COOPERATION PARADIGMS FOR OVERCOMING COMMUNICATION LIMITATIONS IN MULTIROBOT WIDE AREA COVERAGE

by

BRIANA LOWE WELLMAN

MONICA ANDERSON, COMMITTEE CHAIR
SUSAN VRBSKY
XIAOYAN HONG
XIAO YANG
FEI HU

A DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Department of Computer Science
in the Graduate School of
The University of Alabama

TUSCALOOSA, ALABAMA

2011

ABSTRACT

Multi-robot systems are an important research topic in wide area coverage applications such as hazardous waste clean-up, bomb detection, surveillance, and search and rescue missions. They can work in parallel and complete tasks faster than a single robot. Communications can support cooperation to speed up execution, reduce duplication, and prevent interference. Communication among team members is achieved explicitly or implicitly. In explicit communication, messages are intentionally transmitted and received from robot to robot. In implicit communication, robots observe the environment and other robot actions. Although many systems use explicit communications, in exploration of large, open areas (e.g. stadiums and parks), persistent intra-team digital communications is not guaranteed. Therefore, alternative approaches that do not rely upon message passing throughout exploration are needed.

Novel contributions of overcoming communication limitations in wide area coverage include: (1) insight on how information shared between robots that are close has more influence on immediate action selection than information shared between robots that are farther apart. Spatial and temporal locality can be instrumental in determining relevance in subsequent action selection; (2) an approach in which observation leverages spatial and temporal locality to infer state rather than rely on digital messaging; and (3) an approach in which robots use spatial rendezvous to exchange information instead of continuously passing messages. Robots explore an environment in sectors, or designated areas, and periodically meet to communicate map information of what they explored. Simulations and physical experiments were conducted and

results suggest both approaches can serve as alternatives to cooperation based on continuous point-to-point communications.

ACKNOWLEDGMENTS

Although my name appears on the front of this dissertation, there are many people that have contributed to its production. Without the guidance from my advisor, encouragement from my team members and support from my family, I would not have been able to complete it.

First, I am grateful for my advisor, Dr. Monica Anderson. Through her guidance, dedication of time, enthusiasm for research, and funding, she has demonstrated an advisor one should want to be and many aspire to have. I will never forget her willingness to take a chance with me. I also would like to thank my doctoral committee, Dr. Susan Vrbsky, Dr. Xiaoyan Hong, Dr. Xiao Yang, and Dr. Fei Hu. Without their support, cooperation and suggestions, I was able to complete it in a timely manner.

I thank my husband, CB, for encouraging me since the very beginning. He has been my main advocate and has spoken and acted countless times with support. I thank my daughter, Ava. Although she is too young to know what is going on, I hope it inspires her to dream in the future. I thank my entire family for their encouraging words and many hours of baby-sitting.

Finally, I thank Shameka Dawson. She has assisted in experiments, proofread many documents, and listened to my ideas and concerns as a good friend and teammate does. I also thank the many undergraduate and graduate students that have helped including but not limited to Aparna Veluchamy, Quinton Alexander, Andrew McKenzie, Gary Moore, Lalindra Jayatilleke, Jason Carter, and Chris Tubbs.

CONTENTS

ABSTRACT ii

ACKNOWLEDGMENTS iv

LIST OF TABLES x

LIST OF FIGURES xi

1 INTRODUCTION 1

1.1 Motivation 1

1.1.1 Differences Between Explicit and Implicit Communications 1

1.1.2 Limitations of Explicit Communications 2

1.1.3 Limitations of Implicit Communications 2

1.2 Problem Statement 3

1.3 Approach 3

1.3.1 Bio-inspired Cooperation 3

1.3.2 Observation-based Cooperation 4

1.3.3 Sector-search with Rendezvous 4

1.3.4 Focus of Research 4

1.4 Summary of Main Contributions 5

1.5 Outline 5

2 LITERATURE REVIEW 7

2.1 Robot Control Paradigms 7

2.1.1 Individual Robot Control 7

2.1.2 Multirobot Control 8

2.2 Cooperation in Multirobot Systems 8

2.3 Evaluation of Multirobot Performance 9

2.3.1 Baseline Results 9

2.3.2 Metrics 9

2.4 Categorization of Communication 11

2.5 Impact of Communication 11

2.6 Centralized Cooperation Approaches 13

2.7 Distributed Cooperation Approaches Using Explicit Communication14

2.7.1 Explicit Communications: Economy/Market-based Approaches 15

2.7.2 Explicit Communications: Approaches Dealing with Limited Communications 16

2.7.3 Explicit Communications: Rendezvous 18

2.8 Distributed Cooperation Approaches Using Implicit Communications 19

2.8.1 Implicit Communications: Swarm Approaches 19

2.8.2 Implicit Communications: Potential Fields Approaches 20

2.8.3 Implicit Communications: Observation/Vision Approaches 21

2.9 Summary and Conclusion 22

3 EFFECTS OF MESSAGE PASSING ON TEAM PERFORMANCE ..24

3.1 Effects of Packet Loss 24

3.1.1 Simulation Setup 25

3.1.2 Simulation Results 25

3.1.3 Analysis 28

3.2 Effects of Message Loads 29

3.2.1 Simulation and Experimental Setup 29

3.2.2 Simulation and Experimental Results 32

3.2.3 Analysis 32

3.3 Conclusion 33

4 IMPORTANCE OF SPATIAL AND TEMPORAL LOCALITY 34

4.1 Locality in Implicit Cooperation 34

4.1.1 Coverage Algorithm 35

4.1.2 Implications of greedy exploration 39

4.2 Experimental Results 41

4.3 Conclusion 43

5 OBSERVATION-BASED COOPERATION 44

5.1 Approach.....44

5.1.1 Formal Description 44

5.1.2 Initialization and Updates of State 45

5.1.3 Implicit Communication (Observation) 45

5.2 Simulations 49

5.2.1 Simulation Setup and Results 50

5.2.2 Analysis of Simulations 51

5.3 Physical Robot Experiments: Simulated Observation 54

5.3.1 Experimental Setup and Results 54

5.3.2 Analysis 55

5.4 Physical Robot Experiments: Visual Observation 58

5.4.1 Experimental Setup and Results 58

5.4.2 Analysis 62

5.5 Conclusions 62

6 SECTOR-SEARCH WITH RENDEZVOUS 63

6.1 Approach 63

6.1.1 Frontier-based Exploration 63

6.1.2 Sector Search with Rendezvous 64

6.1.3 Approaches for Comparison 68

6.1.4 Role-based with Dynamic Team Hierarchy 68

6.2 Simulations 69

6.2.1 Simulation Setup 69

6.2.2 Simulation Results 70

6.2.3 Analysis 72

6.3 Conclusions 73

7 CONCLUSION 74

7.1 Contributions 75

7.2 Future Work 75

REFERENCES 77

APPENDICES 86

A Simulations and Physical Experiments Set-up 87

A.1 Simulations 87

A.2 Physical Experiments 87

A.2.1 Visual Observation 88

A.3 Localization 88

A.4 Communications 89

A.5 Coverage Algorithm: Frontier-based Exploration 89

LIST OF TABLES

3.1 Three-robot Teams: Comparison of average times (minutes) and standard deviations.....26

3.2 Five-robot Teams: Comparison of average times (minutes) and standard deviations.....27

3.3 Comparison of message delay for messages received 30

3.4 Simulation: Comparison of average coverage time for less vs. more messages 31

3.5 Physical Robots: Comparison of average coverage time for less vs. more messages. 32

5.1 Averages for 50% and 90% coverage for each approach in all environments in simulation.....51

5.2 Averages for 50% and 90% coverage for each approach for the real environments.....55

5.3 Averages for 50% and 90% coverage for each approach for the real environments. 59

5.4 Average percentage of individual robot coverage in the real experiments 60

6.1 Averages for 50% and 90% coverage for each approach 70

LIST OF FIGURES

2.1 In centralized approaches, a single robot or computer acts as a leader in which makes the decisions for coordinating the robot team 13

2.2 In distributed approaches, robot act independently and do no rely on a leader to make decisions for the group. 14

3.1 The environment for simulations was 16x16 meters with four 4x4 meter obstacles. 25

3.2 Coverage vs. time for a three-robot team exploring an environment .26

3.3 Coverage vs. time for a five-robot team exploring an environment....27

3.4 The coverage of a three-team robot with 0% packet loss in communications 28

3.5 The coverage of a three-team robot with 50% packet loss in communications 28

3.6 Environmental configuration for the experiments testing the effect of message loads on latency.....30

3.7 Simulation: Coverage time for low vs. high message load.....31

3.8 Physical Robots: Coverage time for low vs. high message load 31

4.1 Frontier exploration redefined as a search tree. 40

4.2 Goal selection of α and β vertices are examined in physical experiments. 42

4.3 Goal selection of α and β vertices are examined in physical experiments. 42

4.4 Distance between robots when messages were sent versus the time between messages were received and relevant. 43

5.1 Simulation environments 1 and 2 were 16x16 meters with four obstacles representing a stadium and city block 50

5.2 Simulation environment 3 was 6x6 meters with obstacles representing two rooms and a hallway 50

5.3 The 20x27 simulated office environment configuration 50

5.4 Coverage over time for all approaches in the simulated open outdoor environment. 52

5.5 Coverage over time for all approaches in the simulated cluttered outdoor environment.....52

5.6 Coverage over time for all approaches in the simulated indoor environment 53

5.7 Coverage over time for all approaches in the simulated office environment. 53

5.8 Coverage over time for all approaches in the open real world environment 55

5.9 Coverage over time for all approaches in the cluttered real world environment. 56

5.10 The paths of a robot team using Direct Comm 56

5.11 The path of a robot team using Ob Coop 57

5.12 The path of a robot team using Ob Coop w/BackTracking 57

5.13 Robots used the ARToolkit for observation of other robots. 58

5.14 Coverage over time for the physical open environment 59

5.15 Coverage over time for the real cluttered environment 60

5.16 Results for wide area coverage for a robot team using direct communication and observation-based cooperation paradigms 61

6.1 The robots search in sectors, or designated areas, and then rendezvous to exchange map information.....65

6.2 The robots search in sectors, or designated areas, and then rendezvous to exchange map information. 65

6.3 The environment for simulations was 16x16 meters with four 4x4 meter obstacles..... 69

6.4 A comparison of coverage and time for a robot team with no communications, direct communications 71

6.5 A comparison of coverage and time for a robot team with no communications and algorithms of the sector search with rendezvous.....71

6.6 A comparison of coverage and time for a robot team with no communications, direct communications.....72

6.7 Coverage and time for Role-based and Sector Search Rendezvous 72

A.1 K-Team Koala robots were used for robot experiments.....88

CHAPTER 1

INTRODUCTION

This dissertation addresses the problem of coordinating a team of robots performing a coverage task of a wide open area under the condition that persistent intra-team digital communication is not guaranteed. Wide area coverage is considered because many applications such as search and rescue, surveillance, toxic waste cleanup, and planetary exploration require coverage of large unknown environments. Typically, in such tasks, wireless point-to-point communications is used to coordinate robots. However, environmental interference, unpredictable network conditions, and distances between robots can affect the reliability of wireless communication. Therefore, cooperation paradigms that do not rely on continuous message passing are investigated.

1.1 Motivation

Generally, researchers agree that deploying multirobot systems rather than a single robot can be advantageous in coverage applications. Deploying multiple robots instead of a single robot can increase robustness and fault tolerance by allowing redundancy. While single-robot systems are spatially limited, multi-robot systems are able to spread out and work in parallel. Also, multiple robots can be useful in tasks that require considerable amount of resources and capabilities that may be too complex for a single robot. Although multirobot systems are essential in many applications, there are challenges in implementing such systems so that they complete tasks effectively.

1.1.1 Differences Between Explicit and Implicit Communications

To complete coverage tasks so that effort is not duplicated, a robot team must work cooperatively. In cooperative multi-robot teams, communication can speed up completion, reduce duplication, and prevent interference between robots. Communication among team members can

be achieved explicitly or implicitly. In explicit communications, messages are deliberately transmitted and received from robot to robot. Some approaches use messages for synchronous action selection based on maximizing global utility. Other approaches are asynchronous allowing robots to use messages to share state and intent. In contrast, implicit communication allows robots to leverage observations of behaviors of other robots or the environment to infer state and intent.

1.1.2 Limitations of Explicit Communications

The efficiency of explicit communications is subject to the limitations of the communications network. For example, performance is affected by message latency [1] and limited bandwidth [2]. If team members are exchanging large amount of data, then there is a risk of receiving incomplete information. In addition, communications in tasks such as surveillance or hazardous waste cleanup of large areas can be even more challenging. Deployed wireless communications can be chaotic since they are unplanned and unmanaged [3]. Researchers have suggested using cooperation that does not require robots to know other robots' future actions. Instead, robots share state allowing for limited distributed information but individual action selection [4].

Researchers have presented work that use explicit communications with network constraints to coordinate robots and include: [5, 6] in which robots are required to maintain line-of-sight with other robots, [7] in which message size is reduced by allowing robots to communicate polygonal representations of the map, and [8] where rendezvous approaches allow robots to meet up to exchange information about the environment.

1.1.3 Limitations of Implicit Communications

If messaging is unavailable, robots can use environment sensing to communicate indirectly with each other. Robots observe other robots' actions or information conveyed through activities in the environment (stigmergy [9, 10]). For example, common implicit communication approaches gather inspiration from insects where robots leave virtual pheromones or trail markings in the environment to direct robot cooperation [11, 12, 13]. Another approach uses potential fields which

robots are attracted to the goal and repulsed from obstacles and other robots [14, 17]. However, both approaches depend on local interactions where after a certain distance they can no longer influence action choices.

1.2 Problem Statement

Persistent intra-team digital communications are not guaranteed, therefore, cooperation paradigms that do not rely upon message passing throughout exploration are needed.

1.3 Approach

In this dissertation, cooperation paradigms that do not rely upon message passing throughout exploration are investigated. Observation-based cooperation and sector-search with rendezvous approaches are presented. These approaches make use of cues gathered from bio-inspired cooperation paradigms. Simulations and physical experiments were conducted to compare team performance of a robot team using both approaches to others using continuous message passing. In the simulations and physical experiments performed, the goal of the multirobot system is to explore or cover the area of an unknown environment as fast as possible.

1.3.1 Bio-inspired Cooperation

Studying cooperation in nature can provide insight to successful cooperation strategies in multirobot systems. The challenges of multi-robot systems are not unique. Animal systems contend and even thrive where communication is limited and environments are changing. For example, the interactions between social insects have inspired development of cooperation in robots. While navigating, ants deposit pheromone on the ground and follow pheromones previously deposited by other ants. In [11], the concept of pheromones, or chemical communication, to communicate and influence the behavior of other animals is used by robots. While previous bio-inspired approaches have concentrated on social insects, birds and primates foraging techniques can also provide hints to cooperation in multirobot systems.

1.3.2 Observation-based Cooperation

Primates show patterns of communications that can be applied to multi-robot systems. For example, as chimpanzees hunt for food they coordinate their movements and roles by visually observing other team member's actions [15]. In OBSERVATION-BASED COOPERATION, implicit communication in the form of observation is used to infer state to coordinate a robot team. Spatial and temporal localities are leveraged to determine what information is relevant in subsequent coverage activities. Frequently, the choice of next action by each team member is not affected by information communicated by teammates unless they are close.

1.3.3 Sector-search with Rendezvous

Harris' hawks initiate a search by splitting into groups on perch sites such as trees or polls [16]. To search for prey, they use a "leapfrog" movement in which they split and rejoin in groups while moving in one direction. They continue this until a prey is captured. With sector-search with rendezvous, robots explore an environment in sectors, or designated areas, and periodically meet to communicate map information of what they explored. Rather than continuous message passing, communications is only allowed during rendezvous.

1.3.4 Focus of Research

Rather than focus on accurate mapping (i.e. SLAM [18]), the goal is to quickly cover an area to gain situational awareness. This initial situational awareness can be used to plan subsequent actions such as mapping. An example scenario for OBSERVATION-BASED COOPERATION is to have a robot team to quickly survey an area for immediate danger before a human team enters the search.

The aim of this research is not to find a better solution to point-to-point communications, but to investigate alternatives. Every cooperation paradigm can be beneficial in different situations. OBSERVATION-BASED COOPERATION and SECTOR-SEARCH WITH RENDEZVOUS approaches

are different from current research because both allow robots to efficiently cover an area when communications can be unpredictable.

1.4 Summary of Main Contributions

In this research, novel contributions to overcoming the challenges of communications are investigated. Methods of cooperation that do not always rely upon explicit communication are presented. Contributions of overcoming communications issues in wide area coverage include:

- Insight into how spatial and temporal locality affect the result of communications. Information shared between robots that are close can have more influence on subsequent action selection than information shared between robots that are farther apart.
- A cooperation paradigm that uses implicit communications in the form of observation to infer state rather than rely on digital messaging. Observation leverages spatial and temporal locality and is likely to occur when robots are close and information is more useful.
- An cooperation paradigm with spatial rendezvous and sector search. Robots use rendezvous to exchange information instead of continuously passing messages. Robots explore an environment in sectors, or designated areas, and periodically meet to communicate map information of what they explored.

1.5 Outline

This dissertation is organized as follows.

Chapter 2 Literature Review In this chapter, there is an overview of important topics on multirobot systems. Topics include multirobot control, cooperation, metrics used for evaluating performance, impact of communications, and different approaches of multirobot communications.

Chapter 3 Effects of Messages Passing on Team Performance This chapter presents the preliminary data gathered to demonstrate how unreliable message passing can affect robot team performance. In the first section, the effect of packet loss on performance is discussed. In the second section, the effect of message loads on performance is compared in simulated and real robot experiments.

Chapter 4 Importance of Spatial and Temporal Locality This chapter explores the importance of spatial and temporal locality of robots during information exchange. Spatial and temporal locality provide insight on what information is relevant in subsequent coverage activities.

Chapter 5 Observation-based Cooperation Spatial and temporal locality are leveraged in observation-based cooperation since robots are more likely to observe one another when they are close. In addition, there is a discussion on how pruning and backtracking search areas can improve multirobot coverage.

Chapter 6 Sector-search with Rendezvous In this chapter, an approach that uses sector search with rendezvous is presented. Robots explore an environment in sectors, or designated areas, and periodically meet to communicate map information of what they explored.

Chapter 7 Conclusions This chapter summarizes the main contributions of this dissertation and possible future work.

CHAPTER 2

LITERATURE REVIEW

In this literature review, topics important to multirobot cooperation and communication are discussed. There is a brief introduction to robot control and cooperation in multirobot systems. A section on evaluation of multirobot performance presents common metrics used in research of multirobot systems. The remaining of the chapter focusing on communications and provides an overview of communication categorization, the impact of communication, and communication schemes in centralized and distributed approaches. Finally, there is a summary and conclusion.

2.1 Robot Control Paradigms

In literature, robot control is discussed on both individual and group levels [19]. Individual control focuses more on robot behavior modules to achieve specific tasks such as obstacle avoidance or wall following. Group control focuses on issues related to robot teams such as task allocation and communication structures.

2.1.1 Individual Robot Control

Behavior-based control is one of the main paradigms for individual robot control [20]. Instead of a sense-plan-act model, behavior-based control using a subsumption model that consists of separate layers of task-achieving behaviors. Complex behaviors are decomposed into simpler behaviors. Coordination is achieved by allowing complex actions to subsume simpler actions. For example, an exploration behavior is on a higher level than a stall recovery behavior because a robot needs to take into account if the robot is stalled before exploring. Each layer receives sensor data and produces actions for the actuators based on that behavior. This allow for robots to react from stimuli in dynamic environments.

