Leveraging Area Bounds Information for Autonomous Decentralized Multi-Robot Exploration

Tsung-Ming Liu and Damian M. Lyons
Fordham University, New York NY 10105, USA
{tliu17,dlyons}@fordham.edu

Abstract. This paper proposes a simple and uniform, decentralized approach to the problem of dispersing a team of robots to explore an area quickly. The Decentralized Space-Based Potential Field (D-SBPF) algorithm is a potential field approach that leverages knowledge of the overall bounds of the area to be explored. It includes a monotonic coverage factor in the potential field to avoid minima, realistic sensor bounds, and a distributed map exchange protocol.

The D-SBPF approach yields a simple potential field control strategy for all robots but nonetheless has good dispersion and overlap performance in exploring areas with convex geometry while avoiding potential minima. Both simulation and robot experimental results are included as evidence, and performance, speedup and efficiency metrics for each are presented.

Keywords: Multi-robot exploration, potential field path-planning, autonomous systems.

1 Introduction

This paper addresses the problem of developing an effective control strategy when multiple robots are deployed to an unknown environment to explore and collect information about the environment. Practical examples of such missions include search & rescue, location of explosive devices, and internal reconnaissance of a structure. In a counter weapons of mass destruction (C-WMD) mission, the robot mission designer may know there is a bomb in the building. In this case, the critical point of the mission is to search as much of the area within the building as possible in the shortest time to locate the WMD so that it can be assessed. In a search and rescue scenario, where there may be injured victims in the building, the critical point of the mission is also to search, in this case to locate victims and assess their condition so as to quickly address their needs. Such missions are time-critical and detailed environment information is not needed, rather a fast, effective scan with the on-board sensors is much more crucial for the mission. It is reasonable to propose that deploying multiple robots will allow the building to be searched more quickly (Burgard, Moors, Dox, Simmons, & Thrun, 2000). If multiple robots are being used to increase exploration efficiency, **and all areas of the building are considered equally important**, then **prioritizing** dispersion and **reducing** overlap in scanning **are potential strategies to improve performance**. However, if members of the team disperse within the building, then it is likely that the communications to a centralized, remote server will be intermittent. In the case of a decentralized approach, it's likely that the team members will lose and regain contact with each other

from time to time as they explore. For simplicity and efficiency, it is preferred to have the robots depend and communicate with each other as little as possible, and for the control strategy of the robots to be as uniform as possible.

In prior work (Liu & Lyons, 2014), we proposed the following as reasonable and useful assumptions to make about the mission: (1) the robot mission designer knows something about the environment, but not everything. (E.g., the rough dimension of the building to be searched is known but not the indoor floor map details); and (2) some inter-robot communication is possible, but only a minimum amount of data transfer is preferred. That work did not address the decentralized communication between members of the robot team. In this paper, a decentralized system architecture and map update algorithm is proposed and evaluated using the simple and uniform potential field dispersion and navigation mechanism based on that in (Liu & Lyons, 2014).

In the next section relevant prior literature is reviewed. Section 3 presents the proposed decentralized map update and navigation control algorithm. It begins with a review of the potential field mechanism proposed and evaluated in (Liu & Lyons, 2014) as well as the modifications added for the decentralized implementation evaluated in this paper. Section 4 reports a series of evaluation results, where the algorithm is evaluated in simulation using a number of metrics and compared to existing work, as well as the results of implementation on a pair of Pioneer 3-AT robots is presented. Section 5 concludes by summarizing the results and discussing future challenges.

2 Prior Work

Burgard et al. (Burgard, Moors, Dox, Simmons, & Thrun, 2000) introduced a centralized approach to multirobot exploration that assigns target location goals to team members by locating a robot and frontier point (a point on the boundary of explored and unexplored space) that balances exploration utility and travel cost. Rogers et al. (Rogers, Nieto-Granda, & Christensen, 2014) develop a centralized scheme to coordinate frontier-based exploration by large robot teams and investigate which robot coordinate strategies are effective. Decentralized schemes have been introduced by Visser & DeHoog (Visser, de Hoog, Jimenez-Gonzalez, & Martinez de-Dios, 2013) and Chand & Carnegie (Chand & Carnegie, 2013) for heterogeneous teams by assigning different roles to team members.

Potential field methods have been leveraged for robot navigation (e.g., (Arkin R. C., 1998)(Khatib, 1986)(Lyons D. , 1986)) because they offer intuitive and efficient implementations. While primarily used to describe single robot motion (e.g., Arkin (Arkin R. C., 1998)), the potential field approach was also used to specify manipulator configurations (e.g., robot manipulators, Khatib(Khatib, 1986); dexterous hands, Lyons(Lyons D. , 1986)). Multirobot exploration offers challenges beyond single robot exploration(Burgard, Moors, Dox, Simmons, & Thrun, 2000). More recently potential field approaches have been extended to handle multiple robot exploration (e.g., Arkin & Diaz (Arkin & Diaz, 2002), Lau (Lau, 2003)), formation (e.g., Schneider (Schneider, 2003)) and map improvement (e.g., Julia et al.(Julia M. , Gil, L., & Reinoso, 2008)). Our motivation in pursuing a potential field approach is its potential for an intuitive and efficient solution.

Our focus in this paper is addressing the multirobot exploration problem by leveraging a distributed potential-field method. Similar approaches include that of Lau (Lau, 2003), Julia et al. (Julia M. , Gil, L., & Reinoso, 2008), Renzaglia (Renzaglia &

Martinelli, 2010), Baxter (Baxter, Burke, Garibaldi, & Norman, 2007), Cepeda (Cepeda, Chaimowicz, Soto, Gordillo, Alanise-Reyes, & Carrillo-Arce, 2012) and Popa (Popa, Helm, Stephanou, & Sanderson, 2004). Baxter (Baxter, Burke, Garibaldi, & Norman, 2007) presents a potential field approach in which the field is shared among robots. Cepeda (Cepeda, Chaimowicz, Soto, Gordillo, Alanise-Reyes, & Carrillo-Arce, 2012) proposed a behavior based approach for multi-robot exploration, but such an approach has difficulty in robot exploration status synchronization. Popa (Popa, Helm, Stephanou, & Sanderson, 2004) uses potential fields for dispersion of sensor networks, but doesn't discuss exploration or path-planning. Batalin (Batalin & Sukhatme, 2003) presents an approach with good dispersion results but is not as strong on path-planning and searching. Min (Min & Papanikolopoulos, 2011), Mi (Mi, Yang, & Liu, 2011) and Schwager et al. (Schwager, Rus, & Slotine, 2011) all provided novel topological approaches for dispersion and coverage of multi-robot exploration, and Jenson (Jensen & Gini, 2013) proposed using wireless signals for the same purpose; however the detailed path-finding strategy is not part of the solutions

One of the key issues that any potential field method must address is how local minima in the field are handled. These are undesirable locations at which the field sums to zero, resulting in robots being stuck or stalled, delaying or halting exploration. Arkin (Arkin R. C., 1998) uses random noise, to eject robots from minima, and spin fields, to move robots along the surface of obstacles, to address minima issues. Julia et al. (Julia M. , Gil, L. & Reinoso, 2008) and Renzaglia (Renzaglia & Martinelli, 2010) use the potential field method for decentralized robot control, and a frontier-based approach for breaking out of potential field minima. Julia et al. (Julia M. , Gil, L. & Reinoso, 2008) represent the space to be explored using an occupancy grid map enhanced so that each cell also represents the degree of exploration of that area. Each cell generates an attractive force on each robot inversely proportional to the amount of exploration of the cell. Furthermore, landmark locations for landmarks not yet precisely enough known generate attractive force. Whenever a local minimum is detected, the robot plans a path to the nearest frontier cell. Renzaglia (Renzaglia & Martinelli, 2010) has two classes of robot: one (follower) which is potential-field driven, and one (leader) which is always planning the path to the frontier. The follower robots are influenced by the explored area and the position of the other robots, whereas the leader robots are uninfluenced by the other robots (and hence local minima). In both schemes, communication and calculation of information shared between the robots, as well as the transition between roles of leader and follower (Renzaglia) or detection of minima (Julia et al), become crucial and risk failure if any link in this cooperation fails.

