, robustness is also tested in each variant of exploration strategy to assess impact in unknown and potentially hazardous environments.

3.3 Method

3.3.1 Simulation Design

Simulated buildings are represented as occupancy maps with a size of 100m x 100m. The maps have a bounded area, and full exploration is always guaranteed to be possible in randomly generated building maps by ensuring each room has at least one entrance.

Unknown exploration environments could be hostile, and two scenarios were benchmarked to reflect this situation. In the first scenario (disrepair), a robot is entrapped or entangled and can be recovered with the assistance of another robot. The scenario improves the robustness of exploration strategies in situations requiring closer teaming and assesses adaptability in complex, unpredictable circumstances.

In the second variation (Unrecoverable), robots receive irreparable damage (such as an explosive mine). They cannot be rescued by others removing them from the scenario.

The simulation includes many parameters that alter the scenarios. These include the number of robots, exploration strategies, starting scenarios, goal targets, robot failure, and room density within the generated building.

The initialization of building search experiments is based on previously established parameters, as illustrated in Figure 3.1 and 3.2. Within the simulation, robots are denoted by red numerals accompanied by blue dots, while targets are marked by white numerals

with black dots. Each robot employs a modified A* planner. The robot adjusts its planned path upon detecting obstacles, like walls, or if another robot completed the exploration of that target cell.



Fig. 3.1: Three phases of building exploration. Agents start from the top left and are assigned initial search goals at the edge of the map to aid in dispersion. After reaching the initial goals, the next goal is assigned based on the strategy in use.

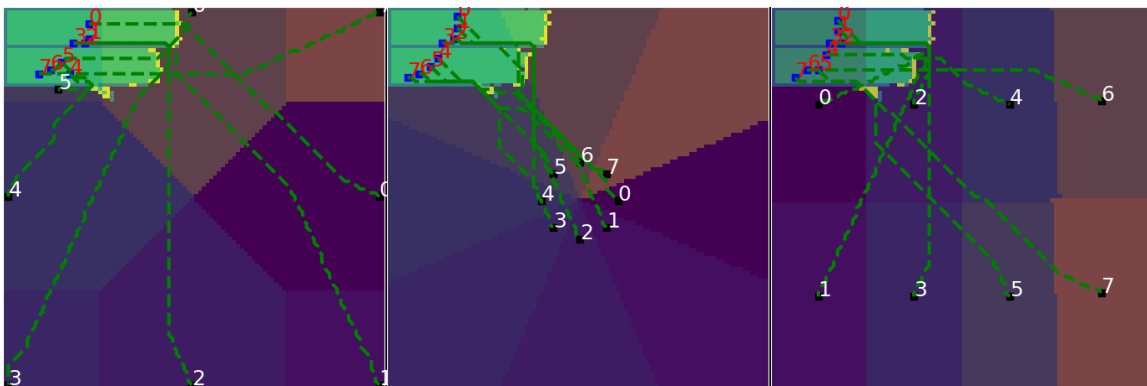


Fig. 3.2: Segmentation of a map among 8 agents. The unknown area is allocated based on Voronoi segments from the initial goal locations (Left - Edge of Map, Center - Center Map, and Right - Equal Spread).

To evaluate the performance of exploration strategies in environments with varying obstacle densities, we generated building layouts with three different minimum room sizes (10, 20, and 30). Figure 3.1 shows an example of a room with a size of 10, where the walls are represented by light-blue colored cells. Smaller room sizes (10) result in a higher density

of rooms.

As the starting position of the robots can impact exploration, five initial configurations are used to test performance: Edges of Map, Equal Spread, Random Location, Center of Map, and Top Left Corner (3.2). Initial dispersion of the robot agents also follows, with the same five locations used as initial navigation targets. These scenarios help understand the effect of dispersion and starting location on exploration efficiency.

Simulated robots are equipped with a sensor range of 25 meters and can travel in any of the eight adjacent cells. Travel speeds are adjustable but are set to $1m/s$ for simulation experiments, enabling rapid traversal.

3.3.2 Exploration Strategies

Twelve exploration strategies are employed, and simplified acronyms will be used to describe them (Table 3.1). A companion publication outlines each method and provides the source code [73]. Exploration strategies access the shared map, where each cell is marked as unknown, empty, or an obstacle. Frontiers, obstacles, and unknown areas (illustrated in Figure 3.3) are sufficient to test the exploration strategies.

Frontier vs Unknown Navigation Strategies

Frontier-based strategies (FC, FR) inform robot navigation targets by only considering available frontiers in the shared map (see Figure 3.3). The FC strategy picks the closest unexplored frontier cell, while the FR strategy randomly chooses from any frontier. If both robots use FC, they potentially select the same frontier cell. This is less common in FR where the random selection encourages wider dispersion. The robots continuously update their targets until the exploration is complete.

In addition to frontier strategies, Figure 3.3 also illustrates goal selection examples in unknown spaces (UC, UR, VUHC, Voronoi Unknown Help Random (VUHR)). Unknown exploration area strategy involves selecting navigation targets from the set of unknown areas and often results in longer navigation, leading to higher dispersion and faster exploration. The wider dispersion also results in mitigating leader-follower situations common to FC

Table 3.1: Exploration Strategies and Descriptions

Acronym	Strategy and Description
FR	Frontier Random: Randomly selects a point from the detected frontiers.
FC	Frontier Closest: Selects the closest frontier point from the detected frontiers.
UR	Unknown Random: Randomly selects a point from the unexplored section of the map.
UC	Unknown Closest: Selects the closest point from the unexplored section of the map.
VFHR	Voronoi Frontier Help Random: Uses the frontier random method paired with Voronoi partitioning task allocation strategy.
VFHC	Voronoi Frontier Help Closest: Uses the frontier closest method paired with Voronoi partitioning task allocation strategy.
VUHR	Voronoi Unknown Help Random: The map is split up into various sections and then the unknown random method is used in tandem with the paired searching strategy.
VUHC	Voronoi Unknown Help Closest: The map is split up into various sections and then the unknown closest method is used in tandem with the paired searching strategy.
DFC	Decision Frontier Closest: Sharing intended goal position with other robots. A goal is assigned to the closest agent.
DEGU	Decay Epsilon Greedy Unknown: Assigns random or closest exploration of a frontier based on an epsilon value that decays over time.
DEGF	Decay Epsilon Greedy Frontier: Assigns random or closest exploration of an unknown area based on an epsilon value that decays over time.
AM	Anti-Majority: Chooses the anti-majority of exploration methods (random or closest) used by other agents.

approaches. However, [UC](#) approaches may not be guaranteed to work without ensuring that all areas within the bounded domain are accessible.

Voronoi Strategies

The addition of Voronoi partitioning is used to allocate map regions based on initial goal locations [36]. Three examples of Voronoi partitioning are shown in Figure 3.2. The Voronoi segmentation methods (Voronoi Frontier Help Closest ([VFHC](#)), Voronoi Frontier Help Random ([VFHR](#)), [VUHC](#), [VUHR](#)) differ in search approach.

When each agent is responsible for their specific voronoi-defined section, agents thoroughly explore assigned areas, allowing for more efficient planning and reducing time spent

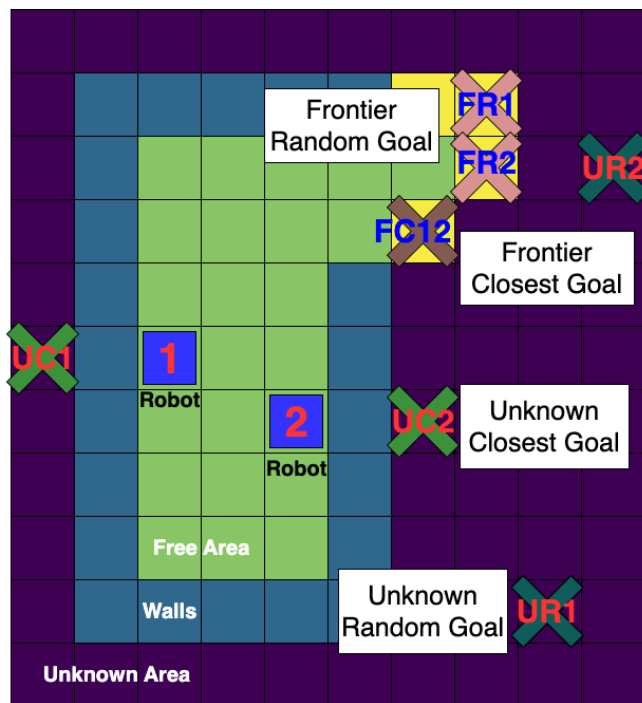


Fig. 3.3: Choices in goal locations as determined by method. Frontier approaches (FC and FR) only consider the yellow frontier cells in determining candidate navigation targets. Unknown approaches (UC and UR) typically have larger areas of consideration as any unexplored, non-frontier region is considered.

recalibrating or adjusting to other agents' movements. By designating assignments, agents are effectively moved where they are most needed. Time and energy traversing large distances between exploration targets are rare, as they will likely be nearby within their assigned zones.

Decision Frontier Closest

Decision Frontier Closest (DFC) aims to optimize the performance of FC by preventing robots from choosing adjacent goals, which minimizes overlap. Each robot updates a centralized database with its new goal locations and plans. If a shorter plan is available for a goal within the Light Detection and Ranging (LiDAR) scan area, the robot with the longer plan has its plan invalidated and triggers a replan to maintain efficiency in the exploration process.

Decay Epsilon Strategies

There are two Decay Epsilon approaches that gradually reduce randomness over time, becoming a greedy search favoring unknown (Decay Epsilon Greedy Unknown ([DEGU](#))) or frontier (Decay Epsilon Greedy Frontier ([DEGF](#))) cells. This approach alters performance through exploration and exploitation by progressively shifting from random goal selection to greedy selection as the known areas increase.

Anti-Majority

The Anti-Majority strategy disperses robots by assigning agents a majority greedy approach ([FC](#)), with a minority of agents choosing the next cells through a random approach ([FR](#)).

3.4 Results

3.4.1 Comparison of Exploration Strategies

To evaluate the success of exploration strategies, we utilize a quality ratio [3.1](#), which provides a quantitative measure for comparing the exploration strategies [[58](#)]. This is useful as a benchmark to show an exploration algorithm’s effectiveness. While the quality ratio positively correlates with the total amount of explored area, it negatively correlates with the total distance traveled.

$$Q = \frac{A}{\sum_{i=1}^n d_i} \quad (3.1)$$

Table [3.2](#) shows the overall results of each tested exploration strategy. The values listed in this table are the average of 81,000 experiments in the standard multi-robot exploration on randomly generated buildings.

Among the strategies that do not consider Voronoi partitioning of a given map, [UC](#) performs the best on average by exploring $8.06m^2$. Similarly, [FC](#) performed well with an average of $7.31m^2$. We note that both high-performance strategies favor greedy goal

Table 3.2: Comparison of strategies with the average quality ratio for a map area of 100×100

Strategy	Distance(m)	$Q(m^2/m)$
VUHC	1104.86	9.05
VFHC	1150.45	8.69
UC	1239.18	8.06
DEGU	1278.00	7.82
VUHR	1318.58	7.58
FC	1366.80	7.31
VFHR	1374.42	7.27
DFC	1400.95	7.13
DEGF	1442.11	6.93
AM	1661.93	6.01
UR	2047.65	4.88
FR	2854.85	3.50

selection. Random strategies, such as **UR** and **FR** performed badly in comparison, having $4.88m^2$ and $3.50m^2$ ratio values, respectively.

Random algorithms (**FR**, **UR**, **VFHR**, **VUHR**) regularly exhibit significantly lower performance than their counterparts for full exploration, contradicting earlier findings [58]. Robots with randomized strategies explore faster initially due to high dispersion but quickly degrade in performance as significant time is spent in long-distance traversal between target cells.

3.4.2 Voronoi Search Time Improvement

Using Voronoi partitioning to create sub-areas results in better allocation of robot agents and significantly reduced large-area traversals between exploration cells. Even for the worst-performing **FR** algorithm, its counterpart strategy (**VFHR**) outperformed many peer algorithms. This finding shows that allocating sub-areas gives all strategies a clear speed advantage. After completing their individual sub-areas, agents with simpler room layouts quickly transitioned to assisting nearby areas, sharing the workload more evenly and contributing to quicker overall exploration.

