Proceedings of the Twenty-Sixth International Florida Artificial Intelligence Research Society Conference

Multirobot Coverage Using Observation-Based Cooperation with Backtracking

Briana Lowe Wellman, Shameka Dawson, and **Monica Anderson**

Abstract

In cooperative robot teams, communications can speed up execution, reduce duplication, and prevent interference. Although many systems use explicit communications, persistent intra-team digital communications is not guaranteed. One approach to this challenge is to use implicit communication to infer state rather than using digital messages. We investigate using implicit communication in the form of observation to infer state to coordinate a robot team in a coverage task. We demonstrate how pruning and backtracking a search tree can improve multi-robot coverage. Experiments were conducted to compare team performance of a robot team using observation-based cooperation to one that uses explicit communications.

Introduction

In cooperative multi-robot teams, communication is essential so that work can be divided to speed up completion, reduce redundancy, and prevent interference. 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 communications allows robots to leverage observations of behaviors of other robots or the environment to infer state and intent.

The efficiency of explicit communications is subject to the limitations of the communications network. For example, performance is affected by message latency (Dawson, Wellman, and Anderson 2010) and limited bandwidth (Rybski et al. 2001). 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 can be even more challenging. Deployed wireless communications can be chaotic since they are unplanned and unmanaged (Akella et al. 2007).

Researchers have presented work on explicit communications with network constraints and include: (Arkin and Diaz 2002) in which robots are required to maintain line-of-sight with other robots, (Meier, Stachniss, and Burgard 2005) in which message size is reduced by allowing robots to communicate polygonal representations of the map, and (Roy and Dudek 2001) where rendezvous approaches allow robots to meet up to exchange information about the environment.

Common approaches to implicit communications include: ant or swarm robots (Koenig, Szymanski, and Liu 2001) where robots leave virtual pheromones or trail markings in the environment to direct robot behavior and potential fields (Howard, Mataric, and Sukhatme 2002) in which robots are attracted to the goal and repulsed from obstacles and other robots. However, both approaches depend on local interactions where after a certain distance they can no longer influence action choices.

In this paper, we investigate using implicit communication in the form of observation to infer state to coordinate a small team of intelligent robots. Rather than using digital messaging, robots use observation of nearby robots to update state. We also discuss the impact of spatial and temporal locality of information exchanged and show how pruning and backtracking of the search tree can improve team performance. Experimental results are presented to compare observation-based and message-based cooperation.

Related Work

Researchers have proposed approaches for coordinating robot teams with consideration of network constraints. One approach involves coordinating robots with potential fields. Howard et al. (Howard, Mataric, and Sukhatme 2002) present a distributed virtual field force where robots are subject to repel other robots and obstacles causing robots to spread out. This approach depends on local interactions which after certain proximity no longer coordinate. In the proposed approach, robots can coordinate after they can no longer sense each other.

Ferranti et al. (Ferranti, Trigoni, and Levene 2009) 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 (Koenig, Szymanski, and Liu 2001) demonstrate coordinating robots to cover a terrain similar to ants. Robots communicate via markings

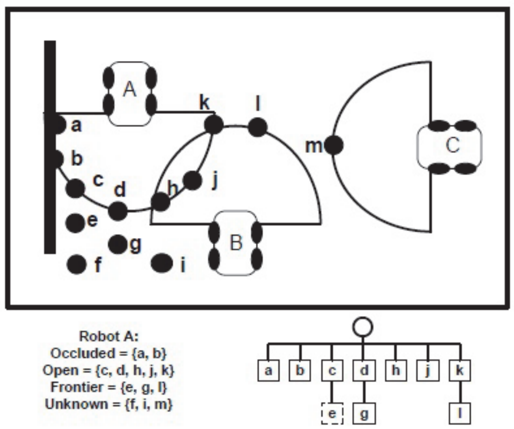

Figure 1: After the state update, the closest vertex, e, is expanded by moving towards e. Robot A next action selection is influenced by Robot B. It is not affected by Robot C since it is not spatially or temporally close. The tree shows nodes that represent occluded and open area as well as frontiers (e,g,l) to be expanded.

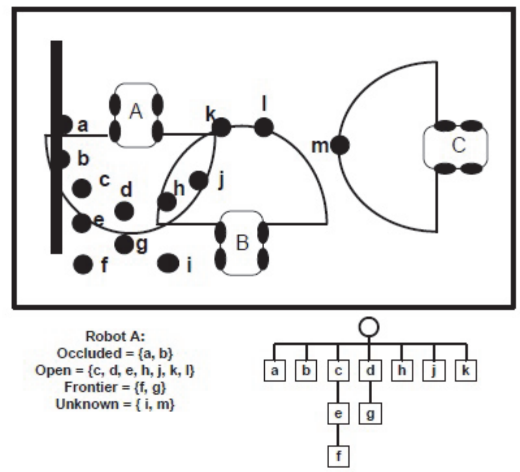

Figure 2: After Robot A visits vertex e, it is removed and a new adjacent vertex f is added to the exploration queue. The tree shows the next frontiers are now at vertex f or g.