The approach proposed in this paper is similar to that of Julia et al. in maintaining an extended occupancy grid with cumulative sensor coverage information per cell, and to Renzaglia's potential field for dispersing follower robots. However, no frontier is maintained, and only one class of robot exists; a substantial simplification. Potential minima will be handled by adding a monotonic coverage factor to the potential field equations of Renzaglia, and by the noise and vortex methodology of Arkin. Furthermore, the exchange of information between robots will be explicitly formalized here in terms of transfer of extended occupancy grids.

In (Ozisik & Yavuz, 2008) Ozisik et al. described an occupancy-grid based SLAM method, however it was not optimized for multiple robot efficiency. In (Birk & Carpin, 2006) Birk shows how to merge occupancy grid maps from multiple robots, and as does Herath (Herath, Kodagoda, & Dissanayake, 2008). However they focus on fea-

ture-recognition and merging data from multiple robots. Lyons et al (Lyons, Shethra, & Liu, 2013) discuss grid-based methods to combine maps so as to avoid transitory readings from other robots. In this paper, the occupancy grid contains information about obstacles sensed and information about exploration coverage. However, no SLAM module will be included; this simplification is made to focus on the coverage, rather than the mapping, issues of the problem.

3 Decentralized Navigation

The proposed approach is presented in two components. The first is the decentralized navigation algorithm – which is based on our prior work (Liu & Lyons, 2014). The second component is the decentralized communication and map update.

3.1 The Space-Based Potential Field Approach

The potential field approach (Arkin R. C., 1998) has been used previously for directing robot motion towards a goal while avoiding obstacles. A key aspect of the approach, as introduced in our prior work (Liu & Lyons, 2014), is that the unexplored area needs to act as an attractive goal for robots. The unexplored area is not modelled as a frontier as in (Renzaglia & Martinelli, 2010)(Julia M. , Gil, L. & Reinoso, 2008) but rather the unexplored area *itself* exerts an attractive force on all robots. Because we assume we know the overall bounds of the area, we know the maximum area to be explored and can calculate the attractive field. All robots are drawn to the unexplored areas of the map.

The proposed approach represents space by a map divided into multiple grids/cells, where each cell has a *potential level* representing the level of exploration or scanning by each robot’s sensors. Initially the potential level for each cell is zero. The effect of a zero potential is to generate an attractive force on all robots. As robot sensors cover an area of the building once or more, the potential level for the grid cells corresponding to that area will increase. This results in the area becoming repulsive, rather than attractive, to all robots, and the more times the area has been covered, the more repulsive it becomes. The potential level will guide the robots to unexplored areas. However, as coverage increases, any remaining cells with zero potential level will generate more attractive potential, and will pull robots toward them more strongly. If any robots are delayed by potential minima, the monotonically increasing field will act to free them without resorting to an external mechanism.

The potential level is also used for avoiding obstacles; if a robot sensor detects an obstacle arising from map structural features such as walls, partitions and columns, it will increase the potential level of the associated cell in the map, so that the area becomes sufficiently repulsive to ensure no robot will enter that space. Additionally, if a robot knows the map cells corresponding to the locations of others members of the robot team, it will add a potential level increment to ensure repulsion from these areas as well.

3.2 Repulsion from Explored Areas, other Robots and Obstacles.

A map grid cell may acquire a larger potential level, and become more repulsive, if it represents any of the following:

- Previously visited or scanned areas;
- The location of other robot(s);
- The location of obstacles (e.g., walls).

In the first two cases, the direction of the repulsive force lies along the direction from the cell to the robot. In the last case, however, the direction is rotated by 90 degrees. The result here is that instead of direct repulsion from the obstacle, the repulsion will form a spin field around the obstacle [6], creating a smoother route away from the obstacle. Furthermore when the robot is near a symmetric shaped obstacle that generates equally repulsion potential on both sides, the robot will go around the obstacle instead of being trapped in an equilibrium state. This vortex effect, along with the addition of a small amount of random noise and the monotonic increasing factor due to coverage in the potential field expressions, are the three mechanisms to address potential field minima issues.

The attractive and repulsive potential field definitions introduced below are based on those in (Renzaglia & Martinelli, 2010). The repulsive potential field is given by

$$U_{rep}(q, q_i) = \begin{cases} \frac{1}{2} \frac{\lambda(q, q_i)(1 - \delta(q_i))}{h_{rep}(c)} \left(\frac{1}{\rho(q, q_i)} - \frac{1}{\rho_0} \right)^2, & \rho(q, q_i) \leq \rho_0 \\ 0, & \rho(q, q_i) > \rho_0 \end{cases} \quad (1)$$

where q is the robot location, q_i is the location of each other cell in the map grid, and $\lambda(q, q_i)$ the difference in potential levels of the cells. $\delta(q_i)$ is the Dirac delta function which yields 1 if q_i has potential level zero and otherwise yields zero. This is used to ensure that only non-zero cells contribute to the repulsive potential field. ρ_0 is the maximum sensing distance of the robot and $\rho(q, q_i) = \|q - q_i\|$. The function, $h_{rep}(c)$ defines a repulsive scaling factor based on the percentage coverage, c , and this is where the area-bounds information is leveraged. In this paper, h_{rep} is selected to be the linear function:

$$h_{rep}(c) = ck_{rep} \quad (2)$$

The principal additions to the field used by (Renzaglia & Martinelli, 2010) are the potential difference factor $\lambda(q, q_i)(1 - \delta(q_i))$ – making more explored areas more repulsive - and the coverage scaling factor h_{rep} – reducing overall repulsion with coverage. The use of the coverage factor in the equation makes the repulsive force larger with smaller coverage and become smaller as coverage grows. At the beginning of the exploration, the robot will be driven away from previously explored areas as well as the other robots. As coverage increases, the repulsion from walls and other robots decreases, allowing the robot to explore more closely into corners and niches as well as closer to other robots. Furthermore, if any robots are stalled in minima, a narrow corridor for example, then as coverage increases the situations giving rise to the minima are also changed.

If the robot is at a location q , then it incurs a repulsive force from each other cell in the map q_i , due to this potential function, $F_{rep}(q) = -\nabla U_{rep}(q)$ as follows:

$$F_{rep}(q, q_i) = \begin{cases} \frac{\lambda(q, q_i)(1 - \delta(q_i))}{h_{rep}(c)} \left(\frac{1}{\rho(q, q_i)} - \frac{1}{\rho_0} \right) \frac{q - q_i}{\rho^3(q, q_i)}, & \rho(q, q_i) \leq \rho_0 \\ 0, & \rho(q, q_i) > \rho_0 \end{cases} \quad (3)$$

The summation of the repulsive force from each other cell with a potential level gives the robot at location q a repulsive force $F_{rep}(q)$ that drives it away from obstacles, other robots and previous visited areas.

$$F_{rep}(q) = \sum_{i=1}^n F_{rep}(q, q_i) \quad (4)$$

3.3 Attraction to Unexplored Areas

When a cell in the occupancy map has zero potential level, then this area has not been scanned or visited by any robots. Such a cell contributes an *attractive potential* to the field for each robot. The proposed approach differs from an approach that models an unexplored frontier, since it treats every unexplored cell as an attractive potential goal. Because an unexplored frontier perimeter does not need to be maintained or calculated, this approach is less demanding computationally. Of course, being area based, it can have larger memory requirements since the entire map, though empty, is present in memory from the start. Note, however, that variable resolution occupancy grids methods such as quad trees or binary space partitioning trees could be used to address this (Dudek & Jenkin, 2000).