While the quality ratio conveniently demonstrates performance, the completion time of exploration is another commonly used success indicator. Table 3.3 lists the average

Table 3.3: Exploration Time and Replan Count Metrics

	Time (Seconds)				Replan Count			
	mean	std	min	max	mean	std	min	max
VUHC	140.5	72.9	3	536	270.6	150.1	52	1370
VFHC	147.5	78.1	3	763	308.4	170.9	55	1352
UC	157.0	93.6	3	1002	300.1	183.8	51	1976
DEGU	165.5	93.7	3	729	226.2	111.3	47	934
VUHR	174.1	108.1	3	966	168.4	81.6	46	739
FC	175.2	117.3	3	1463	399.9	268.0	49	3024
VFHR	183.2	118.9	3	905	166.2	81.0	44	745
DFC	186.4	113.8	3	854	333.8	193.5	53	1621
DEGF	189.7	118.4	3	1228	319.3	194.6	49	1494
AM	220.1	138.5	3	1193	274.6	169.0	46	1317
UR	270.8	162.3	3	1099	176.4	83.5	50	720
FR	390.7	282.7	3	2278	245.3	148.2	47	1227

completion time for explorations and replanning count. Time for completion also benefited significantly from the Voronoi partitioning, even in the worst scenarios (i.e. max completion time).

In addition to the time spent on exploration, another important metric is the number of times agents must replan their routes during exploration. Among the different strategies we’ve analyzed, those based on random choices (**VFHR**, **FR**, **VUHR**, **UR**) tend to require fewer replans than greedy frontier algorithms (**VFHC**, **FC**, **VUHC**, **UC**, **DFC**).

3.4.3 Unknown Strategies Performance

Both **VUHC** and **UC** allow robots to reach full exploration faster than their **VFHC** and **FC** frontier counterparts. The superior performance arises mainly from two factors. First, the randomness inherent in selecting unknown cells ensures a better distribution of robots across the search area. Second, this approach retains the advantage of greedy selection for the closest points, much like frontier strategies. Additionally, unknown-cell targeting algorithms are more effective in mitigating leader-follower issues, a problem often encountered in frontier-based methods.

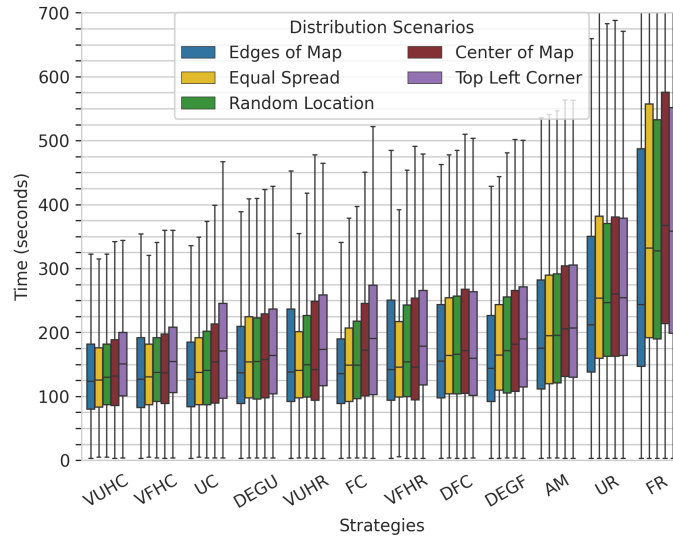


Fig. 3.4: Comparing the impacts of area allocation and distribution (initial navigation targets) over the exploration speed. Initial distribution scenarios do not greatly impact the speed of full exploration. The strategies that allocate sub-areas have a clear improvement in exploration time compared to their counterparts without segmentation.

3.4.4 Distribution Effect For Timing

To explore an environment more quickly, tests were conducted on the impact of the initial starting position of robots across the map (e.g., edges, center, or evenly spread). Figure 3.4 shows that initial robot distribution in a building does not significantly impact exploration speed. While a slight advantage exists when widely dispersing the start point to the "Edges of the Map", performance was minimally impacted.

3.4.5 Exploration Speed in Relation to Robot Team Size

Regardless of the exploration strategy, increasing the number of robots improves exploration speed. Figure 3.5 shows the average exploration performances of different robot group sizes. The linear increase in the number of robots (increasing by four) shows a marginal decrease in performance gain. Increases in team size when teams are small have a larger impact on the exploration speed, as observed by the time decreases more significantly when the team size increases from 4 to 8 compared to the increase from 8 to 12.

When the system has fewer robots, the choice of exploration strategy is more critical

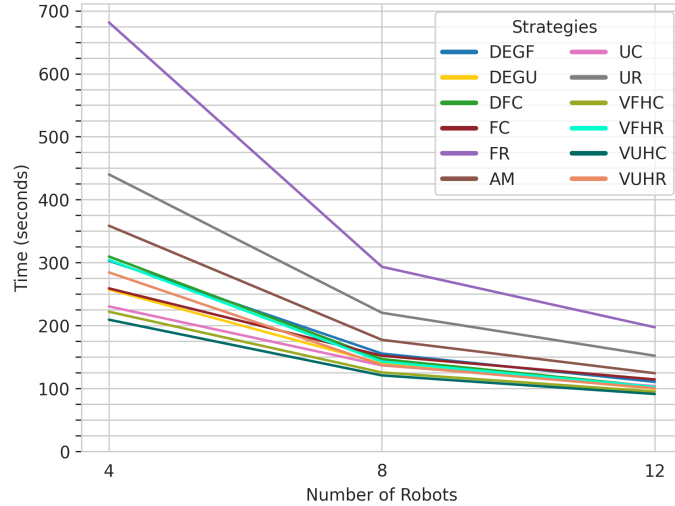


Fig. 3.5: Average exploration time related to robot count. As the number of robots increases, the exploration time reduces.

than large team sizes. Most strategies converge to similar performance when the robot count increases. This finding can be observed in Figure 3.5; when using four robots to explore an area, the average completion time difference between the best algorithm (VUHC) and the worst algorithm (FR) is around 475 seconds. However, if the robot count increases to 12, the performance difference between the two is only 100.

3.4.6 Resilience to Density of the Environment

Figure 3.6 illustrates that segmented strategies perform well across a range of room densities. Additionally, it can be observed that segmented algorithms are more resilient to wall density compared to their unsegmented counterparts.

When the room sizes are small (i.e. high density of rooms in the building), the range for completion time between strategies is quite significant. However, as rooms become larger (i.e. sparse populated map), the performance of all strategies becomes similar, blurring the benefits of exploration strategies. The performance difference between VFHC and FC is significant when the room size is 10, but this significance is largely unnoticeable when larger room sizes are used.

Figure 3.7 illustrates the replanning count change for strategies over time. Once the

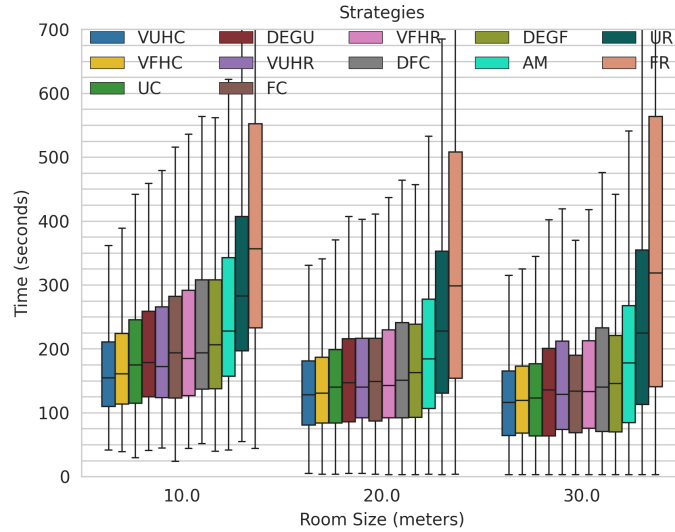


Fig. 3.6: Exploration completion time of the strategies against three different room sizes. A small number (10.0) represents many small rooms, resulting in a densely populated area with many walls. When the room sizes increase, the area between walls becomes more sparse.

area started to become explored around 200 seconds, the replan count slowed down with a curve and became almost straight, shown with points on the upper side of the plot. This stabilization reaches a point shown with a small circle indicating the replanning count becomes so minimal, correlated with the area percentage close to 100 percent exploration.

3.4.7 Resilience to Robot Failure Scenarios

We tested our exploration algorithms for mapping under the Agent scenario, which assumes that robots can continue functioning throughout the exploration task. We built robot failure scenarios (Disrepair and Unrecoverable), each running 81,000 times with the same set of parameters as the standard scenario, to assess the effectiveness of our techniques and other metrics under robot failures. The Disrepair condition allows other team members to come and rescue broken or non-functioning robots, while the Unrecoverable condition assumes that robots cannot be rescued or contacted once they stop functioning.

Out of 81,000 experiments, robots in Disrepair conditions completed the exploration in most cases. Only 130 experiments failed to be completed in the Disrepair condition (robots

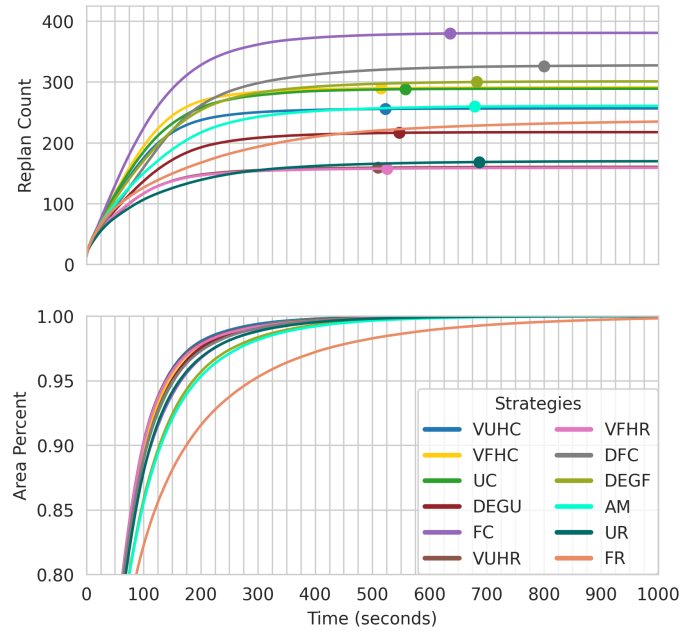


Fig. 3.7: Illustration of replan count over time on various exploration strategies. Each point in the upper plot represents the time when replan counts stabilize.

were all disabled before exploration was completed).

Robots in the Unrecoverable scenario experienced much higher failure to complete exploration. Of 81,000 experiments, only 27,341 experiments successfully completed. The overall success rate for the Unrecoverable scenario is significantly lower (33.7%) as compared to the Disrepair scenario (98.4%).

Figure 3.8a shows these failed experiments' distribution among exploration strategies. While randomized strategies are the worst, with high failure rates (78% and 82%), the best performers only fail 55% of the time.

Since there were many failed attempts for the unrecoverable failure scenario, it is worth investigating how robot counts and exploration strategies affect the percentage of explored areas, even when missions fail. Figure 3.8b shows the impact of robot count over the percentage of area exploration, despite all 53,659 experiments failing to reach 100% exploration. The y-axis ranges from 0.5 to 1.0, representing 50% to 100% of exploration. Increasing the robot count significantly increases the potential of known explored areas. Even using 4 robots allowed robots to explore more than 70% of the map in most experiments. All

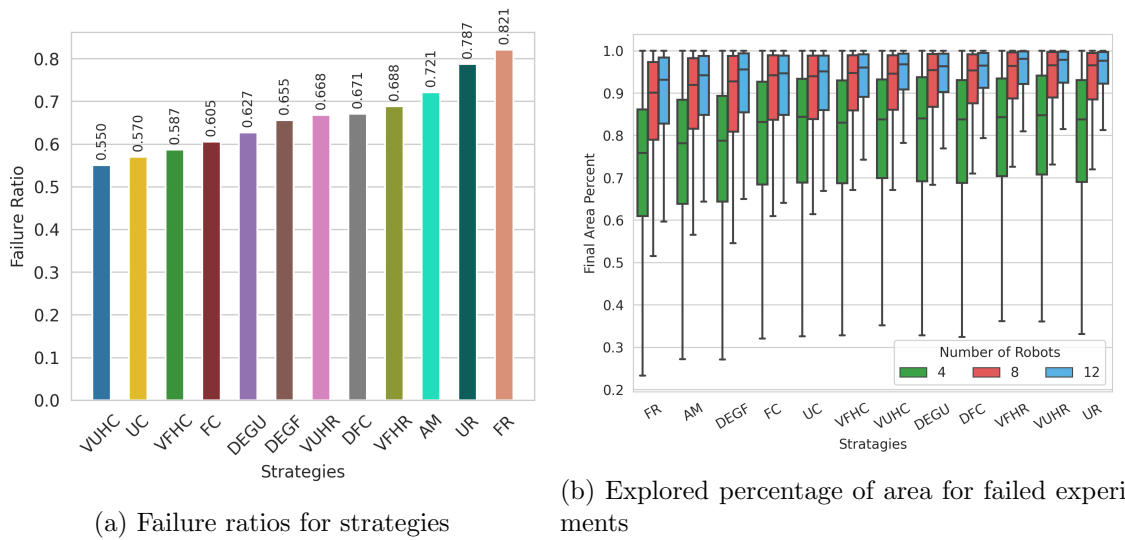


Fig. 3.8: (a) - The failure ratio of experiments based on strategies. While the random strategies failed more often, **VUHC** performed better than others. (b) - The explored area percentage for the failed experiments for the Unrecoverable failure scenario. Using 4 robots allows exploring more than 70% of the area in most experiments before all robots stop functioning. The failed experiments that used 12 robots even reached more than 95% of the area.

strategies except the **FR** strategy using 12 robots reach more than 95% exploration.

