Effects of Multi-Robot Team Formations on Distributed Area Coverage

Prithviraj Dasgupta, Taylor Whipple, Ke Cheng
University of Nebraska at Omaha, USA

ABSTRACT

We consider the problem of distributed coverage of an initially unknown environment using a multi-robot system. We specifically focus on a coverage technique for coordinating teams of multiple mobile robots that are deployed and maintained in a certain formation while covering the environment. We have analyzed our technique theoretically and experimentally to verify its operation and performance within the Webots robot simulator as well as on physical robots. Our experimental results show that our coverage technique with robot teams moving in formation can perform comparably with a technique where the robots move individually while covering the environment. We also quantify the effect of various parameters of the system such as the size of the robot teams, the presence of localization and wheel slip noise, as well as environment related features like the size of the environment and the presence of obstacles and walls on the performance of the area coverage operation.

Keywords: Multi-robot systems, distributed area coverage, team formation, flocking, swarm robotics.

INTRODUCTION

Robotic exploration of an unknown environment using a multi-robot system is an important topic within robotics that is relevant in several applications of robotic systems. These applications include automated reconnaissance and surveillance operations, automated inspection of engineering structures, and even domestic applications such as automated lawn mowing and vacuum cleaning. An integral part of robotic exploration is to enable robots to cover an initially unknown environment using a distributed terrain or area coverage algorithm. The coverage algorithm should ensure that every portion of the environment is covered by the coverage sensor or tool of at least one robot. Simultaneously, to ensure that the coverage is efficient, the coverage algorithm should prevent robots from repeatedly covering the same regions that have already been covered by themselves or by other robots. In most of the current multi-robot area coverage techniques, each robot performs and coordinates its motion individually. While individual coverage has shown promising results in many domains, there are a significant number of scenarios for multi-robot exploration such as extra-terrestrial exploration, robotic demining, unmanned search and rescue, etc., where the system can perform more efficiently if multiple robots with different types of sensors or redundant arrays of sensors can remain together as single or multiple cohesive teams (Cassinis, 2000; Chien *et al.*, 2005; De Mot, 2005). For example, in the domain of robotic demining (Bloch, Milisavljevc & Acheroy, 2007), where

autonomous robots are used to detect buried landmines, the incidence of false positive readings from underground landmines can be significantly reduced if robots with different types of sensors such as ground penetrating radar (GPR), IR (infra-red) sensors and metal detectors are able to simultaneously analyze the signals from potential landmines. In such a scenario, it would benefit if robots, each provided with one of these sensors, are able to explore the environment while maneuvering themselves together as a team. Multi-robot formation control techniques provide a suitable mechanism to build teams of robots that maintain and dynamically reconfigure their formation, while avoiding obstacles along their path (Mastellone, Stipanovic, Graunke, Intlekofer & Spong, 2008; Olfati Saber, 2006; Smith, Egerstedt & Howard, 2009). However, these techniques are not principally concerned with issues related to area coverage and coverage efficiency. To address this deficit, in this paper, we investigate whether multi-robot formation control techniques and multi-robot area coverage techniques can be integrated effectively to improve the efficiency of the area coverage operation in an unknown environment by maintaining teams of multiple robots.

Recently, miniature robots that have a small footprint size are being used for applications such as automated exploration of engineering structures (Rutishauser, Corell &Martinoli, 2009; Tache *et al.*, 2009). Similarly, unmanned aerial vehicles (UAVs) and micro-helicopters that have memory and computation capabilities comparable to these mini-robots are being widely used in several domains such as aerial reconnaissance for homeland security, search and rescue following natural disasters, monitoring forest fires, wildlife monitoring, etc (Anderson *et al.*, 2008). Mini-robots are attractive because they are relatively inexpensive to field and a swarm of several mini-robots can be fielded at a cost comparable to fielding one or a few large robots. A multi-robot system that consists of several mini-robots also improves the robustness of the system. However, coordinating the actions of mini-robots to make them work cooperatively (e.g., move in formation) in a distributed manner becomes a challenging problem. We have approached this problem using a flocking-based technique (Gokce & Sahin, 2009; Balch & Arkin, 1998) to control the movement of robots so that they can move in formation. We have theoretically analyzed our team-formation techniques and identified certain conditions under which team formation improves the efficiency of distributed area coverage. We have also verified our techniques through extensive experiments on the Webots robotic simulation platform as well as using physical robots within an indoor environment. Our analytical and experimental results show that our team-based coverage techniques for distributed area coverage can perform comparably with other coverage strategies where the robots are coordinated individually. We also show that various parameters of the system such as the size of the robot teams, the presence of localization and wheel slip noise[1], as well as environment-related features like the size of the environment and the presence of obstacles and walls significantly affect the performance of the area coverage operation.