2.1.2 Multirobot Control

Multirobot control is usually described as being centralized or distributed. In a centralized approach, a single computer or robot, receives all the information from other robots and makes decisions on task allocation for the team. In a distributed approach, there is no central controller making decisions on individual action selection. Instead, each robot has its own controller and makes its own choices on next actions. Behavior-based individual robot control works well in a distributed paradigm since critical behaviors can be executed across many robots and robots can adapt their behavior [19]. Example work done by Parker [21] focuses on a multirobot-system with behavior-based control architecture and by Balch and Arkin [22] which focus on behavior-based formation control.

2.2 Cooperation in Multirobot Systems

Definitions of multirobot cooperation vary but typically focus on the task, or method of cooperation, that increases system performance. Cao et al. [23] list definitions of robot cooperation found in literature and defines cooperative behavior among robots as "given some task specified by a designer, a multiple-robot system displays cooperative behavior if, due to some underlying mechanism (i.e., the mechanism of cooperation), there is an increase in the total utility of the system." Mataric [24] describes cooperation as "a form of interaction, usually based on some form of communication".

Cooperation can be characterized as implicit or explicit. With implicit cooperation, task agreement is not decided beforehand or communicated between robots. Instead, a robot relies on its perception of the world and its observation of other teammates' actions. With explicit cooperation, goals and actions are directly communicated from robot to robot. State variables used to calculate self utility and the other robots' utilities are communicated so that robots are explicitly aware of other teammates' goal and actions.

2.3 Evaluation of Multirobot Performance

There is no standard metric system for evaluating multirobot system performance. Often, only theoretical analysis are given in work rather than quantitative measurements. Typically, when metrics are used in robotics they evaluate the efficiency of robots performing a specific task. This section will focus on metrics used for evaluation of multirobot coverage tasks.

2.3.1 Baseline Results

Researchers often use baseline results as a control for comparing performance of proposed approaches. Balch and Arkin generated baseline results for the coverage tasks of forage, consume, and graze [25]. The results contained 30 different randomly selected scenarios including a varied number of robots and number of attractor objects for grazing. The baseline measurements were made from when there was no communication between robots and were used for comparison when robots did communicate. Rybski et al. also used baseline results generated from when no communication was used. These results were used to compare to their proposed cooperation and communication schemes [26]. Batalin and Sukhatme gathered baseline results from their previously proposed algorithm which allowed robots to disperse themselves by repelling each other to 'expand coverage' [27]. These baseline results were used for comparison of their new coverage approach which robots drop communication beacons that refer exploration directions for nearby robots [28].

2.3.2 Metrics

Balch and Arkin describe performance as "objectively measurable" and present potential metrics for multirobot systems as cost, time, energy, and reliability/survivability [25]. They argue that metrics must be carefully chosen because there can be competition between them (i.e. cost versus reliability). A system should be built with minimum cost and number of robots. Time measures how long it takes to accomplish the task. The smallest amount of energy should be used

to complete a task. Energy is considered to be more important for where energy stores are limited such as planetary or undersea applications. A reliability/survivability system has the greatest chance to complete the task at the expense of time and cost. Although these metrics were presented, most authors select time to complete as the primary performance metric for their studies [4, 17, 25, 27, 29, 30].

Generally, the goal is for a robot team to cover as much area as possible in the shortest amount of time. Howard et al. present a solution for the sensor network deployment problem. Their metrics for comparisons are described as coverage, what is the area coverage, and time, how long does it take for network deployment [17]. Other researchers that use total coverage and/or time as metrics include [4, 27, 29, 30]. Amount of coverage for individual robots are also used for determining algorithm performance. For example, Poduri and Sukhatme evaluate a mobile sensor network deployment algorithm based on per-node coverage [31].

Another metric used by researchers is distance traveled by the team or individual robots. Zlot et al. demonstrate a market-economy based control for multirobot mapping and exploration [32]. In the market architecture, the goal is to maximize information gain and minimize cost. Distance traveled to reach a goal is used to calculate expected cost. To determine exploration quality, the total of distance traveled and information gain, which is area covered, are evaluated. Distance traveled is also related to energy consumption [33]. Berhault et al. use distance traveled by the entire team and describes it as the amount of energy consumed in an exploration task [34]. Researchers also refer to distance traveled as the total number of moves or time steps [12].

When conducting multirobot simulations or experiments, many researchers measure speed up, or the increased productivity due to adding more robots. They also acknowledge that increasing the number of robots can positively affect performance up to a certain point [35]. Interference, or competition for space between robots, can increase as the number of robots grows. Interference can decrease productivity of a system since more time is spent avoiding [36]. Rosenfeld et al. studied scalability of coordination algorithms with larger groups and found that spatial limitations can increase interference [37] in simulation. Measuring speed up and interference can be used

to determine the effectiveness of a coordination algorithm because scaling the number of robots appropriately can improve team performance.

2.4 Categorization of Communication

Communication in multirobot systems has been categorized using different methods. Iocchi et al. [38] use the terms direct and indirect communication, which direct communication is described as using some hardware onboard dedicated to sending signals that team members can understand, whereas indirect communication make use of stigmergy [10]. Dudek et al. [39] provide a taxonomic organization of communication by range, topology, and bandwidth. Communication range is the maximum distance between two robots in which communication can occur. Communication topology describes the hierarchy at which robots can communicate with other robots. Communication bandwidth is the amount of information that can be transmitted from robot to robot.

Cao et al. [23] characterize robot interaction via environment, sensing, and communications. In interaction via environment, there is no interaction between robots but the environment is the communication medium. Interaction via sensing occurs when robots sense (i.e. through vision or RFID) one another without explicitly communicating. With interactions via communication, there is explicit communication with intentional messaging between robots. Balch and Arkin [25] describe communications as explicit, the deliberate act of signaling between robots, and implicit, the observations of environment and other robots. In this dissertation, these descriptions of explicit and implicit communications are utilized.

2.5 Impact of Communication

Klavins [40] examines communications complexity to estimate scalability for control and communication algorithms in multi robot systems. Communications complexity is analyzed using the bandwidth used and the latency, or time spent waiting, of a communication event. In the worst case complexity, $O(n^2)$, there is constant communications between all robots. It is not considered scalable since the amount of communications increases significantly with the number of robots.

Bullo et al. [41] describe communication complexity as the maximum number of messages transmitted during the entire execution of a task by multirobot systems. Communication complexity is used to analyze the performance of several distributed algorithms.

Other researchers demonstrate and show results on the effects of communication on team performance. Balch and Arkin [25] researched how communication affects multirobot systems performance during the tasks of forage, consume, and graze. The impact of different types of communications is compared and includes both no communications and direct communication. Results suggest that communications appears to be unnecessary in tasks in which implicit communications exists and the more complex communications had little or no benefit over basic communications for these tasks.

Rybski et al. [2] demonstrate how low bandwidth communication channels affect the performance of a robot team on a surveillance task. Miniature scout robots were required to use very low capacity RF communication systems due to their size. Relying on off-board processing, the robots shared bandwidth. Results show that the performance of the system degraded with the increase of communications load. In other work, Rybski et al. [42] demonstrate the effects of simple communications strategies on the performance of a robot performing foraging. The task was for robots to locate a target in an enclosed arena, pick it up, and then drop that target off at a designated area. Robots foraged using random walk with three different communication strategies: no communication, which serves as the baseline, reflexive, which the robot turns on a light beacon as it grabs the target, and deliberative, which the robot turns on a beacon when a target is located but it is not able to grab it. Physical experiment results indicated that with communication there was not a significant increase of overall performance over not having communications. Instead, there was a decrease in the variance of completion time.

Wellman et al. [43] suggest as robots exchange large amounts of information, they run into the risk of receiving incomplete information due to CPU overload from individual message processing. Specifically, in outdoor areas, maps can become large and as robots communicate map information

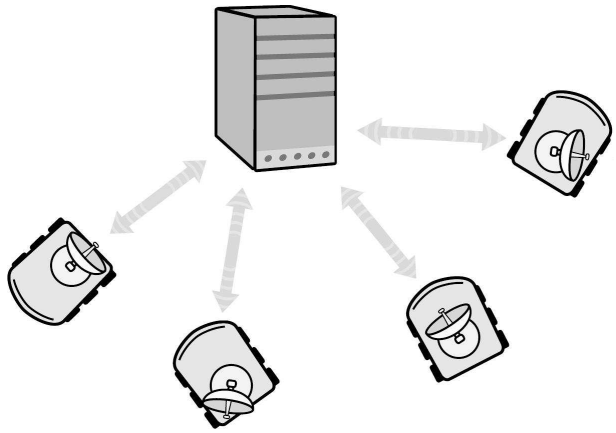

Figure 2.1: In centralized approaches, a single robot or computer acts as a leader in which makes the decisions for coordinating the robot team

the network can quickly become saturated. As message loads increase, team performance degrades [1].

2.6 Centralized Cooperation Approaches

Under centralized communications architectures, a single computer or robot acts as the leader (Figure 2.1). Robots communicate with the leader and the leader is responsible for making the decisions to coordinate the team. Examples of centralized approaches for multirobot exploration include [29, 32, 44].

The advantage of a centralized approach is that there can be optimal planning since one computer coordinates the entire team. However, each team member is required to remain in communication range with the leader so that the leader has the latest information to coordinate the team in which can require a high bandwidth. Therefore, in an unknown or dynamic environment with a large number of robots and limited communications, a centralized approach may not be suitable. In addition, performance of the team relies heavily on the leader. If the leader fails and a new leader is not chosen then the entire team becomes ineffective. Therefore, centralized approaches are more appropriate for teams of small sizes in a static environment with global communications [45].

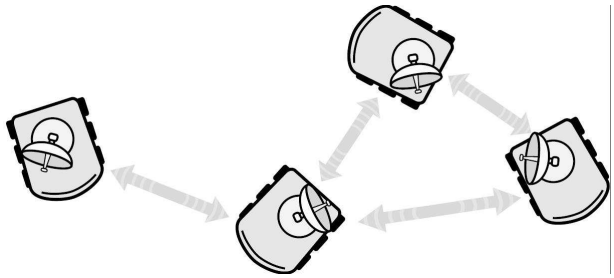

Figure 2.2: In distributed approaches, robots act independently and do no rely on a leader to make decisions for the group. A less stringent communication scheme, such as communication between robots that are close, can be used since they do not have to remain in constant communication range with a leader.

In decentralized approaches, robots act independently and use their sensing abilities to make decisions. Although robots can communicate to improve coordination, results are often sub-optimal since planning is local. However, a decentralized approach does not have many of the disadvantages of a centralized approach such as a single point of failure (Figure 2.2). Researchers have concluded that decentralized approaches have several advantages over centralized approaches [10, 46, 47]. For example, in a task in which robots need to spread out such as in exploration, a decentralized approach does not require robots to remain in communication range with a leader.

2.7 Distributed Cooperation Approaches Using Explicit Communication

Several decentralized approaches for coordinating multirobot systems make use of explicit communications. Yamauchi et al. [48] present a frontier-based coverage algorithm. Robots detect frontiers, or regions on the boundaries between unexplored and open space, and then expand area knowledge by recursively exploring the nearest unvisited frontier. Robots cooperate by communicating state updates, reducing duplicate coverage by the team. Work done by Cortes et al. [49], Kong et al. [50], and Zheng et al. [51] are additional examples of distributed approaches using explicit communications.

2.7.1 Explicit Communications: Economy/Market-based Approaches

In economy/market-based approaches, robots act independently to plan their actions but take in account the resources of the team by negotiating or bidding with other robots to trade tasks. Simmons et al. [29] present a frontier-based exploration algorithm with bidding that coordinates robots in exploration. Each robot creates bids representing information gain and cost of travelling to different areas. A central location receives the bids and use a simple greedy algorithm to assign tasks to robots.

Dias and Stentz [52] present a distributed approach to coordinate a robot team based on a free market architecture. Robots receive revenue and incur costs depending on their contribution to the overall task. They also negotiate through bidding for services to maximize profits and minimize costs. Similar to the traveling salesman problem, robots were given the task to visit a set of pre-selected observation points (cities). Preliminary results suggested there was a reduction in team cost for when robots negotiated in city-for-revenue deals. Additional studies by Dias [47] using market-based architecture for multirobot coordination use robots, called TraderBots, that trade tasks via auctions.

Zlot et al. [32] address multirobot mapping and exploration using a market control architecture to coordinate robots. Robots make decision by communicating price information to a central agent. The agent serves as an interface between the robot team and human operator. If there is communication failure, a robot can continue to explore using a random goal generation strategy.

Sheng et al. [53] demonstrate coordinating robots using a distributed bidding-based algorithm under limited-ranged communications. A robot only communicates and bids with other robots within its own subnetwork. A nearness measure is used to increase the possibility that robots are close enough to communicate with each other. Results from simulations show that robots clustered together due to the nearness measure. They also suggest that the larger the communication range between robots, the quicker the exploration rate.

Combinational bidding [54], which robots bid on bundles of targets, are also utilized for robot coordination. An example can be found in Berhault et al. [34] where results of combinational

and single-item bidding for an exploration task are compared. Robots are assigned predetermined targets and a virtual auctioneer is responsible for holding auctions. When a robot finds new information about the terrain, it shares it with the auctioneer. Each robot bids and the auctioneer determines the winner. Results suggest that combinational bidding provides an increase in team performance.

With economy/market based approaches, lower communication bandwidth can be used. Rather than communicating large amounts of data, prices and bidding act as aggregated information allowing for lower communication costs. However, negotiation and bidding protocols as well as the reward and cost schemes for task allocation can complicate the design of the system. Additional research of economy/market-based approaches include: a survey by Dias et al. [55], coordination for a box pushing task by Gerkey and Mataric [56], exploration task in a dynamic environment with one of the bidders acting as the auctioneer by Sariel and Balch [57], a security sweep task with tight coordination by Kalra et al. [58], selecting appropriate bidding rules for the global objective by Tovey et al. [59], and complex tasks are generalized into tasks trees in which robots can bid on any level of that task abstraction by Zlot and Stentz [60].

2.7.2 Explicit Communications: Approaches Dealing with Limited Communications

Several researchers have presented approaches with consideration for limited communications. Parker [61] presents ALLIANCE, a behavior-based fault tolerant architecture for cooperative control. Robots have a broadcast communication mechanism that is not guaranteed to be available. In ALLIANCE, a robot is allowed to decide on its own actions depending on what the mission is, what other robots are doing, the environment conditions, and its internal state.

Burgard et al. [44] consider coordinating multiple robots during exploration under communication with a limited communication range. To coordinate the robots, target point selection for robots is determined using the cost of moving to an area. When distance between robots is too large for them to communicate, robots divide into sub-teams that can communicate with each other.

Meier et. al. [7] address coordinating multiple robots during exploration using communication with limited bandwidth. Robots only communicate a compact polygonal representation of the environment and only sent incremental changes to the polygon. Experiments were conducted in different bandwidth conditions. Results suggest that the algorithm efficiently coordinates the robots in restricted bandwidth conditions.

Rooker and Birk [62] discuss multirobot exploration with the constraints of wireless networking. The algorithm includes configuration changes, or collection of possible moves, that robots can take. A heuristic utility guides the selection of the configuration change at each time step. Simulations were conducted under two scenarios: robots permanently maintain communication with each other and a base station and in the second scenario the base station constraint was removed allowing robots to move freely while maintaining communication with other robots. In the first scenario, results suggest that the algorithm scales scales well as the number of robots increased the amount of explored area increased. In the second scenario, robots ended up stuck in deadlock due to robots having to remain within communication distance with each other. After a solution to remove deadlock was added, robot were able to explore the entire area.

Roth et al. [63] present an approach to overcome the problem of robots creating a real-time world map of an environment with high-communication latency. Performing robot soccer (RoboCup [64]), robots built an individual map and shared a world map. However, information was shared only on an as-needed basis such as when the ball could not be located. Experimental results include a comparison of how often the ball was lost when the robot team used a shared world map and when it did not use a shared world map. When information was shared between robots the ball was lost for 1.84% compared to 19.47% when information was not shared.

Other researchers use line-of-sight approaches which robots are allowed to communicate only when they are within sight of each other. Arkin et al. [5] investigate how a team of robots can self organize during exploration by maintaining line-of-sight communications. Experiments involved robots searching for hazardous materials with varying degrees of prior knowledge. Exploration

time was not decreased significantly because the approach only allowed one robot to move at a time.

Rekleitis et al. [6] coordinated robots restricted to line-of-sight communication in an unknown environment. The results show that by maintaining the cohesiveness of the team, by allowing only minimal splitting, caused a reduction in repeat coverage. However, the performance of the proposed algorithm depends on the average cell size. If in most cells the team of robots does not fit then the remaining robots will become idle, therefore creating a form of dynamic repeat coverage. In later work, Rekleitis et al [65] discuss algorithm solutions for coverage tasks for when there is limited communications and unlimited communications. When there is unlimited communications they use auction-based global-communications. However, for the limited communications, communication is restricted by line of sight and the team based coverage allowing robots to work on teams that split into two teams only to move around an obstacle.

2.7.3 Explicit Communications: Rendezvous

Another technique researchers have studied to address communication limitations is to use rendezvous. Instead of communicating during the entire execution of the task, robots share information during rendezvous. Roy and Dudek [8] focus on getting robots to meet at a rendezvous location if they do not know each other's initial start positions. The goal is to explore the environment and meet at a location with robots that have noisy sensors.

Hoog et al. [66] introduce a way of calculating rendezvous points so robots can meet and share information. Role-based exploration where some robots (Relay) relay information between robots and a central command centre while others (Explorer) continue to explore using frontier exploration. When a Relay and Explorer meet, they share information about the environment and the Explorer plans the next rendezvous point. The Explorer determines the next rendezvous point by placing it deep into its next choice of frontier. Although their method for determining rendezvous points is effective, they note that in dynamic environments they may have problems

where the Explorer and Relay may not reach each other. The approach is extended to deal with more dynamic environment by including role-swapping [67].

Others have presented approaches with consideration for limited communications network. Wagner and Arkin [30] demonstrate robots performing communication-sensitive renaissance while maintaining network connectivity. Ulam and Arkin [68] focus on robot behaviors, or reactions of robots, for communication recovery in tasks with unknown hostile environments. Visser and Slamet [69] present work in which robots explore frontiers where it is more likely that area will have communication success. Xu et al. [70] focus on large teams and examine the influence of network topology on the efficiency of information sharing between robots. Sheng et al. [71] present a totally distributed bidding-based coordination algorithm to accommodate for limited-range communications.

2.8 Distributed Cooperation Approaches Using Implicit Communications

2.8.1 Implicit Communications: Swarm Approaches

With swarm robotics, groups are usually large and consist of relatively simple robots. Swarm robots are usually biologically inspired, have decentralized control, and use local communications. Typically, the robots are non-intelligent individually, but collectively through local interaction emergent behaviors can emerge such as flocking, dispersion, and aggregation [72]. Several researchers have presented work in swarm robots with inspiration from ants.

Ant-like Swarms

Ferranti et al. [73] investigate agents that do not communicate directly with each other but indirectly by leaving information on tags deployed in the environment. They coordinate by reading and updating traces on the tags. Similarly, Koenig [12] demonstrate coordinating robots to cover a terrain similar to ants. Robots communicate via markings left by other robots and do not coordinate

based on memory or maps. While these approaches do not rely on direct communications, they require local interaction to distribute robots.

Wagner et al. [74] robots use evaporating traces similarly to how insects use chemicals called pheromones to coordinate. The world is represented by vertices and edges in which graph traversal is used. Robots communicate by leaving constant amount of traces. The intensity of traces decreases with time and act as memory of what is the latest time a point was visited. Simulations were conducted with different graph traversal algorithms and the graph was covered in a polynomially bounded time when traces were left at the edges and in exponential upper bound time when traces were left on the vertices.

