Dispersion and exploration algorithms for robots in unknown environments

Steven Damer[a], Luke Ludwig[a], Monica Anderson LaPoint[a], Maria Gini[a], Nikolaos Papanikolopoulos[a], and John Budenske[b]

[a] Dept of Computer Science and Engineering, University of Minnesota, 200 Union St SE, Room 4-192, Minneapolis, MN 55455, USA

[b]Architecture Technology Corporation, P.O. Box 24344, Minneapolis, MN 55424, USA

ABSTRACT

We study the problem of dispersing a group of small robots in an unknown environment. The objective is to cover the environment as much as possible while staying within communications range. We assume there is no central control, the environment is unknown and with complex obstacles, the robots operate without any central control, and have only limited communications with other robots and limited sensing capabilities. We present algorithms and validate them experimentally in the Player/Stage simulation environment.

Keywords: robot behaviors, exploration, dispersion, self-deployment, swarm approaches.

1. INTRODUCTION

Behavior-based architectures are commonly used to control individual robots and teams of robots. They offer the advantage of being flexible, robust, and reactive. However, the flexibility often comes at the cost of difficulty in reusing existing behaviors for new application domains, and in unanticipated interferences among behaviors.

This paper addresses these problems by focusing on methods for developing reusable behaviors to control teams of robots. We focus on the problem of dispersing a group of robots in an unknown environment so as to cover the environment as much as possible while staying within communications range. We assume there is no central control, the environment is unknown but with complex connectivity and populated with obstacles, the robots operate independently, with limited communications with the other team members, and have limited sensing capabilities. We present algorithms and validate them experimentally using a Player/Stage simulation environment.

The primary motivation for this work comes from the need to develop robust methods that are applicable to very small robots that operate in unknown complex human-made environments. One of the major challenges that needs to be addressed when using very small robots is their extremely limited ability to estimate their own location. This is further complicated by the robots limitations in communications range, computing power, and suite of sensors.

The approach we propose uses basic behaviors to control the motions of each robot so that the robots will disperse in the environment without the need for any centralized control. We then present two different distributed algorithms that build a network of robots connected by communication links while they disperse in the environment. The behaviors and algorithms are validated experimentally in simulation.

Further author information: (Send correspondence to Maria Gini.)
Steven Damer: E-mail: damer@cs.umn.edu
Luke Ludwig: E-mail: ludwig@cs.umn.edu
Monica Anderson LaPoint: E-mail: mlapoint@cs.umn.edu
Maria Gini.: E-mail: gini@cs.umn.edu
Nikolaos Papanikolopoulos.: E-mail: npapas@cs.umn.edu
John Budenske: E-mail: jbudenske@atcorp.com

2. MOTIVATIONS

Mobile robots capabilities are improving dramatically while their costs are decreasing. Robot teams hold great promise in providing help for several difficult tasks associated with emergency response. For example, teams of robots can perform urban surveillance after a hurricane, execute effectively remote operations in the case of hazardous spills, participate in decontamination and decommissioning efforts immediately after a nuclear disaster, execute search and rescue operations, and find survivors in collapsed structures.

The primary motivation for this work comes from the Scout project.[1] The scouts are small, two wheeled robots with limited processing capability, that can deployed either by being hauled or launched into the environment by a larger robot. Because of their small size they can get into tight areas and be used in large numbers. Scouts are being considered for search and rescue applications, locating the source of a biological or chemical release, decontamination and decommissioning efforts, and monitoring highly sensitive areas. Scouts are often teleoperated, but their ability to do autonomous operations has been demonstrated in a task of hiding and watching for motion.[1]

Figure 1. The Scout robot.

One of the major issues when using very small robots is their extremely limited ability to estimate their own location and their limited communications range. In addition, their computing power and suite of sensors are limited.

Because of these limitations, we believe that the robots should be controlled not as individuals but as a swarm.[2] In a swarm, each robot uses simple local rules to decide its own actions, without needing any command from a central controller or from any other robot. This makes the overall system highly scalable and robust to failures. On the other side, since robots in a swarm interact with each other in ways often unexpected, unexpected behaviors tend to emerge, which makes it hard to provide guarantees of performance.