RELATED WORK

Much of the formation control research with multi-robot teams (Bahceci, Soysal & Sahin, 2003; Gokce & Sahin, 2009; Olfati Saber 2006; Sahin & Zergeroglu, 2008; Turgut, Celikkanat, Gokce & Sahin, 2008) has been based on Reynolds' model for the mobility of flocks (Reynolds, 1987). Reynolds prescribes three fundamental operations for each team member to realize flocking - *separation*, *alignment* and *cohesion*. In the flocking model, each robot in a robot team adapts its motion and position based on the current position and heading of other team members such as a team leader or an immediate neighbor(s). This allows the robot team to remain in formation

while moving as well as adapt its formation while avoiding obstacles. Following Reynolds' model, Chen & Luh (1994) and Wang (1989) describe mechanisms for robot-team motion while maintaining specific formations where individual robots determine their motion strategies from the movement of a team leader or neighbor(s). In (Balch & Arkin, 1998), the authors describe three reactive behavior-based strategies for robot teams to move in formation, viz., unit-center-referenced, neighbor-referenced, or leader-referenced. In contrast to these approaches, Fredslund & Mataric (2002) describe techniques for robot team formation without using global knowledge such as robot locations, or the positions/headings of other robots, while using little communication between robots. Smith, Egerstedt & Howard (2009) have used a combination of graph theory and control theory-based techniques to effect multi-robot formations However, in most of these approaches, the main objective is to achieve and maintain a certain formation and not to ensure the efficiency of tasks, like area coverage, being performed by the robots. Complementary to these approaches Spears, Kerr & Spears (2006) have used physics-based approaches to form and navigate multi-robot teams.

Distributed coverage of an unknown environment using a multi-robot system has been an active area of research for over a decade and excellent overviews of this area are given in (Choset, 2001; Stachniss, Mozos & Burgard, 2008). Subsequently, several techniques for multi-robot coverage such as using Boustrophedon decomposition (Rekleitis, New, Rankin & Choset, 2008), using occupancy grid maps (Burgard, Moors, Fox, Simmons & Thrun, 2005), using probabilistic Bayesian models of the coverage map, information gain-based heuristics and graph segmentation techniques (Wurm, Stachniss & Burgard, 2008), ant-based coverage algorithms (Koenig, Szymanski & Liu, 2001; Wagner, Altshuler, Yanovski & Bruckstein, 2008) have also been proposed. Tzanov (Tzanov, 2006) provides techniques that can be used by a group of robots to cover an initially unknown environment using either a frontier expansion method when the robots have perfect localization, or, using depth first traversal proceeding along triangulations of the environment when the robots localize themselves only with respect to each other. Recently, several techniques have been proposed where robots incrementally build a map of the environment, using a graph traversal technique (Gabriely & Rimon, 2001) and store the map either within the memory of each robot (Cheng & Dasgupta, 2007, Kaminka, Schechter & Sadov, 2008; Rutishauser, Correll & Martinoli, 2009) or at a central location that can be accessed by all robots (Koenig, Szymanski & Liu, 2001). However, these researchers principally focus on controlling robots individually and designing different coordination strategies between them so that the robots can cover the environment while reducing repeated coverage among the regions covered by different robots. In contrast, our work focuses on achieving area coverage while coordinating teams of robots in a distributed manner instead of coordinating each robot's movements individually

MULTI-ROBOT DISTRIBUTED AREA COVERAGE

We consider a scenario where R mobile robots are deployed into an initially unknown two-dimensional environment. Without loss of generality, we assume that the environment is a square with each side of length D. The obstacles in the environment are assumed to be convex shaped. and the location of these obstacles are not known initially by the robots. Let O be the area within the environment that is inaccessible to the robots because those areas are either occupied by obstacles or too tight for a single robot to fit into. The area of the environment to be covered by robots is given by $D^2 - O$ sq. units. Each robot is equipped with a square coverage tool with a width $d << D$. We define the duration of a single timestep as the time required by a robot to

travel a distance equal to the length of its own footprint measured in the direction of its motion. Let a_r^t denote the action performed by a robot $r \in R$ during a timestep t that results in the robot's motion. Let c_r^t denote the corresponding area covered by robot r's coverage tool because of its motion during the timestep t. The objective of the distributed area coverage problem is to find a sequence of actions for each robot that ensures the following criteria:

1. *Maximum Coverage Criterion.* The area of the environment covered by the coverage tool of at least one robot is maximized, i.e., $max\{\cup_{r \in R} \cup_{t=1...T} c_r^t\} \cap \{D^2 - O\}$.
2. *Minimum Overlap Criterion.* The overlap between the regions covered by different robots is minimized, i.e., $min \cap_{r \in R} \cap_{t=1...T} c_r^t$. This criterion ensures that the system performs efficiently and robots do not expend time and energy to revisit regions that have already been covered by other robots.
3. *Distributed Behavior Criterion.* Each robot should determine its actions autonomously, in a completely distributed manner, so that the system can be scalable and robust.