The attractive potential field, modified from (Renzaglia & Martinelli, 2010), is

$$U_{att}(q, q_i) = \frac{1}{4} h_{att}(c) \delta(q_i) \rho_{goal}^4(q_i) \quad (5)$$

$\delta(q_i)$ ensures that only zero potential cells contribute to the field. The function, $h_{att}(c)$ gives an attractive scaling factor for coverage, c . In this paper, we gain select a simple linear function for this:

$$h_{att}(c) = k_{att}/c \quad (6)$$

The principal addition to the field used in (Renzaglia & Martinelli, 2010) is the coverage factor $h_{att}(c)\delta(q_i)$. The use of the coverage factor in the equation make the attractive force get larger with smaller coverage and smaller as coverage grows. This complements the reduction in repulsion as coverage increases, pulling the robot to explore corners and niches towards the end of the mission. Furthermore, it also acts to dislodge robots from minima by varying the field conditions as coverage changes.

The attractive force is given by

$$F_{att}(q, q_i) = h_{att}(c) \delta(q_i) (q_i - q) \rho_{goal}^2 \quad (7)$$

And summing up all the attractive cells (cells with a potential level of zero) we can get the sum attractive force applied to a robot at map location q ,

$$F_{att}(q) = \sum_{i=1}^n F_{att}(q, q_i) \quad (8)$$

3.4 Sum of forces

This summation of force will attempt to drive the robot away from the other robots, obstacles and previously visited areas, toward unexplored areas.

$$\begin{aligned} F_{total} &= F_{rep,explored\ area} + F_{rep,other\ robots} + \\ &\quad F_{rep,obstacles} + F_{att,unexplored\ area} \\ &= m\ddot{q} - v\dot{q} \end{aligned} \quad (9)$$

where m is the mass of the simulated robot and v a viscous damping factor. These are used as in (Renzaglia & Martinelli, 2010) to generate smooth trajectories.

Due to the coverage factor in the repulsion and attraction force, in the beginning of an exploration mission where the map coverage is small, the robot will tend to focus on spreading out or moving away from the explored area while at a later stage of the mission, the coverage control factor will focus on drawing the robots to un-explored areas, and fill up the empty sections on the map.

Any map that contains a completely enclosed space (i.e., the space internal to a column, elevator shaft, etc.) would form a permanent attractive zone. Since line of sight prevents any robot from scanning such an area, the potential level of grid cells representing the area will remain at zero. This attractive zone will distract robots from dispersing through the building. The approach to this problem laid out in (Liu & Lyons, 2014) will be followed here also: Any area internal to a closed obstacle boundary (as determined by the state of the map) is also considered as an obstacle rather than an attractive region.

3.5 Decentralized Control

A major innovation of the decentralized Space-based Potential Field (D-SBPF) approaches is that it incorporates the knowledge of map exploration into the map and takes that information into consideration while calculating desired motions. Since the potential level of each cell can only increase after sensing (i.e., become more repulsive), the history of the exploration and sensing for each robot is integrated into the map and is easily transferred and compared to the map from a different robot. If the potential level of the same cell from two different robot maps are compared, **then assuming a static map**, the one with the higher potential level represents a later (and more up to date) value. Also, the approach allows easy communication between members of the robot team. It is expected that as the robot team moves through the building, robots will lose contact with each other from time to time and subsequently regain contact. When a robot recovers contact with another robot, they exchange copies of their maps. They each update their maps to the latest values from this exchange, and hence each acquire knowledge of the exploration history.

Before addressing map update, the implementation of the wireless ad-hoc network component of the algorithm is discussed. Since each member of the robot team can handle the path-planning locally with its own sensors and potential field map, they only need to share location information to avoid collisions, and share the change in potential levels of each cell to indicate the level of exploration. This is implemented using an ad-hoc wireless network for robot communication between team members.

Figure 1: Centralized server-controlled (a) versus ad-hoc network connected (b) robot team.

In many traditional robot control schemas, each robot serves as a sensor terminal, taking commands from the central server (see Figure 1(a)), and sending back robot sensor readings as well as other information collected back to the central server. However, in an ad-hoc network scheme, as shown in Figure 1(b), each robot is both an individual terminal as well as a server. The robots do their own obstacle avoidance and path-planning, calculate the potential field map, then share the map with the other robots in the network. This allows stand-alone exploration for each robot in the team, even when the network connectivity is limited. While connected, robots can share information and work collaboratively, and can easily handle the loss and restart of the connection to other robots.

Putting aside for a moment the mechanisms of setting up the ad-hoc network, it is important to consider which network protocol is most useful for this problem. Using the TCP protocol, the sender of a message requires a confirmation from the receiver, or it will continue sending the message; this is the way TCP ensures secure package delivery. In the cases that the connection is not stable, if for example the other robot travels out of wireless signal range, the sender robot will be stuck in the transmission state until the connection is regained. However the UDP protocol requires no confirmation upon receiving the message, therefore the sender can move on to next state of map exploration as soon as the packet is sent out. For this reason, the UDP protocol is used in the D-SBPF algorithm.

3.6 Packet transfer and map update

Let I be the set of robots in the team. Let $Map_i, i \in I$ be the potential level map for robot i in the team. $Map_i(x, y)$ refers to the map cell with coordinates x and y . All robots in the team know the spatial extent of the map, which we simplify here to a rectangle (SX, SY) , and the map resolution (r_x, r_y) . Each robot map therefore has the following number of cells:

$$(X_{max}, Y_{max}) = \left(\left\lceil \frac{SX}{r_x} \right\rceil, \left\lceil \frac{SY}{r_y} \right\rceil \right) \quad (10)$$

Each robot maintains localization information (ox_i, oy_i, θ_i) . In this way, each robot can translate its location (x_i, y_i) to common map coordinates (x, y) using

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \left[\frac{1}{r_x} \right] & 0 \\ 0 & \left[\frac{1}{r_y} \right] \end{bmatrix} \left(R(\theta_i) \begin{bmatrix} x_i \\ y_i \end{bmatrix} + \begin{bmatrix} ox_i \\ oy_i \end{bmatrix} \right) \quad (11)$$

All robots are initialized with an unexplored map, that is with $Map_i(x,y)=0$ for all cells. As each robot explores, it updates its own map to reflect its sensor coverage as described in prior sections. Periodically robots exchange information with each other by sending their current location, so that they can treat each other as repulsive potential sources. Periodically they also send out their current map Map_i and listen for maps from other robots, Map_j for some $j \neq i$. When robot i gets a new map from robot j , it compares its old map to the received map, and updates to a new map as follows:

$$Map_{i,new}(x,y) = \text{MAX}\{Map_{i,old}(x,y), Map_j(x,y)\} \quad x \in 0 \dots X_{max} - 1, y \in 0 \dots Y_{max} - 1 \quad (12)$$

Since MAX is a binary associate operator, it doesn't matter in what order the robots exchange maps, the result from one full set of exchanges will always be unique. Notice that this map rule is a conservative estimate of the exploration level for each map cell. If both robots have scanned the same cell, then a more accurate estimate of the coverage might be the sum of the two levels. However, without tagging the cell with the ID of the robot or robots that generated the coverage level, there is no way to determine if the coverage estimates are independent and hence can be added. **It also assumes a static environment. That latter assumption could be relaxed by associating timestamps with each cell, and by updating based on which map cell had the latest timestamp.**

Consider a robot team exploration mission consisting of three robots: if robot #3 moves too far away from robot #1 and loses direct communication, it is still able to update its own potential field map, as well as getting the map update packet from robot #2 which contains the most recent area exploration data from both robot #1 and #2. Even if any of the robots moves out of wireless communication range, it is still be able to perform stand-alone exploration and attempt to update and synchronize its potential field map with the other members. Once communication resumes, the robot can compare the highest potential level for each cell in the maps it receives, integrating the most recent level of exploration and obstacle information.