3. RELATED WORK

Coverage of terrain during motion is important in many application domains, such as floor cleaning, lawn mowing, demining, harvesting, etc. In such applications usually one needs to cover the terrain only once. Wagner et al.[3] formalize the terrain covering problem and propose two algorithms, one called mark and cover, the second called probabilistic coverage, both for single and multiple robots. They show how several cooperating robots can obtain faster coverage. Algorithms inspired by insect behaviors, such as ants, are becoming popular both for terrain coverage,[4] where robots leave trails and cover the terrain repeatedly, An algorithm for complete coverage of free space with a team of robots is presented in,[5] where it is assumed that robots have odometry to track their own position as they construct a cell decomposition of the free space.

Exploration by multiple robots has been studied extensively. For instance, in[6] an algorithm is presented to build a global map, select goal locations for exploration, and use explicit communication to prevent multiple robots to go after the same goal. We are interested in a combination of exploration and deployment, where robots have to explore but also to maintain coverage of the areas explored and maintain communications with other robots. The applications we are considering are urban search and rescue, where it is important to form a network to allow for transmission of information and monitoring.

Various algorithms have been proposed for dispersion to ensure maximum coverage. For instance, in[7] two dispersion methods are proposed. The methods use beacons on the robots and visual communication, which require direct line of sight, and a simple environment with only convex obstacles. A simple diffusion algorithm has recently been demonstrated[8] with small robots that use their buzzers and microphones to disperse.

Howard et al.[9] address the problem of incremental deployment, where robots are deployed one-at-a-time into an unknown environment, and each robot uses information gathered by previously deployed robots to determine its deployment location. They assume every robot is equipped with an ideal localization sensor. We do not assume the robots know where they are, since for small robots localization is very hard. New technology, such as the Cricket,[10] could provide position information, but it requires placing beacons in the environment, not having walls between the beacons, and has limits on the accuracy of the position information. These requirements rule out using the Cricket in the applications we are envisaging.

The dispersion method presented in[11] is based on artificial potential fields, and assumes that each node is equipped with a sensor that allows it to determine the range and bearing of its neighbors. Obstacles, such as walls, are not modeled. In our work, we disperse the robots in an environment which includes walls and obstacles. We require that each robot remains within communications range of at least one other robot. We are interested in working in very large environments where requiring a higher degree of connectivity might require to use too many robots. We will explore the effect of increasing the degree of connectivity in the future.

Spears et. al[12] have presented a framework for swarms moving in formations based on natural physics laws that can be used to self-organize a large swarm of robots into regular formations. They use the concept of repulsive forces as,[9] and in particular focus on creating specific formations such as hexagonal lattices. This work assumes that each robot is capable of determining the distance away and bearing of its local neighbors, which again requires a 360^o laser range finder or instrument of similar capability.

Hsiang et al.[13] propose methods for dispersing robots from fixed locations to cover the entire environment. They assume a continuous stream of robots would be entering the environment through predetermined locations. The goal of the robots is to position themselves such that the entire accessible area is covered. Through the use of deterministic robot motions and infinite supply of robots the information available is sufficient to guarantee that the robot will make the correct choice for its motion. We do not assume that there are enough robots to cover the entire space and to guarantee that every robot can remain within sensor range of the other robots. We require only that the robots stay within communications range, which is larger that sensor range and can go through walls.

4. PROPOSED APPROACH

We assume the robots only have local knowledge, i.e. they are not under global control (no central source knows the state of all of the robots), and do not have any knowledge of the environment other than what they can detect with their sensors. This assumption is useful for redundancy purposes since central control introduces a single point of failure.

In addition, we prefer that the robots do not need to know how many other robots are operating in the same environment, where those robots are located, and where those robots have been. This serves two purposes. First, it allows for some flexibility since more robots could be introduced after the initial deployment and failures are not catastrophic. Second, it does not put a heavy burden on sensors to either recognize other robots or calculate odometry accurately.