3.5 Conclusions

In this chapter, we delve into the complexities of exploring unknown and occluded spaces within buildings. The integration of Voronoi partitioning techniques, especially when combined with segmentation methods, proves instrumental in enhancing exploration efficiency, enabling faster coverage and optimized agent allocation. Randomized strategies, although providing initial dispersion faster, are not suitable for sustained, efficient exploration due to extensive long-distance traversals.

The research also highlights the importance of initial robot distribution, team size, and choice of exploration strategy. Smaller teams benefit significantly from appropriate strategy selection, emphasizing the need for tailored approaches based on team size. Furthermore, segmented strategies exhibit resilience across varying complexity of building plans, ensuring efficient exploration in diverse indoor environments.

We did not highlight the multiple agents' exploration performance under different communication constraints. In future work, the focus will shift towards robot coordination under different communication constraints. Exploring the impact of limited communication between agents only in a visibility scenario and even full communication loss (no communication) on multi-robot coordination strategies could provide valuable insights for the system behavior under complex and communication restricted environments.

CHAPTER 4

AGENT COORDINATION IN COMMUNICATION RESTRICTED DOMAINS

In the realm of autonomous robotics, the exploration of unknown and intricate spaces within buildings has been a subject of intense research focus. Previous studies have predominantly relied on techniques like frontier-based exploration, necessitating constant data sharing among robots to construct a global map. However, this approach becomes challenging, especially in indoor environments, due to the inherent communication overhead. The existing solutions, such as market-based strategies and decentralized approaches, although promising, often come with their own set of limitations. Coordinating multiple agents efficiently while managing communication restrictions poses a substantial challenge. Furthermore, real-world scenarios demand exploration strategies that can adapt to varying levels of communication allowances and account for scenarios where no communication is possible.

In light of these challenges, we investigate the multiple agent coordination behaviors under diverse communication constraints. Our approach leverages limited communication ranges, enabling robots to share crucial map data within their immediate vicinity. To navigate toward valuable frontiers, robots utilize an onboard agent position detection system, ensuring localized decision-making while optimizing exploration paths. Crucially, we mainly explore the intricate dynamics of multiple robot exploration in communication-restricted environments. We employ a range of communication scenarios, from complete visibility to no communication, to understand the impact on exploration strategies and agent coordination. Additionally, our investigation delves into the significance of area segmentation, where sub-regions are assigned to robots before exploration, providing a comprehensive understanding of its effects on exploration efficiency.

4.1 Earlier Coordination of Multiple Robots in Unknown Environments

Exploring unknown areas using multiple robots has received substantial attention within the research community [78–80]. One widely studied technique in this domain is frontier-based exploration [31, 81, 82], which was initially introduced by Yamauchi for single-agent scenarios [83] and later extended to accommodate multiple agents [30]. This approach necessitates each agent to share its local perception data, enabling the creation of a global map. However, this continuous data sharing results in a significant communication overhead, which can impact performance, particularly in indoor environments.

To alleviate the challenge of excessive map exchanges among agents, Zlot et al. proposed a market-based strategy that coordinates agents with minimal information transfers [84]. While this strategy improves agent distribution [85, 86], it requires knowledge of other agents’ positions in the building, which may not always be practical. Additionally, the computation of multiple combinations of pathways for bid exchanges introduces additional computation overhead.

Some of this computational overhead is relieved using loop closures to account for completed regions. However, due to the difficulty of finding mutual points for the perfect loop closures in unknown environments, this can be difficult even for two robots [87]. Other decentralized approaches focused on assigning independent regions to each robot to reduce the knowledge difficulty [36, 37]. However, these segmentation approaches still require agents to know others’ locations to partition the area [77], limiting their practicality in real-world situations. Instead, dividing the area based on the initial goal locations in the unknown environment overcomes this drawback since robots do not need to know others’ coordinates and only need to be informed about their respective allocated regions.

While various techniques have been developed to enhance frontier selections [31, 70, 88], their investigation has often been confined to specific communication scenarios [34] or particular building structures [32], resulting in findings that lack generalizability. Coordinating multiple agents to select optimal frontiers necessitates coordination and communication, which becomes challenging in areas with restricted outside communication infrastructure.

Agents in such environments often rely on localized communication devices [89], which assist decision-making systems for continuing exploration or meeting with others to share information [90]. Several decentralized studies have explored scenarios where agents share data only within their line of sight [91] or within limited communication bandwidth [92] as many buildings do not have readily accessible communication networks available to robots. However, these devices cannot be used in scenarios where the environment prevents all types of in-between communications [93], or any data should not be released in wireless communication.

In response to these challenges, we investigated unknown area exploration using multiple agents under varying communication allowances. Simulated robots operate under a limited communication range, share collected map data, and detect other agents internally to navigate toward valuable frontiers. Additionally, we explore the significance of area segmentation, where sub-regions are assigned to robots before the exploration. Various team sizes of robots in varying complexity of building maps has been used, which aims to shed light on the intricate dynamics of multiple robot exploration in communication-restricted environments.

4.2 Method

4.2.1 Communication Scenarios

We leverage the Unknown Building Exploration Simulator ([UBES](#)) simulation software [73] to observe the importance and effects of frontier-based exploration, next-frontier selection (incentivized via agent-detected agent positions), and area partitioning under different levels of communication restrictions. [UBES](#) was modified by adding limited and fully denied communication capabilities to the simulator. Simulations were conducted to assess the effectiveness of tested strategies in three levels of building layout complexity.

Oracle-Based Scenario

This communication scenario assumes an oracle (central entity) overseeing all agent

exploration activities. Agents communicate with the oracle, which, in turn, aggregates their information (explored map) and determines the exploration's completion status. This communication scenario is developed to imitate the behavior of a "broadcasting map" shared between agents [30].

Let A denote the area explored by an agent, M is the total map area that is being explored, and E_{oracle} represents the exploration status decided by an oracle. If the total explored area by all agents is equal to the total unknown map area, then exploration is terminated with the oracle.

$$E_{\text{oracle}} = \begin{cases} 1, & \text{if } \sum_{i=1}^n A_i == M \\ 0, & \text{otherwise} \end{cases}$$

Non-Line of Sight (NLOS) Scenario

In the **NLOS** scenario, agents operate fully decentralized without coordination or monitoring. Agents cannot communicate with an outside network provider or with each other during exploration, relying solely on their local sensor data to explore the map individually. This communication scenario tests how agents explore a building when it is not desired to wirelessly release any information, such as the map or location, during the operation. Wireless communication restrictions arise due to security concerns or environmental limitations.

Unlike the oracle scenario, **NLOS** transfers the "terminating experiment decision mechanism" to the individual robots. Every agent explores the area individually without receiving information from others and determines the exploration status, E_{nlos} . Once an agent determines that sufficient exploration is conducted, it will terminate its own exploration independent of any other potential agent, as it assumes that the exploration task has been completed.

$$E_{\text{nlos}} = \begin{cases} 1, & \text{if } \forall i \in \{1, 2, \dots, n\}, A_i == M \\ 0, & \text{otherwise} \end{cases}$$

Line of Sight (LOS) Scenario

In the LOS scenario, agents can only communicate within their line of sight, simulating real-world constraints where communication is possible if no obstacles obstruct the line of sight between agents. To decide if another agent is visually detected, agents scan their surroundings, covering a 360-degree radius within a 10-meter range. Information exchange occurs only when agents can identify each other within their visibility zones.

The exploration status, E_{los} , is determined similarly to E_{oracle} . However, a key distinction in the LOS scenario is that agents cannot broadcast their map to an oracle. Instead, they share their local observations through pairwise communication when one agent detects another within its visibility range.

These scenarios facilitate a systematic analysis of communication mechanisms in unknown building exploration, illuminating the nuanced strategies employed by agents under various communication constraints.

4.2.2 Frontier Incentivization Algorithm

To enhance exploration efficiency within unknown buildings, we utilized the closest frontier-based exploration as presented in [73] and retained it as our baseline strategy for comparison. Furthermore, we designed a frontier incentivization mechanism to direct agents toward frontiers to minimize unnecessary travel over already explored regions. It achieves this by an agent position detection system and uses those coordinates to assign values to the frontiers so that the potential encounter with another agent can be minimized. This reduces cases where leader-follower dynamics occur and more readily disperses agents.

Underlying this frontier incentivization mechanism contains two fundamental factors: the distance to a frontier and the angle between the frontier vector to the agent’s dispersion vector. These factors are instrumental in shaping the robot’s decision-making process, allowing it to explore the unknown terrain effectively.

The incentivization technique we developed is encapsulated in Algorithm 2 and is illustrated in Figure 4.1. When a robot reaches a previous frontier goal and detects one or multiple agents, it calculates new frontier values with this incentivization mechanism.

Algorithm 2: Frontier Incentivization Algorithm

```

1 Input: Frontiers detected  $\mathbf{F}$  for an agent  $i$ , list of detected agents  $\mathbf{A}_i$  within
   radius  $r$ 
2 Output: List of frontiers with incentivized values  $F_u$ 
3 Define function DispersionVector( $\mathbf{A}_i, d_i$ )
4   If  $\mathbf{A}_i == 1$ :
5     Return (coord_x_ $\mathbf{A}_i$ , coord_y_ $\mathbf{A}_i$ )
6   Elif  $\mathbf{A}_i > 1$ :
7     For each detected agent  $a$  in  $\mathbf{A}_i >$ :
8        $i_x, i_y += [a_x \text{dist}, a_y \text{dist}]$ 
9     End For
10    Return ( $i_x, i_y$ )
11 Initialize empty list  $F_u$ 
12 For each agent  $i$  do:
13   Initialize:  $n \leftarrow 0, D \leftarrow \text{DispersionVector}(i), f \leftarrow 0$ 
14   For each frontier  $f$  in  $F$  do:
15      $n \leftarrow n + 1$ 
16      $d \leftarrow \text{frontier\_vector}(f_x, f_j)$ 
17      $\theta \leftarrow \text{angleToDispersionVector}(D, d)$ 
18      $f \leftarrow (1 / \text{dist}_f) * \cos(\theta)$ 
19      $F_u.\text{append}(f)$ 
20   End For
21   If  $n = 1$  then:
22      $f \leftarrow (f_x, f_y)$ 
23   End For
24 Return  $F_u$ 

```

Based on the detected locations of other agents, a dispersion vector is calculated by adding each calculated vector to each other. For a single known agent, this would be a vector pointing 180 degrees away from that agent. The impact of a detected agent's proximity is inversely correlated with its distance, meaning that closer agents exert a more significant influence on the selection of the next explored frontier.

Each frontier is assigned a value when an agent is detected, and the incentivization algorithm takes advantage of the dispersion vector by checking the angle between the frontier vector (the vector between a robot and a frontier) and the dispersion vector of the robot. Next, a cosine function is applied to assign higher values to angles closer to the dispersion vector. The weighted angle and distances to the frontier are combined to generate a frontier

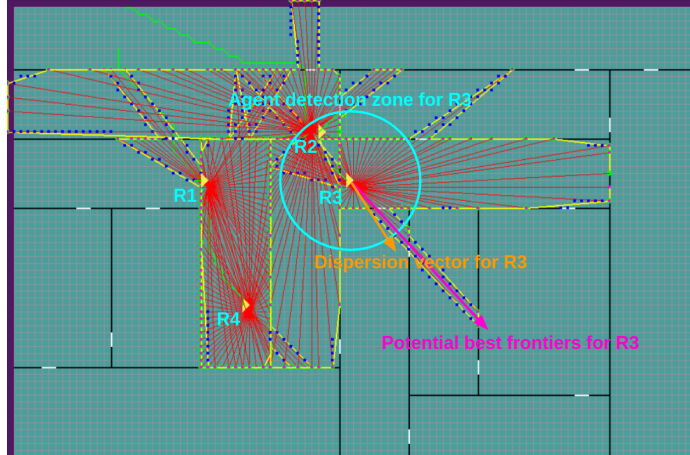


Fig. 4.1: A portion of the simulation interface shows four agents exploring an unknown building. The area gets explored while agents scan their environments with 360-degree LiDAR. We propose a frontier incentivization technique, illustrated in a simple example that two agents cannot see each other, and R3 uses R2’s location data to incentivize its frontiers accordingly.

exploration value for each frontier. The agent chooses to move toward the frontier with the highest exploration value. This design choice empowers agents to prioritize frontiers with smaller angles relative to their optimal direction, improving the exploration process.