There is an additional important advantage of this approach: assuming one robot moves out of the wireless communication signal range of some robots in the team, it can still communicate as long as there is an intermediate robot. The intermediate robot operates effectively as a forwarding node, a relay robot, sending information to the rest of the team. In a centralized server control scenario, once the robot moves out of wireless signal range of its sensor, it will not be able to communicate with the other members of the team. In short, the ad-hoc network significantly increases the robustness of the wireless communication infrastructure of the team.

3.7 System Architecture

The decentralized Space-Based Potential Field algorithm is explained in Figure 2.

Figure 2: Decentralized space-based potential field (D-SBPF) algorithm.

The algorithm is split into two main stages: the initialization stage, and the explore stage. When each robot starts, it enters its initialization stage. At the initialization stage, the robot broadcasts a *request-for-joining* message to all the other robots containing its own IP address and robot information, including position and heading. The robot then waits for the reply to the request message from other robots. When it receives a reply message which contains another robot's information, it adds the other robot to its *local contact list*. Initialization ends when the robot has not received a reply for some threshold time.

After the initialization has finished, the robot then starts the exploration stage. It uses its sensors to gather environment information and constructs its local potential field map. Periodically, the robot communicates with the other robots in its *local contact list*, sending its map and collecting their maps. Upon receiving the potential field maps from the other robots, the robot updates its map using the map update rule (9).

Using the updated map, the robot calculates the potential field force that will drive the robot to explore. It checks the map for overall map coverage. If the map coverage goal has not been met yet, the robot will then go back to the first step of the exploration stage, proceed and repeat until the map has been fully explored.

4 Simulation Results

In this section, simulation results for the D-SBPF algorithm are presented that demonstrate its effective use of multiple robots, deployment locations and times (when available) as well as its effective dispersion of the robot team to search a building quickly. The algorithm is tested on a set of three maps, similar to those used by Renzaglia (Renzaglia & Martinelli, 2010), which offer challenges to potential-field based approaches. However, unlike Renzaglia, we will assume that the robots are equipped with laser sensors that have a field of view constraint of angle $\alpha \leq \pi$. The section begins by introducing the three maps that will be explored. It then presents a series of metrics that will be used to measure performance on each of the maps, and then presents the results for each.

4.1 Maps and Metrics

The algorithm's performance will be evaluated on three examples of building maps (Figure 3(a), (b) and (c) respectively). All of the maps are $10 \times 10 \text{ m}^2$ and are represented on a 30×30 grid. The first example uses a map of an empty building. The purpose of this example is to show how the algorithm directs the robot team to spread out and head toward unexplored areas in an open area. While missions may occasionally need to be carried out in such examples (e.g., an empty warehouse), the main purpose is demonstrate nominal behavior for the algorithm for homogeneous teams of 1 to 3 robots, as well as with temporal and spatial starting delays (leveraging the capability of the decentralized algorithm to handle the loss and regaining of contact with team members).

Figure 3: The three example building maps as displayed by *MobileSim* during a 1 robot exploration. The robot is shown as the shaded pentagon, with 16 sonar signals (unused here) and with lightly-shaded laser field of view

In the second example map, two rooms with open doors are added to the empty space of the first example. The robots need to go in and out of two rooms in order to explore the building fully. This example is one that can give rise to potential field minima between the two rooms. Also, because each room is surrounded by the walls, the net repulsion from the walls outside of the room will form a force preventing a robot from entering the room. In addition, once the robot is inside the room, the net repulsion from the surrounding easily trap a robot inside.

The final map adds several enclosed obstacles strewn across the empty space of the first map. The robots will attempt to reach the other side of a building to complete the

search despite several internal obstacles. But also, the effect of the enclosed space inside obstacles needs to be discounted or it will attract, and distract, the team from finishing the search.

The percentage of coverage, that is the area of the room that has been searched, will be used as the principal metric of performance. If $c(t)=\alpha$ is the coverage at time t , then we define the *performance* to reach that percentage of coverage as $P_\alpha=1/t$. This has the conventional effect of defining faster coverage as larger (and hence better) performance. In the case where more than one robot is being used, then the *speedup* associated with using two robots is defined for coverage $c(t)=\alpha$, by analogy with parallel computing, as the performance of the n robot case $P_{n,\alpha}$ divided by that of the one robot case $P_{1,\alpha}$ for the same coverage:

$$S_{n,\alpha} = P_{n,\alpha} / P_{1,\alpha} \quad (13)$$

We will typically adopt a coverage of $\alpha=95\%$ as our ‘mission succeeded’ coverage threshold. This is an arbitrary threshold chosen to allow timely generation and comparisons of multiple runs. Speedup is a measure of how much more quickly a multirobot team operates than a single robot. A linear speedup, i.e., $S_{n,\alpha} = P_{1,\alpha}/n$, is the desirable case where using n robots is n times as quick as using a single robot.

Another useful metric that will be calculated is efficiency, the benefit of each additional robot that is added, or how effectively on average each additional robot contributes to the mission.

$$E_{n,\alpha} = \frac{S_{n,\alpha}}{n} = \frac{P_{n,\alpha}}{nP_{1,\alpha}} = \frac{T_{1,\alpha}}{nT_{n,\alpha}} \quad (14)$$

where n is the number of robots, $T_{1,\alpha}$ is the time required to finish the exploration using one robot, and $T_{n,\alpha}$ is the time required to finish the exploration using n robots. An efficiency of 1.0 means that on average each robot contributes fully and equally to the mission performance. An efficiency less than 1.0 means that on average each robot is not as efficiently used as in the one robot case, while an efficiency greater than 1.0 means that on average each robot is more efficiently used than in the one robot case. This latter can be due to the team leveraging a characteristic of the problem: For example, two people folding sheets can typically do the job in less than half the time of one because of the nature of the task.

The next sections present the simulation and physical robot results. The D-SBPF algorithm is programmed using C++ under Linux. The *MobileSim* software from Adept *MobileRobots* is used to simulate the performance of the Pioneer 3-AT equipped with SICK laser. The D-SBPF program sends motion commands to, and receives sensory data from *MobileSim* as it would to and from one or more physical Pioneer 3-AT robots. The same D-SBPF implementation is used in the physical robot experiment at the end of this section, with the physical robots substituted for *MobileSim*. A graphing utility called GNUPlot is used to visualize and output the potential field map as well as the information for the robots.