With swarm robots, distributed control architecture is used so there is not a single point of failure. The number of robots can easily be increased or decreased allowing for scalability. However, local interaction is required since the robots are not intelligent enough individually to accomplish the task. Additional studies in swarm robotics include the following. Kube and Zhang [75] present work on cooperative box pushing. Drogoul and Fereber [76] discuss foraging and chain making robots. Payton et al. [77] describe the concept of virtual pheromones. Hsiang et al. [78] develop and analyze swarm algorithms. McLurkin [79, 80] provides distributed algorithms for swarms of hundreds of robots. Morlok and Gini [81] examine how the choice of movement algorithms affects swarm performance. Wagner et al. [82] present work on cleaning ant robots that use whether or not dirt is present in an area as a method for implicit communication.

2.8.2 Implicit Communications: Potential Fields Approaches

The idea of potential fields applied to robots was introduced by [83]. Potential fields involve imaginary forces that attract robots to targets and repels them from obstacles. The sum of all forces determines the direction and the speed a robot travels [84].

Researchers have demonstrated dispersing robots using potential field methods. Balch and Hybinette [14] use potential field approaches in robot navigation for formation control. Formation behaviors are implemented with motor schemas vectors which attracts robots to the closet position

in the formation and repel them from obstacles. Different formations (column, diamond, line) were tested in simulations and results indicate that the column formation strategy provides the best performance.

Howard et al. [17] use potential fields to distribute a mobile sensor network. Robots spread out as they repel other robots and obstacles. Howard et al. [85] also present work on maximizing coverage while simultaneously ensuring that robots retain line-of-sight relationships with one another. Agents were deployed into an unknown area one at a time, each one making use of data gathered from the previous robot in an attempt to determine the location that will produce the maximum coverage in an area. Poduri and Sukhatme [31] also investigate the use of potential fields where robots could repel or attract one another.

While the potential field approach is simple, there are some issues that arise when used to coordinate robots. A robot can become trapped in a cyclic behavior or local minima in which can occur if it runs into a dead end. A robot may experience unstable motion or may not find a path in the presence of closely spaced obstacles or narrow passages if forces are acting against each other [84].

2.8.3 Implicit Communications: Observation/Vision Approaches

Researchers have introduced approaches which robots use observation or vision to communicate. Pagello et al. [86] examine cooperative behavior of a robot soccer (RoboCup [64]) team that uses the observations of other robots behaviors to coordinate. The architecture takes into account individual value, or what is important to the individual robot, and the social value in which is more relevant to the team. The state of the environment is represented by macroscopic parameters that can be modified by the robots to hint to other robots as a form of communication. Batalin and Sukhatme [27] present a coverage approach that allow robots to coordinate by moving in a direction away from all its immediate sensed neighbors.

Other work that have used vision/observation among robots for coordination include: Parker et al. [87] which use vision for robot detection in mobile sensor network deployment, Das et al.

[88] present vision-based formation control, Rybski et al. [26] study foraging robots that illuminate beacons as a form of implicit communication, and Meng et al. [89] present a hybrid approach that using vision to locate the host in case of communication loss.

2.9 Summary and Conclusion

Many multirobot approaches presented have been successful in coordinating robot. However, each one may be appropriate in a certain situation while in another it is unsuitable. Centralized approaches can be unreliable with a large robot team or unlimited communications since all robots must communicate with a single computer or robot that coordinates the team. Although with economy/market based approaches information is aggregated allowing for lower communication cost, they can become computational inefficient as the negotiation and bidding protocols become complicated. Line-of-sight approaches can be inadequate in large environments because coordination depends on robots remaining close or visible to each other preventing them from spreading out. In general, with all explicit communication approaches, maintaining point-to-point communications can be challenging.

The challenges of using explicit communications are often neglected in multirobot systems research. One issue is that most simulations are conducted with the assumption that there is a perfect communications network. Another issue is that many physical experiments are conducted only in indoor areas. In environments (large outdoor and/or dynamic) where there can be increases in distances between robots and environmental interference, message delays, packet loss, and loss of links become more unpredictable. Even under various ad-hoc routing protocols, wireless communications have been shown to be unreliable in outdoor areas when teleoperating one robot [90]. It could be even more unreliable if robots are executing autonomously as the need for computational resources increases. Although with implicit communications there is less concern for digital communication failure, there are other challenges. Approaches using swarm robots and potential fields require local interaction between robots making it difficult for them to independently explore a large area.

As more robot teams are developed for critical missions, additional approaches that overcome communication limitations are needed. In particular, approaches for multirobot systems performing in wide open area where communications can be more unpredictable or unreliable would be useful. The contributions of this research include opportunities for overcoming this challenge. With OBSERVATION-BASED COOPERATION, instead of using point-to-point communications, robots use implicit communication in the form of observation to infer state to coordinate in a coverage task [91]. With SECTOR SEARCH WITH RENDEZVOUS, instead of continuously passing messages, robots explore an area in sectors and then communicate what was found during periodic rendezvous [92].

CHAPTER 3

EFFECTS OF MESSAGE PASSING ON TEAM PERFORMANCE

Many researchers use direct communications to coordinate robot teams performing coverage tasks. However, the effects of message passing on team performance are often ignored. In Burgard et al. [94], each robot maintains a cumulative team state and computes utilities for all team members in parallel. Although it allows for optimal action selection, it becomes less scalable as the number of robot increases. It possesses a communications complexity of $O(n^2)$ each time step where n is the number of robots using naive point to point message passing.

Communication complexity in economy/market-based approaches should also be considered. Scalability is limited by communication complexity as the number of auctions and bid complexity increases. In systems using single auctions, a polynomial number of messages is sent. An exponential number of messages can be required with combinatorial auctions because of the number of task bundles to be considered [55].

As the number of messages increases, messages can be lost or delayed creating a communications bottleneck. Therefore, robots may not receive all messages which can result in degradation in team performance. In this chapter, the effects of message loss and delay on robot team performance are evaluated.

3.1 Effects of Packet Loss

In this section, there is an investigation on the effects of packet loss on robot team performance. Results from simulations with perfect network conditions are compared to those with network failure. Packet loss is introduced into the network to compare robot team performance of a system with network constraints to one with a optimum network condition. It is proposed that packet loss influences robot team performance during a coverage task.

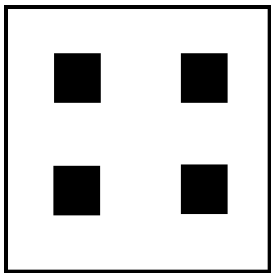

Figure 3.1: The environment for simulations was 16x16 meters with four 4x4 meter obstacles.

There were four experiments conducted using different rates of dropped messages. To implement dropped messages in simulations, code was written to deliberately drop messages as they were received by the emitter.

3.1.1 Simulation Setup

Simulations were conducted in the 3-D physics-based simulator, Webots [93]. The controller was written in the C programming language and experiments were performed on a Dual Core 2.33 GHz machine running Linux with 2GB of RAM. An emitter and receiver were added to each robot for point-to-point communications. The environment (Figure 3.1) for simulations was 16x16 meters with 4x4 meter obstacles. Twenty trials for each case were conducted with a three-robot and five-robot team. Additional details on experimental setup are reported in Appendix A.

Even though it is intuitive that message loss can affect team performance, results of it are often ignored in algorithm design. In this set of experiments, robot teams using direct communications to coordinate during exploration were conducted with different rates of dropped packets. Since the amount of packet loss can be unpredictable in a network, simulations were conducted with the percentage of packets dropped ranging from 0% to 50%.

3.1.2 Simulation Results

Each simulation was executed until coverage was complete. The simulation at which robots use no communications serves as the baseline experiment for comparison against the simulations with direct communications. Performance is measured by the percentage of area covered versus

	50% Cover Time (min)	σ	90% Cover Time (min)	σ
No Comm	6.63	1.42	20.90	3.83
0% dropped packets	5.86	1.37	13.89	3.38
25% dropped packets	6.09	1.70	18.29	2.90
50% dropped packets	6.61	1.63	19.69	3.93

Table 3.1: Three-robot Teams: Comparison of Average Times (minutes) and Standard Deviations

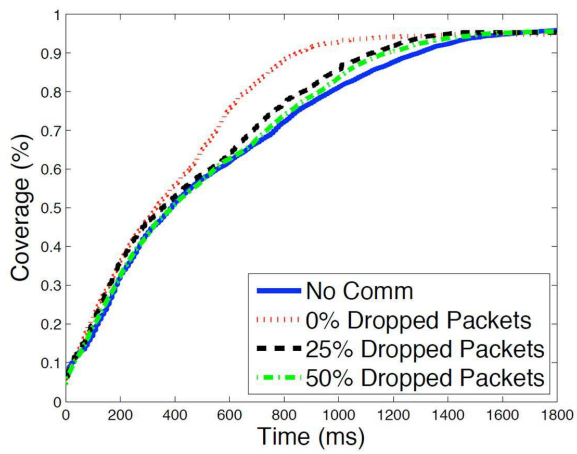

Figure 3.2: Coverage vs. Time for a three-robot team exploring an environment

time. Each set of simulations will be compared with another to determine if the hypothesis is true. The following simulations were conducted:

- No Comm
- 0% Packets Dropped
- 25% Packets Dropped
- 50% Packets Dropped

It is hypothesized that packet loss has an impact on robot team performance. Data collected include the average time that the team explored at least 50% and 90% of the environment with different network conditions. Data also includes all averages of coverage over time.

Table 3.1 shows the average times for when at least 50% and 90% of the area was covered by the three-robot team. On average the more packets that were dropped, the higher the time it took to cover the environment. Overall, the robot team that had no packet loss performed the best (Figure 3.2).

	50% Cover Time (min)	σ	90% Cover Time (min)	σ
No Comm	6.69	1.45	19.46	3.68
0% dropped packets	3.91	0.78	10.50	1.79
25% dropped packets	4.29	0.93	12.86	1.70
50% dropped packets	5.81	1.13	16.96	2.32

Table 3.2: Five-robot Teams: Comparison of Average Times (minutes) and Standard Deviations

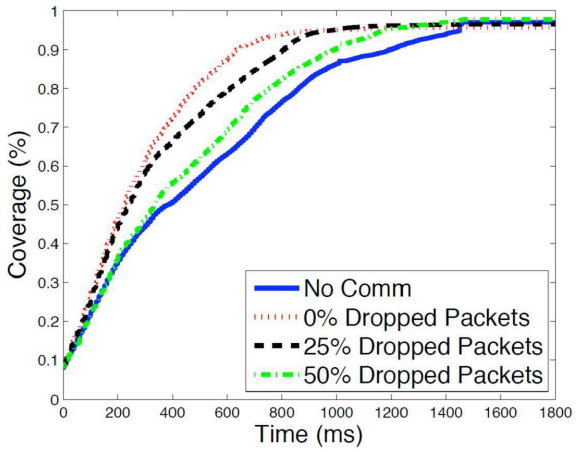

Figure 3.3: Coverage vs. Time for a five-robot team exploring an environment.

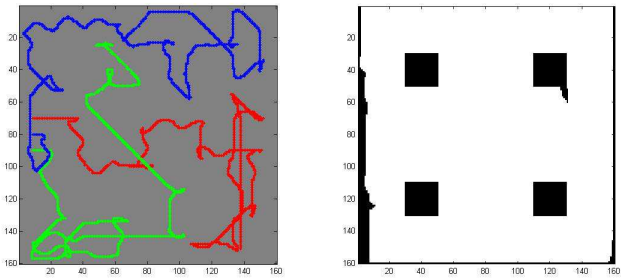

Figure 3.4: The coverage of a three-team robot with 0% packet loss in communications.

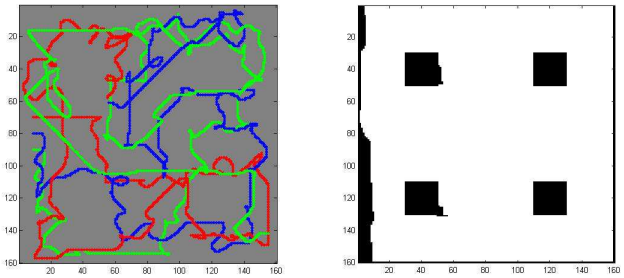

Figure 3.5: The coverage of a three-team robot with 50% packet loss in communications.

There are similar results for the five-robot team. The higher the percentage of packet loss, the worse the team performed (Table 3.2). Figure 3.3 illustrates the overall coverage for the five-robot team and it shows the less packet loss the better the performance. The five-robot team covered the environment quicker simply because there were more robots. However, there was a larger difference in the effect of packet loss when compared to the three-robot team (Figures 3.2 and 3.3). For example, the average difference of the coverage times for 50% and 90% for the three-robot and five-robot teams was 1.64 minutes and 2.84 minutes, respectively.

3.1.3 Analysis

The results support the hypothesis because packet loss did affect robot team coverage in an exploration task [43]. Performance degraded linearly with the rate of packet loss. As seen in Figure

3.4, robots did not overlap coverage when network conditions were perfect. However, when there was 50% packet loss, coverage was duplicated (Figure 3.5).

Transmitting more messages does not always improve performance. The robots in the experiment transmitted information on an area only once. These simulations were performed with three and five sized robot teams with one environment. There may be different results for a larger robot team and different environments. It is presumed that when robots communicate they share information that may be inaccurate. When robot uses direct communication, they share maps based on odometry with some error.

3.2 Effects of Message Loads

Simulations and physical experiments were conducted to compare team performance of a multirobot systems using point-to-point communications with an optimum communications network to one with intermittent communications.

Although simulations are beneficial because of the time and resources it takes to maintain multiple robots, they are often performed with optimum network conditions. As a result, there can be discrepancies between the results of simulation and physical experiments. The impact of network failure on team performance of a multirobot system is examined. Simulations and physical experiments were conducted to illustrate the effect of message loads on performance of a team of robots using point-to-point communications to coordinate during an exploration task.

3.2.1 Simulation and Experimental Setup

The control program, written in C, was essentially the same for both the simulated and real experiments. The coverage algorithm used was frontier-based exploration (explained in more detail in A.5 of the appendix). The simulation environment used was Webots [93], a 3-D physics-based mobile robot simulator. K-team Koala robots were used for physical experiments. The range of the distance sensor used for both simulations and physical experiments was two meters. More detail of experimental setup is presented in Appendix A.

To compare the effects of message loads, experiments were performed using two direct communication scenarios. In the first scenario, robots were allowed to transmit messages about a newly discovered area. In the second scenario, robots were allowed to broadcast more by transmitting information about an area every time step it was observed. The data collected includes the percentage of area covered, average times at 50% and 90% coverage, and average message delay for messages received.

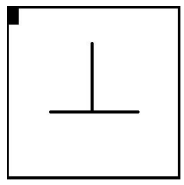

Figure 3.6: Environmental configuration for the experiments testing the effect of message loads.

Table 3.3: Comparison of message delay for messages received.

	Low Message Load (s)	σ	*High Message Load (s)*	σ
SIMULATIONS	1.446	0.153	3.247	0.067
PHYSICAL EXPERIMENTS	1.791	2.022	3.620	2.715

To compare the effects of message loads, experiments were performed using two different direct communication scenarios: LOW MESSAGE LOAD and HIGH MESSAGE LOAD. In the LOW MESSAGE LOAD experiments, robots were allowed to transmit once whether an area was open or closed. In the HIGH MESSAGE LOAD experiments, robots were allowed to broadcast more by transmitting information about an area every time step it was observed.

A team of three robots was dispersed into an unknown environment to search via frontier exploration. The environmental configuration for the experiment was 6m x 6m with a T-shaped obstacle that represent walls of an office building (Figure 3.6).

The message delay averages are comparable between simulations and real experiments (Table 3.3). There is a 0.344s and 0.372s difference between message delay for the LOW MESSAGE LOAD

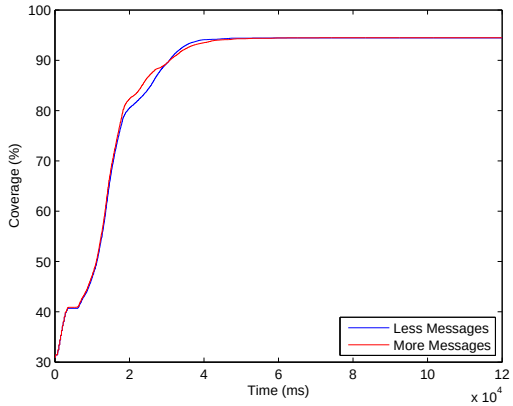

Figure 3.7: Simulation: Coverage time for low message load vs. high message load simulations.

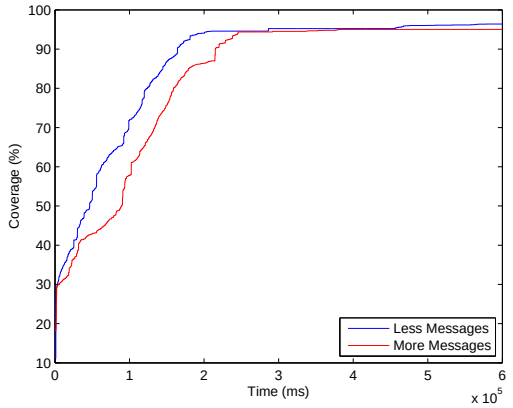

Figure 3.8: Physical Robots: Coverage time for low message load vs. high message load experiments.

Table 3.4: Simulation: Comparison of average coverage time for low vs. high message load.

Messages	*50% coverage time (s)*	σ	*90% coverage time (s)*	σ
LOW MESSAGE LOAD	12.323	2.539	24.523	7.754
HIGH MESSAGE LOAD	11.971	2.261	26.072	8.565

Table 3.5: Physical Robots: Comparison of average coverage time for low vs. high message load.

Messages	*50% coverage time (s)*	σ	*90% coverage time (s)*	σ
LOW MESSAGE LOAD	48.580	22.766	168.054	77.611
HIGH MESSAGE LOAD	89.639	17.284	202.057	31.256

and HIGH MESSAGE LOAD experiments, respectively. However, the standard deviations for the real robot experiments are significantly higher than those in simulation.

3.2.2 Simulation and Experimental Results

To examine the impact of message loads on performance, percentage of area covered and average times at 50% and 90% coverage are compared. The coverage times for simulations indicate that there is not a significant difference between when robots pass less or more messages (Figure 3.7). For example, at 50% coverage, robots that sent more messages were out performing those that sent less messages (Table 3.4). However, in the real robot experiments, completion time increased as the number of messages increased (Figure 3.8).

Tables 3.4 and 3.5 show the average times robots reached 50% and 90% coverage. For the LOW MESSAGE LOAD experiments, 90% coverage was reached in simulation 143.531s (2.39 min) faster than in real robot experiments. With the HIGH MESSAGE LOAD experiment, 90% coverage was reached in simulation 175.985s (2.93 min) faster than in real robot experiments.

3.2.3 Analysis

Results suggest that message delay has a greater impact on team performance in physical robot experiments than in simulation [1, 95]. The higher standard deviations for message delay in the real robot experiments indicate that it is more unpredictable and variable when using real robots. Therefore, when conducting simulations the effects of message loads should be modelled for more realistic performance prediction.

3.3 Conclusion

Traditionally, point-to-point communications is used to coordinate cooperative robots. However, it is often in simulation with optimum network conditions. It is hypothesized that team performance is affected by packet loss. Simulations were conducted with different amounts of packet loss. Results suggest that the more that packet are dropped, the worse the team performs. also examined team performance of when there was low and high messages loads in both simulations and physical robot experiments. Results indicate that team performance is influenced more from message loads in physical robot experiments that in simulations.

CHAPTER 4

IMPORTANCE OF SPATIAL AND TEMPORAL LOCALITY

In this chapter, there is a discussion on the impact of spatial and temporal locality in implicit cooperative multirobot coverage. Frequently, the choice of next action by each team member is not affected by information communicated by team mates unless they are close. Spatial and temporal locality can be instrumental in determining the relevance of data to immediate coverage activities, precisely the data provided by observation. Observation can be used to infer state rather than relying upon digital messages. Observation is proposed as a mechanism to propagate state in the absence of digital communications.