Our results have been obtained in simulation, using Player/Stage.[14] For simplicity, our simulated robots are pioneers, each equipped with a sonar array that provides distances to obstacles and some immediate environmental information.

Simulation allowed us to run a large number of tests with different behaviors before doing any testing on the actual robots. We used multiple simulation experiments to validate our initial hypotheses regarding the effectiveness of various types of behaviors to disperse, cover, and maintain communications in an unknown environment.

A new Stage device was created to simulate the communications using an onboard Bluetooth device. The discovery process, simulated using the WiFiInterface, gives each robot the names of other robots within approximately 8 meters. Degradation experienced due to intervening walls was implemented using a discounting scheme that deducted for any walls between robots.

We developed a number of basic behaviors and measured how well they succeed in dispersing the robots. In particular, we measured the total area coverage achieved by the robots in an assigned amount of time and the instantaneous coverage.

The basic behaviors are designed to keep the robots in motion at all times, so that they spread and explore the environment. Since we are interested in combining exploration with surveillance of the areas explored, we developed two additional algorithms, named CLIQUE-INTENSITY and BACKBONE DISPERSION, to combine movements to spread the robots with methods for keeping the robots in place once they have dispersed enough. The algorithms effectively deploy a sensor network. We describe them later in Section 6

5. BASIC BEHAVIORS

We have found three basic behaviors that are the most effective for robot dispersion. The first behavior is a RANDOM WALK behavior. Given no obstacles, the robot moves on a slightly curved path by turning a random amount between 10° and $-10°$ per time step. When the robot detects an obstacle, it stops, turns by a larger random amount (in the range 120° to 240°), and then resumes moving while turning a small random amount.

The second behavior is a FIND OPENINGS behavior that uses sensors to locate openings such as doorways or halls. Although, many sensors that measure range can be used, in the simulation, we used a sonar array that is similar to the sonar array found on the pioneer. The ranges in front of the robot are compared and the robot turns in the direction of the two adjacent ranges with the highest change in value.

The third behavior is a comparison behavior, AVOID ROBOTS. It simulates a robot that is able to recognize another robot that is nearby and move away from it. The simulation is implemented by putting a target on each robot that any robot can recognize. Physically, this could be implemented with varying level of effectiveness with vision or infrared sensors. Range sensors cannot be used since robots cannot distinguish between another robot and an obstacle.

More details on those behaviors and experimental results are reported in.[15]

In addition, we developed two behaviors that react to a robot moving outside the communications range of all other robots. The RANDOM RANGE control algorithm moves the robot randomly until communications is reestablished. The BACKTRACK RANGE control algorithm attempts to move the robot back to previous positions. The rationale is that at a previous time step, the robot was in range so moving back to that position should reestablish communications, assuming the other robots did not move too far.

Number of Robots	RANDOM RANGE CONTROL	BACKTRACK RANGE
10	101.04	126.05
20	123.31	138.69
30	48.20	115.33
40	158.33	143.70
50	126.05	159.13

Table 1. Mean time to reestablish communications by number of robots and by range recovery behavior.

The RANDOM RANGE CONTROL is no more effective than the BACKTRACK RANGE behavior (see Table 1). One reason may be that BACKTRACK RANGE had to rely solely on odometry to move to previous positions.

Behavior	Cumulative		Instantaneous	
	10 Robots	50 Robots	10 Robots	50 Robots
RANDOM WALK[1]	41.37%	90.28%	16.91%	46.82%
RANDOM WALK[2]	61.19%	78.52%	18.87%	41.98%
AVOID ROBOTS[1]	71.13%	91.57%	18.9%	45.97%
AVOID ROBOTS[2]	60.91%	85.18%	19.46%	50.28%
FIND OPENINGS[1]	94.32%	98.74%	20.29%	50.33%
FIND OPENINGS[2]	92.96%	98.08%	16.93%	50.51%

Table 2. Percentage of area covered in the home environment from Player/Stage after a running time of 30 minutes. Cumulative indicates the total area covered in the entire experiment, Instantaneous is the maximum instant coverage. [1] employs the RANDOM RANGE control. [2] employs BACKTRACK RANGE control.