Figure 4: Decentralized-SBPF map coverage examples for the empty room map.

The axes labels are the grid indices for the map and are labelled 0 to 30.

4.2 Open Space Example

The first example is empty map with no obstacles. This example shows how the D-SBPF algorithm directs the robot team to spread out and head toward unexplored areas. The algorithm is carried out with different numbers of robots from 1 to 3. Figure 4 shows three snapshots from the execution of the D-SBPF exploration of this map with one (top), two (middle) and three (bottom) robots. The left column of Figure 4 is the initial situation, with all robots starting in the upper left. The center column shows a time intermediate to full coverage, and the right column shows the final situation, where 95% coverage is the mission conclusion threshold. The time to achieve this coverage varies in each case, of course, and the time (in seconds) at which image in Figure 4 is take appears in the title for that image. In Figure 4 the arrows indicate the direction of potential field at each point in space (i.e., at that cell in the occupancy grid), the thick black lines show the regions that robot sees as obstacles or walls. Each robot is indicated by grey dot, the shaded cone-shapes indicate the sensor-sensing region for each robot, and the darkness of the shade of each cells represents the value of the potential level for that cell, where a darker cell means higher potential level values. If the map is mostly covered then most cells will have been heavily shaded.

In Figure 4 (e,h), we can see that the robots have successfully spread out at the early stage of exploration. In Figure 4 (f,i) we see the robots successfully spread out at the final stage of exploration. In addition to the robots dispersion in location, the robots also have different headings and have minimally-overlapping sensing areas.

Figure 5: Combined graph of coverage vs. time for the empty room scenario.

Num. Robots	Performance	Speedup	Efficiency
1	$P_{1, 95\%} = 0.012$	$S_{1, 95\%} = 1.0$	$E_{1, 95\%} = 1.000$
2	$P_{2, 95\%} = 0.022$	$S_{1, 95\%} = 1.8$	$E_{2, 95\%} = 0.916$
3	$P_{3, 95\%} = 0.058$	$S_{1, 95\%} = 4.8$	$E_{3, 95\%} = 1.610$

Table 1: Exploration performance, multiple robots on empty room map.

Overlaying the coverage versus time graphs for the 1, 2 and 3 robot cases, as shown in Figure 5, where each graph is the average of 10 simulation runs, we can see that the 3-robot-team is able to reach 95% coverage at 17 seconds, the two-robot-team at 46 seconds while the 1-robot-team requires 82 seconds to reach the same level of coverage. The overall exploration performance P_{95} for single-robot team is 0.012, for two-robot team is 0.022 and for three-robot team is 0.058. Table 1 shows the performance, speedup and efficiency metrics for this example.

In the two robot case, the improvement in performance is almost doubled. We observe some strong synergy when multiple robots are working as a team in a map as open as this, as evidenced by the speedup and impressive efficiency for the three robot case. By the end of the single robot case (Figure 4(c)), the robot has circled the center and made forays to each corner. However, in the multiple robot case (Figure 4(f,i)), when one robot finishes exploring the diagonal, it does not need to move to the other sides of the room; instead the second or third robot has already taken care of that due to the empty nature of the room.

Figure 6: Graph of coverage versus time for the open room map with robots starting at different times.

Num. Robots	Start	Performance	Speedup	Efficiency
1	-	$p_{1_i} = 0.012$	$S_1 = 1.00$	$E_1 = 1.000$
2	-	$p_2 = 0.022$	$S_2 = 1.80$	$E_2 = 0.916$
2	Delay	$P_{2d} = 0.014$	$S_{2d} = 1.16$	$E_{2d} = 0.561$
3	Delay	$P_{3d} = 0.019$	$S_{3d} = 1.58$	$E_{3d} = 0.536$

Table 2: Exploration performance for open room map with delayed starts. (Coverage=95%; omitted from subscripts for clarity).

Observing the final phase of exploration shown in Figure 5, we can see very little difference from the result shown in (Liu & Lyons, 2014), the centralized control version of SBPF algorithm. The decentralized version simply moves the calculation of the potential function to the robots from the controlling server, and implements the communication functionality and map update. However, it is now possible to dynamically change the size of the robot team, and hence determine the effect of this on the coverage performance compared to the coverage of a fixed-size robot team.

Figure 6 shows the coverage graph for 1, 2 and 3 robots for the empty room map. However, now each robot starts 10 seconds *later* than the previous one. This allows preceding robots to explore the map some amount before later robots start. All robots start in the same upper left area on the map as in Figure 4. Table 2 shows the metrics.

Figure 7: Graph of coverage versus time for the open room map with robots starting at different times and different start locations.

Num. Robots	Start	Performance	Speedup	Efficiency
1	-	$P_1 = 0.012$	$S_1 = 1.00$	$E_1 = 1.000$
2	-	$P_2 = 0.022$	$S_2 = 1.80$	$E_2 = 0.916$
2	Delay	$P_{2d} = 0.014$	$S_{2d} = 1.16$	$E_{2d} = 0.56$
2	Delay, different location	$P_{2ds} = 0.037$	$S_{2ds} = 3.08$	$E_{2ds} = 1.01$

Table 3: Exploration performance for open room map, delayed start and different start location.

The delayed start strategy with 2 robots reached 95% coverage at 73 seconds, and has an overall exploration performance $P_{2d,95}$ of 0.014; the exploration performance is better than using single-robot team ($P_{1,95}=0.012$), but still not as good as using 2 robots from the beginning with $P_{2,95}=0.022$. The performance when using delayed additional robots is still higher than using single robot, although the rate of increase of coverage over time is not as high as when all start at the beginning of experiment,. Note that at the early stage of the exploration, since the robots start from the same starting point they inevitably have to move through some previously-explored area, the improvement in exploration performance is not significant. Using one additional, delayed robot raises the exploration performance $P_{2d,95}$ from 0.012 to 0.014, and using two additional, delayed robots allows the team to complete the exploration in 51 seconds, and further raised the overall exploration performance $P_{d3,95}$ to 0.019. However, in each case, it is clear from the efficiency metrics (Table 2) that the additional deployed robot is always less efficient than in the simultaneous start case ($0.56<0.96$, and $0.53<1.6$).

Let us now consider the case where that the robots that join the exploration later start from a different location. The additional, delayed robots join the exploration team from the lower left corner of the map, while the initial robots start the mission from the upper left corner of the map. In this scenario the robot team is able to reach 80% coverage in 19 seconds, and finish the exploration at 27 seconds. The overall exploration

performance $P_{2ds,95\%}$ is 0.037, higher than the cases using two robots or one additional delayed robot. The additional delayed robots do not re-cover any previously explored area, and therefore there is an improvement in coverage efficiency above that of separately starting robots in the same starting region (Figure 7).

4.3 Two Room Example

In the next example map, the two room map, the robots need to go in and out of each of the rooms to explore the building fully. Recall that this is a map that can give rise to potential field minima and possibly trap a robot that uses potential field method for path-planning.

Figure 8: Decentralized-SBPF map coverage examples, two room map.

Figure 8 shows examples of the D-SBPF algorithm results for the one, two and three robot cases for the two room map. It is organized similar to Figure 4. The simulation correctly captures the line-of-sight constraint for the laser sensors, i.e., robots are unable to acquire information through walls. (This is visible in the shading of the grid cells; but the sensor cone is always drawn with the same shape).