4.2.3 Simulation Setup

To implement our exploration strategies effectively, we established a simulation setup characterized by homogeneous robots, each possessing identical capabilities such as sensors, communication mechanisms, speed, and size. Each robot was equipped with a simulated LiDAR sensor for obstacle detection and exploration, and the agents’ local maps were iteratively updated with these LiDAR scans, forming the foundation of our exploration framework.

Robots started their exploration from a designated entrance on the map (top left corner) in every experiment performed, and no initial goals were given. However, only in scenarios where Voronoi partitioning was employed were robots allocated initial goal locations due to the necessity of partitioning the environment based on the goal locations. This approach ensured a systematic division of the unknown map among the participating agents and

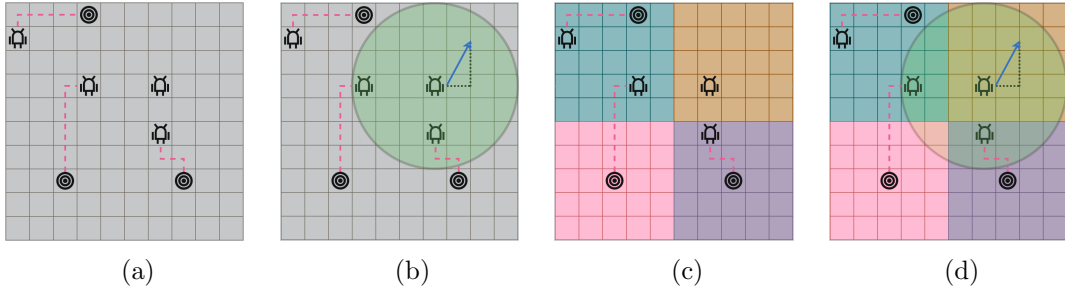


Fig. 4.2: Illustration of the four techniques evaluated in simulation. While robots in (a) and (c) do not utilize frontier incentivization via agent detection, (a) and (b) do not consider using voronoi sub-partitioning. Robots in technique (d) uses both agent detection and voronoi partitioning and (a) only uses frontier-based exploration without any upper parameters.

significantly contributed to the optimization of completion times.

Frontier Incentivization and Voronoi Partitioning

To comprehensively explore the interplay between communication scenarios and coordination techniques (frontier incentivization and Voronoi partitioning), we introduced two boolean arguments that allow certain experiment benefits using these techniques or not. These arguments are "agent detection" and "Voronoi partitioning". By toggling these parameters on and off, we systematically measured each of these techniques' importance under the listed communication scenarios.

Figure 4.2 demonstrates the usage of these techniques from one agent's perspective. Whenever an agent reaches a selected destination over the map, it chooses the next destination (a frontier) using one of these techniques illustrated in the subfigures. Subfigure 4.2a shows the baseline frontier-based exploration that does not use agent detection and Voronoi partitioning. An agent is shown to have reached its goal and picks the next frontier based on the distances of frontiers, and the closest one is chosen. This serves as a benchmark for our assessment of the effectiveness of frontier incentivization and Voronoi partitioning under different communication constraints.

Subfigure 4.2b shows the usage of frontier incentivization when an agent detects two other agents within its limited sensor range (shown in green). The dispersion vector has

been illustrated with a blue arrow, and the agent decides its next frontier destination by calculating the values of its known frontiers as explained in Algorithm 2. Similar to subfigure 4.2a, subfigure 4.2c also picks the next frontier similarly using frontier-based exploration. However, the picked frontier needs to be in its assigned sub-region. Last, subfigure 4.2d shows the usage of both frontier incentivization and Voronoi partitioning, which is the combination of both techniques. The agent remains restricted to picking the next goal in its assigned subregion but it can also incentivize itself towards frontiers when it detects others.

Robot Numbers and Room Density of Buildings

The synergy between these strategies and their performance under different team sizes (4, 6, 8) allowed us to explore the nuanced influence of agent numbers on exploration outcomes.

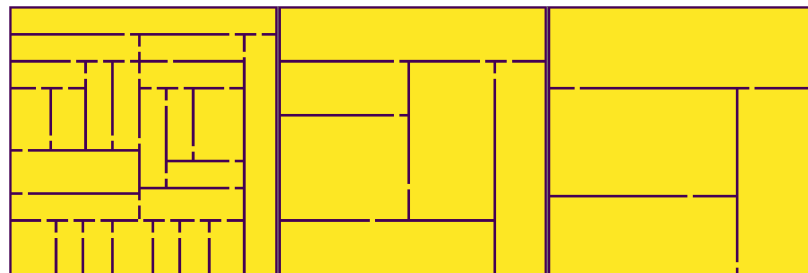


Fig. 4.3: Three examples of building floor plans were generated using minimum room sizes of 10, 20, and 30, from left to right, respectively.

Each unknown map, spanning 100 by 100 meters, was designed to encapsulate different complexities, as illustrated in Figure 4.3. These complexities were derived from automatic floor plan generation based on minimum room sizes, ensuring a diverse array of environments for our experiments. We generated three distinct maps for each complexity to provide generalizable findings, ensuring consistent experimental conditions while exploring various scenarios, strategies, and team sizes.

4.3 Results

It is logical to evaluate the impacts of agent detection and Voronoi segmentation techniques individually within each scenario. This approach allows for a comprehensive understanding of the effectiveness of these strategies in varying communication contexts. In the subsequent subsections, we delve into the effects of these techniques under each communication scenario.

4.3.1 Oracle Based Monitoring Scenario

Table 4.1 provides a detailed overview of the average completion times, including standard deviations and minimum-maximum completion times, for experiments conducted with an oracle-based communication scenario. The oracle monitored each experiment, which sent termination signals to the agents once it determined that the area was fully explored. We note that the agents cannot access the oracle’s map.

Table 4.1: Exploration performance in Oracle Scenario

Agent Detection	Voronoi Partitioning	Mean	Std	Min	Max
False	False	266.59	115.72	110.0	506.0
True	False	189.93	87.21	92.0	345.0
False	True	136.96	34.14	90.0	217.0
True	True	133.44	35.80	87.0	234.0

Our observations revealed a key trend: incentivizing frontier exploration based on detected locations led to a dispersion effect among the robots. This dispersion effect is because the robots were motivated to maintain distance from each other during exploration, thus ensuring the exploration of diverse areas. The experiments in this communication scenario significantly benefited from this incentivization technique as each agent was motivated to explore a different part of the map, and the oracle monitored these individual experiments to send a termination signal once it decided the area was fully explored. Our findings strongly support the efficacy of frontier incentivization, showcasing significant improvement compared to experiments that do not use it.

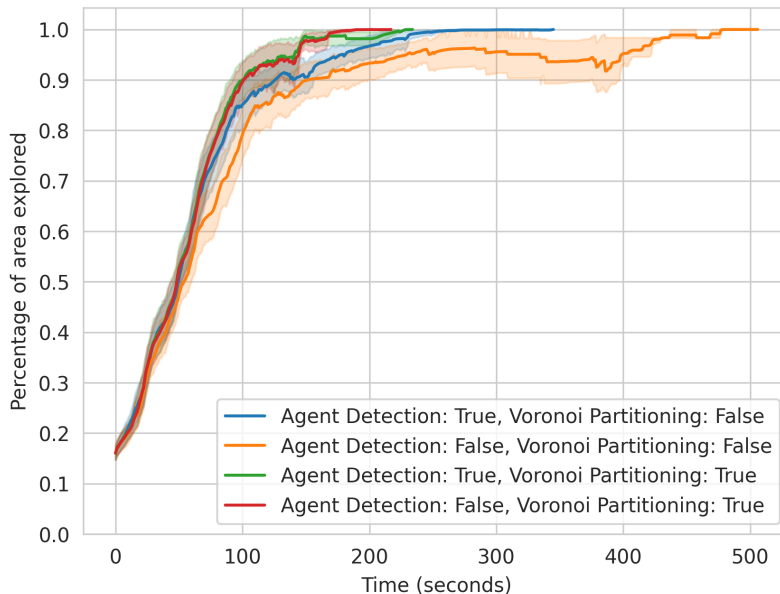


Fig. 4.4: Average area exploration percentage over time in the Oracle communication scenario. This compares the effects of agent detection and Voronoi partitioning in an unknown area. Once the map is fully explored, the oracle terminates the exploration.

Frontier algorithms perform significantly faster when agents explore independent sub-areas generated by Voronoi partitioning. Allowing agents to operate within these Voronoi subregions led to an average completion time of 136.96 seconds. While combining Voronoi partitioning and agent detection enhanced the completion times, the impact was not substantial. This lack of significant effect can be attributed to the conflicting nature of the behaviors exhibited by these techniques, both of which aim to disperse robots across unknown areas. Although assigning individual subsections of an unknown map resulted in faster completion times on average, adding the agent detection mechanism did not yield substantial benefits when an Oracle was present.

The impact of frontier incentivization on regular Frontier-Based exploration becomes apparent when examining Figure 4.4. Similarities in the experiments persisted until approximately 60 percent of the total area was explored. After this point, frontier incentivization came into play, motivating agents toward frontiers that facilitated their dispersion. Voronoi strategies exhibited similarities regardless of the use of agent detection. This can be attributed to Voronoi partitioning’s ability to distribute agents effectively, rendering agent

detection less influential in encouraging their dispersion.

4.3.2 NLOS (Non-Line of Sight) Scenario

This subsection delves into the results obtained under the **NLOS** scenario. Table 4.2 provides a comprehensive overview of the average completion times for experiments conducted in **NLOS** scenarios. These experiments were conducted without communication, requiring agents to rely solely on local decision systems based on their locally collected information to terminate exploration.

Table 4.2: Exploration performance in **NLOS**

Agent Detection	Voronoi Partitioning	Mean	Std	Min	Max
False	False	499.93	188.92	263.0	791.0
True	False	861.19	540.43	320.0	2239.0
False	True	256.07	137.39	92.0	557.0
True	True	224.52	104.47	89.0	443.0

Without an oracle or any communication mechanism, agents operated within a local decision framework, necessitating independent exploration of the entire area. Surprisingly, introducing an agent detection system in this scenario had a detrimental effect, leading to a 72% decrease in performance compared to regular frontier-based exploration. This negative impact can be attributed to the conflicting goals of frontier incentivization and experiment termination requirements in **NLOS**.

However, experiments employing Voronoi partitioning still demonstrated significant improvement in completion times, even in **NLOS** scenarios, as detailed in Table 4.2. The noteworthy performance achieved in these experiments was mainly due to each agent being assigned specific subregions to explore independently. Combining Voronoi partitioning with agent detection yielded the best completion times in the **NLOS** scenario.

The exploration progress over time is visually represented in Figure 4.5. The plot reveals that agents utilizing agent detection without Voronoi partitioning explored the slowest. In contrast, experiments employing Voronoi partitioning achieved full exploration more

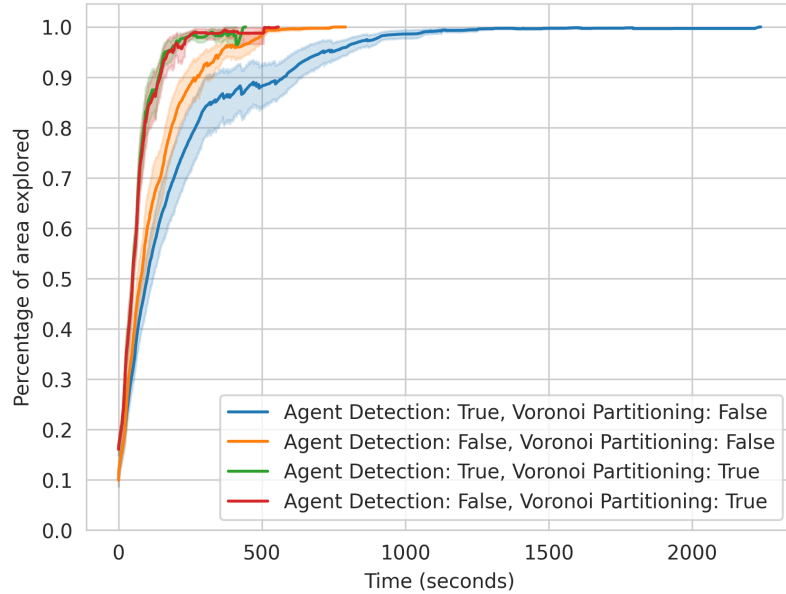


Fig. 4.5: Average area exploration percentage in time for the experiments in **NLOS** scenarios.

rapidly, even in the early phases. This accelerated exploration can be attributed to the dispersion effect facilitated by Voronoi partitioning and the allowance for agents to explore only specific subregions of the overall map.