Since odometry is not reliable and moving to previous positions may be hampered by obstacles and errors, the robots may not have managed to move very far from the position where communications was lost.

The effectiveness of the dispersion and exploration behaviors varied with the deployment environment. AVOID ROBOTS and FIND OPENINGS were both very successful in the Player/Stage house environment (see Table 2) as measured by total coverage. FIND OPENINGS needed less time to explore the environment than the other behaviors. However FIND OPENINGS did not do as well in other larger environments. FIND OPENINGS had a tendency to move the robot into the edge of doorways which caused a stall. RANDOM WALK tended to have more variable results.

We measured dispersion success by the cumulative area covered by the robots during the entire experiment, and by the total area covered by the robots at any instant in time. Results are given in Table 2. These numbers are naturally bounded by the number of robots and the environmental constraints (i. e. doors, walls, etc). The maximum instant coverage is essentially the same for all behaviors in the house environment.

Another finding in our experiments was the natural tradeoff between maintaining communications and exploring the area. The actions taken to reestablish communications slowed exploration progress, as shown by the plateaus in the coverage graphs at points were communications were lost by one or more robots.

6. DISPERSION ALGORITHMS

We have developed two distributed dispersion algorithms, the CLIQUE-INTENSITY algorithm and the BACKBONE DISPERSION algorithm. Both algorithms attempt to accomplish dispersion and to establish a sensor network in a distributed way. Both algorithms use the wireless signal intensity to aid a robots in dispersing. Typically it is assumed that the relative distance and bearing of neighboring robots is known.[7, 9, 12] Signal intensity gives only a rough approximation of distance and no bearing information, making the problem harder.

Theoretically, signal intensity varies according to the law of inverse signal propagation, which simply means the signal intensity is proportional to the inverse square of the distance it travels. In a practical setting, the environment plays a huge role since obstacles cause noise in this signal. However, it is unnecessary for the signal intensity to be very accurate in order to provide some indication to a robot for which way it should travel.

If it is known that one of the neighbors of a robot is stationary, then the robot could specifically reference the stationary robot's intensity and attempt to move in a direction of decreasing signal intensity until some threshold is reached. This is a key concept used in both our algorithms.

6.1. Clique-Intensity Algorithm

The CLIQUE INTENSITY algorithm is designed for a distributed homogenous swarm. The knowledge of each robot is a graph with robots as nodes and signal intensities between robots as weights. This graph is referred to as the connectivity graph. Robots share portions of their connectivity graphs with their neighbors.