4.1 Locality in Implicit Cooperation

There are many methods for accomplishing distributed, cooperative exploration. Most methods can be categorized as using either explicit or implicit cooperation. In explicit cooperation, robots within the team choose actions by considering self-utility as well as the utility of actions accomplished by teammates. In order to maintain accurate data on the utility of others' actions, communication of action utility or the state variables that are used to calculate utility must be communicated through the team. If the communications are intermittent, due to failures, overloading or distance, it is difficult to keep data current and these methods may experience a degradation in performance. In implicit cooperation, there is no agreement on task assignment. A specific task is not directly communicated to other members. Therefore, the focus is on implicit cooperation.

4.1.1 Coverage Algorithm

In an effort to reduce or eliminate reliance on digital messaging, it is instructive to understand when and how communicated data is used in implicit cooperation. Many models for multirobot teamwork assume robots have a shared joint utility with synchronized task execution and communication [96]. I start with the a formal event-driven model for asynchronous control and communication from [97].

Definition 1 *Event-driven control and communication law has the sets:*

(i) A*, a set containing the null element, called communication alphabet–elements of A are called messages;*
(ii) $W^{[i]}$*,* $i \in I$*, called the processor state sets; and*
(iii) $W_o^{[i]} \subseteq W^{[i]}$*,* $i \in I$*, sets of allowable initial values;*

with the following definition for coverage:

- *I = {1,...,n}, I is called the set of unique identifiers (UIDs) for each robot,*
- *group of robots are moving in bounded exploration region* $\mathbb{R}^2$*,*
- $X^{[i]}$*:* $X^{[i]} \in \mathbb{R}^2$*, each robot's position within exploration region,*
- p*:* $p \in \mathbb{R}^2$ *defines positions within the exploration region,*
- $occ - map$*:* p *x {Open, Occluded, Unknown}, defines occupancy status of the position,*
- $f - map$*:* p *x {True, False, Off}, defines whether position is a frontier,*
- $W^{[i]}$*:* $occ - map$*,* $f - map$*, state for each robot,*
- A*:* p *x {Open, Occluded}* $\cup$ *{null}, state update messages*

and the following maps:

(i) Message functions
  (a) $msg - gen^{[i]}$*:* $W^{[i]}$ *x* $\mathbb{A}_{env}^{[i]} \rightarrow A$*,* $i \in I$*, called message-generation function.*
  (b) $msg - rec^{[i]}$*:* $W^{[i]}$ *x* $A \rightarrow W^{[i]}$*,* $i \in I$*, called message-reception function.*
(ii) $stf^{[i]}$ *:* $W^{[i]}$ *x* $\mathbb{A}_{env}^{[i]} \rightarrow W^{[i]}$*,* $i \in I$*, called the state-transition function.*
(iii) $ctl^{[i]}$*:* $X^{[i]}$ *x* $W^{[i]} \rightarrow U^{[i]}$*,* $i \in I$*, called (motion) control functions, where* $U^{[i]}$ *is the selected action based on the state.*

Each robot has access to its own physical position, $X^{[i]}$, and state, $W^{[i]}$. $W^{[i]}$ contains two state structures, an occupancy map (occ-map) and frontier status (f-map). The occupancy map contains a discretized state indicating whether a point has been visited (open or occluded) or not (unknown). In frontier-based exploration [98], the edge between open and unknown space is targeted for visiting in order to expand the known space. The list of frontier points to visit can be maintained as a list (discussed in the next section) or in a corresponding spatial map. The advantage of a spatial map is the addition of frontier specific states (i.e. true, false, off). The frontier states of true, false, and off indicate whether the space is a frontier, not a frontier, or a previous frontier, respectively. Each map is represented by a binary grid indicating the state of that point or cell.

The *state transition function* and the *message generation function* are combined into the STF_MSG-GEN for clarity. The combined function, STF_MSG-GEN, updates the state based on local sensing of open and occluded areas as well as generates messages based on the updates. The sensor covers the area which is modeled using a range-limited visibility sensor [97]. This sensor produces a set of points $\mathbb{A}_{env}$ that are visible from the current position $X^{[i]}$ and within the range r. Using this information, the state is updated for visited points to indicate both open and occluded status (Algorithm 4.1, lines 6-27). As the status of points changes from UNKNOWN to OPEN or OCCLUDED, messages are generated and sent to the team (lines 15 and 17 in Algorithm 4.1). The MSG-REC function maps the messages to state updates $W^{[i]}$ with the incoming status (lines 28 - 40 in Algorithm 4.1).

Another approach advocated in [4] is to select frontiers only from points that are adjacent to self-discovered open cells (results in removing lines 34-38 in Algorithm 4.1). This results in robots only having a partial view of globally unvisited locations. However, this approach reduces the likelihood of multiple robots selecting the same exploration region which ultimately decreases interference and speeds coverage.

The CTL function (shown in Algorithm 4.1, lines 41-50) maps the state and the current position to the next unvisited goal location. The next goal location is chosen from the frontier list

Algorithm 4.1: Frontier-based Exploration Part 1

```
Robot Network Robot team with range-limited sensing, r, and sensing of own
  position in unknown bounded environment, p ∈ ℝ^2 ;
Alphabet A: p x {Open, Occluded} ∪ {null} ;
Processor State W^[i] = (occ-map, f-map), where
occ-map: p x {OPEN, OCCLUDED, UNKNOWN}, initially unknown for all p ;
f-map: p x {FALSE, TRUE, OFF}, initially false for all p
/* Combine state transition with message generation
   function since both rely upon the same input        */
function stf_msg-gen(W^[i], 𝔸_env);
  msgs= null;
  foreach o ∈ 𝔸_env do
      if occ-map(o) is UNKNOWN then
          if f-map(o) is TRUE then
              f-map(o) is OFF
          end
          if perimeter(o) and dist(X^[i], o) < r then
              occ-map(o) = OCCLUDED;
              msgs = msgs ∪{W_o^[i], OCCLUDED};
          else
              occ-map(o) = OPEN;
              msgs = msgs ∪{W_o^[i], OPEN};
              foreach a ∈ adjacent(o) do
                  if f-map(a) is FALSE and status(a)==UNKNOWN then
                      f-map(a) = TRUE
                  end
              end
          end
      end
  end
return (W^[i], msgs);
```

Algorithm 4.1: Frontier-based Exploration Part 2

```
function msg-rec( W^[i], msgs);
    foreach non-null message (W_rcvd,status) ∈ msgs do
        occ-map(rcvd) = status ;
        if f-map(rcvd) is TRUE then
            f-map(rcvd) is OFF
        end
        foreach a ∈ adjacent(rcvd) do
            if f-map(a) is FALSE and status(a)==UNKNOWN then
                f-map(rcvd) is TRUE
            end
        end
    end
return W^[i]

function ctl(X^[i],W^[i], 𝔸_env);
    pGoal={null};
    pLength=∞;
    foreach pt ∈ {f-map(p)==TRUE} do
        if pLength < pathLength(X^[i],pt) then
            pLength=pathLength(X^[i],pt);
            pGoal=pt
        end
    end
return followPath(X^[i],p_goal,𝔸_env);
```

that represents the edge between open space and unknown space (f-map(p) == TRUE). The goal is selected as the point with the shortest path to the current robot position. Once the exploration goal is determined to be open or occluded, a new goal is selected.

4.1.2 Implications of greedy exploration

Action selection can be analyzed to determine when information from teammates affects action choice. The coverage algorithm can be described as a graph coverage algorithm that uses a greedy exploration strategy to completely traverse the graph [99]. Figure 4.1 illustrates this notion. As points (also referred to as vertices) are expanded, their unvisited child vertices are added to the queue of future vertices to be visited. The algorithm is greedy since the next goal vertex is selected based on shortest path from the current robot location.

Observation 1 OFTEN THE NEXT GOAL VERTEX IS ADJACENT TO A RECENTLY VISITED VERTEX Some vertices (referred to as α) are vertices on the frontier that are adjacent to vertices within the sensor footprint (distance of $r + \tau$ where τ is small). These vertices have been added to the unvisited vertices list during this time step. The rest of vertices (referred to as β) on the frontier result from unexpanded vertices discovered during previous time steps. The shortest path to α vertices will always be less than or equal to the shortest path to β vertices (there are no obstacles between this vertex and the current robot position according to the sensors). Only once all current α vertices have been visited will the nearest β vertex be visited (backtrack). Visiting a β vertex may result in the addition of new α vertices. If the branching factor > 1, expansions of α vertices equal backtracking to β vertices. For all branching factors > 2, expansions of α vertices outnumber backtracking to β vertices.

Observation 2 ONLY INFORMATION RECEIVED FROM TEAMMATES THAT CONCERNS VERTICES THAT ARE NEAR THE CURRENT ROBOT POSITION AFFECT THE NEXT ACTION SELECTION. To affect action selection via intra-team communication, a message must denote that the current goal vertex has been visited. A robot, $rbt_i, i \in I$, at $X^{[i]}$ determines its closest unvisited

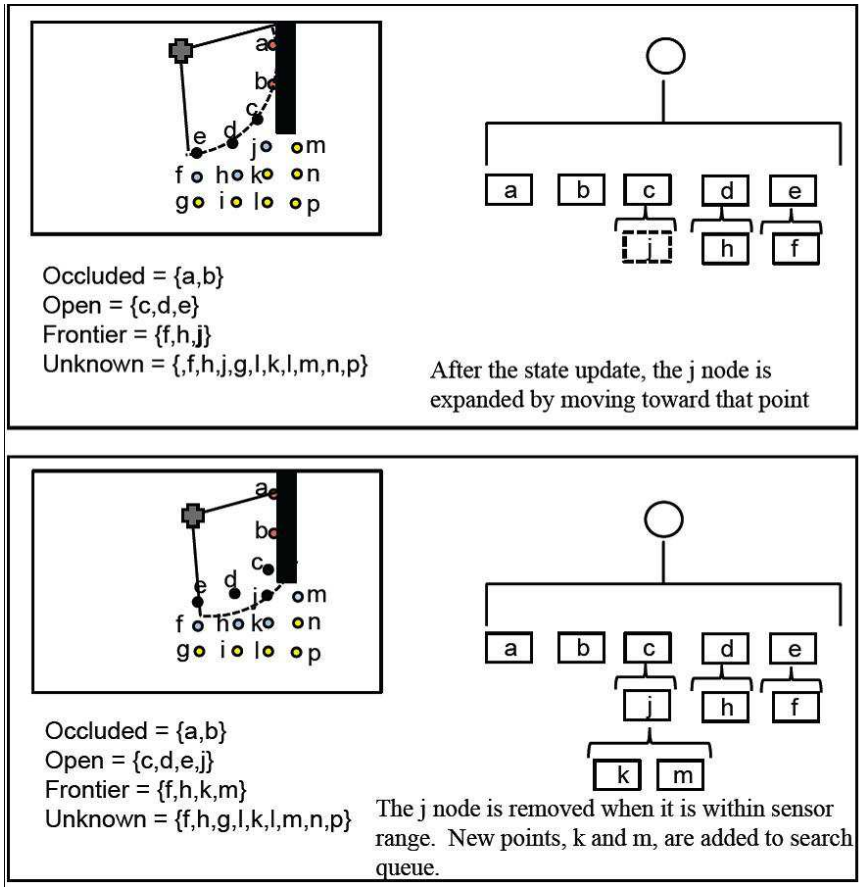

Figure 4.1: Frontier exploration redefined as a search tree. As vertices are expanded, their unvisited child vertices are added to the queue of future vertices to be visited. The algorithm is greedy since the next goal vertex is selected based on shortest path from the current robot location.

vertex is p_i. Based on the MSG-REC mapping function, a message that denotes p_i as being OPEN or OCCLUDED would remove it from the unvisited vertex list of rbt_i. Based on the STF_MSG-GEN mapping function, this message is only generated when the vertex in question falls with the sensor range of another robot, $rbt_j, j \neq i, j \in I$. If p_i is an α vertex, then $|X^{[i]} - X^{[j]}| < 2r + \tau$, denoting the proximal sensor ranges. Removing p_i from the unvisited list of rbt_i a new unvisited goal is set to the next closest unvisited vertex.

Incidentally, action selection can also be affected if a message results in a new unvisited vertex closer to the robot than the current goal vertex. In the Yamauchi [98] approach, the list of unvisited vertices is generated based on the edge between OPEN and UNKNOWN space. Only messages that change vertices (OCCUPIED$\longrightarrow$OPEN) that are closer to rbt_i than p_i create a new edge that becomes the goal vertex. Changing the status from OPEN $\longrightarrow$OCCUPIED has the effect of removing the edge and any attached UNKNOWN vertices are not visited. Also the new path to the goal vertex may be made longer (circumventing obstacles). Generally, vertices are not changed from OPEN to OCCUPIED as these are usually transient obstacles. Instead a probabilistic method (such as occupancy grids) [100] is appropriate.

4.2 Experimental Results

Analysis of goal selection data collected from robots using Algorithm 4.1 is shown in Figures 4.2 and 4.3. The grid is defined as 0.25 cells over a 6m x 6m experiment area. Detail of physical experimental setup is reported in Appendix A. Distance between the previous goal and the new goal is calculated and graphed using a histogram. Since travel, goal calculation, and path planning take time, α goals tend to be slightly more than the cell width but less than 1 meter.

Communication between teammates should affect action selection to reduce duplicate coverage. However, unless the teammates are close ($2r + \tau$), the selection of α vertices is unaffected. Such communications impacts β vertices when backtracking to previously unexpanded vertices. Figure 4.4 illustrates the distance between robots when messages are sent and the time between

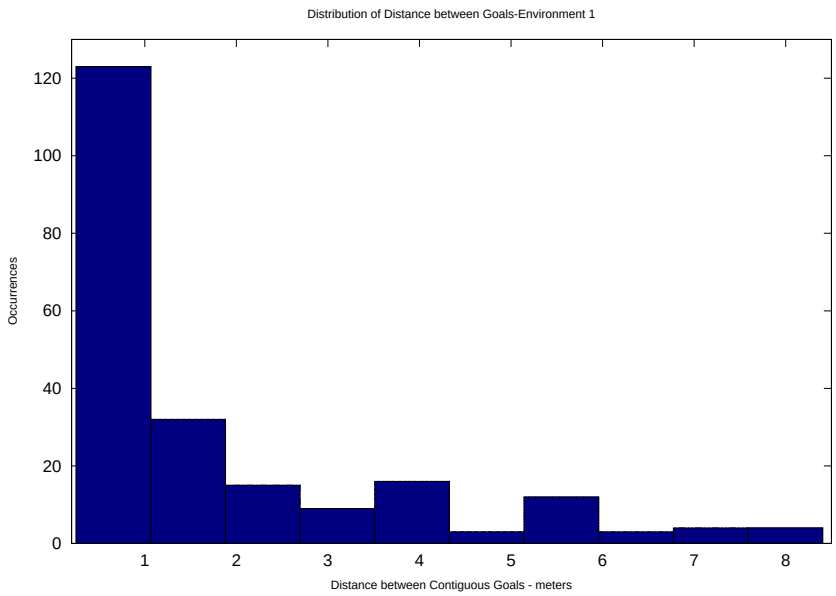

Figure 4.2: Goal selection of α and β vertices are examined in physical experiments. α vertices (defined here as distances below 1 meter) dominate the goals. The environment is a 6m x 6m obstacle free test region.

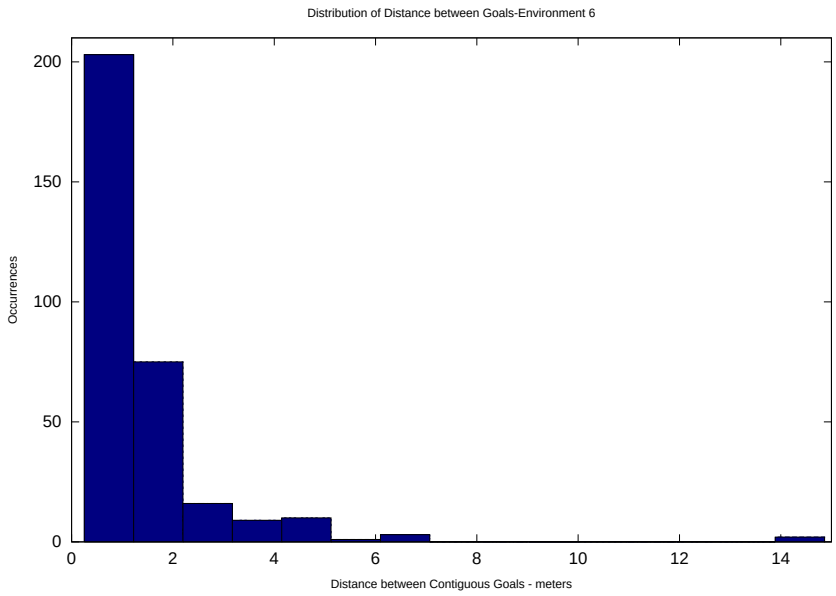

Figure 4.3: Goal selection of α and β vertices are examined in physical experiments. α vertices (defined here as distances below 1 meter) dominate the goals. The environment is a 6m x 6m test region containing four obstacles.

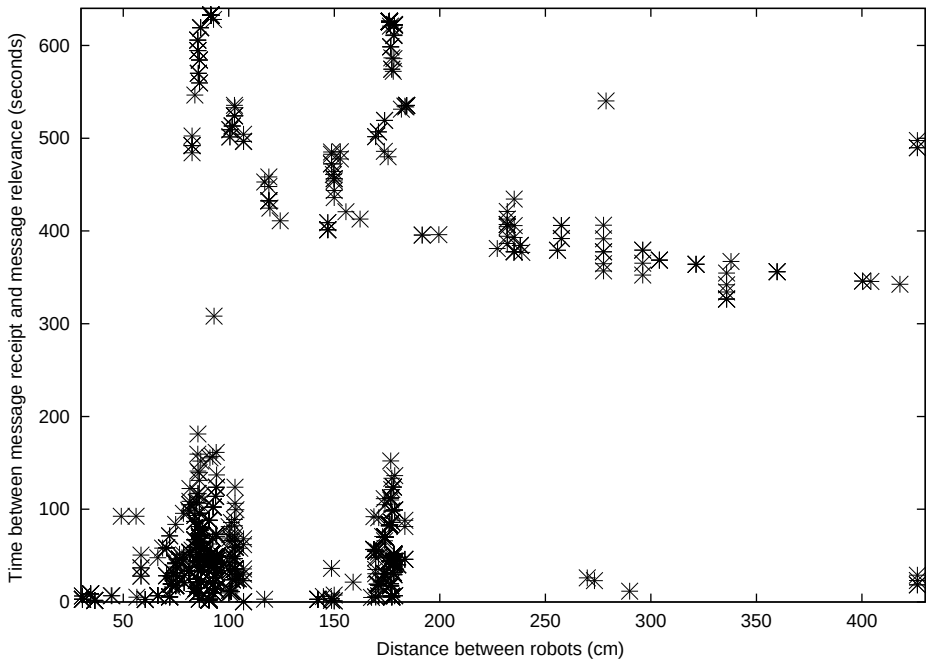

Figure 4.4: Distance between robots when messages were sent versus the time between messages were received and relevant. Messages that are communicated between robots that are close have a shorter time between being sent and being needed. The environment was 6x6 with one obstacle in the middle.

message receipt and message relevance. A message becomes relevant when it is used to determine goal selection. Messages that are communicated between robots that are close have a shorter time between being sent and being used. Given that α vertices occur more frequently than β vertices, many communications do not have an immediate impact unless the two robots are already relatively close.