4.3.3 LOS (Line of Sight) Scenario

Due to limited information transfer in pairwise communications, the **LOS** setting allows agents to achieve expedited exploration times. Our primary focus is to examine the additional performance gains enabled by significantly limited communication between agents, specifically in the context of Voronoi partitioning and agent detection techniques.

Table 4.3: Exploration performance in **LOS**

Agent Detection	Voronoi Partitioning	Mean	Std	Min	Max
False	False	352.33	133.65	172.0	613.0
True	False	307.11	128.27	138.0	560.0
False	True	240.37	115.76	93.0	539.0
True	True	212.96	90.22	91.0	402.0

Experiments conducted in **LOS** scenarios exhibited an average of significantly improved exploration completion times compared to their **NLOS** counterparts. Table 4.3 provides a detailed breakdown of the average completion times for each strategy, presenting associated standard deviations and minimum-maximum completion times. Notably, agent detection positively influenced performance by 15% compared to scenarios where agent detection was not employed. This contrasts with our observations in **NLOS** scenarios, underlining the pivotal role of agent detection in enhancing exploration efficiency when direct communication channels are available.



Fig. 4.6: Average area exploration percentage in time for the experiments in **LOS** scenarios.

Figure 4.6 visually represents exploration progress over time. Interestingly, the plot reveals a striking similarity between the exploration paths of agent detection and no detection strategies, especially until 75 percent of the area is explored. This suggests that agent detection mechanisms might inadvertently redirect agents in disparate directions in later phases of exploration, hindering effective communication among them. In contrast, scenarios without agent detection exhibit superior performance due to the increased likelihood of inter-agent communication.

However, the most compelling results emerge from the combined implementation of Voronoi partitioning and agent detection. Despite Voronoi partitioning being the dominant mechanism, the synergy with agent detection yields the best exploration times in **LOS** scenarios. This integrated approach maximizes the benefits of Voronoi partitioning and mitigates the communication challenges posed by agent detection, resulting in optimal exploration efficiency.

These findings underscore the balance required in **LOS** scenarios considering the earlier context. While communication between agents offers advantages, effective coordination mechanisms, such as the synergy between Voronoi partitioning and agent detection, are vital for harnessing its full potential. These insights provide valuable guidance for optimizing exploration strategies in scenarios with direct inter-agent communication.

4.3.4 Agent Team Size and Map Complexity Impacts

The relationship between agent team size and environmental complexity is a significant aspect of this chapter. We highlight the impacts of these factors on exploration efficiency. Room complexity and agent count are critical, mainly when communication constraints exist, as they necessitate robust, decentralized maps. These maps fundamentally influence the exploration performance, and the following discussion will treat the number of agents and room complexity separately.

Impact of Number of Robots

Analyzing the performance across diverse robot team sizes reveals consistent **LOS** and Oracle-based scenario improvements. However, in the **NLOS** scenario, the trend remains relatively stable across different agent sizes, as depicted in Figure 4.8. The lack of inter-agent communication, including termination signals, mandates that each agent independently covers the entire area, leading to a uniform performance irrespective of team size. Notably, agents assigned to Voronoi partitioned subareas exhibit remarkable success rates even within the constraints of **NLOS** scenarios.

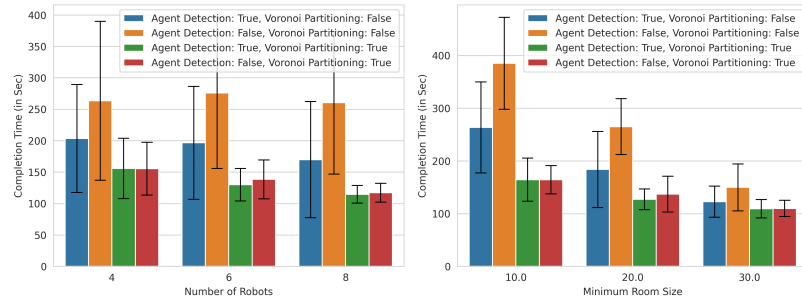


Fig. 4.7: Oracle Scenario: Monitoring agents overall in the background to terminate exploration.

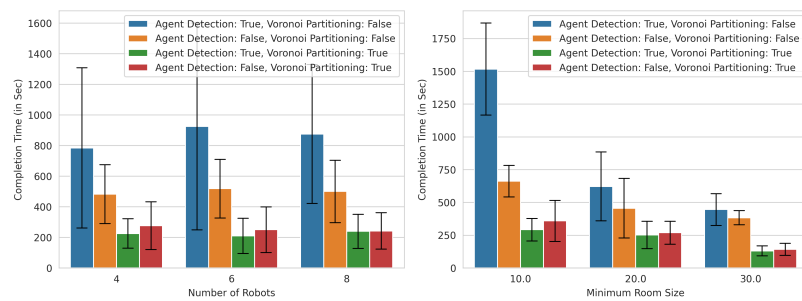


Fig. 4.8: **NLOS**: No type of communication allowed between agents. Local decisions have been taken by agents.

Introducing line-of-sight communication within buildings (**LOS** scenario) offers a notable advantage for every agent size compared to **NLOS** conditions, especially for agents employing agent detection systems. A comparison of completion times between Figure-4.8 (no communication) and Figure-4.9 (limited communication) underscores this shift. In **NLOS** scenarios, teams relying solely on agent detection systems typically experience completion times of around 800 seconds for every agent size. However, enabling limited communication in **LOS** scenarios significantly enhances exploration performance by reducing completion times to approximately 300 seconds. Remarkably, the success rates achieved by agent detection systems under limited communication align closely with those observed in oracle-based scenarios (Figure 4.7).

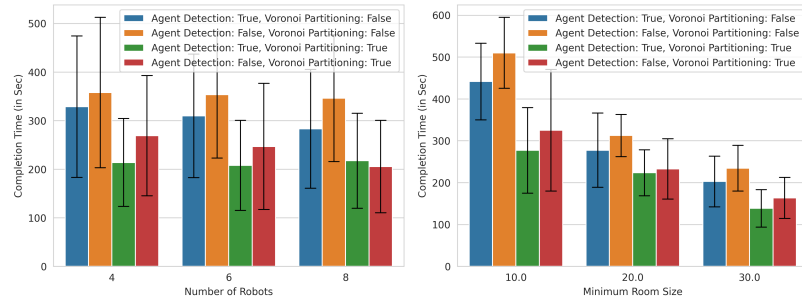


Fig. 4.9: **LOS**: Limited communication allowed only in line of sight scenarios. Local decisions have been taken by agents.

Impact of Environmental Complexity

Considering environmental complexities, such as varying room sizes and densities, elucidates interesting patterns in exploration efficiency. As expected, environments comprising numerous small rooms pose challenges due to the limited reach of **LiDAR** beams, resulting in prolonged exploration times. This is evident in the plots on the right side of Figure 4.7, 4.8, and 4.9, which illustrate agent performance across three distinct wall densities. In these scenarios, larger room sizes translate to superior exploration efficiency.

Agents face substantial challenges in scenarios characterized by small rooms, such as those with dimensions of 10. Particularly in **NLOS** conditions, the exploration timings are notably suboptimal, specifically for the frontier incentivization technique. As illustrated in the right side of Figure 4.8, completion times range from 700 to 1500 seconds when the room sizes are set to 10 for the strategies that do not employ Voronoi partitioning. These extended durations highlight agents' difficulty navigating confined spaces without direct line-of-sight communication capabilities.

However, it is worth pointing out that experiments employing Voronoi partitioning and agent detection consistently demonstrate optimal performance across diverse environmental complexities. Even in densely populated maps, where exploration is inherently challenging, Voronoi partitioning significantly influences completion times. Introducing limited communication in **LOS** scenarios further amplifies this effect, tripling the performance, especially for agents utilizing agent detection systems.

CHAPTER 5

Conclusion and Discussion

This dissertation demonstrates our core algorithmic contributions to the field of robotics. We start presenting [PMVC](#), an algorithm designed to achieve full city roadway coverage using multiple [UAV](#) drones concurrently. [PMVC](#) operates on known road graphs, identifying equal-sized regions suitable for various drone capacities. Through simulated tests on three cities, the algorithm demonstrates its scalability for city-wide drone deployment. By analyzing different drone types and their costs, it has been shown increasing the drone count significantly reduces road coverage time. [PMVC](#) enables city planners to assess trade-offs between drone numbers, capabilities, and costs, which suggests obtaining more low-capability drones can be cost-effective for rapid city-wide road coverage, allowing cities to tailor their drone systems based on specific needs and budget constraints.

Further studies focus on exploring interior but closed-shaped bounded domains (buildings) in various complexities when agents are allowed to broadcast information to a central communication server, which is responsible for merging maps and sharing those with agents. Twelve exploration strategies for multi-robot systems in complex, unknown indoor environments were evaluated, considering variables like robot count, map types, and failure conditions. While frontier-based strategies showed effectiveness initially, they exhibited limitations in full exploration scenarios due to oscillatory behaviors. Segmentation-based approaches speed up exploration and enhance resilience against failures, maintaining utility throughout the exploration process.

Later, the complexities of robotic exploration were systematically investigated across various team sizes and communication scenarios using closest frontier-based exploration, frontier incentivization, and Voronoi partitioning techniques. Frontier incentivization, based on detecting other agents' locations, significantly improved exploration effectiveness, especially in limited communication setups, although it faced challenges in ([NLOS](#) - no com-

munication) situations. On the other hand, Voronoi partitioning consistently demonstrated substantial performance improvements across communication scenarios by dividing the unknown area into manageable sub-maps, ensuring robust exploration outcomes even without inter-agent communication. Voronoi partitioning emerges as a universally effective tool for autonomous robotic exploration, offering consistent and reliable results, and providing a foundational approach for the development of more refined and efficient exploration systems.

The challenge of coordinating decentralized agents in unknown environments with limited communication capabilities is addressed with a frontier incentivization technique that provides implicit coordination among agents. Exploring an unknown building optimally is quite challenging, and frontier incentivization achieves close-to-optimal performance without significant computational overhead. When an oracle is present to act as a central agent to collect and distribute maps, the incentivization technique could add a significant benefit to regular frontier exploration due to its distribution effect, and it even improves the VFHC algorithm with a slight edge. However, it shows its true potential when the oracle is not present, and agents are only allowed to communicate with each other in LOS. In this case, the proposed method outperforms the other two methods by a significant margin. This highlights the effectiveness of the proposed approach in addressing the coordination challenges in both oracle-based and LOS scenarios.

Overall, we developed techniques in this dissertation to better coordinate agents in uncertain and unknown situations, communication-restricted domains, and fixed/unknown geometries. Uncertainty in area exploration could occur due to the potential of losing robots in buildings, potential communication failures, and the possibility of landing agents in different locations. Both known geometries such as city road networks, and unknown areas such as building exploration could be used as a baseline to explore or cover with the strategies we employed. It is worth noting that agents' decision to terminate exploration is based on the assumption that the outside boundary is known in advance.

For the simulation experiments, we implemented many software components such as

a simulation framework for unknown building exploration, created a front-end application, parallel experiment runs for faster experiment trials, core exploration strategies such as frontier exploration, random walk, and explorations under voronoi partitioned sub-regions. We made the code publicly available as open-source [73]. This software could ease the process of testing new strategies in unknown building exploration since it includes many core techniques of exploration and automatic map generation in different complexities.

Many of the techniques we developed in this dissertation bring valuable pieces into the field and can be adapted into challenges such as DARPA Subterranean (SubT) Challenge [94]. Since this challenge focuses on developing new approaches to rapidly map, navigate, search, and exploit complex underground environments such as human-made tunnel systems, urban underground, and natural cave networks; the techniques we developed specifically for unknown environments could be used as a baseline in the future research to ease the development process.