Figure 9 shows the coverage graph for this example for one to three robots on a team. As before, the graph and metrics in Table 4 are averaged from 10 runs of the algorithm in each case. The graph shows in every case the robot team can achieve its objective. A noteworthy improvement in performance is seen when using two robots compared to just using one robot. The repulsion from one robot is able to push the other robot to unexplored areas faster. Using two robots, the coverage reaches 80% in 80 seconds and

the exploration finishes at 169 seconds; less time than required when using a single robot. That case took 152 seconds to reach 80% coverage and finishes the exploration at 193 seconds. The single-robot team has an overall exploration performance $P_{1,95}=0.0052$, and the two-robot team has $P_{2,95}=0.006$; a speedup of 1.14, much less impressive than the empty room example of course.

Figure 9: Graph of coverage versus time for the two-room scenario.

Num. Robots	Performance	Speedup	Efficiency
1	$P_1 = 0.0052$	$S_1 = 1.00$	$E_1 = 1.00$
2	$P_2 = 0.0060$	$S_2 = 1.15$	$E_2 = 0.57$
3	$P_3 = 0.0074$	$S_3 = 1.42$	$E_3 = 0.47$

Table 4: Exploration performance for the two room map.

In this map the doorway prevents two or more robots from moving forward to unexplored areas while maintaining dispersion at the same time. Using three robots allows the team to reach 80% coverage at 43 seconds, but later the rate of growth decreases dramatically and does not finish exploration until 134 seconds, resulting in an overall exploration performance $P_{3,95}=0.0074$, for speedup of 1.4. Although the increase in exploration performance suggested a linear growth with respect to the number of robots used, the geometry prevents the robots from getting as good dispersion and high exploration performance as in the open space example.

It is interesting to note that in all cases, the coverage graph has a shelf. The $P_{n,80}$ numbers are substantially better than the $P_{n,95}$ numbers. The robots disperse through the map quickly, but leave unexplored space in the narrow corridors at the bottom of the map that takes a while to finish (Figure 8(c,f,i)), and in all cases, only one robot is dispersed to that area. The $S_{2,80}$ metric is 1.92, and $S_{3,80}$ is an impressive 3.54.

The graphs in Figure 10 are for the same two room map using different number of robots. However, each robot starts 10 seconds later than the previous one - allowing the preceding robot(s) to explore the map somewhat before the later ones start.

Figure 10: Graph of coverage versus time for the two-room map with delayed start.

Num. Robots	Start	Performance	Speedup	Efficiency
1	-	$P_1 = 0.0052$	$S_1 = 1.00$	$E_1 = 1.000$
2	Delay	$P_{2d} = 0.0054$	$S_2 = 1.03$	$E_2 = 0.518$
3	Delay	$P_{3d} = 0.0067$	$S_{3d} = 1.29$	$E_3 = 0.428$

Table 5: Exploration performance for the two room map with delayed start.

The results reported in Figure 10 and Table 5 show that in the early stage of the exploration, in which the second robot joins 10 seconds after the first robot, the rate of growth in coverage has no improvement compared to the single-robot case. Using an additional, delayed robot halved the time required to reach 85% map coverage, from 171 to 86 seconds, and ultimately allowed the exploration to finish at 186 seconds, with exploration performance $P_{2d,95} = 0.0054$. Using a second additional, delayed robot further decreased the time required to reach the same level of map coverage by 36 seconds, and increased the overall exploration performance $P_{3d,95}$ to 0.0067.

This is because the narrow doorway limits the dispersion of robots, and the following robots have to travel through the area already explored by previous robots. Thus, they cannot make contributions to map coverage until they have moved to unexplored areas. This effect is even more serious for the third robot. Since the previous two robots (that started twenty and ten seconds earlier) have moved and covered any undiscovered region ahead, the robot that joins the team later cannot reach an undiscovered regions quickly and is not able to provide any extra coverage capability until later in the exploration mission. Thus, we propose that having a second (or third) robot start from a different place will have a bigger impact on performance; even better perhaps than using two robots that start at the same time.

Figure 11: Graph of coverage versus time for the two-room map with delayed start and different start locations.

Num. Robots	Start	Performance	Speedup	Efficiency
1	-	$P_1 = 0.00520$	$S_1 = 1.00$	$E_1 = 1.000$
2	-	$P_2 = 0.0060$	$S_2 = 1.15$	$E_2 = 0.570$
2	Delay	$P_{2d} = 0.0054$	$S_{2d} = 1.03$	$E_{2d} = 0.518$
2	Delay, different location	$P_{2ds} = 0.0071$	$S_{2ds} = 1.37$	$E_{2ds} = 0.689$

Table 6: Exploration performance for two room map with delayed start and different start locations.

Consider the scenario where a second delayed robot joins the exploration team from the lower left corner of the map, while the initial robots start the mission from the upper left corner of the map. This robot team that reaches the threshold coverage earlier than the other robot teams (Figure 11), and finishes the exploration at 140 seconds. This is earlier than the 169 seconds obtained when using two robots from the beginning and the 186 seconds obtained with one additional, delayed robot with same starting position. This team achieved an overall exploration performance of $p_{2ds,95}=0.0071$, a speed of 1.37 compared to 1.03 for just a delayed start, and comparable with the exploration performance of using three robots, $p_{3,95}=0.0074$ (Table 6).

4.4 Internal Obstacle Example

In the third example map, as shown in Figure 12, the robot team must explore to reach the other side of a building that has several internal obstacles and walls including a permanently enclosed area.

Figure 12: Decentralized-SBPF map coverage examples, internal obstacle map.

Comparing the intermediate phase and final phase of Figure 12 with 1, 2 and 3 robots, it can be seen that the robots move toward different sub-regions based on the number of the robots. In Figure 12 (b), the robot moved to the lower half of the map, the largest unexplored sub-region, and finished the exploration after making a circle. On the other hand, the robots separate as shown in Figure 12 (e) and Figure 12 (h), heading towards the upper half and lower half respectively (two robot team), and upper half, middle half and lower half (three robot team). They later complete coverage of the map after the robot that traveled to the lower half makes a circle and returns.

Figure 13: Graph of coverage versus time for the internal obstacle map.

Num. Robots	Performance	Speedup	Efficiency
1	$p_1 = 0.0035$	$S_1 = 1.00$	$E_1 = 1.00$
2	$p_2 = 0.00650$	$S_2 = 1.80$	$E_2 = 0.92$
3	$p_3 = 0.0080$	$S_3 = 2.29$	$E_3 = 0.76$

Table 7: Exploration performance for the internal obstacle map.

As reported in Figure 13, coverage shows an almost linear increase rate at the early stage of the mission, as the robots are driven to the larger blank areas. The rate then decreases slightly, moving to 95% coverage as the robots are drawn towards and see behind the obstacles. Since the robot sensors are line-of-sight sensors, the map coverage will increase by a greater amount when the robot goes around an occlusion and is able to see behind it.

From the results shown in Figure 13 and in Table 7, we can see that the two-robot team reaches 95% map coverage at 155 seconds with an overall exploration performance $P_{2,95}=0.0065$. With a speedup of 1.8, this is close to half the time required when using one robot (287 seconds and exploration performance $P_{1,95}=0.0035$.) Due to the crowdedness of the map, the performance of the third robot in this case is even more constrained. Using three robots has only a slight improvement in performance over using two robots, compared to the improvement from one robot to two robots. The team of three robots finished the exploration and reached 95% map coverage at 125 seconds and the exploration performance $P_{3,95}$ is 0.008. With a speedup of 2.29, the time saved and improvement of exploration performance from upgrading a two-robot team to three-robot team is smaller than the case of switching from a one-robot team to two-robot team with each robot being use with 0.76 efficiency as opposed to 0.92.