4.3 Conclusion

Information from robots that are close in both time and space are more likely to be immediately useful. One approach may be to use a spatially distributed message storage. Coverage status for a region is stored locally (i.e. RFID) and can be received when local to the transmitter. Another approach allows robots to observe locations of nearby robots only to infer state and/or intent, rather than transmit messages. Observation is more likely to occur when robots are close, precisely when the information is the most useful.

CHAPTER 5

OBSERVATION-BASED COOPERATION

In this chapter, observation as a mechanism to propagate state in the absence of digital communications is proposed. Observations is used to update state based on the predicted actions of nearby robots. Areas that are observed as being searched by another robot are pruned from the search tree. Robots backtrack to pruned search nodes after there is no new area to visit. Although observation cannot provide a complete global picture of the coverage state, it is proposed that observation can support cooperation and provide significant action insight.

An experimental comparison between observation-based and message-based implicit cooperation is provided. Observation-based cooperation exploits temporal and spatial locality and provides better exploration rates than non- communicative approaches and comparably to direct communication approaches.

5.1 Approach

5.1.1 Formal Description

The event-driven model for asynchronous control and communication from [97] is augmented to update the state and action control based on the proximity of teammates. The most important change is the replacement of the message functions with observation-based functions.

Definition 2 *Observation-based Cooperation Control and Communication Law:*

(i) $W^{[i]}$, $i \in I$, *called the processor state sets;*
(ii) $W_o^{[i]} \subseteq W^{[i]}$, $i \in I$, *sets of allowable initial values;*
(iii) $\mathbb{A}_{rbt}^{[i]}$, *a set containing the null element, called robot sensing alphabet; and*
(iv) $\mathbb{A}_{env}^{[i]}$, *a set containing the null element, called environment sensing alphabet;*

with the following definition for coverage:

- *I = {1,...,n}, I is called the set of unique identifiers (UIDs) for each robot,*
- *group of robots are moving in bounded exploration region* $\mathbb{R}^2$*,*
- $X^{[i]}$*:* $X^{[i]} \in \mathbb{R}^2$*, each robot's position within exploration region,*
- p*:* $p \in \mathbb{R}^2$ *defines positions within the exploration region, and*
- $occ - map$*:* p *x {Open, Occluded, Unknown}, defines occupancy status of the position,*
- $f - map$*:* p *x {True, False, Pruned, Off}, defines whether position is a frontier,*
- $W^{[i]}$*:* $occ - map$*,* $f - map$*, state for each robot,*

and the following maps:

(i) *(*$obs - trig^{[i]}$*,* $obs - rec^{[i]}$*) are called the observation-trigger function and observation-reception function*
   (a) $obs - trig^{[i]}$*:* $i \in I$*,* $\mathbb{A}^{[i]}_{rbt} \rightarrow$ *{true, false} ,*
   (b) $obs - rec^{[i]}$*:* $i \in I$*,* $X^{[i]}$ *x* $W^{[i]}$ *x* $\mathbb{A}^{[i]}_{rbt} \rightarrow W^{[i]}$

(ii) $stf^{[i]}$*:* $W^{[i]}$ *x* $\mathbb{A}^{[i]}_{env} \rightarrow W^{[i]}$*,*$i \in I$*, called the state-transition function*

(iii) $ctl^{[i]}$*:* $X^{[i]}$ *x* $W^{[i]}$ *x* $\mathbb{A}^{[i]}_{env} \rightarrow U^{[i]}$ *,*$i \in I$*, called (motion) control functions, where* $U^{[i]}$ *is the selected action based on the state.*

5.1.2 Initialization and Updates of State

In OBSERVATION-BASED COOPERATION, each robot knows its physical position, $X^{[i]}$, and state, $W^{[i]}$. The state, $W^{[i]}$, includes the occupancy map ($occ - map$) and frontier map ($f - map$). Possible states for the $occ - map$ are unknown, open, and occluded. Possible states for $f - map$ include unknown, frontier, visited, and pruned. The visited state refers to visited frontiers and the pruned state refers to area covered by other robots. The $occ - map$ and $f - map$ are initially set to unknown. The *state-transition function* updates the state based on local sensing of open and occluded area and maintains the frontier map (Algorithm 4.1, lines 3 - 23).

5.1.3 Implicit Communication (Observation)

The OBS-REC function maps observations to state updates by prohibiting points by the observed robot from becoming frontiers (lines 24 - 33). In addition, if the observed is already in the

Algorithm 5.1: Observation-based Cooperation Exploration Part 1

```
Robot Network Robot team with range-limited sensing, r, and sensing of own
position in unknown bounded environment, p ∈ ℝ^2 ;
Alphabet A: p x {Open, Occluded} ∪ {null} ;
Processor State W^[i] = (occ-map, f-map), where
occ-map: p x {OPEN, OCCLUDED, UNKNOWN}, initially unknown for all p
;
f-map: p x {FALSE, TRUE, PRUNED, OFF}, initially false for all p
function stf(W^[i], 𝔸_env);
  foreach o ∈ 𝔸_env do
      if occ-map(o) is UNKNOWN then
          if f-map(o) is TRUE then
              f-map(o) is OFF
          end
          if perimeter(o) and dist(X^[i], o) < r then
              occ-map(o) = OCCLUDED;
          else
              occ-map(o) = OPEN;
              foreach a ∈ adjacent(o) do
                  if f-map(a) is FALSE and occ-map(a) is UNKNOWN then
                      f-map(a) = FRONTIER
                  end
              end
          end
      end
  end
return (W^[i]);
```

Algorithm 5.1: Observation-based Cooperation Exploration Part 2

```
function obs-rec( W^[i], A_rbt);
    foreach obs ∈ A_rbt do
        {obs_region} = proximity(obs);
        foreach o_r ∈ {obs_region} do
            if f-map(o_r) is FRONTIER then
                f-map(o_r) = PRUNED
            end
        end
    end
return W^[i]  function ctl(X^[i],W^[i], A_env);
    pGoal={null};
    pLength=∞;
    foreach pt ∈ {f-map(p)==TRUE} do
        if pLength < pathLength(X^[i],pt) then
            pLength=pathLength(X^[i],pt);
            pGoal=pt
        end
    end
    if pGoal==null then
        foreach pt ∈ {f-map(p)==PRUNED} do
            if pLength < pathLength(X^[i],pt) then
                pLength=pathLength(X^[i],pt);
                pGoal=pt
            end
        end
    end
return followPath(X^[i],p_goal,A_env);
```

$FRONTIER$ state, then the robot updates the state of the area as being $PRUNED$. Pruning involves removing subtrees, or search area, that may be irrelevant at the time [101]. This indicates that the area is assumed to be covered by another robot and can be visited, or backtracked to, at a later time. Observations affect frontiers but do not affect occupancy maps. Updates to the occupancy maps may cause problems with path planning.

Goal Selection

The CTL function creates a path from the robot position to its next goal or unvisited location. The closest goal to the current robot position is selected (lines 41 - 50). Previous work suggests that observation errors affect coverage completion. As a robot observe another robot it makes assumptions that are at times inaccurate resulting in a degradation in performance at the end of a coverage task [91]. Therefore, first the robot attempts to choose the closet vertex in the $FRONTIER$ state. If there are no vertices in the $FRONTIER$ state left, then the robot backtracks the closest vertex in the $PRUNED$ state. Backtracking corrects some observation errors by eliminating the possibility that it was not searched by another robot [101]. As a result, robots return back to the frontiers they marked off as being covered by other robots (lines 42 - 50).

Benefits of Pruning and Backtracking

Pruning only occurs when robots are temporally and spatially close enough to observe one another. As a robot observes another robot, it removes search area that is assumed covered by another robot allowing coverage to expand. Although those areas are removed from their search, backtracking enables robots to revisit these areas for some assurance they are being covered. Backtracking results in a more complete coverage by providing some redundancy toward the end of the search.

This algorithm is referred to as OB-BASED COOPERATION W/ BACKTRACKING. The algorithm in which robots do not backtrack is referred to as OB-BASED COOPERATION.

Practical Considerations

There are many practical considerations to successfully using observation to support cooperation. First, the ability to accurately identify the position (range and bearing) of nearby robots is a challenge. Although this is an active field of research, several approaches seem promising including shape classifiers [102, 103], features classifiers [104], or neural network classifiers [105].

Second, no robot in the team has a complete view of the work to be completed. This may prevent the team from completing the exploration. Previous work on similar algorithms (partial shared state) shows the initial performance boost gained through reduced interference overshadows slower coverage rates after 90% of the exploration is complete.

Finally, although the algorithms presented presume a simplistic spatial decomposition (points or cells), other methods for persisting coverage progress and occupancy are more advantageous. Occupancy grids, [100], provide a probabilistic view of occupancy incrementally composed from sensor data. Thresholding is used to determine cell status (unknown, open, occupied). Stachniss and Burgard [106] prefer to use probabilities to indicate the amount of coverage in a cell rather than the probability of coverage. Either method can be used within the presented algorithms.

5.2 Simulations

To show the usefulness of observation in cooperative exploration, present algorithm comparisons in simulation. There is a comparison of implicit cooperation approach based on messaging to share state (DIRECT COMM described in Section 5.1), with the observation-based approaches without backtracking (OBSERVATION-BASED COOPERATION) and with backtracking (OBSERVATION-BASED COOPERATION W/ BACKTRACKING), and NO COMM where robots each complete a frontier exploration uninformed by the actions or state of others.

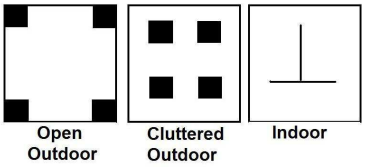

Figure 5.1: Simulation environments 1 and 2 were 16x16 meters with four obstacles representing a stadium and city block.

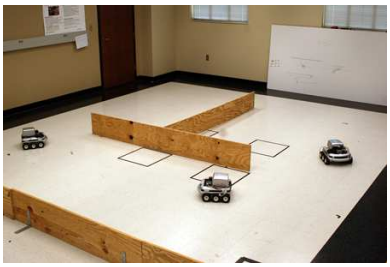

Figure 5.2: Simulation environment 3 was 6x6 meters with obstacles representing two rooms and a hallway.

5.2.1 Simulation Setup and Results

To investigate the properties of each approach, experiments were conducted in the 3-D physics-based simulator, Webots. The robots' locations were determined by the robots' odometry, a global positioning system (GPS), and a compass. To simulate robots operating in similar real world environments, a GPS was added for localization. The controller was written in the C programming language and experiments were performed on a Dual Core 2.33 GHz machine running Linux with 2GB of RAM. Additional information on simulation setup can be found in Appendix A.

Twenty trials for each approach were conducted with a three-robot team in both large and small environments for comparison. The large environments include cluttered and open areas

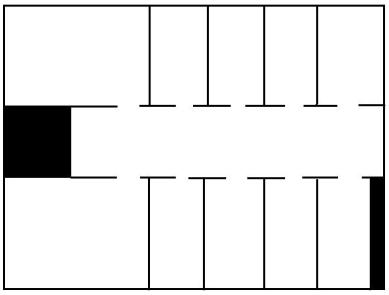

Figure 5.3: The 20x27 simulated office environment configuration

	Approach	50% Cover Time (m)	σ	90% Cover Time (m)	σ
Open	No Comm	13.24	2.98	34.41	2.90
	Direct Comm	7.51	1.23	19.99	2.79
	Ob Coop	9.19	2.27	25.40	6.76
	Ob Coop w/ Backtracking	7.81	1.39	20.46	3.39
Cluttered	No Comm	11.27	4.10	34.13	11.89
	Direct Comm	7.51	3.35	23.70	12.01
	Ob Coop	7.19	2.80	18.58	4.68
	Ob Coop w/ Backtracking	9.07	2.21	22.84	6.26
Indoor	No Comm	1.02	0.08	2.94	0.85
	Direct Comm	0.67	0.02	2.79	0.58
	Ob Coop	0.68	0.02	2.94	0.66
	Ob Coop w/ Backtracking	0.71	0.06	2.45	0.78
Office	No Comm	30.87	11.75	-	-
	Direct Comm	19.01	5.02	52.76	15.87
	Ob Coop	20.28	18.67	-	-
	Ob Coop w/ Backtracking	16.06	6.91	122.92	74.45

Table 5.1: Averages for 50% and 90% coverage for each approach in all environments in simulation.

that represent large outdoor areas (Figure 5.1 and 5.2) and an office environment (Figure 5.3) similar to one used in [94]. The small indoor environment is used for comparison to the real world environment (Figure 5.2). Figures 5.4, 5.5, 5.6, and 5.7 compare average coverage over time for each approach. Table 5.1 shows the averages for 50% and 90% coverage of the environment.

5.2.2 Analysis of Simulations

In NO COMM, robots each complete the frontier exploration without sharing state or intent. Without a cooperative strategy, robots duplicate exploration. NO COMM achieves slow rates of exploring new areas. In the cluttered and indoor environments in Figure 5.1, OB-BASED COOPERATION W/ BACKTRACKING performed the best (Figures 5.5 and 5.6). However, in the open outdoor environment, DIRECT COMM and OB-BASED COOPERATION W/ BACKTRACKING performed comparably (Figure 5.4). In OBSERVATION-BASED COOPERATION, the state updates remove frontiers around the observed position of teammates to reduce duplicate work without direct messaging. There are three phases that influence performance. Since the robots start close to each other, robots have the ability to use their sensing of nearby robots to disperse allowing for initially

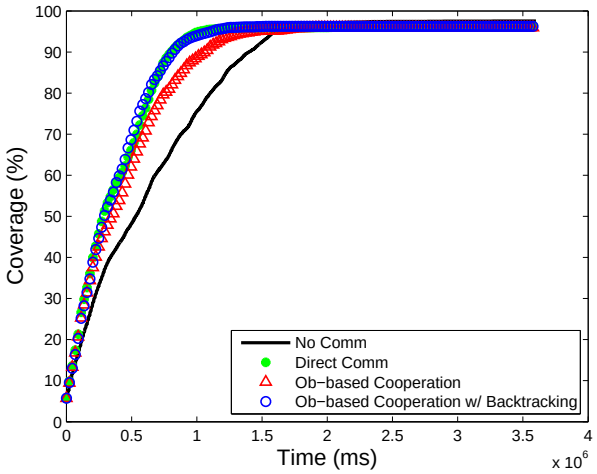

Figure 5.4: Coverage over time for all approaches in the simulated open outdoor environment.

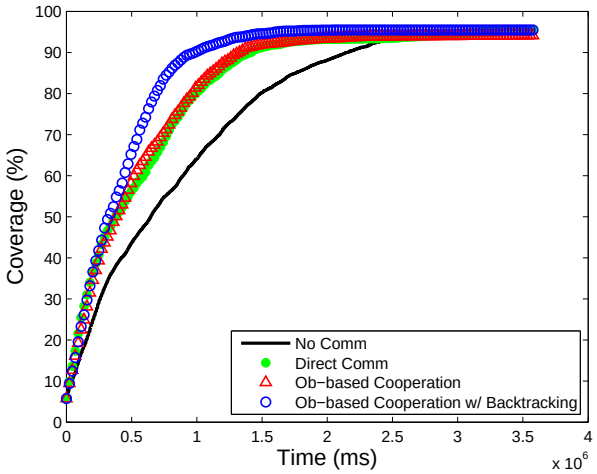

Figure 5.5: Coverage over time for all approaches in the simulated cluttered outdoor environment

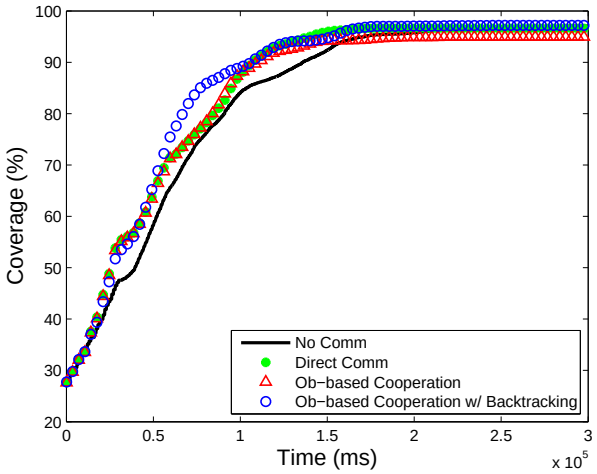

Figure 5.6: Coverage over time for all approaches in the simulated indoor environment.

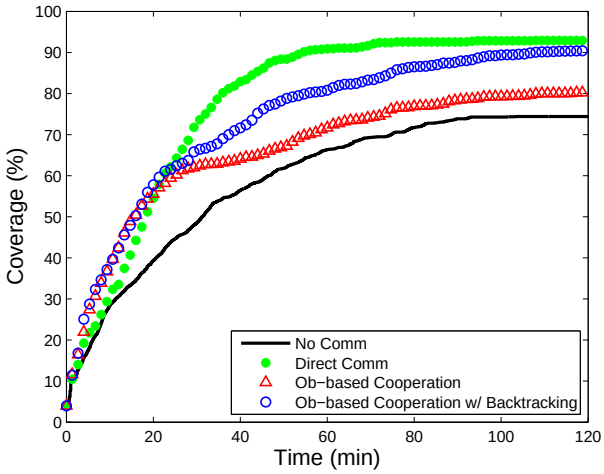

Figure 5.7: Coverage over time for all approaches in the simulated office environment.

fast coverage. Once robots no longer sense each other, visiting unpruned areas allows for comparable coverage. As robots backtrack through pruned areas, the rates at which new area is explored increases slowly.

To investigate the use of OBSERVATION-BASED COOPERATION, simulations were conducted in a more complex environment (Figure 5.3). The difference between the approaches is more significant in the cluttered office environment as DIRECT COMM performance increases over the other approaches (Figure 5.7). Since the environment contains many walls, robots are less likely to observe one another affecting coverage. In some cases with NO COMM and OB-BASED COOPERATION, 90% coverage was not reached after two hours (Table 5.1). However, in all the environments, the use of pruning and backtracking allows for better coverage in OB-BASED COOPERATION W/ BACKTRACKING than if robots do not backtrack (OB-BASED COOPERATION).

5.3 Physical Robot Experiments: Simulated Observation

To validate the results gathered from simulations, real robot experiments were conducted. The physical robot experiments provide realistic view of the performance given the inaccuracies of the incorporated sensors. In this set of experiments, observation is simulated by having a central server communicate to each robot when there is another robot within two meters. This approach allows evaluation of the algorithm with perfect observation in which robots can observe with a 360° field of view.

5.3.1 Experimental Setup and Results

Experiments were conducted using K-Team Koala robots equipped with a Dual Core 1.60 GHz machine running Ubuntu with 2GB of RAM (Appendix A reports detail on physical robot experimental setup). The environments used in the experiments included a 6x6 open area and an area similar to the indoor environment in simulations (Figure 5.2). Five trials with a three-robot team were executed.

	Approach	50% Cover Time (m)	σ	90% Cover Time (m)	σ
Open	No Comm	0.60	0.11	3.17	0.43
	Direct Comm	0.57	0.60	2.72	3.20
	Ob Coop	0.73	0.07	3.57	0.70
	Ob Coop w/ Backtracking	0.75	0.06	2.96	6.26
Cluttered	No Comm	1.77	0.43	5.96	1.34
	Direct Comm	1.57	0.43	3.71	0.74
	Ob Coop	1.56	0.05	4.68	1.42
	Ob Coop w/ Backtracking	1.23	0.48	4.50	1.08

Table 5.2: Averages for 50% and 90% coverage for each approach for the real environments.