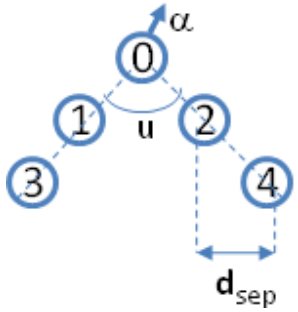


Figure 1: A robot team showing the position identifiers of each robot. The angular separation in the team is u, *the separation between adjacent robots is* d_{sep} *and* α *is the heading of the team.*

Each robot in our system is a two-wheeled robot equipped with forward-facing distance sensors to avoid obstacles and is capable of wireless communication. Each robot is also provided with a local positioning system (a GPS node in the simulator or an overhead camera-based positioning system in the physical experiments) to determine its position in the environment.

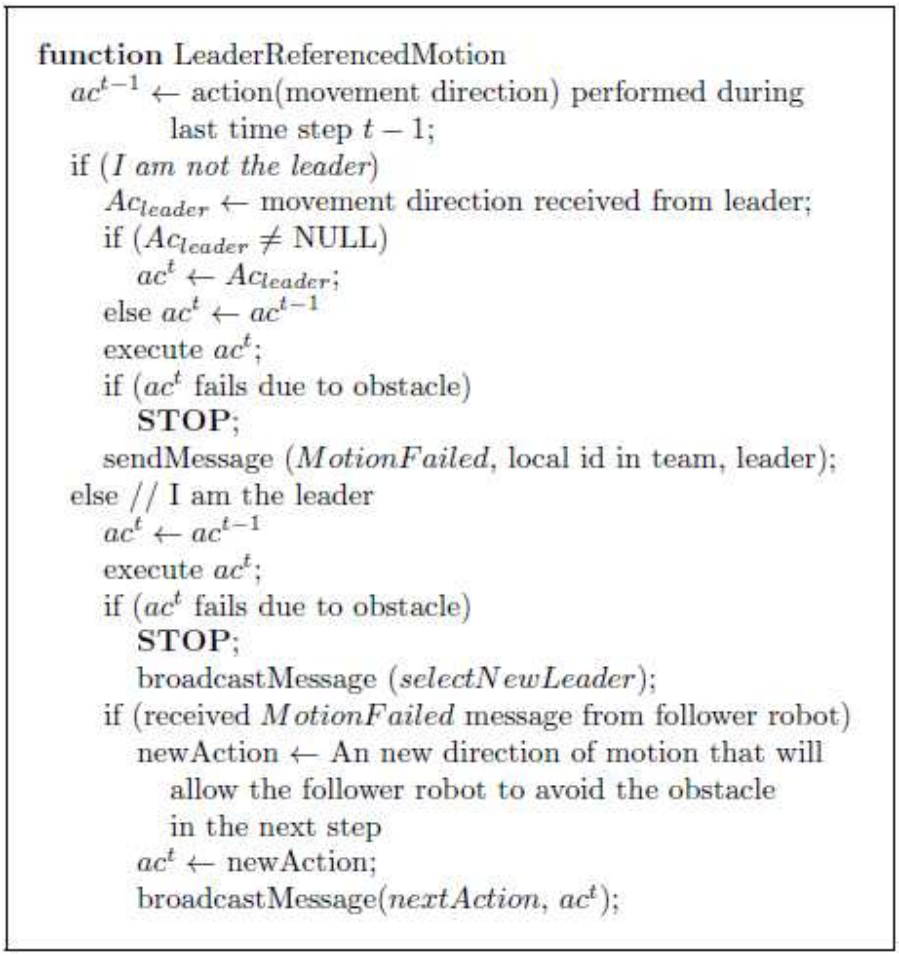
Team Representation

We have defined a robot team as a set of robots (>=2) that are able to navigate within an environment while avoiding obstacles and while preserving the team's shape and configuration. The essential parameters related to a robot team are described as follows**: (1) Team Leader**. Our robot team formation technique is inspired by the leader-referenced motion described in Reynolds' flocking model (Reynolds, 1987). In the leader referenced motion, one robot in a team of robots is selected as the leader. The leader robot guides the motion of the rest of the robots in the team by communicating its direction of movement to all other team members. A robot that is not the team leader is called a follower robot. **(2) Team Position Identifiers**. To interact with the follower robots in a team, each leader robot assigns a local position identifier to each robot within the team. The leader robot's position identifier is 0, robots to the left of the leader robot are assigned odd integer identifiers starting from 1, while robots to the right of the leader robot are assigned even integer identifiers starting from 2. **(3) Team Shape**. To enable efficient movement of a team, the number of follower robots that are located on either side of a leader robot in a team are equally balanced. The angular separation between the two sets of followers robots, denoted by $u \in [0,\pi]$, denotes the shape of the team, where u is measured in radians. When $u = 0$, the team is organized into a vertical line-shape, when $u = \pi$, the team is organized

into a horizontal line-shape, and when $0 \leq u \leq \pi$, we get a V-shape formation in the team, as shown in Figure 1. **(4) Team Configuration**. To maintain the shape of the team while in motion, each robot has to ensure that its relative position within the team does not change when the team moves. To achieve this, every pair of robots in the team maintains a separation of d_{sep} units between each other.

Single-Team Flocking

In our single-team flocking technique, the leader robot communicates the direction it is moving as the prescribed direction of motion for each follower robot in the team. Each follower robot then attempts to move in the prescribed direction. If any follower robot fails to move in this direction, it stops and communicates to the leader robot that its motion failed. Depending on the position of the follower robot in the team and its attempted direction of motion, the team leader then selects a new direction