REFERENCES

- [1] G. Lee, K. Cho, and J.-H. Kim, “Multi-robot simultaneous localization and mapping: A review,” in *Robotics and Autonomous Systems*, vol. 90, 2017, pp. 90–103.
- [2] M. Guzek and W. Szykiewicz, “Multi-robot exploration strategies: A review,” *Robotics and Autonomous Systems*, vol. 131, p. 103628, 2020.
- [3] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005.
- [4] J. Ko, B. Song, and K. Cho, “A survey of vision-based formation control of multiple unmanned aerial vehicles,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 1, pp. 4–20, 2007.
- [5] M. Schwager, J. Djugash, J.-J. E. Slotine, and D. Rus, “Coordinated multi-robot exploration,” *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 726–736, 2011.
- [6] C. Vasile, C. Birsan, M. R. Dogar, J. Spletzer, and D. Rus, “Adaptive informative path planning for multiple robots with costly observations,” *The International Journal of Robotics Research*, vol. 36, no. 10, pp. 1051–1068, 2017.
- [7] J. Vilela, Y. Liu, and G. Nejat, “Semi-autonomous exploration with robot teams in urban search and rescue,” in *2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. IEEE, 2013, pp. 1–6.
- [8] C. Sampedro, A. Rodriguez-Ramos, H. Bavle, A. Carrio, P. de la Puente, and P. Campoy, “A fully-autonomous aerial robot for search and rescue applications in indoor environments using learning-based techniques,” *Journal of Intelligent & Robotic Systems*, vol. 95, pp. 601–627, 2019.
- [9] B. Lindqvist, C. Kanellakis, S. S. Mansouri, A.-a. Agha-mohammadi, and G. Nikolakopoulos, “Compra: A compact reactive autonomy framework for subterranean map based search-and-rescue operations,” *Journal of Intelligent & Robotic Systems*, vol. 105, no. 3, p. 49, 2022.
- [10] S. Hayat, E. Yanmaz, T. X. Brown, and C. Bettstetter, “Multi-objective uav path planning for search and rescue,” in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 5569–5574.
- [11] H. Huang, A. V. Savkin, and C. Huang, “Decentralized autonomous navigation of a uav network for road traffic monitoring,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 4, pp. 2558–2564, 2021.
- [12] H. Kocabas, C. Allred, and M. Harper, “Divide and survey: Observability through multi-drone city roadway coverage,” in *2022 IEEE International Smart Cities Conference (ISC2)*. IEEE, 2022, pp. 1–7.

- [13] M. Diwanji, S. Hisvankar, and C. Khandelwal, "Autonomous fire detecting and extinguishing robot," in *2019 2nd International Conference on Intelligent Communication and Computational Techniques (ICCT)*. IEEE, 2019, pp. 327–329.
- [14] T. Ghosh Mondal, M. R. Jahanshahi, R.-T. Wu, and Z. Y. Wu, "Deep learning-based multi-class damage detection for autonomous post-disaster reconnaissance," *Structural Control and Health Monitoring*, vol. 27, no. 4, p. e2507, 2020.
- [15] M. S. Bahraini, A. Zenati, and N. Aouf, "Autonomous cooperative visual navigation for planetary exploration robots," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 9653–9658.
- [16] S. Azimi, E. Zemler, and R. Morris, "Autonomous robotics manipulation for in-space intra-vehicle activity," in *Proceedings of the ICAPS Workshop on Planning and Robotics*, vol. 3, 2019.
- [17] Y. Gao and S. Chien, "Review on space robotics: Toward top-level science through space exploration," *Science Robotics*, vol. 2, no. 7, p. eaan5074, 2017.
- [18] I. Colomina and P. Molina, "Unmanned aerial systems for photogrammetry and remote sensing: A review," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 92, pp. 79–97, 2014.
- [19] A. Schmuck, M. Broxvall, and A. J. Lilienthal, "Distributed robotic mapping and exploration in chemical plumes: a comparative study," *Frontiers in Robotics and AI*, vol. 6, p. 113, 2019.
- [20] T. Fraichard and O. Aycard, "Shortest paths for a car-like robot: A practical approach," in *Proceedings of the IEEE*, vol. 87, no. 7, 2018, pp. 1201–1212.
- [21] G. A. Hollinger and V. Isler, "Sampling-based multi-robot path planning," in *Robotics: Science and Systems*, 2014.
- [22] A. U. H. Qureshi, I.-H. Lee, and C.-Y. Yoon, "A survey on multi-robot systems: Communication, coordination, and control," *Information Sciences*, vol. 546, pp. 126–147, 2021.
- [23] A. Hernandez, A. Sanfeliu, and J. Andrade-Cetto, "Information gain maximization for active visual slam," *Robotics and Autonomous Systems*, vol. 76, pp. 39–49, 2016.
- [24] J. Yang, X. Li, H. Yan, H. He, L. Yuan, and P. Zhang, "Multi-robot exploration strategy with voronoi diagram in unknown environments," *Robotics and Autonomous Systems*, vol. 123, pp. 16–26, 2019.
- [25] H. Wang, H. Ma, Y. Jia, L. Zhang, X. Wei, H. Zhang, and H. Su, "Multi-robot exploration strategy in unknown environments based on the improved gaussian potential field model," *Sensors*, vol. 20, no. 7, 2020.
- [26] S. X. Yang, H. Ma, R. Xiong, J. Yang, and M. Song, "Efficient decentralized multi-robot exploration under communication constraints," *Robotics and Autonomous Systems*, vol. 115, pp. 18–29, 2019.

- [27] H. Chung and S. Lee, “A survey on multi-robot systems under localization error,” *Robotics and Autonomous Systems*, vol. 99, pp. 76–92, 2018.
- [28] A. Crespi, V. Lippiello, and B. Siciliano, “Cooperative target tracking with a group of mobile robots with limited sensing capabilities,” *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1674–1686, 2013.
- [29] C. Ostafew, B. Hiebert, B. Englot, S. Manjanna, and M. Broucke, “Secure and scalable multirobot coordination and control,” *Journal of Field Robotics*, vol. 32, no. 3, pp. 306–326, 2015.
- [30] B. Yamauchi, “Frontier-based exploration using multiple robots,” in *Proceedings of the second international conference on Autonomous agents*, 1998, pp. 47–53.
- [31] K. M. Han and Y. J. Kim, “Autoexplorer: Autonomous exploration of unknown environments using fast frontier-region detection and parallel path planning,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 10 536–10 541.
- [32] A. Soni, C. Dasannacharya, A. Gautam, V. S. Shekhawat, and S. Mohan, “Multi-robot unknown area exploration using frontier trees,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 9934–9941.
- [33] W. Burgard, M. Moors, and F. Schneider, “Collaborative exploration of unknown environments with teams of mobile robots,” in *Advances in Plan-Based Control of Robotic Agents: International Seminar Dagstuhl Castle, Germany, October 21–26, 2001 Revised Papers*. Springer, 2003, pp. 52–70.
- [34] W. Burgard, M. Moors, C. Stachniss, and F. E. Schneider, “Coordinated multi-robot exploration,” *IEEE Transactions on robotics*, vol. 21, no. 3, pp. 376–386, 2005.
- [35] N. Jadhav, M. Behari, R. Wood, and S. Gil, “Multi-robot exploration without explicit information exchange,” in *In Preparation*, In Preparation.
- [36] K. M. Wurm, C. Stachniss, and W. Burgard, “Coordinated multi-robot exploration using a segmentation of the environment,” in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2008, pp. 1160–1165.
- [37] A. C. Kapoutsis, S. A. Chatzichristofis, and E. B. Kosmatopoulos, “Darp: divide areas algorithm for optimal multi-robot coverage path planning,” *Journal of Intelligent & Robotic Systems*, vol. 86, pp. 663–680, 2017.
- [38] B. Galkin, J. Kibilda, and L. A. DaSilva, “Coverage analysis for low-altitude uav networks in urban environments,” in *GLOBECOM 2017-2017 IEEE Global Communications Conference*. IEEE, 2017, pp. 1–6.
- [39] M. A. Khan, B. A. Alvi, A. Safi, and I. U. Khan, “Drones for good in smart cities: a review,” in *Proc. Int. Conf. Elect., Electron., Comput., Commun., Mech. Comput.(EECCMC)*, 2018, pp. 1–6.

- [40] H. P. D. Nguyen and D. D. Nguyen, “Drone application in smart cities: The general overview of security vulnerabilities and countermeasures for data communication,” *Development and Future of Internet of Drones (IoD): Insights, Trends and Road Ahead*, pp. 185–210, 2021.
- [41] H. Menouar, I. Guvenc, K. Akkaya, A. S. Uluagac, A. Kadri, and A. Tuncer, “Uav-enabled intelligent transportation systems for the smart city: Applications and challenges,” *IEEE Communications Magazine*, vol. 55, no. 3, pp. 22–28, 2017.
- [42] X. Li, M. C. Chuah, and S. Bhattacharya, “Uav assisted smart parking solution,” in *2017 international conference on unmanned aircraft systems (ICUAS)*. IEEE, 2017, pp. 1006–1013.
- [43] H. Huang and A. V. Savkin, “A method of optimized deployment of charging stations for drone delivery,” *IEEE Transactions on Transportation Electrification*, vol. 6, no. 2, pp. 510–518, 2020.
- [44] S. Lloyd, “Least squares quantization in pcm,” *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.
- [45] F. Aurenhammer, “Voronoi diagrams—a survey of a fundamental geometric data structure,” *ACM Computing Surveys (CSUR)*, vol. 23, no. 3, pp. 345–405, 1991.
- [46] T. Ai, W. Yu, and Y. He, “Generation of constrained network voronoi diagram using linear tessellation and expansion method,” *Computers, Environment and Urban Systems*, vol. 51, pp. 83–96, 2015.
- [47] OpenStreetMap contributors, “Planet dump retrieved from <https://planet.osm.org> ,” <https://www.openstreetmap.org>, 2017.
- [48] J. Edmonds and E. L. Johnson, “Matching, euler tours and the chinese postman,” *Mathematical programming*, vol. 5, no. 1, pp. 88–124, 1973.
- [49] M. Kerekrety, “everystreet algorithm,” <https://github.com/matejker/everystreet>, 2020.
- [50] C. Stachniss, G. Grisetti, and W. Burgard, “Information gain-based exploration using rao-blackwellized particle filters.” in *Robotics: Science and systems*, vol. 2, 2005, pp. 65–72.
- [51] J. De Hoog, A. Jimenez-Gonzalez, S. Cameron, J. R. M. de Dios, and A. Ollero, “Using mobile relays in multi-robot exploration,” in *Proceedings of the Australasian Conference on Robotics and Automation (ACRA’11), Melbourne, Australia, 2011*, pp. 7–9.
- [52] M. Juliá, A. Gil, and O. Reinoso, “A comparison of path planning strategies for autonomous exploration and mapping of unknown environments,” *Autonomous Robots*, vol. 33, pp. 427–444, 2012.

- [53] S. Shen, N. Michael, and V. Kumar, “Stochastic differential equation-based exploration algorithm for autonomous indoor 3d exploration with a micro-aerial vehicle,” *The International Journal of Robotics Research*, vol. 31, no. 12, pp. 1431–1444, 2012.
- [54] K. Cesare, R. Skeelee, S.-H. Yoo, Y. Zhang, and G. Hollinger, “Multi-uav exploration with limited communication and battery,” in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 2230–2235.
- [55] B. Zhou, H. Xu, and S. Shen, “Racer: Rapid collaborative exploration with a decentralized multi-uav system,” *IEEE Transactions on Robotics*, 2023.
- [56] F. Amigoni, “Experimental evaluation of some exploration strategies for mobile robots,” in *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 2818–2823.
- [57] M. Keidar and G. A. Kaminka, “Robot exploration with fast frontier detection: Theory and experiments,” in *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, 2012, pp. 113–120.
- [58] R. Zlot, A. Stentz, M. B. Dias, and S. Thayer, “Multi-robot exploration controlled by a market economy,” in *Proceedings 2002 IEEE international conference on robotics and automation (Cat. No. 02CH37292)*, vol. 3. IEEE, 2002, pp. 3016–3023.
- [59] M. Elbanhawi and M. Simic, “Sampling-based robot motion planning: A review,” *Ieee access*, vol. 2, pp. 56–77, 2014.
- [60] D. Duberg and P. Jensfelt, “Ufoexplorer: Fast and scalable sampling-based exploration with a graph-based planning structure,” *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2487–2494, 2022.
- [61] M. Kegeleirs, D. Garzón Ramos, and M. Birattari, “Random walk exploration for swarm mapping,” in *Towards Autonomous Robotic Systems: 20th Annual Conference, TAROS 2019, London, UK, July 3–5, 2019, Proceedings, Part II 20*. Springer, 2019, pp. 211–222.
- [62] Y. Wang, A. Liang, and H. Guan, “Frontier-based multi-robot map exploration using particle swarm optimization,” in *2011 IEEE symposium on Swarm intelligence*. IEEE, 2011, pp. 1–6.
- [63] Z. Yan, N. Jouandeau, and A. A. Cherif, “A survey and analysis of multi-robot coordination,” *International Journal of Advanced Robotic Systems*, vol. 10, no. 12, p. 399, 2013.
- [64] M. Otte, M. J. Kuhlman, and D. Sofge, “Auctions for multi-robot task allocation in communication limited environments,” *Autonomous Robots*, vol. 44, no. 3, pp. 547–584, 2020.
- [65] F. Pratissoli, B. Capelli, and L. Sabattini, “On coverage control for limited range multi-robot systems,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 9957–9963.