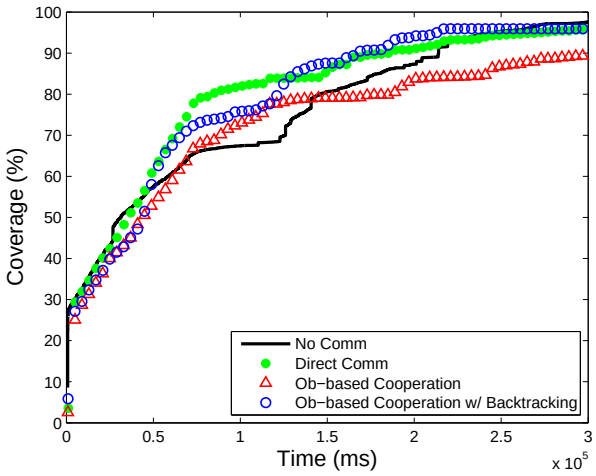

Figure 5.8: Coverage over time for all approaches in the open real world environment.

Data collected is presented in Table 5.2 and includes the average amount of time it takes the robot team to cover the area by 50% and 90%. Figures 5.8 and 5.9 show averages of coverage over time. Averages were calculated from the ground truth position data gathered by the Stargazer localization system.

5.3.2 Analysis

The results from the experiments largely matched the simulated results. The OB-BASED COOPERATION W/ BACKTRACKING provided better exploration rates than The OB-BASED COOPERATION and comparable to DIRECT COMM. Figures 5.10, 5.11, and 5.12 show robot dispersal and coverage of a full deployment in the real cluttered environment. Robots using DIRECT COMM and

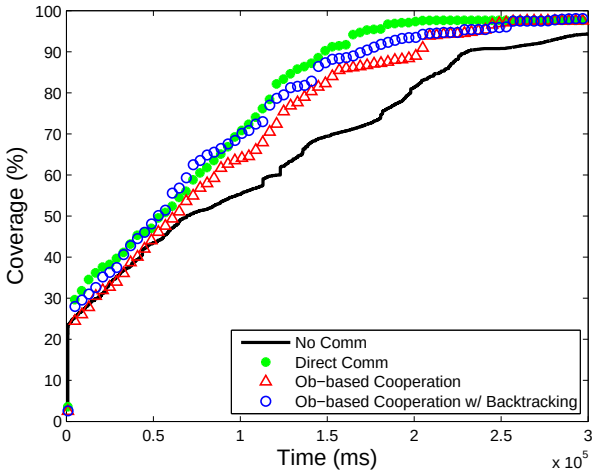

Figure 5.9: Coverage over time for all approaches in the cluttered real world environment.

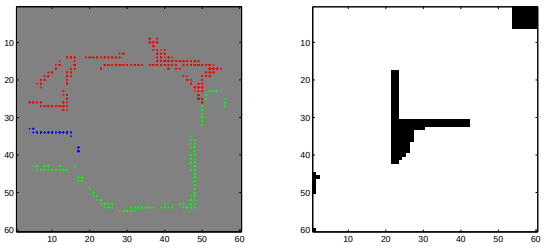

Figure 5.10: The paths of a robot team using DIRECT COMM.

OB-BASED COOPERATION did not cover part of the environment. The advantages of backtracking to pruned areas is illustrated with OB-BASED COOPERATION W/ BACKTRACKING as robots have more complete coverage.

Both approaches have elements that affect performance more in real experiments than in simulations. DIRECT COMM is subject to communication network limitations [1] and propagated odometry error [107]. OB-BASED COOPERATION W/ BACKTRACKING is sensitive to assumptions based on observation as robots assume areas are being covered [91]. Results suggest that observation-based approaches provide alternatives to digital messaging.

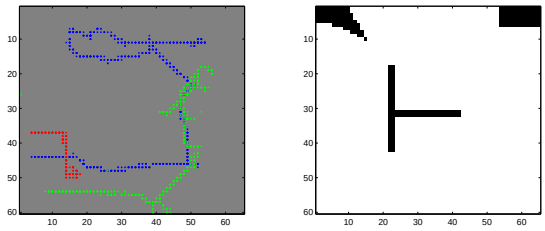

Figure 5.11: The path of a robot team using OB COOP.

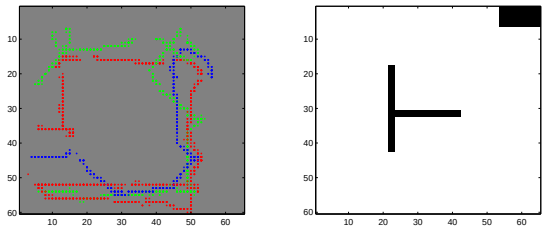

Figure 5.12: The path of a robot team using OB COOP W/BACKTRACKING.

Figure 5.13: Robots used the ARToolkit for observation of other robots. The ARToolkit uses image processing to detect markers placed throughout the environment.

5.4 Physical Robot Experiments: Visual Observation

In the previous set of experiments, observation was mimicked using simulation of observed position. In this set of experiments, robots use vision to observe one another. In addition, while the previous set of experiments were performed in small indoor environments, this section presents results for a larger area.

5.4.1 Experimental Setup and Results

Three K-team Koala robots were used for physical experiments. The real robots were equipped with a Hokuyo URG laser range finder to detect distances. The robots were also equipped with a Dual Core 1.6GHz machine running Ubuntu with 2GB of RAM. The controller was written in the C programming language and integrated in the Robot Operating System (ROS), an open-source software framework that provides libraries and tools to help in robot software development [108]. ROS was used because it has tools for image processing incorporated.

For observation, markers from ARToolkit [109] were placed on top of the robots along with a Playstation Eye camera with a 60° field of view to detect markers (Figure 5.13). The ARToolKit has libraries that use image processing to identify markers and to calculate positional data. Placing

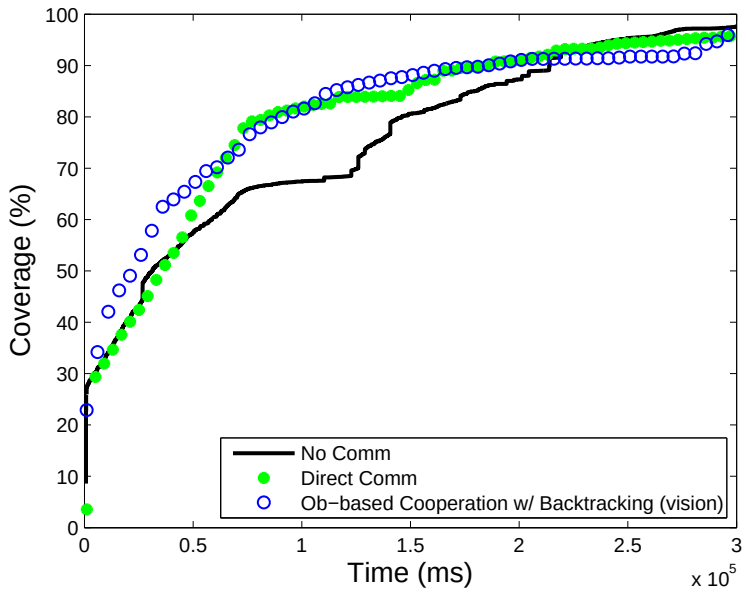

Figure 5.14: Coverage over time for the physical open environment.

	Approach	50% Cover Time (m)	σ	90% Cover Time (m)	σ
Open	No Comm	0.60	0.11	3.17	0.43
	Direct Comm	0.57	0.60	2.72	3.20
	Ob Coop w/ Backtracking (vision)	0.33	0.19	3.11	1.18
Cluttered	No Comm	1.77	0.43	5.96	1.34
	Direct Comm	1.57	0.43	3.71	0.74
	Ob Coop w/ Backtracking (vision)	1.23	0.48	4.50	1.08

Table 5.3: Averages for 50% and 90% coverage for each approach for the real environments.

markers on top of robots allowed estimation of distance between and bearing of nearby robots. Additional details on experimental setup can be found in the appendix.

Open and Cluttered Indoor Environment

For comparison, the OBSERVATION-BASED COOPERATION WITH BACKTRACKING algorithm using vision was conducted in the same open and cluttered environment as the previous experiments. Figures 5.14 and 5.15 shows the average coverage over time for all robots. In the open

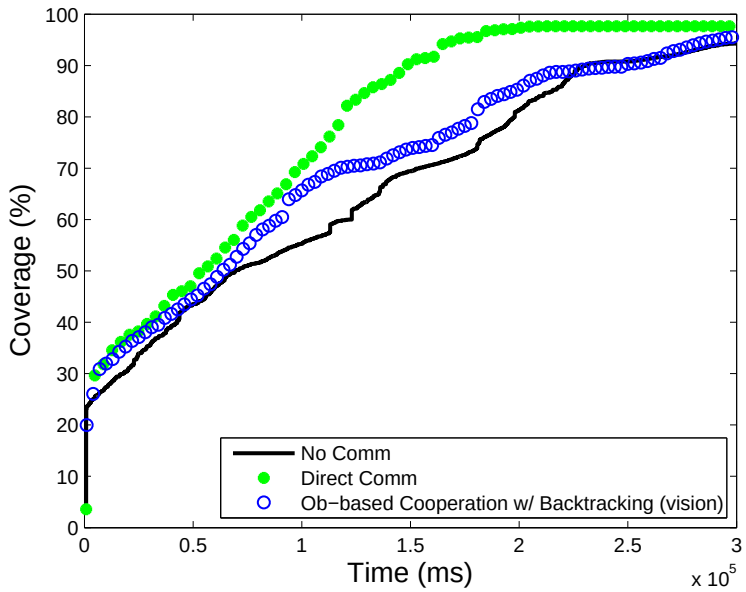

Figure 5.15: Coverage over time for the real cluttered environment.

environment, team performance was comparable to when robots used direct communications (Table 5.3). However, in the cluttered environment, there was better team performance with direct communications during all trials. Robots were not able to see each other as often in the cluttered environment because of obstacles.

Observation of other robots varied greatly between trials. In some trials a robot may not observe any nearby team members but in other trials as many as 200 nodes would be pruned as a

Table 5.4: Nodes that are pruned but not eventually searched by team mates can cause an observation-based search to be incomplete.

	Pruned Nodes (mean)	σ	Pruned but not searched nodes (mean)	σ	Error Rate
Open	150.83	64.56	12.75	18.35	8.5%
Cluttered	121.11	58.42	25.45	24.30	21.0%
Wide-open Area	142.50	60.62	55.00	4.24	38.6%

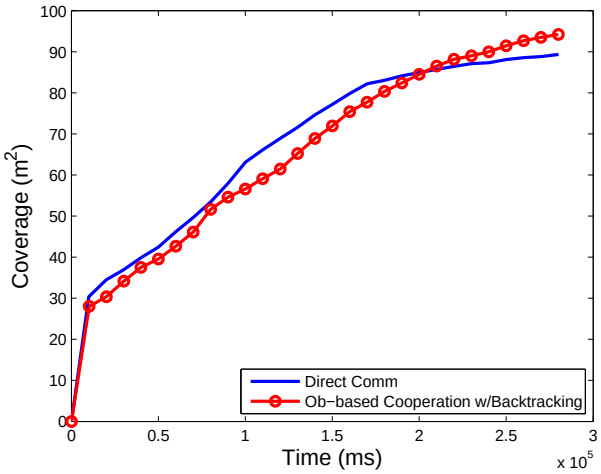

Figure 5.16: Results for wide area coverage for a robot team using direct communication and OBSERVATION-BASED COOPERATION paradigms.

result of observations. In the cluttered environments, robots observed team mates less often than in the open environment.

Pruning areas that are not subsequently searched can cause a search not to complete. Table 5.4 shows the average number of nodes that were pruned correctly (searched) during observation-based exploration OBSERVATION-BASED COOPERATION. Fewer errors occurred in open areas (8% of nodes) where previously pruned areas were not searched by team mates. Cluttered areas had both a higher error rate (21% of pruned nodes remained unsearched by team mates) as well as a lower rate of pruning areas.

Wide open environment

In this set of experiments, robots explored a wide open indoor area of a building atrium. Figure 5.16 illustrates the total area covered versus time for direct communications and OBSERVATION-BASED COOPERATION in wide-open area. The results suggest that direct communication has a comparable performance to OBSERVATION-BASED COOPERATION. Although there were 38.6% of pruned area that was not searched, robots were able to efficiently search the area. In the beginning of exploration, robots spread out as they observe one another. This allows them to search and expand in different sub-trees.

5.4.2 Analysis

Similarly to the previous sections' results, observation-based cooperation demonstrates a three-slope phase with faster coverage towards the end. In the beginning, exploration is slow as robots observe other robots and prune those areas from their search tree. In the middle of exploration, robots expand on their search by determining the nearest unpruned α vertices that is least likely to be visited by another robot (determine by previously observed bearing). Towards the end of exploration, robots move further apart allowing for faster coverage as they expand on separate parts of the search tree.

5.5 Conclusions

An approach that uses observation to infer state to coordinate robots is presented. Observation is used to direct immediate and long term action selection through state updates. It is shown how observation can leverage spatial and temporal locality and use pruning and backtracking to improve coverage. Experiments illustrate the usefulness of OBSERVATION-BASED COOPERATION to coordinate robots in exploration. Results suggest that observation can provide an alternative to direct communications in wide open areas.

OBSERVATION-BASED COOPERATION is discussed as a viable strategy for multirobot exploration. Although observation does not support exploration throughout the process, it does provide a boost to the exploration rate through a reduction in duplicate coverage for a good portion of the exploration. Given it is counter intuitive that observation can provide sufficient direction for exploration, a rationale is presented for how observation can leverage spatial and temporal locality in much the same way as other direct messaging implicit cooperation strategies.

CHAPTER 6

SECTOR-SEARCH WITH RENDEZVOUS

In cooperative multirobot systems, communication can speed up completion, reduce redundancy, and prevent interference between robots. Interference occurs when robots try to occupy the same space resulting in time spent avoiding. Therefore, reducing interference can improve team performance. In this chapter, an approach that overcomes communication limitations while reducing interference is presented.

SECTOR SEARCH WITH RENDEZVOUS allows robots to explore in sectors, or designated areas, resulting in less interference. Instead of continuously passing messages throughout the entire exploration, robots explore and then rendezvous to communicate what was found. Small teams of intelligent robots were used to cover a large open area that represents an outdoor environment such as a city block. SECTOR SEARCH WITH RENDEZVOUS is compared to other direct communication and rendezvous approaches. It is hypothesized that SECTOR SEARCH WITH RENDEZVOUS is efficient in coordinating robots in coverage tasks of wide open areas.

6.1 Approach

6.1.1 Frontier-based Exploration

All approaches in this chapter use the coverage algorithm, frontier-based exploration. In a frontier-based [98] algorithm, robots recursively explore an unknown area while building a cellular representation [100]. Robots detect frontiers and visit them to gain more knowledge about the environment thereby recursively exploring. Frontiers, unknown, open, and occupied space are represented with occupancy grids. Robots use a distance sensor to determine whether a space is occupied, open, or a frontier cell. When the robots communicate, they explicitly broadcast messages

to all other robots. These messages are deliberately transmitted and received through point-to-point communications. Open cells information is shared. Additional details on the exploration algorithm and communications used are reported in Appendix A.4 and A.5.

6.1.2 Sector Search with Rendezvous

The SECTOR SEARCH WITH RENDEZVOUS approach is designed to allow a robot team to explore an unknown environment. The goal of the robots is to collaboratively explore a large open area without using continuous message passing throughout the entire exploration. Instead, robots rendezvous to exchange information about what they have discovered. The following assumptions are made:

- Robots are equipped with a global positioning sensor (GPS) for localization and to share a global map.
- The environment is wide open and unknown (e.g. outdoor).
- Robots start close to each other and share a common frame of reference.
- Robots are homogeneous and capable of exploring the environment on their own.
- Robot teams are small allowing for a limited number of messages at rendezvous.
- When robots communicate, messages are intentionally transmitted and received from robot to robot.

The Algorithm

With sector search, robots explore a pre-agreed coverage areas using frontier-based exploration. As an asynchronous approach, robots select frontier areas based on individual utility allowing fault tolerance against individuals being disable or out of range.

The area is divided into designated areas for each robot. The team decides on designated areas from similar start locations. For example, a robot may be designated to explore the most eastern part of the environment while another is designated to explore the most northern part. Because dividing the area into designated areas can cause uneven assignment, robots can explore outside their sectors after completing their own sector.

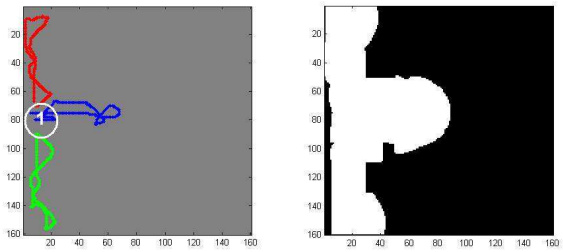

Figure 6.1: The robots search in sectors, or designated areas, and then rendezvous to exchange map information. The paths of the robots are shown, with the starting point and initial rendezvous location at point 1. The diagram on the right shows the area that has been covered.

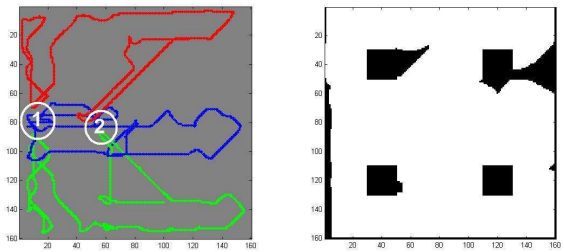

Figure 6.2: The robots search in sectors, or designated areas, and then rendezvous to exchange map information. The paths of the robots are shown, with the starting point and initial rendezvous location at point 1. Subsequent rendezvous locations are determine after each meeting. The second rendezvous location is shown at point 2. The diagram on the right shows the area that has been covered.

Although the robots are exploring designated areas within sectors, they still execute the frontier-based exploration algorithm within their area. Sector search is beneficial in large environments with few robots because robots spread out more in their assigned areas. If the environment is large and they do not explore in designated areas then they may end up exploring in the same area. In addition, when robots rendezvous, it is more likely that they are sharing new area information if they explore in designated areas.

Sector Designation

Although the environment is unknown, robots can be designated bearings from the start point. In the beginning of coverage, robots start at adjacent locations. This area of the environment can be represented as a semicircle. The arc of a semicircle always measures to 180°. If there are N robots then $180°/N$ indicates the angles of the sectors robots should cover. For example, in the experiments of this work, there are three robots. Therefore, robots distribute at 60° angles from each other.

Rendezvous Locations

One of the most important factors in rendezvous is choosing rendezvous locations. It is difficult to determine locations in an unknown environment. Robots explore and build a map individually to share with other robots. If a robot receives conflicting map information from other robots, it uses the latest information received to update its map. Although, they share map information when they meet, they may not always have the complete map. For example, robots may not agree on a location because the location does not lie in every robot's map. In addition, robots may not make it to a location in time. Both cases result in robots having partial and varying maps. However, either way robots can still effectively cover the environment since they search in separate areas.

In our algorithm, rendezvous locations are determined individually by each robot. The robots start exploration in adjacent locations. The initial rendezvous location is at the start point. Subsequent rendezvous locations are decided at each rendezvous. If robots have successful rendezvous, they will have similar maps after meeting. On the contrary, if they do have successful rendezvous and are not able to communicate, then they will not have similar maps. However, after each scheduled rendezvous, robots choose the most central point in their map as the next rendezvous location.

Robots are equipped with a global positioning sensor (GPS) for position estimation since odometry errors can accumulate over time. Although a solution for multirobot SLAM [18] could be used instead, there are other issues that would need to be addressed [110]. For instance, in larger areas, there can be a data association problem if landmarks cannot be uniquely identified. However,

sector search can be used in conjunction with a SLAM solution. For example, sector search mimics how human teams coordinate search to cover an area to gain situational awareness quickly. This initial situational awareness can be used to plan subsequent actions using other approaches such as mapping.

Figures 6.1 and 6.2 illustrate the paths of a three-robot team exploring in sectors and their first two rendezvous locations. The location at point 1 shows the starting points of the robots and the initial rendezvous location. After the initial rendezvous, the robots each decide on their next rendezvous located at point 2.