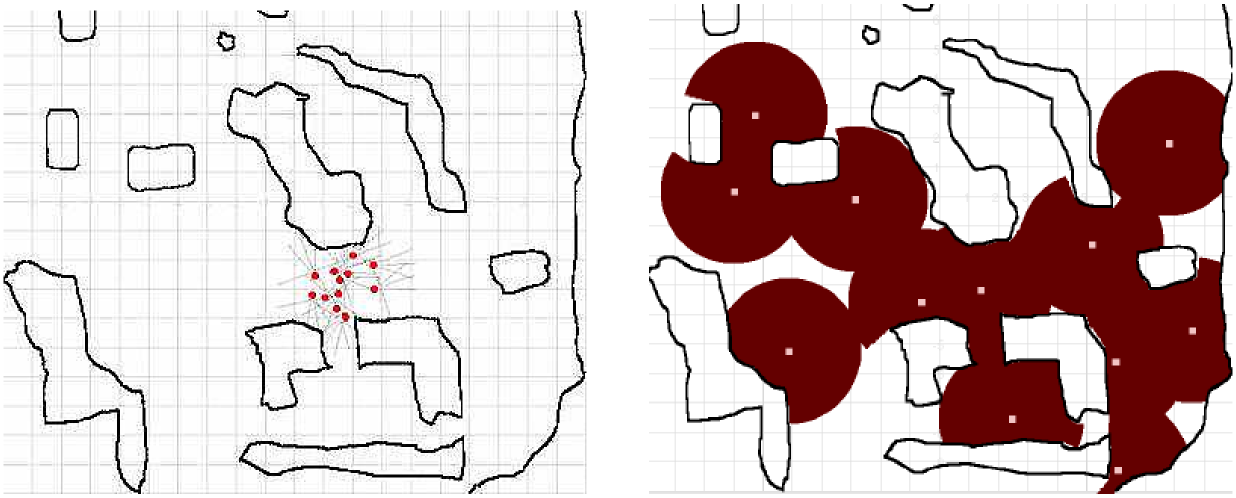
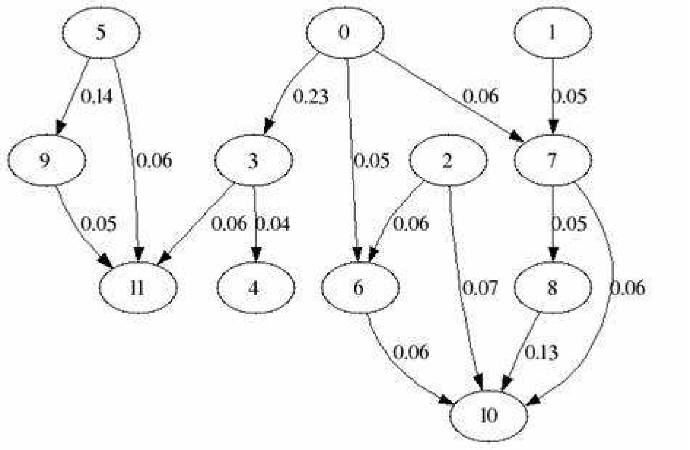


Figure 2. Experiment A. Left: Starting configuration of experiment with 12 robots in a cave-like setting. A total of 25 square meters is covered at the start. Lines emanating from the robots indicate the range of the proximity sensors. Right: The output of the post-processing area coverage step for one run. Each robot covers an area within a 2.5 meter radius. This run resulted in an area coverage of 141 square meters in 80 seconds.



Clique: Set([0, 3])
Clique: Set([0, 6])
Clique: Set([0, 7])
Clique: Set([1, 7])
Clique: Set([2, 10, 6])
Clique: Set([3, 4])
Clique: Set([11, 3])
Clique: Set([9, 11, 5])
Clique: Set([8, 10, 7])

Figure 3. An example connectivity graph where nodes are robots and weights are signal intensities. This graph reflects the state of the graph after it reached an equilibrium in which robots were no longer moving. This was generated during a simulation experiment with 12 robots by querying a robot's web server for its knowledge and passing the knowledge onto the Dot program. This should be interpreted as an undirected graph, so ignore the arrows.

A clique is a graph or subgraph in which every node is connected to every other node. A maximal clique is one that is not a subgraph of another clique. For each maximal clique in the connectivity graph a single robot is chosen to be the sentry for the clique, meaning it remains stationary. The other robots in the clique attempt to move away from the sentry. This is done by monitoring the change in the signal intensity over time. Each robot behaves in such a way that causes the entire swarm to disperse in an attempt to create cliques in the connectivity graph of size three or two. This is an attempt to triangulate the map which is known to be the most effective static configuration for the area coverage problem.[16]

The primary decision to make is whether or not a robot is a sentry, and if not then the robot must decide which neighbor will be its sentry. This decision is made individually by each robot examining its connectivity graph and following a set of rules. The rules are structured such that each robot will arrive at the same decision as to which are sentries and which are in motion.

We would like as few robots as necessary to be sentries, so if a robot is a sentry for one clique, then it is considered a sentry for all of its cliques. We want to encourage robots on the perimeter of the dispersion effort to continue dispersing. This is done by choosing the robot in a clique which is in the most cliques to be the sentry. This follows the idea that robots on the perimeter will be in fewer cliques. The tie-breaker for this situation is to simply choose the robot with the lowest id.