- [66] V. G. Nair and K. Guruprasad, “Mr-simexcoverage: Multi-robot simultaneous exploration and coverage,” *Computers & Electrical Engineering*, vol. 85, p. 106680, 2020.
- [67] J. Hu, H. Niu, J. Carrasco, B. Lennox, and F. Arvin, “Voronoi-based multi-robot autonomous exploration in unknown environments via deep reinforcement learning,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 14 413–14 423, 2020.
- [68] N. Palmieri, F. De Rango, X. S. Yang, and S. Marano, “Multi-robot cooperative tasks using combined nature-inspired techniques,” in *2015 7th International Joint Conference on Computational Intelligence (IJCCI)*, vol. 1. IEEE, 2015, pp. 74–82.
- [69] T. Andre and C. Bettstetter, “Collaboration in multi-robot exploration: to meet or not to meet?” *Journal of intelligent & robotic systems*, vol. 82, pp. 325–337, 2016.
- [70] M. Keidar and G. A. Kaminka, “Efficient frontier detection for robot exploration,” *The International Journal of Robotics Research*, vol. 33, no. 2, pp. 215–236, 2014.
- [71] A. Stentz, “Optimal and efficient path planning for partially-known environments,” in *Proceedings of the 1994 IEEE international conference on robotics and automation*. IEEE, 1994, pp. 3310–3317.
- [72] C. Tovey and S. Koenig, “Improved analysis of greedy mapping,” in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, vol. 4. IEEE, 2003, pp. 3251–3257.
- [73] C. Allred, H. Kocabas, and M. Harper, “Unknown building exploration simulator (ubes),” *Software Impacts*, p. 100576, 2023.
- [74] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, “Receding horizon” next-best-view” planner for 3d exploration,” in *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2016, pp. 1462–1468.
- [75] M. Dharmadhikari, T. Dang, L. Solanka, J. Loje, H. Nguyen, N. Khedekar, and K. Alexis, “Motion primitives-based path planning for fast and agile exploration using aerial robots,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 179–185.
- [76] L. Schmid, M. Pantic, R. Khanna, L. Ott, R. Siegwart, and J. Nieto, “An efficient sampling-based method for online informative path planning in unknown environments,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1500–1507, 2020.
- [77] R. J. Alitappeh and K. Jeddisaravi, “Multi-robot exploration in task allocation problem,” *Applied Intelligence*, vol. 52, no. 2, pp. 2189–2211, 2022.
- [78] I. M. Rekleitis, G. Dudek, and E. E. Milios, “Multi-robot exploration of an unknown environment, efficiently reducing the odometry error,” in *International joint conference on artificial intelligence*, vol. 15. LAWRENCE ERLBAUM ASSOCIATES LTD, 1997, pp. 1340–1345.
- [79] E. Olcay, J. Bodeit, and B. Lohmann, “Sensor-based exploration of an unknown area with multiple mobile agents,” *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 9621–9627, 2020.

- [80] T. Luo, B. Subagdja, D. Wang, and A.-H. Tan, “Multi-agent collaborative exploration through graph-based deep reinforcement learning,” in *2019 IEEE International Conference on Agents (ICA)*. IEEE, 2019, pp. 2–7.
- [81] W. Gao, M. Booker, A. Adiwahono, M. Yuan, J. Wang, and Y. W. Yun, “An improved frontier-based approach for autonomous exploration,” in *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. IEEE, 2018, pp. 292–297.
- [82] G. Best, R. Garg, J. Keller, G. A. Hollinger, and S. A. Scherer, “Resilient multi-sensor exploration of multifarious environments with a team of aerial robots,” *Robotics: Science and Systems XVIII*, 2022.
- [83] B. Yamauchi, “A frontier-based approach for autonomous exploration,” in *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA’97. Towards New Computational Principles for Robotics and Automation’*. IEEE, 1997, pp. 146–151.
- [84] R. Zlot, A. Stentz, M. B. Dias, and S. Thayer, “Multi-robot exploration controlled by a market economy,” in *Proceedings 2002 IEEE international conference on robotics and automation (Cat. No. 02CH37292)*, vol. 3. IEEE, 2002, pp. 3016–3023.
- [85] F. Zhang, W. Chen, and Y. Xi, “Improving collaboration through fusion of bid information for market-based multi-robot exploration,” in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. IEEE, 2005, pp. 1157–1162.
- [86] W. Sheng, Q. Yang, J. Tan, and N. Xi, “Distributed multi-robot coordination in area exploration,” *Robotics and autonomous systems*, vol. 54, no. 12, pp. 945–955, 2006.
- [87] J. G. Mangelson, D. Dominic, R. M. Eustice, and R. Vasudevan, “Pairwise consistent measurement set maximization for robust multi-robot map merging,” in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 2916–2923.
- [88] A. Renzaglia, J. Dibangoye, V. Le Doze, and O. Simonin, “Combining stochastic optimization and frontiers for aerial multi-robot exploration of 3d terrains,” in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 4121–4126.
- [89] P. Yang, R. A. Freeman, and K. M. Lynch, “Multi-agent coordination by decentralized estimation and control,” *IEEE Transactions on Automatic Control*, vol. 53, no. 11, pp. 2480–2496, 2008.
- [90] L. Bramblett, R. Peddi, and N. Bezzo, “Coordinated multi-agent exploration, rendezvous, & task allocation in unknown environments with limited connectivity,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 12 706–12 712.
- [91] D. Fox, J. Ko, K. Konolige, B. Limketkai, D. Schulz, and B. Stewart, “Distributed multirobot exploration and mapping,” *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1325–1339, 2006.

- [92] J. Bayer and J. Faigl, “Decentralized topological mapping for multi-robot autonomous exploration under low-bandwidth communication,” in *2021 European Conference on Mobile Robots (ECMR)*. IEEE, 2021, pp. 1–7.
- [93] S. J. Kim, G. J. Lim, and J. Cho, “Drone relay stations for supporting wireless communication in military operations,” in *Advances in Human Factors in Robots and Unmanned Systems: Proceedings of the AHFE 2017 International Conference on Human Factors in Robots and Unmanned Systems, July 17- 21, 2017, The Westin Bonaventure Hotel, Los Angeles, California, USA 8*. Springer, 2018, pp. 123–130.
- [94] “DARPA Subterranean (SubT) Challenge.” [Online]. Available: <https://www.darpa.mil/program/darpa-subterranean-challenge>

APPENDICES

APPENDIX A

Simulators Development

This appendix gives a quick brief that aims to give quick directions to set up the simulation tools for potential future coding developments. Also, it shares a quick overlook of the simulators' major parts, therefore, further developments can easily find and focus on the interested code pieces.

All the below explanations have been tested on an Arch GNU/Linux operating system. However, the instructions should be the same, or similar with small command modifications to use in MAC OSX and Windows operating systems.

A.1 City Coverage Simulator

A.1.1 Dependencies Installation and Quick Run

First, we want to get some useful packages to play around with. We'll add the conda-forge channel to our conda package manager so we can have access to their massive repository of packages. Following commands add the conda-forge channel and then update conda to the latest version.

```
$ conda update -n base conda
$ conda config --prepend channels conda-forge
```

Now, let's generate a specific conda environment named "citycoverage" for this simulator, and activate it to use it. The reason we are generating a specific conda environment for this simulator is to create an abstraction layer from other projects so that it would not be affected by having older or newer library dependency issues.

```
$ conda create --name citycoverage
$ conda activate citycoverage
```

In this conda environment, we need to install a couple of new packages into an isolated conda environment. The first is OSMnx, a package to download, analyze, and visualize street networks from OpenStreetMap. The second is Jupyterlab, which lets us interact with Python code in handy Jupyter notebooks.

```
$ conda install osmnx jupyterlab
```

Install network library through pip.

```
$ pip install network
```

All the dependencies could be ready to start running the simulation tool. The application can be run with the following command, which will start generating CSV files for later analysis.

```
$ python city_coverage_multi.py
```

A.1.2 Postman Total Distance Calculation

This simulator mainly consists of two parts. The first part is the calculation of a total postman route distance of a given [OSM](#) map object, which uses the algorithm presented in [\[49\]](#). This algorithm takes a map object shown in [Figure A.1](#), and calculates the shortest route that visits all the nodes in the map.

This route calculation can be applied to any [OSM](#) map object, however, the computation has significantly arisen due to the algorithm's complexity is $O(n^3)$, where n is the number of nodes in the map. Therefore, the algorithm is not suitable for large-scale maps.

At the end of this calculation, the shortest distance to travel to a specific neighborhood or a city can be calculated. To cover a city with multiple agents such as drones, a city can be divided into sub-regions that are roughly equal in size so that each agent completes their missions around the same time.

A.1.3 Voronoi Division of A City

The second part of the simulator aims to divide a city into sub-regions that are roughly equal in size. To do this, a certain number of coordinates are distributed over the map, and

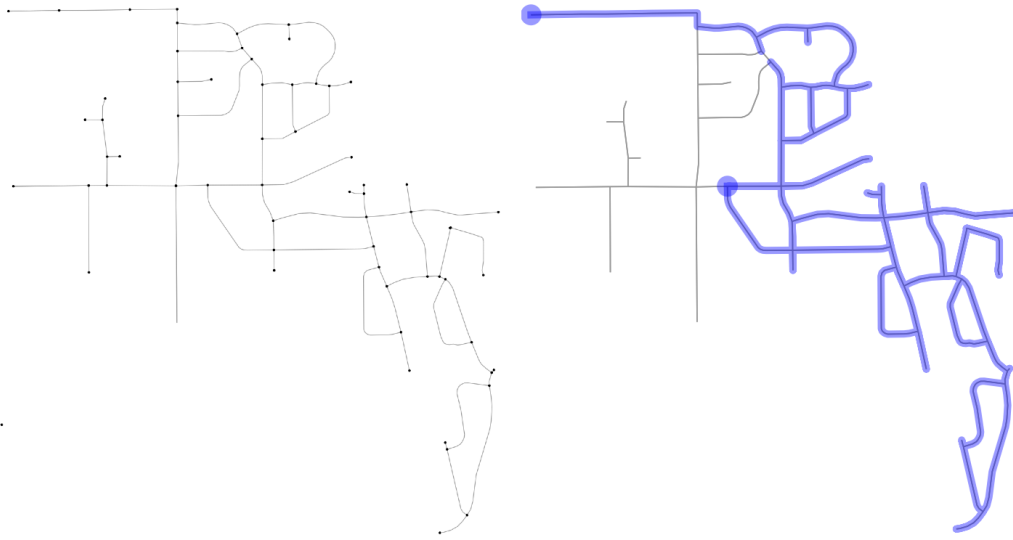


Fig. A.1: Left: An [OSM](#) map object of a neighborhood in Logan, Utah as an input to postman route calculation. Right: The output of the postman route is fully calculated, and an instance of the postman moving along the route is illustrated.

this number can be adjustable based on the agent count. Then, these coordinates are used as centroids for the Voronoi partitioning of the area. Each node and edge in the Voronoi diagram is assigned to the nearest centroid, and the resulting partitioning is shown in [Figure A.2](#). Centroids are manually placed in this figure for demonstration purposes.

The problem with this partitioning is that the sub-regions are not equal in size. To solve this problem, the centroids are moved closer/away from neighbors depending on their total distance size, and the partitioning is recalculated iteratively. This process is repeated until the sub-regions are roughly equal in size.

Using this strategy, cities with different complexities are tested with various numbers of agents, and the sub-regions are optimized using the above method. This technique can be implemented in many challenging real-life problems such as placing the bus stations in optimized centroid locations could allow individuals to walk a short distance to reach the bus stations. Also, these types of optimized sub-regions can be scanned with drones or other autonomous moving agents roughly at the same time due to the total route distance of each region is roughly equal.

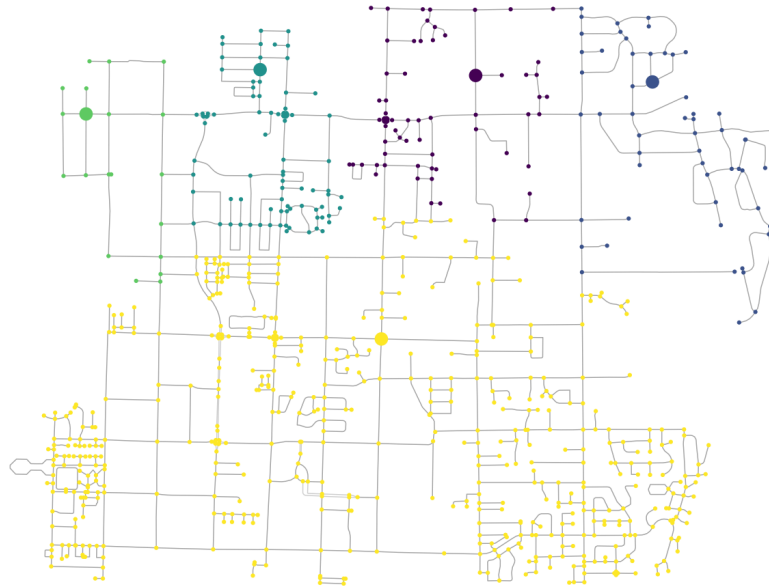


Fig. A.2: Dividing a city into sub-regions using Voronoi partitioning. The centroids are shown with bigger colored dots, and the resulting partitioning is shown in different colors.