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. Instead, in our approach robots are capable of covering an environment individually or cooperatively.

In similar work, Arkin et al. (Arkin and Diaz 2002) 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. The line-of-sight approaches work well when there is a requirement for robot cohesiveness, but in general will not be as efficient in a large environment when a small number of robots have to spread out more to cover the environment.

Meier et al. (Meier, Stachniss, and Burgard 2005) present a technique for assigning targets to robots and deciding what information to transmit when using communication with limited bandwidth. Each robot explores an unknown environment and creates a polygonal approximation of a map. To reduce message sizes, polygonal representations of the map are communicated. Using their approach, they were able to effectively coordinate a team of robots under bandwidth limitations. Nevertheless, communicating polygonal representations of a map can still result in an overhead of communication efforts.

Background

The greedy algorithm, frontier-based exploration (Yamauchi 1997), 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 (Elfes 1989). Frontiers, or boundaries between open and unexplored area, are detected and visited to gain more knowledge about the environment. 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. As a greedy algorithm, a robot chooses the unvisited vertex, p_i, with the shortest path from its current location. Therefore, information that is shared between robots that are close in space (spatial) and time (temporal) is more useful in determining subsequent actions. We demonstrate how spatial and temporal locality can be leveraged in observation-based cooperation.

Temporal and Spatial Locality

Vertices that are frontiers and adjacent to vertices within the robot's sensor footprint will be referred to as α. These vertices are added to the unvisited vertices list during the current time step. For example, in Figure 1, vertices e, g, and l are α nodes because they are frontiers and adjacent to the nodes (a, b, c, d, h, j, and k) that are in the robot's sensor footprint.

To affect action selection via intra-team communication, a message must indicate the current goal vertex has been visited. Robot, $rbt_i, i \in I$, determines its closest unvisited vertex is p_i. If a robot receives a message denoting p_i is open or occluded then it would remove it from the unvisited vertex list. This message is only generated when that vertex falls within the sensor range of another robot. If p_i is an α vertex then removing p_i from the unvisited list of $robot_i$ results in a new goal set to the next closed unvisited vertex. For example, in Figure 1, communications from Robot B to Robot A results in Robot A removing vertex l from its frontiers to an open vertex (Figure 2). Subsequent action selection is affected when Robot A chooses the next closest vertex as its goal.

Since robots select the closest vertices in which are normally adjacent vertices, the next action selections are affected by information received from robots that are close. If they are not close, the selection of α vertices is not affected. For example, communication about vertex m from Robot C to Robot A would not affect subsequent action selection since they are not close (Figures 1 and 2).

Information that is close in both time and space is useful in immediate action selection. In the case that digital messaging in not available, one approach is to use observation to infer state. Observation is likely to occur when robots are close when information is more useful.

Observation-based Cooperation

With observation-based cooperation, robots do not depend on digital messages. Instead, robots update their state with predicted actions of other robots. In observation-based cooperation, we consider spatial and temporal locality of nodes in determining which events will affect action selection. The most important change is the replacement of the message functions with observation-based functions. A brief explanation of this change follows.

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 1, lines 4 - 22).

Implicit Communication (Observation)

The OBS-REC function maps observations to state updates by prohibiting points covered by the observed robot from becoming frontiers (lines 23 - 32). In addition, if the observed is already in the $FRONTIER$ 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 (Skiena 2008). This indicates that the area is assumed to be covered by another robot and needs to be backtracked to later. Observations affect frontiers but do not affect occupancy maps.