A simple experiment with 12 robots was performed on a cave-like map. The clique distance was set at 5, and it was assumed the robots covered an area within a radius of 2.5 meters. The robots disperse from the starting configuration shown in Figure 2 until the algorithm reaches an equilibrium in which the entire swarm is stationary. The initial area coverage is 25 square meters. This experiment was ran 10 times and resulted in a mean area coverage of 128 square meters with a standard deviation of 6.27 square meters. It took between 60 and 100 seconds for the algorithm to reach an equilibrium. Notice there are no disconnections in the area coverage. Figure 3 represents the connectivity graph corresponding to the final equilibrium.

More details on the algorithms and additional experimental results are reported in.[17]

6.2. Backbone Dispersion Algorithm

The Backbone Dispersion algorithm is our second algorithm to disperse robots while maintaining contact. It is fundamentally based upon the assumption that two moving robots will not simultaneously lose contact with a stationary robot. Since the speed and range of communication between robots much greater than the speeds attainable by the robots this is a reasonable assumption.

The Backbone Dispersion algorithm maintains a core group of stationary robots such that every robot is in direct contact with the backbone. Additional robots are added to the backbone when they are needed to maintain connectivity. Robots which are not in the backbone wander around freely. The only distinction between robots in the backbone and robots not in the backbone is in their behavior. Robots not in the backbone can join the backbone, but once in the backbone, a robot never leaves it. In addition to providing connectivity, the backbone can also be used as a source of primitive navigation for the wandering robots.

Robots which are assigned to the backbone follow simple rules. They remain in one place, and broadcast their status as members of the backbone and their distance in hops from other members of the backbone. The behavior of wandering robots is more complex. If a wandering robot is in direct contact with one robot in the backbone and at least one other robot it will wander freely. If it is in contact with exactly one member of the backbone and no other robots, it will stop moving and wait for another robot to approach. If it is not in contact with any member of the backbone, but it is in contact with at least one other robot, it will join the backbone, and send a message to one of its neighbors causing that robot to join the backbone. If it is in contact with no other robots, it will wander around until it regains contact with the backbone.

Robot is in direct contact with:	Action	Goal
Two or more robots, at least one in the backbone	Wander randomly, avoiding obstacles	Find uncovered areas
Exactly one robot which is in the backbone	Remain stationary until another robot wanders into range	Avoid losing contact with the backbone
No robots in the backbone and at least one not in the backbone	Join the backbone, and send a message to one neighbor telling it to join the backbone	Extend the backbone to an uncovered area
No robots at all	Wander randomly, avoiding obstacles	Regain contact with the backbone

Table 3. Summary of rules for non-backbone robots.

These rules allow a group of robots to disperse while maintaining connectivity. If two robots simultaneously wandered out of range of the backbone while still in range of each other, they could become unconnected from the backbone, but this is fairly unlikely. If it were a significant problem, it can be easily avoided by having robots stop and join the backbone just before they get out of range (presumably at a point where signal strength is degraded, but before the connection is lost). The dispersion is effective because for every pair of robots which join the backbone, one of the new nodes is completely unconnected to the existing nodes of the backbone.