of motion that would possibly allow the affected follower robot to avoid the obstruction in its path. The team leader then broadcasts this newly selected direction as the prescribed direction for the next time step to all the follower robots in the team. In some scenarios, due to communication noise, a follower robot might fail to receive the communication containing the prescribed direction of motion from the leader robot. Then the follower robot just continues to move in the same direction it moved during its previous time step. The pseudo-code algorithm used by a team of robots in the leader-referenced motion strategy is described in Figure 2.



```
function LeaderReferencedMotion
  ac^{t-1} ← action(movement direction) performed during
        last time step t − 1;
  if (I am not the leader)
    Ac_leader ← movement direction received from leader;
    if (Ac_leader ≠ NULL)
      ac^t ← Ac_leader;
    else ac^t ← ac^{t-1}
    execute ac^t;
    if (ac^t fails due to obstacle)
      STOP;
    sendMessage (MotionFailed, local id in team, leader);
  else // I am the leader
    ac^t ← ac^{t-1}
    execute ac^t;
    if (ac^t fails due to obstacle)
      STOP;
      broadcastMessage (selectNewLeader);
    if (received MotionFailed message from follower robot)
      newAction ← An new direction of motion that will
        allow the follower robot to avoid the obstacle
        in the next step
      ac^t ← newAction;
      broadcastMessage(nextAction, ac^t);
```

Figure 2: Algorithm used by a robot to realize the leader-referenced formation control.

Formation Maintenance. When a team of robots moves in formation, the wheel slip noise and encoder readings can cause one or more of the team members to lose their desired positions which destroys the configuration in the team. To address this problem, a leader robot uses a dynamic formation maintenance protocol to ensure that each follower robot retains its position in the team. In this protocol, the team leader first calculates the desired positions (DP_i) of every follower robot i relative to its own position and sends it to follower robot i. Each follower robot i compares its desired position DP_i with its actual position AP_i. A follower robot i adjusts its speed (move faster or slower) proportionally to $\|AP_i - DP_i\|$ so that it can reach its desired position and maintain the configuration of the team. The calculation of DP_i for follower robot i is given below. In these formulae, the actual position AP_i is represented by (x_{APi}, y_{APi}), the desired position DP_i is represented by (x_{DPi}, y_{DPi}), α is the direction of motion of the team, u is the angular separation in the team, d_{sep} is the linear separation between adjacent robots and i is the local identifier of a follower robot in a team: (a) Case 1: $0 \leq a < \pi$

$$x_{DP_i} = \begin{cases} x_{AP_i} - \frac{i}{2} \times d_{sep} \times \cos(\alpha - \frac{u}{2}) & \text{if } i \text{ is odd} \\ x_{AP_i} + \frac{i}{2} \times d_{sep} \times \cos(\alpha - \frac{u}{2}) & \text{if } i \text{ is even} \end{cases}$$

$$y_{DP_i} = y_{AP_i} - \frac{i}{2} \times d_{sep} \times \sin(\alpha - \frac{u}{2})$$