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 33 - 40). Previous work suggests that observation errors affect coverage completion. As a robot observes another robot it makes assumptions that are at times inaccurate resulting in a degradation in performance at the end of a coverage task (Wellman et al. 2010).Therefore, first the robot attempts to choose the closest 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 illuminating the possibility that it was not searched by another robot (Skiena 2008). As

Algorithm 1: Observation-based Cooperation

```
1  Robot Network Robot team with range-limited
   sensing, r, in unknown bounded environment, p ∈ ℝ^2 ;
2  Alphabet A: p x {Open, Occluded} ∪ {null} ;
3  Processor State W^[i] = (occ-map, f-map), where
   occ-map: p x {OPEN, OCCLUDED, UNKNOWN},
   f-map: p x {FRONTIER, VISITED, PRUNED,
   UNKNOWN}, initially unknown for all p
4  function stf(W^[i], 𝔸_env);
5     foreach o ∈ 𝔸_env do
6        if occ-map(o) is UNKNOWN then
7           if f-map(o) is FRONTIER then
8              f-map(o) is VISITED
9           end
10          if perimeter(o) and dist(X^[i], o) < r then
11             occ-map(o) = OCCLUDED;
12          else
13             occ-map(o) = OPEN;
14             foreach a ∈ adjacent(o) do
15                if f-map(a) is UNKNOWN and
                  occ-map(a) is UNKNOWN then
16                   f-map(a) = FRONTIER
17                end
18             end
19          end
20       end
21    end
22 return (W^[i]);
23 function obs-rec( W^[i], 𝔸_rbt);
24    foreach obs ∈ 𝔸_rbt do
25       {obs_region} = proximity(obs);
26       foreach o_r ∈ {obs_region} do
27          if f-map(o_r) is FRONTIER then
28             f-map(o_r) = PRUNED
29          end
30       end
31    end
32 return W^[i];
33 function ctl(X^[i],W^[i], 𝔸_env);
34    pGoal={null}; pLength=∞;
35    foreach pt ∈ {f-map(p)==FRONTIER} do
36       if pLength < pathLength(X^[i],pt) then
37          pLength=pathLength(X^[i],pt);
38          pGoal=pt
39       end
40    end
41    if pGoal==null then
42       foreach pt ∈ {f-map(p)==PRUNED} do
43          if pLength < pathLength(X^[i],pt) then
44             pLength=pathLength(X^[i],pt);
45             pGoal=pt
46          end
47       end
48    end
49 return followPath(X^[i],p_goal,𝔸_env);
```

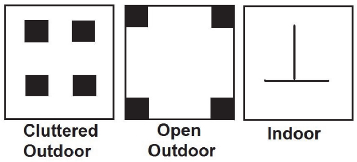

Figure 3: Simulation environments 1 and 2 were 16x16 meters with four obstacles. Simulation environment 3 was 6x6 meters with obstacles representing two rooms and a hallway.

a result, robots return back to the frontiers they marked off as being covered by other robots (lines 41 - 49).

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.

Simulations and Physical Experiments

We compare performances between NO COMM, DIRECT COMM, OB-BASED COOPERATION, and OB-BASED COOPERATION W/ BACKTRACKING in simulation and physical experiments. The performance was measured for each approach by examining two metrics: coverage or the area visible by the robots' distance sensor and the amount of time to cover an area. In DIRECT COMM, robots transmit open cell information once.

Simulation Setup and Results

Simulations were conducted in the 3-D physics-based simulator, Webots (Michel 2004). In Webots, a wheel encoder noise of based on a Gaussian distribution was added to resemble error in the real world. Global positioning sensor (GPS) was used for localization as well as a laser range finder. 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. Twenty trials for each approach were conducted with a three-robot team in both large and small environments for comparison (Figure 3). The two large environments include cluttered and open areas that represent large outdoor areas. The small indoor environment is used for comparison to the real world environment.

Results are summarized in Table I and graphed in Figures 4, 5, and 6. Results suggest that OB-BASED COOPERATION W/ BACKTRACKING performed the best in the cluttered outdoor and indoor environments. However, it performed comparable to DIRECT COMM in the open outdoor environment. Note that OB-BASED COOPERATION performance degraded and was worse than all approaches towards the end of execution but OB-BASED COOPERATION W/ BACKTRACKING improved.

Physical Experiments 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. The 802.11b wireless card on each machine allowed for wireless communications. The

	Approach	50% Cover Time (m)	σ	90% Cover Time (m)	σ
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
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
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

Table 1: Averages for 50% and 90% coverage for each approach in all environments in simulation.

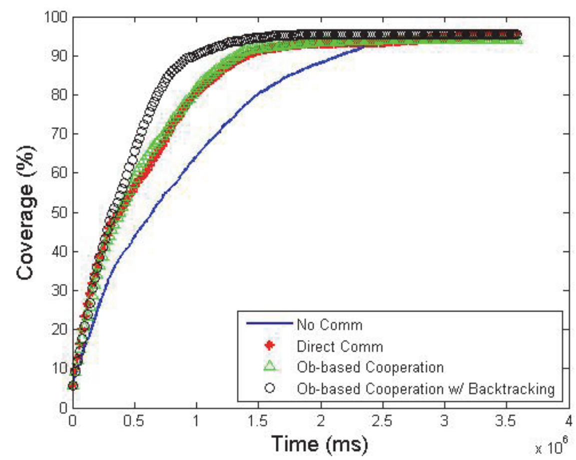

Figure 4: Coverage over time for all approaches in the simulated cluttered outdoor environment.