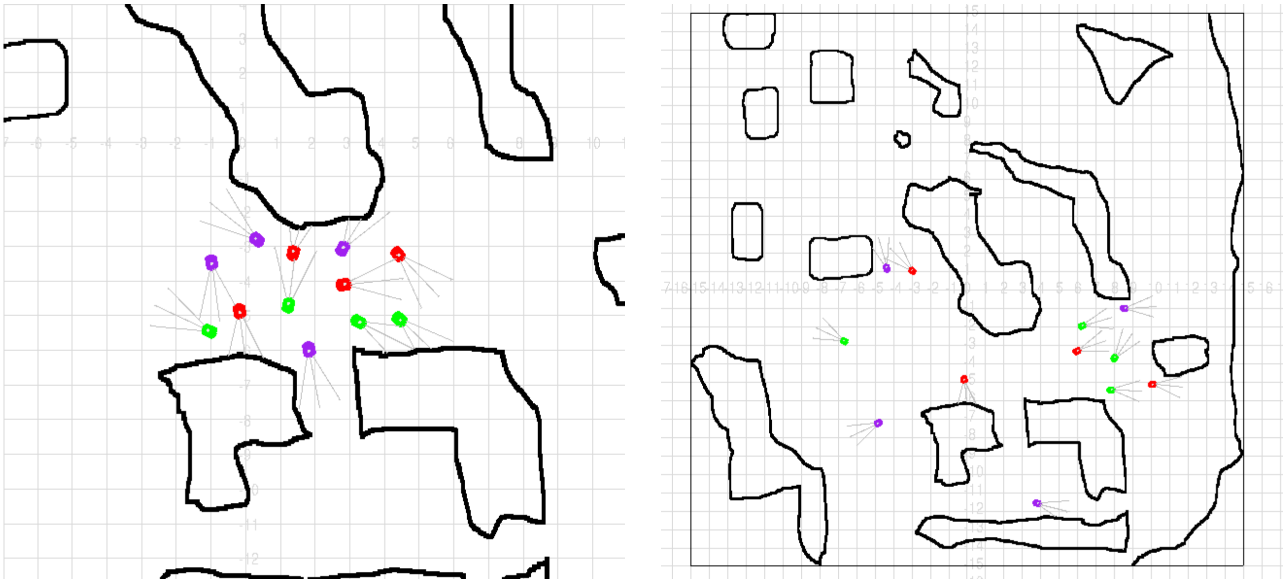


Figure 4. Initial position of robots (left) and part way through the dispersion (right).

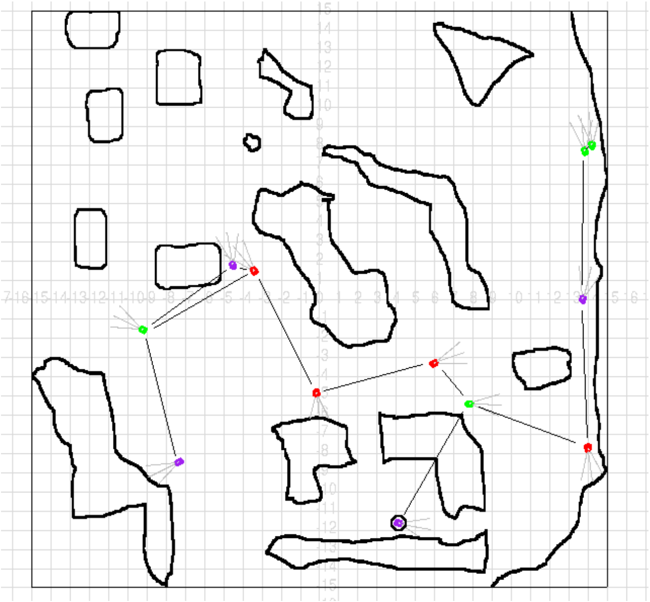


Figure 5. Final position of robots

Figures 4 and 5 show the progression of backbone based dispersion in three stages. Figure 4 shows the initial positions of the robots and the the robots part way through the dispersion. Figure 5 shows the final positions of the robots once they have all stopped moving. The lines in Figure 5 indicate which robots are directly connected. The circled robot is the only remaining robot which is not part of the backbone. It will remain stationary forever, because there is no other robot which could approach it.

The structure the backbone provides can be used to do more than simply maintain connectivity. Robots can track an estimated position by observing which backbone nodes they have a direct connection to. They can then use that information to achieve a simple level of navigation by reversing direction when they find themselves in contact with a node which is further away from their destination. This has the potential to be used to achieve more efficient dispersion by directing wandering robots towards areas of the backbone which have not been fully explored.

7. CONCLUSIONS AND FUTURE WORK

We have presented different behaviors that can used by a group of robots to disperse in an unknown environment while remaining within communications range. We make minimal assumptions on the capabilities of the robots. In particular, we do not assume the robots know their own position or know where the other robots are or have been. We have then presented two distributed algorithms that disperse the robots while maintaining connectivity, until all the robots reach a stationary position.