Rendezvous Parameters

Other important aspects of rendezvous are choosing the time between when robots should rendezvous and the duration of the rendezvous itself. If robots do not meet often enough then large amount of messages may be passed when they do meet. However, if they meet too often, they spend more time meeting than exploring.

The duration of a rendezvous can also be influential on performance. If the duration is not long enough, then robots may not have enough time to make it to the rendezvous point and exchange information. In contrast, if the duration is too long, then they are wasting time after communicating. These two parameters are tested by two variations of the algorithm: SECTOR REND 1 and SECTOR REND 2.

In SECTOR REND 1, communication is limited in that only information that was discovered after the last rendezvous will be shared. In other words, environment information is shared only once at the next rendezvous. Therefore, if robots are not able to make it to the rendezvous location, they will miss the information sent by other robots. However, with SECTOR REND 2, robots send all the map information they have gathered from beginning of execution every time they meet. Therefore, SECTOR REND 2 should be assigned a longer rendezvous duration than SECTOR REND 1.

For both SECTOR REND 1 and SECTOR REND 2, the initial rendezvous is set to happen after two minutes of exploring. Subsequent rendezvous are every five minutes. This gives the robots enough time to cover a good portion of the environment and return. The rendezvous durations for SECTOR REND 1 and SECTOR REND 2 are two minutes and six minutes; respectively. Additional time is needed for SECTOR REND 2 since the robots communicate all environment information that was found from the beginning of the search every rendezvous.

6.1.3 Approaches for Comparison

To demonstrate SECTOR SEARCH WITH RENDEZVOUS, it is compared to NO COMM, PROX COMM, FULL COMM, which are frontier-based exploration approaches with no rendezvous, and the ROLE-BASED approach described in [111].

In the NO COMM approach, there is no direct communications between robots. Each robot only relies on their perception of the environment. However, with the FULL COMM and PROX COMM approaches, robots explicitly broadcast messages to all other robots. The only difference between the FULL COMM and PROX COMM is that with PROX COMM robots only communicate with other robots that they sense within five meters. The following section describes the ROLE-BASED presented in [111] in more detail.

6.1.4 Role-based with Dynamic Team Hierarchy

In [111], de Hoog et al. demonstrated role-based exploration with a dynamic team hierarchy. As noted in the related work section, some robots (Relays) relay information between robots and a central command centre while others (Explorers) continue to explore using frontier exploration. Subsequent rendezvous points are pushed deeper and deeper into the environment, leading to full exploration. When two robots enter one another's communication range, they examine whether it is advantageous for them to swap roles, i.e. for each to take on the other's task. If yes, they swap roles and exploration continues.

The main differences between sector-based exploration and role-based exploration are:

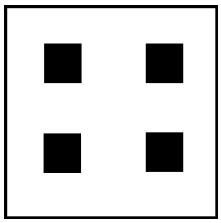

Figure 6.3: The environment for simulations was 16x16 meters with four 4x4 meter obstacles.

- Role-based exploration requires a team hierarchy whereas sector-based exploration does not
- In role-based exploration, meetings occur between robots pairwise whereas in sector-based exploration all robots meet simultaneously
- Role-based exploration typically aims to gather all information at a single, central location, which is not the case in sector-based exploration

6.2 Simulations

6.2.1 Simulation Setup

The SECTOR REND, NO COMM, FULL COMM and PROX COMM approaches were simulated in the 3-D physics-based simulator, Webots [93]. The controller was written in the C programming language and experiments were performed on a Dual Core 2.33 GHz machine running Linux with 2GB of RAM. More detail of simulation setup is reported in Appendix A.

The performance of each approach is measured by examining the percentage of coverage and amount of time taken to cover an area. The NO COMM approach serves as the baseline experiment for comparison against the other approaches. The goal of the experiments was to determine whether communicating only during rendezvous is a good alternative to continuous message passing. The environment used to run the simulations was 16×16 meters with four 4×4 meter obstacles representing a city block (Figure 6.3). Twenty trials for each approach were conducted with a three-robot team.

The best approach for SECTOR REND and the ROLE-BASED approach were simulated in MRESim, a JAVA-based simulation environment [66]. Currently, MRESim assumes perfect sensor data and localization. However, the simulator is used for the purpose of comparison to previous

	50% Cover Time (m)	σ	90% Cover Time (m)	σ
No Comm	6.63	1.42	20.97	3.83
Full Comm	4.71	0.44	16.98	3.17
Prox Comm	5.35	0.63	15.92	3.32
Sector Rend 1	7.04	0.33	15.37	3.47
Sector Rend 2	7.06	0.31	14.95	2.17

Table 6.1: Averages for 50% and 90% coverage for each approach.

work. Each approach was simulated in the same environment shown in Figure 6.3. Three trials for each approach were conducted with a three-robot team.

6.2.2 Simulation Results

Data collected include the average time that the team required to explore at least 50% and 90% of the environment for each approach. Data also includes all averages of coverage over time.

Figure 6.4 illustrates the comparison between robots using NO COMM, FULL COMM and PROX COMM. Inspecting the figure, FULL COMM and PROX COMM have similar performance. However, FULL COMM covers more area at first and PROX COMM covers more at the end of exploration. On average, PROX COMM reaches 90% almost one minute faster than FULL COMM (Table 6.1).

Figure 6.5 illustrates the SECTOR SEARCH WITH RENDEZVOUS algorithms. Overall, when compared with no communications they performed better. However, there is a point in the beginning of execution where the robots using no communications covered more than those using SECTOR REND 1 and SECTOR REND 2. In Table 6.1, it shows that at 50% coverage, no communications is outperforming by more than 0.41 seconds. In addition, SECTOR REND 1 and SECTOR REND 2 are very similar in performance to each other until about 90% at which then SECTOR REND 2 had better performance.

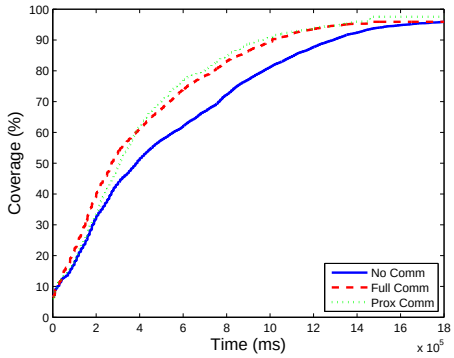

Figure 6.4: A comparison of coverage and time for a robot team with no communications, direct communications during an entire exploration, and direct communications only when they are within close proximity of another robot.

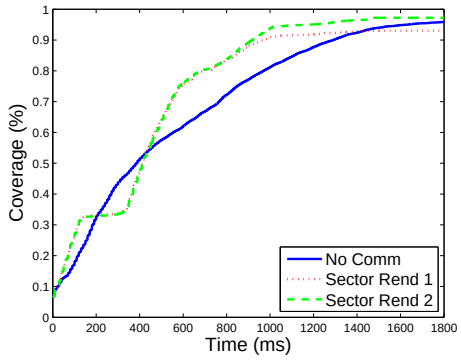

Figure 6.5: A comparison of coverage and time for a robot team with no communications and algorithms of the SECTOR SEARCH WITH RENDEZVOUS.

As shown in Figure 6.6, SECTOR REND 2 and PROX COMM have comparable results. SECTOR REND 2 on average covered 90% of the area quicker than PROX COMM (Table 6.1). However, PROX COMM demonstrated better coverage time between 30% and 70% (Figure 6.5).

Table 6.1 shows the average times for when at least 50% and 90% of the area was covered by the three-robot team. On average, FULL COMM reached 50% coverage the faster. However, SECTOR REND 2 approached 90% quicker. Figure 6.5 shows PROX COMM and SECTOR REND 2 performed similarly after 90%.

Finally, Figure 6.7 illustrates coverage and time for three runs each of the SECTOR REND and the ROLE-BASED approaches. While ROLE-BASED exploration outperforms sector-based exploration in the early stages, the advantages of sector based exploration soon become evident as

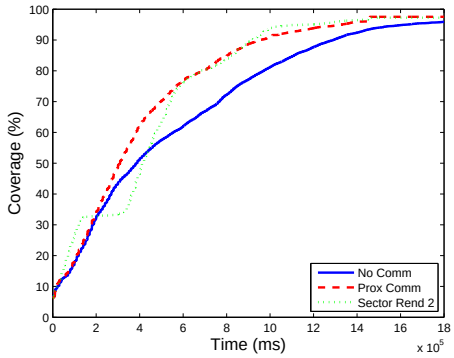

Figure 6.6: A comparison of coverage and time for a robot team with no communications, direct communications only when they are within close proximity of another robot, and SECTOR SEARCH WITH RENDEZVOUS approach.

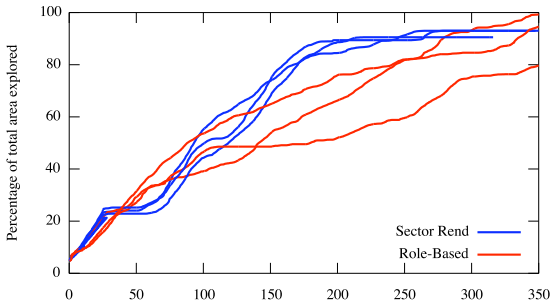

Figure 6.7: Coverage and time for Role-based and Sector Search Rendezvous.

its coverage improves. This confirms that sector-based exploration is a better solution for open, outdoor environments (whereas role-based exploration typically has the greatest advantages in communication-limited, narrow and obstacle filled environments).

6.2.3 Analysis

Results suggest that the alternative of having robots only communicate when in close proximity PROX COMM with each other performs better than communicating during the entire exploration (FULL COMM). With FULL COMM, robots have more messages to process allowing for less time to explore. It also demonstrates how information that is close in both time and space is useful in immediate action selection.

SECTOR REND 1 and SECTOR REND 2 had comparable performance but show a decrease in performance to the point where it was under no communications during the beginning of execution (Figure 6.5). It illustrates how taking the time to rendezvous can affect coverage. Instead of exploring, robots are meeting to exchange information. In addition, SECTOR REND 1 decrease towards the end of execution due to the amount of communications and longer rendezvous duration.

Although PROX COMM and SECTOR REND 2 had the best performance (Figure 6.6), they both have advantages and disadvantages. When robots only communicate when they are in close proximity with each other, they do not have to waste time to rendezvous. However, in wide open environments with small teams, there is a chance that they may not ever be in close distance of each other. In wide open environments, rendezvous is beneficial because robots make an effort to meet instead of relying on the chance they may come in close proximity of other robots.

The utilization of rendezvous to overcome communication limitations is illustrated in both the SECTOR REND and the ROLE-BASED approaches. They are similar because they use rendezvous but task assignment is different. For example, in ROLE-BASED, robots are heterogeneous and assigned the specific roles of Explorers and Relays. In SECTOR REND, robots are homogeneous and are assigned designated areas to search. The combination of sector search and rendezvous is beneficially unique. Although sector search could be implemented without rendezvous, it offers another way to coordinate robots by preventing additional interference and duplication.

6.3 Conclusions

An approach that uses SECTOR SEARCH WITH RENDEZVOUS to coordinate robots in an exploration task of a unknown large open environment is presented. Our approach is compared to other communication paradigms in simulation. Results suggest that SECTOR SEARCH WITH RENDEZVOUS leads to gains over approaches having no communications, is comparable to when robots communicate only with other robots in close proximity, and outperforms role-based exploration in open environments.

CHAPTER 7

CONCLUSION

Cooperative multirobot systems are advantageous in wide area coverage applications such as search and rescue, surveillance, and toxic waste clean-up. Deploying robots instead of humans can prevent human casualties as well as speed up completion. To cooperatively complete tasks so that work is not duplicated, robots must communicate.

Communication among team members is achieved explicitly or implicitly. In explicit communication, messages are intentionally transmitted and received from robot to robot. In implicit communication, robots observe the environment and other robot actions. Although many systems use explicit communications, persistent intra-team digital communications is not guaranteed. For example, limited communications bandwidth is a common issue with explicit communications. If robots are exchanging large amounts of information then they run into the risk of receiving incomplete information due to limited bandwidth. In addition, latency of message loads has been found to affect team performance. As message loads increase, team performance degrades. Therefore, alternative approaches other than continuous message passing throughout exploration is needed.

In this dissertation, first the importance of spacial and temporal locality of information shared is discussed. In Chapter 5, there is description of how OBSERVATION-BASED COOPERATION leverages spatial and temporal localities of information to coordinate a robot team. Information shared between robots that are close has more influence on action selection than information shared between robots that are farther apart. In Chapter 6, SECTOR SEARCH WITH RENDEZVOUS is presented. In this approach, robots explore an area in sectors and then communicate what was found

during periodic rendezvous. Instead of continuous message passing, robots meet and exchange information after exploring their sector. Results from simulations and physical robot experiments indicate that both OBSERVATION-BASED COOPERATION and SECTOR-SEARCH WITH RENDEZVOUS can serve as alternatives to explicit communications.

7.1 Contributions

The following are the main contributions to robotics literature.

- An explanation of how spatial and temporal locality provide insight on what information is relevant in subsequent coverage activities. Information shared between robots that are close has more influence on action selection than information shared between robots that are farther apart.
- A cooperation paradigm that uses implicit communications in the form of observation to infer state rather than rely on digital messaging. Observation leverages spatial and temporal locality and is likely to occur when robots are close and information is more useful.
- An cooperation paradigm with spatial rendezvous and sector search. Robots use rendezvous to exchange information instead of continuously passing messages. Robots explore an environment in sectors, or designated areas, and periodically meet to communicate map information of what they explored.

7.2 Future Work

This dissertation describes cooperation paradigms that overcome communications in multi-robot wide area coverage. The work presented can be extended in the following ways.

- Evaluating performance of cooperation algorithms in the metrics of energy consumption (path length), memory overhead, and the speedup of adding additional robots.

- Conducting experiments in larger outdoor environments with additional robots including unmanned aerial vehicles.
- Implementation of hybrid communication schemes that utilizes both implicit and explicit communications. In one approach, robots implicitly communicate to other robots that are close and explicitly communicate with robots that are farther away.
- Applying biological and cognitive sciences for inspiration for controlling robots. Since many previous biological inspired approaches are concentrated on social insects, studying other animal societies could be beneficial.

REFERENCES

[1] S. Dawson, B. Wellman, and M. Anderson, "Using simulation to predict multi-robot performance on coverage tasks," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, pp. 202–208.

[2] P. E. Rybski, S. A. Stoeter, M. Gini, D. F. Hougen, and N. Papanikolopoulos, "Effects of limited bandwidth communications channels on the controlof multiple robots," in *2001 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2001. Proceedings*, vol. 1, 2001.

[3] A. Akella, G. Judd, S. Seshan, and P. Steenkiste, "Self-management in chaotic wireless deployments," *Wireless Networks*, vol. 13, no. 6, p. 737755, 2007.

[4] M. Anderson and N. Papanikolopoulos, "Implicit cooperation strategies for multi-robot search of unknown areas," *Journal of Intelligent and Robotic Systems*, vol. 53, no. 4, pp. 381–397, 2008.

[5] R. C. Arkin and J. Diaz, "Line-of-sight constrained exploration for reactive multiagent robotic teams," in *AMC 7th International Workshop on Advanced Motion Control*, 2002, pp. 455–461.

[6] I. Rekleitis, V. Lee-Shue, A. P. New, and H. Choset, "Limited communication, multi-robot team based coverage," in *IEEE International Conference on Robotics and Automation*, vol. 4, 2004, pp. 3462–3468.

[7] D. Meier, C. Stachniss, and W. Burgard, "Coordinating multiple robots during exploration under communication with limited bandwidth," in *ECMR*, 2005, pp. 26–31.

[8] N. Roy and G. Dudek, "Collaborative robot exploration and rendezvous: Algorithms, performance bounds and observations," *Autonomous Robots*, vol. 11, no. 2, pp. 117–136, 2001.

[9] P. Grassé, "La reconstruction du nid et les coordinations interindividuelles chezbellicositermes natalensis etcubitermes sp. la théorie de la stigmergie: Essai d'interprétation du comportement des termites constructeurs," *Insectes sociaux*, vol. 6, no. 1, pp. 41–80, 1959.

[10] R. Beckers, O. Holland, and J. Deneubourg, "From local actions to global tasks: Stigmergy and collective robotics," in *Artificial life IV*, vol. 181. Citeseer, 1994, p. 189.

[11] T. Kazama, K. Sugawara, and T. Watanabe, "Collecting behavior of interacting robots with virtual pheromone," in *Distributed Autonomous Robotic Systems 6*, 2007, pp. 347–356.

[12] S. Koenig, B. Szymanski, and Y. Liu, "Efficient and inefficient ant coverage methods," *Annals of Mathematics and Artificial Intelligence*, vol. 31, no. 1, pp. 41–76, 2001.

[13] K. Sugawara, T. Kazama, and T. Watanabe, "Foraging behavior of interacting robots with virtual pheromone," in *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 3. IEEE, 2004, pp. 3074–3079.

[14] T. Balch and M. Hybinette, "Social potentials for scalable multi-robot formations," in *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, vol. 1. IEEE, 2000, pp. 73–80.

[15] C. Boesch and H. Boesch,'Hunting behavior of wild chimpanzees in the tai national park," *American Journal of Physical Anthropology*, vol. 78, no. 4, pp. 547–573, 1989.

[16] J. C. Bednarz, "Cooperative hunting harris' hawks (Parabuteo unicinctus)," *Science*, vol. 239, no. 4847, p. 1525, 1988.

[17] A. Howard, M. J. Mataric, and G. S. Sukhatme, "Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem," *Distributed autonomous robotic systems*, vol. 5, pp. 299–308, 2002.

[18] A. Howard, "Multi-robot simultaneous localization and mapping using particle filters," *The International Journal of Robotics Research*, vol. 25, no. 12, p. 1243, 2006.

[19] M. Matarić, *The robotics primer*. Mit Pr, 2007.

[20] R. Brooks, "A robust layered control system for a mobile robot," *Robotics and Automation, IEEE Journal of*, vol. 2, no. 1, pp. 14–23, 1986.

[21] L. Parker, "On the design of behavior-based multi-robot teams," *Advanced Robotics*, vol. 10, no. 6, pp. 547–578, 1995.

[22] T. Balch and R. Arkin, "Behavior-based formation control for multirobot teams," *Robotics and Automation, IEEE Transactions on*, vol. 14, no. 6, pp. 926–939, 1998.

[23] Y. U. Cao, A. S. Fukunaga, and A. Kahng, "Cooperative mobile robotics: Antecedents and directions," *Autonomous robots*, vol. 4, no. 1, pp. 7–27, 1997.

[24] M. Matari, "Issues and approaches in the design of collective autonomous agents* 1," *Robotics and autonomous systems*, vol. 16, no. 2-4, pp. 321–331, 1995.

[25] T. Balch and R. C. Arkin, "Communication in reactive multiagent robotic systems," *Autonomous Robots*, vol. 1, no. 1, pp. 27–52, 1994.

[26] P. E. Rybski, A. Larson, H. Veeraraghavan, M. Anderson, and M. Gini, "Performance evaluation of a multi-robot search & retrieval system: Experiences with MinDART," *Journal of Intelligent and Robotic Systems*, vol. 52, no. 3, pp. 363–387, 2008.

[27] M. A. Batalin and G. S. Sukhatme, "Spreading out: A local approach to multi-robot coverage," *Distributed Autonomous Robotic Systems*, vol. 5, pp. 373–382, 2002.

[28] M. Batalin and G. Sukhatme, "Multi-robot dynamic coverage of a planar bounded environment," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (Submitted)*. Citeseer, 2002.

[29] R. Simmons, D. Apfelbaum, W. Burgard, D. Fox, M. Moors, S. Thrun, and H. Younes, "Coordination for multi-robot exploration and mapping," in *Proceedings of the National Conference on Artificial Intelligence*, 2000, p. 852–858.