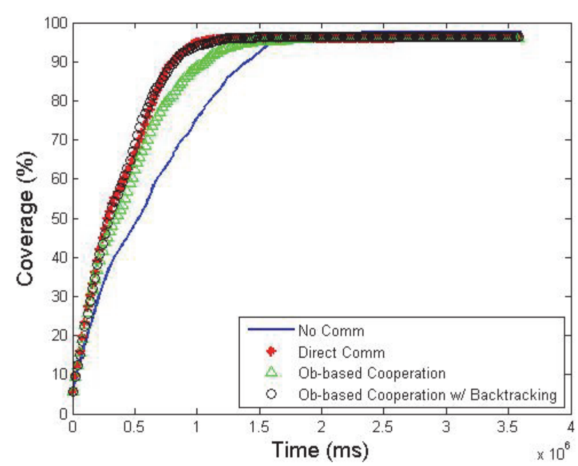

Figure 5: Coverage over time for all approaches in the simulated open outdoor environment.

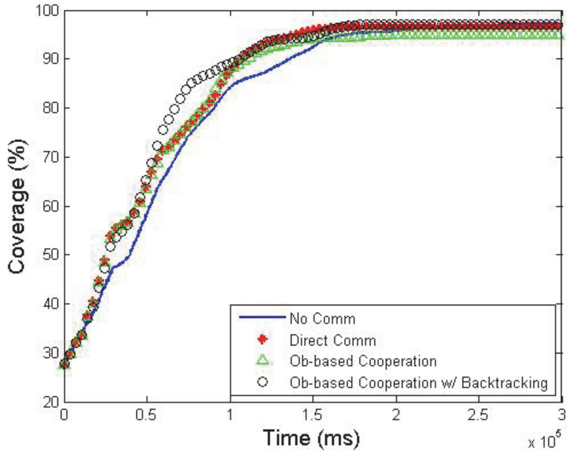

Figure 6: Coverage over time for all approaches in the simulated indoor environment.

real robots were augmented with a Hokuyo URG laser range finder.

The environments used in the experiments included a 6x6 open area and an area similar to the indoor environment in simulations (Figure 3). Five trials with a three-robot team were executed. The environment was augmented with the Hagisonic Stargazer localization system to gather ground truth positioning.

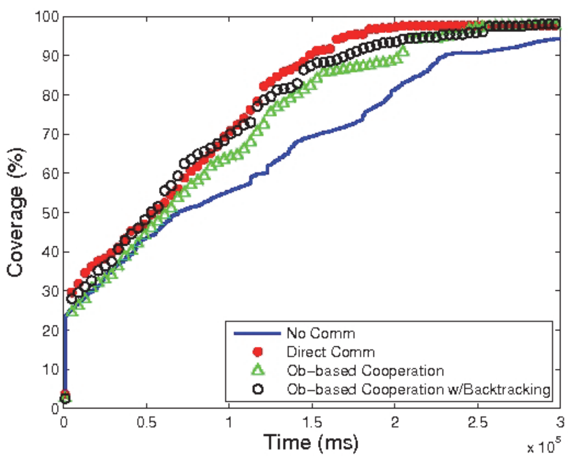

Figure 7: Coverage over time for all approaches in the cluttered real world environment.

Results are illustrated in Figure 7, Figure 8, and Table II. In the cluttered environment, DIRECT COMM performed the best overall by covering quicker. However, OB-BASED COOPERATION W/ BACKTRACKING was comparable and performed better at times (Figures 7 and 8). In addition, OB-BASED COOPERATION W/ BACKTRACKING covered quicker than OB-BASED COOPERATION.

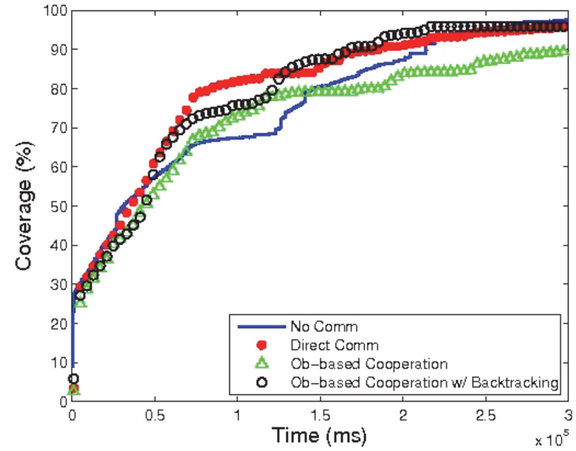

Figure 8: Coverage over time for all approaches in the open real world environment.