The results obtained from our experiments indicate that a swarm of small robots can be dispersed effectively through the use of wireless signal intensities, without knowing the relative locations of neighboring robots. Using small robots instead of larger ones means that many more robots can be deployed for the same cost.

ACKNOWLEDGMENTS

Partial funding for this work was provided by the National Science Foundation Industry/University Safety, Security, and Rescue Research Center, and by the Air Force under Contract No. FA8651-04-C-0191 to Architecture Technology Corporation.

REFERENCES

1. P. E. Rybski, S. A. Stoeter, M. Gini, D. F. Hougen, and N. Papanikolopoulos, "Performance of a distributed robotic system using shared communications channels," *IEEE Trans. on Robotics and Automation* **22**, pp. 713–727, Oct. 2002.
2. E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*, Oxford University Press, 1999.
3. I. A. Wagner, M. Lindenbaum, and A. M. Bruckstein, "MAC vs PC – determinism and randomness as complementary approaches to robotic exploration of continuous unknown domains," *Int'l Journal of Robotics Research* **19**(1), pp. 12–31, 2000.
4. S. Koenig, B. Szymanski, and Y. Liu, "Efficient and inefficient ant coverage methods," *Annals of Mathematics and Artificial Intelligence* **31**, pp. 41–76, 2001.
5. I. Rekleitis, V. Lee-Shue, A. P. New, and H. Choset, "Limited communication, multi-robot team based coverage," in *Proc. of the IEEE Int'l Conf. on Robotics and Automation*, Apr. 2004.
6. R. Simmons, D. Apfelbaum, W. Burgard, D. Fox, M. Moors, S. Thrun, and H. Younes, "Coordination for multi-robot exploration and mapping," in *Proc. Nat'l Conf. on Artificial Intelligence*, pp. 852–858, 2000.
7. M. A. Batalin and G. S. Sukhatme, "Spreading out: A local approach to multi-robot coverage," in *Proc. Int'l Symp. on Distributed Autonomous Robotic Systems"*, pp. 373–382, 2002.
8. S. Bergbreiter and K. Pister, "Cotsbots: An off-the-shelf platform for distributed robotics," in *Proc. IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*, 2003.
9. A. Howard, M. J. Mataric, and G. S. Sukhatme, "An incremental self-deployment algorithm for mobile sensor networks," *Autonomous Robots* **13**, pp. 113–126, Sept. 2002.
10. N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The cricket location-support system," in *Proc. 6th ACM MOBICOM*, Aug. 2000.
11. S. Poduri and G. S. Sukhatme, "Constrained coverage for mobile sensor networks," in *Proc. of the IEEE Int'l Conf. on Robotics and Automation*, pp. 165–172, 2004.

12. W. Spears, D. Spears, J. Hamann, and R. Heil, “Distributed, physics-based control of swarms of vehicles,” *Autonomous Robots* **17**(2–3), 2004.
13. T.-R. Hsiang, E. Arkin, M. A. Bender, S. Fekete, and J. Mitchell, “Algorithms for rapidly dispersing robot swarms in unknown environments,” in *Proc. 5th Workshop on Algorithmic Foundations of Robotics (WAFR)*, 2002.
14. B. P. Gerkey, R. T. Vaughan, K. Stöy, A. Howard, G. S. Sukhatme, and M. J. Matarić, “Most valuable player: A robot device server for distributed control,” in *Proc. IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*, pp. 1226–1231, Oct. 2001.
15. R. Morlok and M. Gini, “Dispersing robots in an unknown environment,” in *Proc. Int'l Symp. on Distributed Autonomous Robotic Systems”*, pp. 241–249, 2004.
16. B. Kadrovach and G. Lamont, “Design and analysis of swarm-based sensor systems,” in *Proc. of the Midwest Symposium on Circuits and Systems*, Aug. 2001.
17. L. Ludwig and M. Gini, “Robotic swarm dispersion using wireless intensity signals,” in *Proc. Int'l Symp. on Distributed Autonomous Robotic Systems”*, 2006.