Figure 14 reports on the results of using team members that are delayed, and delayed with different start locations. Since this map geometry is divided into smaller regions by the obstacles, the first robot will not quickly achieve high coverage of the entire region, due to the occlusions blocking the line of sight of the sensors. The robot

has to choose one portion (which has higher attractive potential) for exploration. When the second robot joins the exploration, it is able to cover the other portion of the un-explored map and thus able to increase the map coverage soon.

Figure 14: Graph of coverage versus time for the internal obstacle map with delayed start and different start locations.

Num. Robots	Start	Performance	Speedup	Efficiency
1	-	$P_1 = 0.0035$	$S_1 = 1.00$	$E_1 = 1.00$
2	-	$P_2 = 0.0065$	$S_2 = 1.80$	$E_2 = 0.92$
2	Delay	$P_{2d} = 0.0053$	$S_{2d} = 1.54$	$E_{2d} = 0.77$
2	Delay, different location	$P_{2ds} = 0.0071$	$S_{2ds} = 2.02$	$E_{2ds} = 1.02$

Table 8: Exploration performance for internal obstacle map with delayed start and different start locations.

The robot team using one additional, delayed robot finished the exploration at the 186 seconds. This team increased the overall exploration performance of the single-robot team from $P_{1,95}=0.0035$ to $P_{2d,95}=0.0053$. Furthermore, if the additional, delayed robot joins the exploration from a different starting position, the robot team is able to finish the exploration at 141 seconds. This has an overall exploration performance $P_{2ds,95}=0.007$, higher than two-robot team's exploration performance, $P_{2,95}=0.0065$. Again we can see that if the robots start from different locations in the map, the exploration performance is much better than the case where two robots start from the same starting point. The improvement in exploration performance is superior to the case of using one robot and two robots that start from the same place, finishing the exploration earlier than the other cases.

4.5 Robot Implementation

To demonstrate that the D-SBPF method transfers from simulation to real implementation, results are presented from carrying out this algorithm with a team of one and two

Pioneer 3-AT robots. The room to be explored is $5 \times 5 \text{ m}^2$, arranged as shown in Figure 15. The central partition is placed to block the robot's line of sight; complete coverage cannot be attained unless the robots move into out of the enclosed area. Each robot is equipped with the pre-installed ARIA library control and a SICK LMS-200 laser sensor with 180 degrees sensing field of view and 10 m sensing range. Because *MobileSim* was used for the simulation experiments reported previously, the same implementation of the D-SBPF algorithm can be used with the physical robots.

Figure 15: Experimental map (a) and snapshots (b-e) taken during exploration.

However the Pioneer 3-AT implementation differs from the simulation in one important aspect: The robot takes its motion direction from the potential field, and the magnitude depends on both the potential force magnitude and direction. But, when the direction points directly behind the robot, then the robot will *reverse* and back off in addition to turning away. Furthermore, the robots are allowed to move at 1 m/s maximum velocity, and will slow down or stop if another object is in the path of movement. The program terminates normally if the map is 95% explored, and terminates in failure if any robot stays in the same position for more than 10 seconds due to collision with robots or obstacles.

Figure 16: Example of a potential field map output towards the final phase during one trial of the two robot mission.

In the two experiment, the robots will start asymmetrically. Robot #1, as shown on the right of Figure 15 (b), will start first. Ten seconds after robot #1 starts, robot #2, as shown on the left of Figure 15 (b), will join the exploration. The experiment was carried out 5 times and map coverage percentage was recorded. Figure 17 shows the graph of coverage over time for the average of the 5 runs for both one and two robot teams. **The averaged performance shows 95% coverage in 145 seconds for the one**

robot team and 125 seconds for the two robot team, for $P_{1,95}=0.0068$, and $P_{2,95}=0.008$ and $S_{2d}=1.16$.

The potential field map (e.g., Figure 16) showed some differences with the actual map geometry, for example the section between the sub-regions is distorted while it should remain a straight line. The error is caused by bad localization: No separate mapping and localization module was run as part of the algorithm; not a problem for the simulation, but an issue for the real robots. Although the error makes the potential field map inaccurate compared to the actual map, the map is still sufficient for indoor exploration missions and allows the robot to reach the threshold map coverage without collisions.

Figure 17: Potential Field Map coverage versus time during the exploration mission for one (triangles) and two robot teams.

Comparing the result shown in Figure 17 with the centralized SBPF method reported in (Liu & Lyons, 2014): first we can see both algorithms finishes the exploration in a similar amount of time. The two robot case results in one robot exploring part of the map while the robot that starts later can just head toward the unexplored region. However, the speedup is very impressive in this case given the small space explored and the issues with localization.

5 Conclusion and Future Work

The principal contribution of this paper is a novel approach to motion planning for multi-robot search problems, the decentralized Space-Based Potential Field (D-SBPF) method. The method assumes that the overall size of the area to be searched is known and this information is used to develop a simple, uniform potential field framework. The method uses an extended occupancy grid to represent the space, where each cell can be attractive (if undiscovered) or repulsive (if discovered). A non-monotonic fieldscale factor proportional to coverage is also used to improve searching of corners and niches and to assist in moving robots out of potential minima. A second contribution of the paper is a fast, decentralized map exchange and update method. An advantage of this approach is that robots can leave/join/rejoin the team at any stage, as might be expected as robots traverse a building and lose mutual connectivity.

Several simulation results were presented to show the performance of the approach. The non-monotonic field approach introduced here can be compared successfully with

other potential field approaches in terms of the effect of potential minima. Despite the opportunity in maps two and three for many field minima as discovered by Renzaglia (Renzaglia & Martinelli, 2010), our one, two and three robot teams were able to navigate the maps without stalling or needing a role-based approach to pull robots from minima.

Metrics of performance, speedup and efficiency, based on similar parallel computation metrics, were introduced and calculated for all these examples. These metrics tease out the benefits of team size, allowing robots to start at different times, and selecting differing starting locations made possible by the decentralized algorithm. The results can be summarized as follows:

1. In the cases where the robots start together, we can see that for the empty room map and the room with obstacles, the robot team with higher number of robots has better coverage efficiency and speed.
2. For maps that allow robots to disperse earlier in the exploration stage (maps one and two), the improvement in exploration performance of using multiple robots is better than the maps that have complex geometry (map two and three) and constrain robot dispersion.
3. In the scenarios where another robot joins the exploration shortly after the first robots start, one would expect the rate of map coverage would surge immediately after the second robot joins. However, this is not seen here. Since the later robots start from the same starting region, they will inevitably have to move through previously explored areas to get to unexplored areas before they are able to make contributions and increase the map coverage.
4. Simulation results were also collected with robots joining the team later from different starting locations. These results show that the rate of map coverage increase is raised to a level similar to the rate when using multiple robots at the beginning of the exploration.

The three maps explored in this paper were all relatively small compared to the robot ($10 \times 10 \text{ m}^2$). This restriction comes from what could also be replicated with the Pioneer team. An important aspect of ongoing work is whether these results can be replicated for larger areas and conducting more Pioneer team experiments.