	Approach	50% Cover Time (m)	σ	90% Cover Time (m)	σ
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
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

Table 2: Averages for 50% and 90% coverage for each approach for the real environments.

In the open environment, OB-BASED COOPERATION W/ BACKTRACKING performed the best covering the most towards the end. Similar to the simulations results, OB-BASED COOPERATION performance degraded below DIRECT COMM and OB-BASED COOPERATION W/ BACKTRACKING towards the end.

Discussion

In simulations and physical experiments, results suggest that using observation to infer state coordinates robots comparably to message-based communications. In previous work (Wellman et al. 2010), with OB-BASED COOPERATION, robots make assumptions that areas are being covered by other robots. As a result, the average of coverage degrades towards the end because some of those assumptions are inaccurate. However, with OB-BASED COOPERATION W/ BACKTRACKING, after a robot visits all the frontiers on its list, it continues by backtracking to frontiers that were pruned as being covered by other robots. For example, in Figures 4, 5 and 6, the simulated robots using OB-BASED COOPERATION W/ BACKTRACKING improved at the end of coverage due to backtracking. Backtracking results in a more complete coverage by providing some redundancy toward the end of the search.

Although the cluttered indoor environment was configured similarly in simulations and physical experiments, the results did not track the same (Figures 6 and 7). In addition, in the real open environment, OB-BASED COOPERATION W/ BACKTRACKING performed better towards the end in the open environment but not with the cluttered environment (Figures 7 and 8). Previous research shows that when using multiple robots, results can differ significantly between simulations and physical experiments. Environmental configuration and communication paradigms effects on robot interference are more prominent in real robot experiments (Dawson, Wellman, and Anderson 2010). In the real environments, interference slowed the progress of robot teams using DIRECT COMM. With OB-BASED COOPERATION W/ BACKTRACKING, robots interfered less because they repelled one another.

OB-BASED COOPERATION W/ BACKTRACKING shows a three slope search by repelling, pruning, and then backtracking. Robots repel each other allowing for the fastest initial coverage. Then as they prune their search trees they search in different areas allowing for comparable search coverage. At the end of coverage, backtracking has slower coverage but robots continue to make progress.

Overall, DIRECT COMM and OB-BASED COOPERATION W/ BACKTRACKING had comparable results. Both approaches have elements that affect team performance. DIRECT COMM is subject to communication network limitations and OB-BASED COOPERATION W/ BACKTRACKING is sensitive to observation errors. However, results suggest that OB-BASED COOPERATION W/ BACKTRACKING approach can be a good alternative to DIRECT COMM.

Future Work and Conclusion

An approach that uses observation to infer state to coordinate robots is presented. Observation is used to direct immediate action selection through state updates. We show 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. Future work includes performing exploration just-in-time message exchange through rendezvous.

Acknowledgements

The authors gratefully acknowledge the support of the following NSF grants: IIS-0846976 and CCF-0829827.

References

Akella, A.; Judd, G.; Seshan, S.; and Steenkiste, P. 2007. Self-management in chaotic wireless deployments. *Wireless Networks* 13(6):737755.

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

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

Elfes, A. 1989. Using occupancy grids for mobile robot perception and navigation. *Computer* 22(6):46–57.

Ferranti, E.; Trigoni, N.; and Levene, M. 2009. Rapid exploration of unknown areas through dynamic deployment of mobile and stationary sensor nodes. *Autonomous Agents and Multi-Agent Systems* 19(2):210–243.

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

Koenig, S.; Szymanski, B.; and Liu, Y. 2001. Efficient and inefficient ant coverage methods. *Annals of Mathematics and Artificial Intelligence* 31(1):41–76.

Meier, D.; Stachniss, C.; and Burgard, W. 2005. Coordinating multiple robots during exploration under communication with limited bandwidth. In *ECMR*, 26–31.

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

Roy, N., and Dudek, G. 2001. Collaborative robot exploration and rendezvous: Algorithms, performance bounds and observations. *Autonomous Robots* 11(2):117–136.

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

Skiena, S. S. 2008. *The Algorithm Design Manual*. Springer.

Wellman, B. L.; Dawson, S.; Veluchamy, A.; and Anderson, M. 2010. Observation-based cooperation in mobile sensor networks: A bio-inspired approach for fault tolerant coverage. In *The 35th IEEE Conference on Local Computer Networks (LCN 2010)*.

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