(b) Case 2: $\pi < a \leq 2\pi$

$$x_{DP_i} = \begin{cases} x_{AP_i} + \frac{i}{2} \times d_{sep} \times \cos(\alpha - \frac{u}{2}) & \text{if } i \text{ is odd} \\ x_{AP_i} - \frac{i}{2} \times d_{sep} \times \cos(\alpha - \frac{u}{2}) & \text{if } i \text{ is even} \end{cases}$$

$$y_{DP_i} = y_{AP_i} + \frac{i}{2} \times d_{sep} \times \sin(\alpha - \frac{u}{2})$$

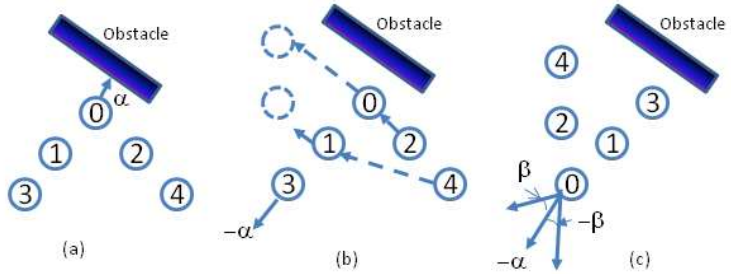


Figure 3: (a) The leader robot (id=0) in a team of five robots encounters an obstacle. (b) A new leader is selected (id=3); robots (id=2, 4) are the robots that have to move the minimum distance to get into the new formation (c) New robot id-s are assigned and the new leader robot selects its heading from randomly between $-\alpha \pm \beta$.

Team Reconfiguration. A leader robot that encounters an obstacle ahead of it will fail to move in its direction of motion. In such a scenario, the team leader stops and communicates to the follower robots to stop moving. Then, the leader robot selects a new leader. If the obstacle is encountered by the old leader using its forward-facing distance sensors on its righthand (lefthand) side going clockwise from current heading, then the follower robot that is farthest from the leader on its lefthand (righthand) side is selected as the new leader. If the old leader robot approaches the obstacle orthogonally resulting in comparable readings on both pairs of the forward-facing (left and right) distance sensors, then one of the two follower robots that is farthest from the old leader robot and has the lowest identifier is selected to become the new leader. Sometimes a team of robots may end up in a tight space such as concave shape where two walls of an obstacle converge. Such scenarios are difficult for reformation because the team is surrounded by obstacles on both its left and right sides. To handle such scenarios, when the leader robot encounters an obstacle the entire team stops and all the robots in the team back up a certain distance by reversing the direction of rotation of their wheels but not changing their

heading. The team attempts to reform only after backing up a fixed distance after none of the robots in the team pick up an obstacle on their IR proximity sensors.

A scenario illustrating team reconfiguration is shown in Figure 3. One of the principal objectives of the team reconfiguration is to enable rapid reconfiguration of the team when the leader encounters an obstacle. To enable this, the follower robots between and including the new and old leader robots do not change their relative position in the team while reconfiguring (Figure 3(b)). The old leader then calculates the relative positions of the remaining follower robots in the new team so that the sum of the distances traveled by these robots to get into their desired positions under the new team leader is minimized. (Figure 3(c)). It then communicates these desired positions to the respective follower robots. The new leader robot also selects a new heading for itself and the team based on a Braitenberg controller that uses the perceived location of the obstacle on the old leader robot's sensor and calculates an appropriate turning angle to ensure that the team turns away from the obstacle and does not encounter it again after forming a new team The new leader robot adds a certain amount of random noise to the new direction calculated by the Braitenberg controller - if α is the turning angle calculated, it selects a value in the range of $\alpha \pm \beta$, where $\beta \in U[0, 10]$ degrees.

In certain scenarios, the obstacles encountered by the robot team might have a complex shape. This can result in the desired positions of one or more the follower robots being unreachable or being occupied by an obstacle and the robots might have to re-attempt several times, perhaps unsuccessfully to get into the desired configuration. To avoid repeated looping by the follower robots to get into their desired position and thereby reduce the stoppage time of the team following reconfiguration, the new leader and the follower robots do not wait to get into their new positions before start to move as a new team. Instead, as soon as the new leader robot reaches its desired position, it starts to move in its new direction. If the path of the follower robots to their desired positions while reconfiguring is occluded or occupied by an obstacle, the robots attempt to avoid the obstacle by turning away from the obstacle using the perceived location of the obstacle from their IR distance sensors, moving a random distance away from the obstacle, and reattempting to resume its desired motion as the team. The new leader robot adjusts its speed to give the follower robots that have not yet reached their desired positions more time to catch up with the rest of the team. After starting to move in its new direction, the new leader updates and communicates the desired positions of the follower robots so that they can move directly towards their new position and retain the formation of the new team. Finally, if the new leader is unable to reach its desired position after repeated tries, it aborts the movement and attempts to go in a direction in which it does not perceive any obstacles. It moves for a random distance in this new direction and tries to reform the team from its new position.