The D-SBPF uses a simple max operator to map coverage information. However, this is conservative and may ignore contributions to coverage. This causes a shelf in the coverage graphs for maps two and three. An improvement is to further extend the occupancy grid so that not only is cell coverage stored, but also a tag which indicates which robot made the coverage. The additional tag allows a less conservative update rule to be evaluated. **The method introduced is limited to static maps, but an extension to dynamic maps was briefly mentioned.**

No SLAM module was included in the system architecture used to evaluate D-SPBF so as to focus attention on just the coverage problem. However, a crucial step in conducting more physical robot evaluations is to add a SLAM implementation. Julia et al. (Julia M. , Gil, L, & Reinoso, 2008) for example integrate their extended occupancy map approach with FastSLAM. Future work will include the additional of SLAM and the evaluation of less conservative map update rules.

Acknowledgement

This research was supported by the Defense Threat Reduction Agency, Basic Research Award #HDTRA1-11-1-0038

REFERENCES

- Arkin, R. C. (1998). *Behavior-Based Robots*. Cambridge MA: MIT Press.
- Arkin, R., & Diaz, J. (2002). Line-of-sight constrained exploration for reactive multigent robotic teamss. *Int. Workshop on Adv. Motion Control*. Maribor, Slovenia.
- Batalin, M. A., & Sukhatme, G. S. (2003). "Coverage, Exploration and Deployment by a Mobile Robot and Communication Network. *Intl. Workshop on Info. Processing Sensor Networks*.
- Baxter, J., Burke, E., Garibaldi, J., & Norman, M. (2007). Multi-Robot Search and Rescue: A Potential Field Based Approach. *Autonomous Robot and Agents* , 76, 9-16.
- Birk, A., & Carpin, S. (2006). Merging Occupancy Grid Maps from Multiple Robots. *Proceedings IEEE* , 94 (1384-1397).
- Burgard, W., Moors, M., Dox, D., Simmons, R., & Thrun, S. (2000). Collaborative Multirobot Exploration. *IEEE Int. Conf. Rob. & Aut.*
- Cepeda, S., Chaimowicz, L., Soto, R., Gordillo, J. L., Alanise-Reyes, E. A., & Carrillo-Arce, L. C. (2012). A Behavior-Based Strategy for Single and Multi-robot Autonomous Exploration. *IEEE Sensor Journal* , 12 (12773-12797).
- Chand, P., & Carnegie, D. (2013). Mapping and Exploration in a Hierarchical Heterogeneous Multi-robot System using Limited Capability Robots. *Robotics and Autonomous Systems* , 61 (6), 565-579.
- Dudek, G., & Jenkin, M. (2000). *Computational Principles of Mobile Robotics*. Cambridge: Cambridge University Press.
- Herath, D., Kodagoda, D., & Dissanayake, G. (2008). New Framework for Simultaneous Localization and Mapping: Multi map SLAM. *IEEE Intl. Conf. Robotics Automation*.
- Jensen, E., & Gini, M. (2013). Rolling Dispersion for Robot Teams. *Joint Conf. Artificial Intelligence*.
- Julia, M., Gil, A., & Reinoso, O. (2012). A comparison of path planning strategies for autonomous robot exploration and mapping of unknown environments. *Autonomous Robots* , 33, 427-444.
- Julia, M., Gil, A., L. P., & Reinoso, O. (2008). Potential Field Based Integrated Exploration for Multi-robot Teams. *Intl. Conf. Informatics Control, Robotics Automation*.
- Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research* , 5 (1).
- Lau, H. (2003). Behavioral Approach for Multi-Robot Exploration. *Australasian Conf on Robotics and Automation*.
- Liu, T.-L., & Lyons, D. (2014). Leveraging Area Bounds Information . *13th Int. Conf. on Intelligent Autonomous Systems*. Padua Italy.
- Lyons, D. (1986). Tagged potential fields: An approach to specification of complex manipulator configurations. *Int. Conf. on Robotics and Automation*.
- Lyons, D., Shethra, K., & Liu, T.-L. (2013). Fusion of Ranging Data from Robot Teams Operating in Confined Areas. *SPIE Defense, Security and Sensing: Multisource, Multisensor Information Fusion*. Baltimore MD.

- Mi, Z., Yang, Y., & Liu, G. (2011). Coverage Enhancement of Mobile Multi-agent Networks while Preserving Global Connectivity. *IEEE Intl. Conf. Robotics Automation*.
- Min, H., & Papanikolopoulos, N. (2011). The Multi-robot Coverage Problem for Optimal Coordinated Search with an Unknown Number of Robots. *IEEE Intl. Conf. Robotics Automation*.
- Ozisik, O., & Yavuz, S. (2008). An Occupancy Grid Based SLAM Method. *Intl. Conf. CIMESA*.
- Popa, D. O., Helm, C., Stephanou, H. E., & Sanderson, A. C. (2004). Robotic Deployment of Sensor Net-work Using Potential Fields. *IEEE Intl. Conf. Robotics Automation*.
- Renzaglia, A., & Martinelli, A. (2010). Potential Field Based Approach for Coordinate Exploration with a Multi-robot Team. *IEEE Intl. Workshop on Safety Security Rescue Robotics*.
- Rogers, J., Nieto-Granda, C., & Christensen, H. (2014). Coordination Strategies for Multi-robot Exploration and Mapping. *Int. J. Rob. Res.* , 33 (4), 519-533.
- Schneider, F. (2003). A potential field based approach to multirobot formation navigation. *IEEE Int. Conf. Intelligent Systems and Signal Processing*.
- Schwager, M., Rus, D., & Slotine, J. (2011). Unifying Geometric, Probabilistic, and Potential Field Approaches to Multi-robot Coverage Control. *Intl. Symposium ISRR*.
- Visser, A., de Hoog, J., Jimenez-Gonzalez, A., & Martinez de-Dios, J.-R. (2013). Discussion of Multirobot Exploration in Communication Limited Environments. *Workshop on Towards Fully Decentralized Multi-Robot Systems: Hardware, Software and Integration, ICRA 2013*. Karlsruhe Germany.

BIOGRAPHY

Dr. Damian M. Lyons is an Associate Professor of Computer Science at Fordham University and the Director of Fordham's Robotics and Computer Vision Laboratory. Dr. Lyons has undergraduate degrees in Math (B.A., 1980) and Electrical Engineering (B.A.E., 1980). He earned his master's degree in Computer Science (M.Sc., 1981) from Trinity College, University of Dublin, Ireland and his Ph.D. in Computer Science from the University of Massachusetts at Amherst (Ph.D., 1986).

Dr. Lyons' research interests are in Robotics and Computer Vision. In robotics his interests include formal methods approaches to robot program representation/analysis and multi-robot exploration systems. In Computer Vision, his interests include target tracking, camera handoff, multisensory fusion and behavior recognition. His background includes over 15 years as a researcher and research program manager at the US division of Philips corporate research laboratories; he was Department head for the Video and Display Processing research department, responsible for technical leadership and funding for this diverse group, and project leader for Philips' research activities in Automated Video Surveillance. He joined Fordham in 2002, and served as chair of the Department of Computer and Information Science at Fordham from 2006 to 2011. He has served on numerous program committees, has published over 80 technical papers in conferences, journals and books, and is inventor/co-inventor of 13 US patents. Dr. Lyons is a senior member of the IEEE.

Tsung-Ming (James) Liu is a Software Engineer at Visible World NY. He earned a Bachelor's degree in Mechanical Engineering from National Taiwan University, a Master's degree in Mechanical Engineering from Columbia University NY (MS 2010) and a Master's degree in Computer Science from Fordham University NY (MS 2014). His Fordham MS thesis was entitled "Multi-Robot Exploration with Space-Based Potential Field Map." The centralized version of this algorithm was published as a conference paper in the 2014 Int. Conf. on Intelligent Autonomous Systems in Padua, Italy.

PHOTOS