A.2 Unknown Building Exploration Simulator (UBES)

A.2.1 Dependencies Installation and Quick Run

This project is investigating how to explore a building map with decentralized robots in different communication scenarios. Each agent will be designed to explore independently by using frontier exploration as well as incentivization techniques to better adjust its frontiers.

```
$ conda create -y --name multiagent python==3.9.2
```

```
$ conda activate multiagent
```

```
$ pip install -r requirements.txt
```

Once all the dependencies are installed, the simulation can be started with the following command:

```
$ python main.py
```

Without any change in the code, the simulation tool is designed to run in headless mode, which spins many experiments in the processor, and only shows the experiments' running status. Changing the self. Debug parameter to True in the parameters.cfg.py file

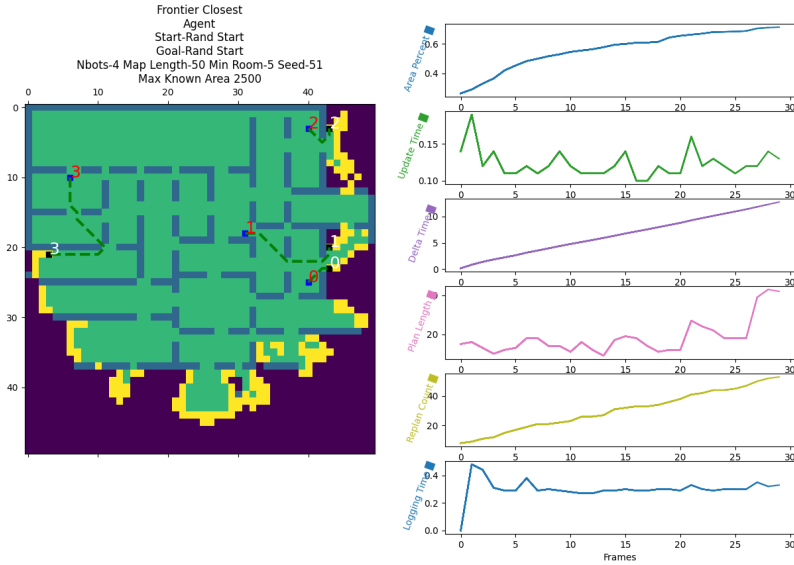


Fig. A.3: Exploration of an unknown building with the size of 50 by 50 meters with four agents, each working on the Frontier Closest algorithm as an exploration strategy. The purple-colored segments of the map represent the unknown areas, the green-colored areas represent the explored known empty areas, and the blue areas represent the explored objects (i.e., walls). The yellow cells on the map are the frontier areas. The red numbers with blue square cells represent the agent locations, and the corresponding white numbers are their goal locations, respectively. The right part of the figure shows the visualization of some values we record, which might be helpful for preliminary analysis.

can enable the visualization of the experiments, which can generate the main visualization window like Figure A.3.

A.2.2 Components of the Simulator

This simulator includes the following sub-components. Each component carries significant importance for unknown area exploration experiments to test various needs.

1. **Exploration Strategies:** The software offers a wide range of exploration methods to choose from, including "Frontier Random", "Frontier Closest", "Unknown Random", "Unknown Closest", and many more. Users can experiment with these strategies to observe their effectiveness to achieve full exploration.
2. **Initial Start Locations:** Users can specify various initial start locations for the agents, such as "Manual Start", "Random Start", "Edge Start", "Top Left Start",

”Center Start”, ”Distributed Start”, allowing to observe how the initial start conditions can affect the completion times of exploration.

3. **Initial Goal Locations:** The software enables users to define initial goal locations for the agents, including options like ”Manual Start”, ”Rand Start”, ”Edge Start”, ”Top Left Start”, ”Center Start”, ”Distributed Start”, helps to make assessments whether distributing agents over the building has a role over the completion times of explorations.
4. **Agent Loss Scenarios:** In addition to the regular unknown exploration with no threat, [UBES](#) also introduces agent loss scenarios to analyze exploration efficiency in hazardous conditions. Users can simulate scenarios like ”Unrecoverable,” where robots hit random mines and become not functional, and ”Disrepair,” where robots can help and fix other robots, enhancing the understanding of exploration strategies in operation-challenging environments.
5. **Visualization and Graphing:** The software provides visualizations and graphs to help users analyze exploration outcomes and compare strategies. Users can gain valuable insights into exploration efficiencies and the impact of various parameters on exploration performance.
6. **Extensive Parameter Control:** Users can control various parameters such as the number of agents, map size, room complexity, iteration repeat count for experiments, and exploration algorithms, allowing for a comprehensive exploration of different scenarios and settings.

A.2.3 Simulation Tool’s Configuration Settings

This section aims to explain the configuration settings that can be adjusted to run the simulation tool in different scenarios. The configuration settings are shown in [Table A.1](#).

Setting	Value	Expected Range of Values	Description
DRAW_PYGAME_SIM	false	true/false	Enable/disable Pygame simulation visualization
GRAPH_LOG_PLOTS	false	true/false	Enable/disable graph log plots
USE_THREADS	false	true/false	Use multithreading for parallel processing
CREATE_GIF	false	true/false	Generate GIFs of the exploration process
USE_PROCESS	true	true/false	Use multiprocessing for parallel processing
SEED	42	Integer	Seed for random number generation: $seed = map.length + experiment.iteration$
N_BOTS	8	Positive Integer	Number of robotic agents
MAP_NP_COLS	50	Positive Integer	Number of columns in the map grid
MAP_NP_ROWS	50	Positive Integer	Number of rows in the map grid
AGENT_OBSTACLE	3.0	Positive Float	Obstacle value for agents on the map
MINE	2.0	Positive Float	Value for mine areas on the map
EMPTY	1.0	Positive Float	Value for empty areas on the map
OBSTACLE	0.0	Positive Float	Value for obstacles on the map
UNKNOWN	-1	Integer	Value representing unknown areas on the map
KNOWN_WALL	0	Integer	Value representing known walls on the map
KNOWN_EMPTY	1	Integer	Value representing known empty areas on the map
FRONTIER	2	Integer	Value representing frontier areas on the map
ROBOT_LOSS_TYPE	Agent	Agent, Unrecoverable, Disrepair, Float (0-1)	Type of robot loss handling
MINE_DENSITY	0.01	Float (0-1)	Density of mine areas on the map
GRID_CELL_THICK	10	Positive Integer	Thickness of Pygame grid cells
PYG_SCREEN_WIDTH	500	Positive Integer	Pygame screen width in pixels
PYG_SCREEN_HEIGHT	500	Positive Integer	Pygame screen height in pixels
PYG_MIN_ROOM_SIZE	120	Positive Integer	Minimum room size for Pygame visualization
BACKGROUND_COLOR	...	RGB Color Value	Background color for visualization

Table A.1: Configuration Settings

CURRICULUM VITAE

HUZEYFE M. KOCABAS

huzeyfe.kocabas@usu.edu

1118 E. Stadium Dr. Logan, UT, 84341

(435) 919-2342

EDUCATION

- Utah State University**, Logan, UT, USA Jan. 2019 — Expected: Jan. 2024
- Ph.D. in Computer Science
- Indiana University**, Bloomington, IN, USA Aug. 2016 — May 2018
- M.S. in Computer Science
- Indiana University**, Bloomington, IN, USA Oct. 2014 — Oct. 2015
- English Education as a Second Language
- Yildiz Technical University**, Istanbul, TURKIYE Feb. 2014 — Aug. 2014
- School of Foreign Languages
- Firat University**, Elazig, TURKIYE Nov. 2008 — June 2012
- B.S. in Computer Engineering

PROFESSIONAL EXPERIENCE

- Graduate Research Assistantship, Utah State University** Logan, UT, USA
Jan. 2023 - June 2023
- Direct-Lab Robotics
- Volunteer Teaching Assistantship, Indiana University** Bloomington, IN, USA
May 2017 - June 2017
- Introduction to Programming Course (Summer-2017)

HONORS AND AWARDS

- Scholarship from the Republic of Turkey, Ministry of National Education, Ankara, Turkey (2013)
- Last Year of Firat University, Honor Student Degree (2012)

COMPUTER AND PROGRAMMING SKILLS

- **Programming Languages:** Python, C/C++, Java, Visual Basic, Assembly, Lisp, Prolog, Matlab, Adobe Flex, Actionscript 2.0
- **Web Design Languages:** Javascript, Php, Css, Html, Xml
- **Databases:** MySQL, PostgreSQL, SQL Server
- **Operating Systems:** Linux (Debian and Arch Based Distros), Mac OSX, Windows Operating Systems

PROJECTS and EXPERIENCES

- | | |
|--|-----------------|
| Code Injection Attacks and Memory Protection Mechanisms | Bloomington, IN |
| • Independent Study with Yan Huang at Indiana University | 2018 |
| Buffer Overflow in Microsoft Windows RPC-DCOM Interface | Bloomington, IN |
| • Term Project in Network Security course at Indiana University | 2017 |
| Active Learning for Fine-Grained Localization | Bloomington, IN |
| • Poster Presentantion in Computer Vision course at Indiana University | 2017 |
| Penetration Testing Tools Learning in KALI | Bloomington, IN |
| • Systems and Protocol Security course at Indiana University | 2017 |
| FileSystem Creation on Xinu: A Small Operating System | Bloomington, IN |
| • Advanced Operating Systems course at Indiana University | 2016 |
| B.S Graduation Project | Elazig, TURKIYE |
| • Secure Video Streaming by using Adobe Flex, Java, ActionScript | 2012 |
| Keylogger Software connected through FTP servers | Elazig, TURKIYE |
| • Independent Research on Security in summer | 2011 |
| Exceptional Project (Software Engineering Group Project) | Elazig, TURKIYE |
| • Academic Paper Analyzer and Popular Projects Finder | 2011 |
| Arkanoid Game Project | Elazig, TURKIYE |
| • Retro Atari Game implementation in Java | 2009 |

Published Papers

- Allred, C., Kocabas, H., & Harper, M. (2023). Unknown Building Exploration Simulator (UBES). *Software Impacts*, 100576.
- Allred, C., Kocabas, H., Harper, M., & Pusey, J. (2022, December). Terrain Dependent Power Estimation for Legged Robots in Unstructured Environments. In *2022 Sixth IEEE International Conference on Robotic Computing (IRC)* (pp. 329-333). IEEE.
- Kocabas, H., Allred, C., & Harper, M. (2022, September). Divide and Survey: Observability Through Multi-Drone City Roadway Coverage. In *2022 IEEE International Smart Cities Conference (ISC2)* (pp. 1-7). IEEE.
- Al-Ameen, M. N., Chauhan, A., Ahsan, M. M., & Kocabas, H. (2021). A look into user's privacy perceptions and data practices of IoT devices. *Information & Computer Security*, 29(4), 573-588.
- Al-Ameen, M. N., Tamanna, T., Nandy, S., & Kocabas, H. (2021, July). Understanding user behavior, information exposure, and privacy risks in managing old devices. In *International Conference on Human-Computer Interaction* (pp. 281-296). Cham: Springer International Publishing.
- Kocabas, H., Nandy, S., Tamanna, T., & Al-Ameen, M. N. (2021, July). Understanding User's behavior and protection strategy upon losing, or identifying unauthorized access to online account. In *International Conference on Human-Computer Interaction* (pp. 310-325). Cham: Springer International Publishing.
- Al-Ameen, M. N., Kocabas, H., Nandy, S., & Tamanna, T. (2021). "We, three brothers have always known everything of each other": A Cross-cultural Study of Sharing Digital Devices and Online Accounts. *Proc. Priv. Enhancing Technol.*, 2021(4), 203-224.

- Al-Ameen, M. N., & Kocabas, H. M. (2020, October). A First Look into Users' Perceptions of Digital Medicine Technology. In Conference Companion Publication of the 2020 on Computer Supported Cooperative Work and Social Computing (pp. 203-207).
- Al-Ameen, M. N., & Kocabas, H. (2020). "I cannot do anything": User's behavior and protection strategy upon losing, or identifying unauthorized access to online account. In Symposium on Usable Privacy and Security (Poster Session).
- Al-Ameen, M. N., Chauhan, A., Ahsan, M. M., & Kocabas, H. (2020). "Most Companies Share Whatever They Can to Make Money!": Comparing User's Perceptions with the Data Practices of IoT Devices. In Human Aspects of Information Security and Assurance: 14th IFIP WG 11.12 International Symposium, HAISA 2020, Mytilene, Lesbos, Greece, July 8-10, 2020, Proceedings 14 (pp. 329-340). Springer International Publishing.
- Seng, S., Kocabas, H., Al-Ameen, M. N., & Wright, M. (2019, November). Poster: Understanding User's Decision to Interact with Potential Phishing Posts on Facebook using a Vignette Study. In Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (pp. 2617-2619).