Single-Team Coverage Technique

After a team of multiple robots is assimilated using the technique described above, the next step is to enable the team to cover the environment using a coverage technique. The coverage technique for a robot team is implemented by the team's leader robot. Each robot in the team, including the leader robot records the coordinates of the locations it has covered over the last H time steps within a data structure called its coverage history. Each follower robot communicates this coverage information at intervals of H time steps to the leader robot. The finite size H of the coverage information recorded makes the coverage technique amenable to implementation within on-board memory limitations of robots. To combine the coverage information of the team, the leader robot uses a node counting technique (Koenig, Szymanski & Liu, 2001). In the node-

counting technique, the leader robot uses a data structure called a coverage map that contains the locations or coordinates visited by itself and the follower robots. Each location is associated with a real number that is initialized to zero. Every time a location appears in the coverage history of the leader robot or one of the follower robots, the number associated with the location is incremented. This results in the formation of a landscape within the leader robot's coverage map. Locations associated with large number or a 'high altitude' on this landscape indicate regions that have been covered multiple times, while regions with a smaller number or zero associated with them denote infrequently visited and unvisited regions respectively. To navigate the team, the leader robot selects a direction that will take it towards the lowest (least covered) point on this coverage landscape. A detailed description of the coverage technique used by the team leader to navigate a single team is given in (Cheng & Dasgupta, 2007). When the leader robot of a team has to change because the team encountered an obstacle, the old leader robot communicates its coverage history to the new leader robot so that it can continue efficient coverage without re-covering regions already covered by the team in its previous configuration.

Multi-Team Distributed Coverage Technique

The single-team coverage technique described above provides a mechanism for multiple robots to move together as a single team. However, when team sizes are large (for example, greater than 10 robots per team), it becomes challenging for the robots to maintain the configuration of the follower robots in the team because of frequent reformations of the team to avoid obstacles, and the motion and communication noise in the follower robots. To prevent the formation of large teams, we limit the maximum allowable size of a team to T_{max} robots. We then use multiple teams to perform the coverage operation in the environment.

These multiple teams of robots need to be coordinated appropriately, in a distributed manner, to ensure that each team covers the environment efficiently while reducing the overlap of regions previously covered by that and other teams. We have used a potential field-based navigation strategy that also uses the recent coverage history of the teams' leader robots to enable multiple teams navigate themselves and perform coverage of the environment. In this strategy, each leader robot of a team has a virtual potential field of radius χ_r around it. When the leader robots of two teams get within the communication range χ_r of each other the leader robots of the teams exchange their current coverage maps, including the maps received from their respective follower robots, with each other. They then fuse each other's coverage information using the node counting technique described in the previous section. Finally, each team leader selects the region that closest to its team that has been least visited, and adjusts its heading to move towards that region. The pseudo-code of the algorithm for implementing the multi-team coverage technique is shown in Figure 4.

```
function PotentialFieldNavigation
  if (I am the leader)
    if (there is another leader robot within radius χ_r)
      Receive location, heading and coverage map
        from the leader robot of the other team
      Select a new direction to move that has
        the least overlap with the coverage history
        of my team and that obtained from the other team
      Perform team reconfiguration
    else
      Use single-team coverage technique to navigate
        until an obstacle is encountered
```

Figure 4: Algorithm used by a leader robot to disperse from other teams in the multi-team coverage technique.

ANALYSIS

In this section, we investigate analytically whether area coverage using multiple robots organized as a team is more efficient than an area coverage technique that uses the same number of robots that are not configured into teams and perform coverage individually. We refer to this latter scenario as coverage with individually coordinated robots. Using the notation introduced in the section titles "Multi-Robot Distributed Area Coverage", we consider a square environment where D is the length of a side of a square, O is the area within the environment occupied by obstacles and $d << D$ is the width of the coverage tool of a robot. We let D_{free} denote the area of the free space in the environment that needs to be covered by the robots, i.e., $D_{free} = D^2 - O$, As mentioned before, the values of D, O and D_{free} are unknown to each robot. To simplify our analysis, we consider that covering the surface of an environment with a coverage tool of width d is analogous to painting stripes in a two dimensional space with a "brush" of width d. The actual length of each such stripe depends on the number of obstacles and the number of robots in the environment. For our analysis, we let $\bar{l}$ denote the average length of a stripe.