[30] A. Wagner and R. Arkin, "Multi-robot communication-sensitive reconnaissance," in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, vol. 5. IEEE, 2004, pp. 4674–4681.

[31] S. Poduri and G. Sukhatme, "Constrained coverage for mobile sensor networks," in *IEEE International Conference on Robotics and Automation*, vol. 1, 2004, pp. 165–171.

[32] R. Zlot, A. Stentz, M. B. Dias, and S. Thayer, "Multi-robot exploration controlled by a market economy," in *IEEE International Conference on Robotics and Automation, 2002. Proceedings. ICRA'02*, vol. 3, 2002.

[33] N. Heo and P. Varshney, "A distributed self spreading algorithm for mobile wireless sensor networks," in *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE*, vol. 3. IEEE, 2003, pp. 1597–1602.

[34] M. Berhault, H. Huang, P. Keskinocak, S. Koenig, W. Elmaghraby, P. Griffin, and A. Kleywegt, "Robot exploration with combinatorial auctions," in *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 2. IEEE, 2003, pp. 1957–1962.

[35] K. Lerman and A. Galstyan, "Mathematical model of foraging in a group of robots: Effect of interference," *Autonomous Robots*, vol. 13, no. 2, pp. 127–141, 2002.

[36] B. W. Shameka Dawson and M. Anderson, "Categorizing Interference in Real Robot Experiments," in *Systems Man and Cybernetics (SMC), 2011 IEEE International Conference on*. IEEE, 2011, submitted.

[37] A. Rosenfeld, G. Kaminka, and S. Kraus, "A study of scalability properties in robotic teams," *Coordination of large-scale multiagent systems*, pp. 27–51, 2006.

[38] L. Iocchi, D. Nardi, and M. Salerno, "Reactivity and deliberation: a survey on multi-robot systems," *Balancing reactivity and social deliberation in multi-agent systems*, pp. 9–32, 2001.

[39] G. Dudek, M. Jenkin, E. Milios, and D. Wilkes, "A taxonomy for multi-agent robotics," *Autonomous Robots*, vol. 3, no. 4, pp. 375–397, 1996.

[40] E. Klavins, "Communication complexity of multi-robot systems," *Algorithmic Foundations of Robotics V*, vol. 7, p. 275–292, 2003.

[41] F. Bullo, J. Cortés, and S. Martínez, *Distributed Control of Robotic Networks*, ser. Applied Mathematics Series. Princeton University Press, 2009, electronically available at http://coordinationbook.info.

[42] P. E. Rybski, A. Larson, H. Veeraraghavan, M. LaPoint, and M. Gini, "Communication strategies in multi-robot search and retrieval: Experiences with MinDART," in *Proceedings of the Seventh International Symposium on Distributed Autonomous Robotic Systems (DARS-04)*, 2004, pp. 301–310.

[43] B. L. Wellman, S. Dawson, and M. Anderson, "The effect of packet loss on cooperative multirobot systems." in *CAINE'10*, 2010, pp. 291–296.

[44] W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun, "Collaborative multi-robot exploration," in *IEEE International Conference on Robotics and Automation, 2000. Proceedings. ICRA'00*, vol. 1, 2000.

[45] M. Dias and A. Stentz, "A comparative study between centralized, market-based, and behavioral multirobot coordination approaches," in *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 3. IEEE, 2003, pp. 2279–2284.

[46] R. Arkin, "Cooperation without communication: Multiagent schema-based robot navigation," *Journal of Robotic Systems*, vol. 9, no. 3, pp. 351–364, 1992.

[47] M. Dias, "Traderbots: A new paradigm for robust and efficient multirobot coordination in dynamic environments," Ph.D. dissertation, Carnegie Mellon University, 2004.

[48] B. Yamauchi, "Frontier-based exploration using multiple robots," in *Proceedings of the second international conference on Autonomous agents*. Minneapolis, Minnesota, United States: ACM, 1998, pp. 47–53.

[49] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.

[50] C. Kong, N. Peng, and I. Rekleitis, "Distributed coverage with multi-robot system," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*. IEEE, 2006, pp. 2423–2429.

[51] X. Zheng, S. Jain, S. Koenig, and D. Kempe, "Multi-robot forest coverage," in *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*. IEEE, 2005, pp. 3852–3857.

[52] M. B. Dias and A. Stentz, "A free market architecture for distributed control of a multirobot system," *Intelligent autonomous systems 6*, p. 115, 2000.

[53] 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.

[54] T. Sandholm, "Algorithm for optimal winner determination in combinatorial auctions* 1," *Artificial Intelligence*, vol. 135, no. 1-2, pp. 1–54, 2002.

[55] M. Dias, R. Zlot, N. Kalra, and A. Stentz, "Market-based multirobot coordination: A survey and analysis," *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1257–1270, 2006.

[56] B. Gerkey and M. Mataric, "Sold!: Auction methods for multirobot coordination," *Robotics and Automation, IEEE Transactions on*, vol. 18, no. 5, pp. 758–768, 2002.

[57] S. Sariel and T. Balch, "Real time auction based allocation of tasks for multi-robot exploration problem in dynamic environments," in *Proceedings of the AAAI-05 Workshop on Integrating Planning into Scheduling*, 2005, pp. 27–33.

[58] N. Kalra, D. Ferguson, and A. Stentz, "Hoplites: A market-based framework for planned tight coordination in multirobot teams," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*. IEEE, 2005, pp. 1170–1177.

[59] C. Tovey, M. Lagoudakis, S. Jain, and S. Koenig, "The generation of bidding rules for auction-based robot coordination," *Multi-Robot Systems. From Swarms to Intelligent Automata Volume III*, pp. 3–14, 2005.

[60] R. Zlot and A. Stentz, "Market-based multirobot coordination for complex tasks," *The International Journal of Robotics Research*, vol. 25, no. 1, p. 73, 2006.

[61] L. E. Parker, "ALLIANCE: an architecture for fault tolerant multirobot cooperation," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 2, p. 220–240, 1998.

[62] M. N. Rooker and A. Birk, "Multi-robot exploration under the constraints of wireless networking," *Control Engineering Practice*, vol. 15, no. 4, p. 435–445, 2007.

[63] M. Roth, D. Vail, and M. Veloso, "A real-time world model for multi-robot teams with high-latency communication," in *Proceedings of IROS*, 2003, pp. 2494–2499.

[64] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, and E. Osawa, "Robocup: The robot world cup initiative," in *Proceedings of the first international conference on Autonomous agents*. ACM, 1997, pp. 340–347.

[65] I. Rekleitis, A. P. New, E. S. Rankin, and H. Choset, "Efficient boustrophedon Multi-Robot coverage: an algorithmic approach," *Annals of Mathematics and Artificial Intelligence*, vol. 52, no. 2, p. 109–142, 2008.

[66] S. C. Julian de Hoog and A. Visser, "Role-based autonomous multi-robot exploration," in *International Conference on Advanced Cognitive Technologies and Applications (COGNITIVE)*, November 2009.

[67] J. de Hoog, S. Cameron, and A. Visser, "Selection of rendezvous points for multi-robot exploration in dynamic environments," in *International Conference on Autonomous Agents and Multi-Agent Systems*, 2010.

[68] P. Ulam and R. Arkin, "When good communication go bad: communications recovery for multi-robot teams," in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, vol. 4. IEEE, 2004, pp. 3727–3734.

[69] A. Visser and B. Slamet, "Including communication success in the estimation of information gain for multi-robot exploration," in *Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks and Workshops, 2008. WiOPT 2008. 6th International Symposium on*. IEEE, 2008, pp. 680–687.

[70] Y. Xu, M. Lewis, K. Sycara, and P. Scerri, "An efficient information sharing approach for large scale multi-agent team," in *Information Fusion, 2008 11th International Conference on*. IEEE, 2008, pp. 1–8.

[71] W. Sheng, Q. Yang, S. Ci, and N. Xi, "Multi-robot area exploration with limited-range communications," in *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 2. IEEE, 2004, pp. 1414–1419.

[72] M. Mataric, "Designing emergent behaviors: From local interactions to collective intelligence," in *From Animals to Animats 2: Int. Conf. on Simulation of Adaptive Behavior*, 1993, pp. 423–441.

[73] E. Ferranti, N. Trigoni, and M. Levene, "Rapid exploration of unknown areas through dynamic deployment of mobile and stationary sensor nodes," *Autonomous Agents and Multi-Agent Systems*, vol. 19, no. 2, pp. 210–243, 2009.

[74] I. A. Wagner, M. Lindenbaum, and A. M. Bruckstein, "Distributed covering by ant-robots using evaporating traces," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 5, pp. 918–933, 1999.

[75] C. Kube *et al.*, "Collective robotics: From social insects to robots," *Adaptive Behavior*, vol. 2, no. 2, p. 189, 1993.

[76] A. Drogoul and J. Ferber, "From tom thumb to the dockers: Some experiments with foraging robots," in *From Animals to Animats 2: Proceedings of the 2nd Int. Conf. on Simulation of Adaptive Behavior*, 1993, pp. 451–459.

[77] D. Payton, M. Daily, R. Estowski, M. Howard, and C. Lee, "Pheromone robotics," *Autonomous Robots*, vol. 11, no. 3, pp. 319–324, 2001.

[78] T. Hsiang, E. Arkin, M. Bender, S. Fekete, and J. Mitchell, "Algorithms for rapidly dispersing robot swarms in unknown environments," *Algorithmic Foundations of Robotics V*, pp. 77–94, 2003.

[79] J. McLurkin, "Stupid robot tricks: A behavior-based distributed algorithm library for programming swarms of robots," Ph.D. dissertation, Massachusetts Institute of Technology, 2004.

[80] J. McLurkin and J. Smith, "Distributed algorithms for dispersion in indoor environments using a swarm of autonomous mobile robots," *Distributed Autonomous Robotic Systems 6*, pp. 399–408, 2007.

[81] R. Morlok and M. Gini, "Dispersing robots in an unknown environment," *Distributed Autonomous Robotic Systems 6*, pp. 253–262, 2007.

[82] I. Wagner, Y. Altshuler, V. Yanovski, and A. Bruckstein, "Cooperative cleaners: A study in ant robotics," *The International Journal of Robotics Research*, vol. 27, no. 1, p. 127, 2008.

[83] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The international journal of robotics research*, vol. 5, no. 1, p. 90, 1986.

[84] Y. Koren and J. Borenstein, "Potential field methods and their inherent limitations for mobile robot navigation," in *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*. IEEE, 1991, pp. 1398–1404.

[85] A. Howard, M. J. Matari, and G. S. Sukhatme, "An incremental self-deployment algorithm for mobile sensor networks," *Autonomous Robots*, vol. 13, no. 2, pp. 113–126, 2002.

[86] E. Pagello, A. D'Angelo, F. Montesello, F. Garelli, and C. Ferrari, "Cooperative behaviors in multi-robot systems through implicit communication," *Robotics and Autonomous Systems*, vol. 29, no. 1, pp. 65–77, 1999.

[87] L. Parker, B. Kannan, X. Fu, and Y. Tang, "Heterogeneous mobile sensor net deployment using robot herding and line-of-sight formations," in *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 3. IEEE, 2003, pp. 2488–2493.

[88] A. Das, R. Fierro, V. Kumar, J. Ostrowski, J. Spletzer, and C. Taylor, "A vision-based formation control framework," *Robotics and Automation, IEEE Transactions on*, vol. 18, no. 5, pp. 813–825, 2002.

[89] Y. Meng, J. Nickerson, and J. Gan, "Multi-robot aggregation strategies with limited communication," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*. IEEE, 2006, pp. 2691–2696.

[90] F. Zeiger, N. Kraemer, and K. Schilling, "Commanding mobile robots via wireless ad-hoc networksa comparison of four ad-hoc routing protocol implementations," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*. IEEE, 2008, pp. 590–595.

[91] B. Wellman, S. Dawson, A. Veluchamy, and M. Anderson, "Observation-based cooperation in mobile sensor networks: A bio-inspired approach for fault tolerant coverage," in *Local Computer Networks (LCN), 2010 IEEE 35th Conference on*. IEEE, pp. 184–187.

[92] B. Wellman, S. Dawson, and M. Anderson, "Using Rendezvous to Overcome Communication Limitations in Multirobot Exploration," in *2011 IEEE Int Conf on Systems, Man, and Cybernetics - Systems Science and Engineering Track*. IEEE.

[93] O. Michel, "WebotsTM: professional mobile robot simulation," *Arxiv preprint cs/0412052*, 2004.

[94] 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.

[95] S. Dawson, B. Wellman, and M. Anderson, "The effect of multiple robots on simulation accuracy," in *Proceedings of International Conference on Robotics and Automation (ICRA) Workshop on the Role of Experiments in Robotics Research*, 2010.

[96] D. V. Pynadath and M. Tambe, "The communicative multiagent team decision problem: Analyzing teamwork theories and models," *Journal of Artificial Intelligence Research*, vol. 16, no. 1, pp. 389–423, 2002.

[97] F. Bullo, J. Cortés, and S. Martinez, *Distributed Control of Robotic Networks*. Princeton University Press, 2009.

[98] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, 1997, pp. 146–151.

[99] S. Koenig, C. Tovey, and W. Halliburton, "Greedy mapping of terrain," in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 4. IEEE, 2001, pp. 3594–3599.

[100] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, 1989.

[101] S. S. Skiena, *The Algorithm Design Manual*. Springer, 2008.

[102] V. Ferrari, F. Jurie, and C. Schmid, "From Images to Shape Models for Object Detection," *International Journal of Computer Vision*, vol. 87, no. 3, pp. 284–303, July 2009.

[103] J. Knopp, M. Prasad, and L. Van Gool, "Orientation invariant 3D object classification using hough transform based methods," in *Proceedings of the ACM workshop on 3D object retrieval*. ACM, 2010, pp. 15–20.

[104] L. Chang, M. Duarte, L. Sucar, and E. Morales, "Object Class Recognition Using SIFT and Bayesian Networks," *Advances in Soft Computing*, pp. 56–66, 2010.

[105] U. Kaufmann, G. Mayer, G. Kraetzschmar, and G. Palm, "Visual robot detection in robocup using neural networks," *RoboCup 2004: Robot soccer world cup VIII*, pp. 262–273, 2005.

[106] C. Stachniss and W. Burgard, "Mapping and exploration with mobile robots using coverage maps," *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, pp. 467–472, 2003.

[107] M. Anderson and N. Papanikolopoulos, "Reducing Sensitivity to Dead-Reckoning Error through Local Search," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), San Diego CA*, 2007.

[108] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, "Ros: an open-source robot operating system," in *ICRA Workshop on Open Source Software*, 2009.

[109] H. Kato and M. Billinghurst, "Marker tracking and hmd calibration for a video-based augmented reality conferencing system," in *Augmented Reality, 1999.(IWAR'99) Proceedings. 2nd IEEE and ACM International Workshop on*. IEEE, 1999, pp. 85–94.

[110] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (slam): Part ii," *Robotics & Automation Magazine, IEEE*, vol. 13, no. 3, pp. 108–117, 2006.

[111] S. C. Julian de Hoog and A. Visser, "Dynamic team hierarchies in communication-limited multi-robot exploration," in *IEEE International Workshop on Safety, Security, and Rescue Robotics (SSRR)*, July 2010.

Appendices

APPENDIX A

Simulations and Physical Experiments Set-up

Simulations and physical experiments were conducted to demonstrate the approaches. The control program, written in the C programming language, was essentially the same for both the simulated and real experiments. The robotic platform used was the K-Team Koala (Figure A.1).

A.1 Simulations

The simulation environment used was Webots [93], a 3-D physics-based mobile robot simulator. While other 3-D simulators could have been utilized, Webots was chosen as the tool for comparison because it provides a fast, easy to use, high fidelity simulation. It also includes typical features that are found in popular simulators. In addition, Webots provides a K-Team Koala robotic model.

The simulations were performed on a Dual Core 2.33GHz Linux machine with 2GB of RAM. A wheel encoder noise (based on a Gaussian distribution) was added to the trials ran in simulation to compensate for error in the real world. The robots in simulation used global positioning sensor (GPS) for localization as well as a laser range finder.

A.2 Physical Experiments

The K-team Koala robots were also equipped with a Dual Core 1.6GHz machine running Ubuntu with 2GB of RAM. Twenty trials were run in simulations and five trials were run in physical environment for each environment. The real robots were equipped with a Hokuyo URG laser range finder and a Hagisonic StarGazer Localization System (used to mitigate sensor error). A driver was written for the K-team Koala platform for communication between the controller and the robot's sensors and actuators.

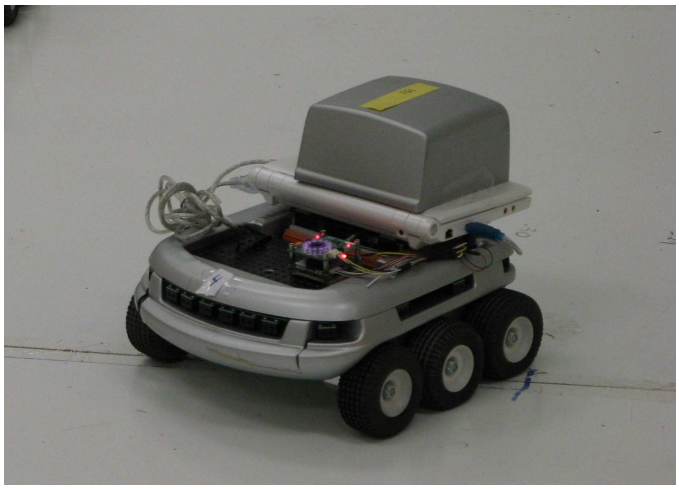

Figure A.1: K-Team Koala robots were used for robot experiments.

A.2.1 Visual Observation

The experiments for which robots used visual observation had a different implementation. The same controller was used but it was integrated with the Robot Operating System (ROS). ROS is an open-source software framework that provides libraries and tools to help in robot software development [108]. ROS was used because it has tools for image processing incorporated.

For observation, markers from ARToolkit [109] were placed on top of the robots. The ARToolKit has libraries that use image processing to identify markers and to calculate positional data. Placing markers on top of robots allowed them to estimate the distance between another robot as well as the bearing of that robot. A Playstation Eye camera with a $60°$ field of view was place under the bumper of the robots to observe markers.

A.3 Localization

Robots were equipped with a global positioning sensor (GPS) in simulations and the Hagisonic StarGazer Localization System in physical experiments for position estimation since odometry errors can accumulate over time. Although a solution for multirobot SLAM [18] could be used instead, there are other issues that would need to be addressed [110]. For instance, in larger

or homogeneous areas, there can be a data association problem if landmarks cannot be uniquely identified. As a result, loop-closing may be difficult if robots cannot return to a trajectory starting point.

A.4 Communications

In simulations, robots were equipped with an emitter and receiver for communications. In physical robot experiments, robots rely on a wireless network based on the IEEE 802.11b standards for communications.

Robots are capable of exploring the area individually. However, with cooperation and communications, the area can be covered quicker. When direct communications is utilized, robots deliberately transmits and receives messages. Robots commmunicate map information and only send messages containing whether an area is open. Anderson and Papanikolopoulos [4] demonstrate that sharing only open cells reduces the number of messages during search allowing for better performance.

A.5 Coverage Algorithm: Frontier-based Exploration

The greedy algorithm, frontier-based exploration [48], is often used for robot exploration. In the frontier-based algorithm, robots recursively explore an unknown area while building a cellular representation of a map [100]. Frontiers, or boundaries between open and unexplored area, are detected and visited to gain more knowledge about the environment. Occupancy grids [48] are used to represent the environment's unknown, occupied, open space, and frontier areas. Initially, each cell is set to unknown. Each robot uses a distance sensor to determine if an area is occupied, open space, or a frontier. As a greedy algorithm, robots choose the closet frontier to visit.