Proposition 1. Coverage using a single robot. *With a single robot performing the coverage of the environment, there is no guarantee of the robot covering previous uncovered terrain after it has made* $\frac{4\bar{l}}{d} - 1$ *stripes.*

Proof. We consider that the single robot travels in a straight line until it encounters a wall or an obstacle. It then turns away from the wall at an angle determined by the Braitenberg controller from the sensor data of its proximity sensors. Using this technique, when the robot starts the i-th stripe, it has already encountered $(i-1)$ walls or obstacles. This means that there are $(i-1)$ points along the boundaries of the environment or on the obstacles within the environment that have been encountered by the robot. The i-th stripe partitions this set of $(i-1)$ points into two disjoint subsets, one subset lying to the left (or counter-clockwise) from the endpoint of the i-th stripe, and the other subset lying to the right (or clockwise) from the endpoint of the i-th stripe. We

denote these two subsets of points on either side of the i-th stripe as CCW_i and CW_i respectively. Let $|\ CW_i\ | = p_k$, and, consequently, $|\ CCW_i\ | = (i-1)-p_k$. Now, if the $(i+1)$-th stripe is made to the left of the i-th stripe, then the $(i+1)$-th stripe will intersect the points in CCW_i. The expected number of intersects the $(i+1)$-th stripe will have is given by: $E(intersects^{i+1}) = |\ p_k\ | \times 2 - E$(stripes in CCW_i). Now, because the angle at which a robot turns is distributed uniformly over $[0, \pi]$, we can assume that the robot has an equal probability of 0.5 of making the $(i+1)$-th stripe to the left or to the right of the i-th stripe. This gives us $|\ p_k\ | = [i/2]$. Also, the uniform distribution of the turning angle implies that the average number of stripes made by the robot in each of the sets CW_i and CCW_i are equal. Therefore, we can write E(stripes in CCW_i) $= [i/2]$. Therefore, $E(intersects^{i+1}) = [i/2]\times 2 - [i/2] = [i/2]$. Then, the expected number of intersects between the i-th stripe and previous stripes is given by $[1/2] + [2/2]+[3/2]+....[i/2] = [1/2] \times[(i(i+1))/2] = [(i(i+1))/4]$ Because each stripe is of width d, every time two stripes intersect there is an overlap of d^2 square units. Correspondingly, the area overlap between the i-th stripe and previous stripes is given by: $d^2 \times[(i(i+1))/4]$. In general, if the average stripe length is $\bar{l}$, then the area of the new region covered till the i-th stripe by the single robot is given by:

$$R^{i}_{new,SR} = (i\bar{\times}l\times d) - (d^2 \times \frac{i(i+1)}{4}) \quad (1)$$

In Equation 1, the first term on the r.h.s. indicates the area of the region covered until the i-th stripe while the second term indicates the area of the region over which there was repeated coverage until the i-th stripe. The value of i after which the second term exceeds the first indicates the number of stripes after which a single robot performs more repeated coverage than covering new region. To find the duration in number of stripes (denoted by $\hat{\imath}_{SR}$) when this happens, we differentiate the expression in Equation 1 w. r. t. i and set the differential equal to zero. This gives us $\hat{\imath}_{SR} = \frac{4l}{d} - 1$.

Proposition 2. Multi-robot Non-flocking Coverage. *When multiple memoryless robots are coordinated individually to perform distributed coverage in an unknown environment, increasing the number of robots by a factor R results in a speedup that is less than R.*

Proof. Consider R robots, each with a coverage tool of width d. As before, let $\bar{l}$ denote the average length of a stripe if there was a single robot in the environment. The robots use the navigation strategy described in the previous section to cover the environment. For this multi-robot scenario, let i_{enc} denote the frequency with which any two robots encounter each other and let $\bar{l}_{frac}$ denote the average length of the incomplete stripe for each robot at that point. The proof follows in a manner similar to the proof of Proposition 1. In the multi-robot case, when a robot does not encounter another robot it makes a stripe of average length $\bar{l}$. Since the frequency of encountering another robot is $[1/(i_{enc})]$, therefore, out of i stripes, there are $1 - [1/(i_{enc})]$ stripes of length $\bar{l}$. The new area covered by these stripes can be obtained by substituting i with $i \times(1 - [1/(i_{enc})])$ in Equation 1. For, the remaining $[i/(i_{enc})]$ stripes, a robot encounters another robot after doing an incomplete stripe of average length l_{frac} and moves away from the robot, thereby starting a new stripe. Combining these $i \times(1 - [1/(i_{enc})])$ complete and $[i/(i_{enc})]$ incomplete stripes, we can get $R_{new,MR}{}^{i}$, the amount of new area covered till the i-th stripe in the multi-robot case as:

$$R^{i}_{new,MR} = \left(i - \left\lfloor \frac{i}{i_{enc}} \right\rfloor\right)\bar{\times}l\times d\times R$$

$$
\begin{aligned}
&-\frac{(i-\lfloor\frac{i}{i_{enc}}\rfloor)(i-\lfloor\frac{i}{i_{enc}}\rfloor+1)}{4}\times d^2R \\
&+\lfloor\frac{i}{i_{enc}}\rfloor\bar{l}_{frac}\times d\times R \\
&-\frac{(\lfloor\frac{i}{i_{enc}}\rfloor)(\lfloor\frac{i}{i_{enc}}\rfloor+1)}{4}\times i_{enc}\times d^2R
\end{aligned} \tag{2}
$$

The stripe $\hat{i}_{MR}$ in the multi-robot case after which a robot covers more previously covered region than new region can be obtained by differentiating the expression in Equation 2 w. r. t. i and setting the differential equal to zero. This gives:

$$
\hat{i}_{MR}=\frac{4(1-\frac{1}{i_{enc}})\bar{l}+4\frac{\bar{l}_{frac}}{i_{enc}}-(2-\frac{1}{i_{enc}})d}{2d\left((1-\frac{1}{i_{enc}})^2+\frac{1}{i_{enc}}\right)} \tag{3}
$$

Comparing the values of the time measured in number of stripes after which a robot covers more previously covered region than previously uncovered region for the single and multi-robot cases, we can get an estimate of the speedup between these two settings. The speedup is given by the following expression:

$$
\begin{aligned}
speedup_{MR-SR} &= \frac{\hat{i}_{MR}}{R\times\hat{i}_{SR}} \\
&= \frac{1}{R}\times\left[\frac{(1-\frac{1}{i_{enc}})+\frac{4\bar{l}_{frac}}{i_{enc}}-d}{2\left((1-\frac{1}{i_{enc}})^2+\frac{1}{i_{enc}}\right)}\right]
\end{aligned} \tag{4}
$$

Since, the average stripe length $\bar{l}$ is greater than the d ($l>d$), and $i_{enc}>=1$, the factor within square brackets in the above expression is <1. Consequently, we get sub-linear speedup by increasing the number of robots by a factor of R.

The result of sublinear speedup obtained above can be attributed to the fact that when the number of robots is increased, although they are able to cover more region in less time, the repeated coverage done by the robots over regions previously covered by other robots also increases.

Proposition 3. Multi-robot, Multi-Team Flocking-based Coverage. *When multiple robots are organized to form teams to perform distributed coverage in an unknown environment, the coverage improves by a factor proportional to the size of each team τ.*

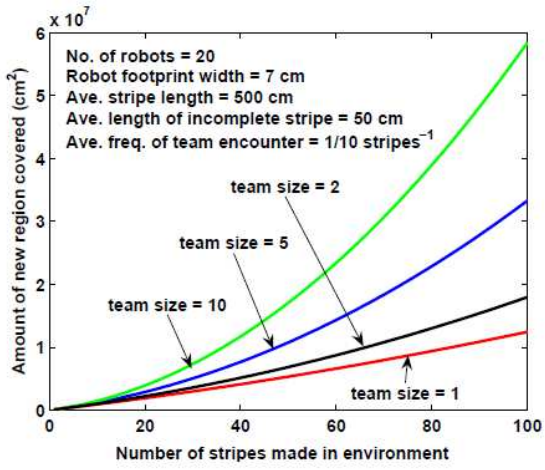
Proof. Let us suppose that R robots are organized to form teams of size τ. This yields $\frac{R}{\tau}$ teams. The footprint of each team is then $d\times\tau$. This setting is similar to the multi-robot case analyzed in proposition 2, with the following changes - each 'unit' of coverage is not a single

robot but a team of τ robots with a footprint of $d\times\tau$, the number of teams is $\frac{R}{\tau}$ and the teams encounter each other after every $\tau\times i_{enc}$ stripes, where i_{enc} is number of stripes after which two robots encounter each other in the individually coordinated multi-robot case. The values of the new region covered till the i-th stripe, $R^{i}_{new,team}$ and the stripe after which a robot covers more previously covered region than previously uncovered region, $\hat{\imath}_{team}$, can be obtained from the corresponding values in the multi-robot case given in Proposition 2, as shown below:

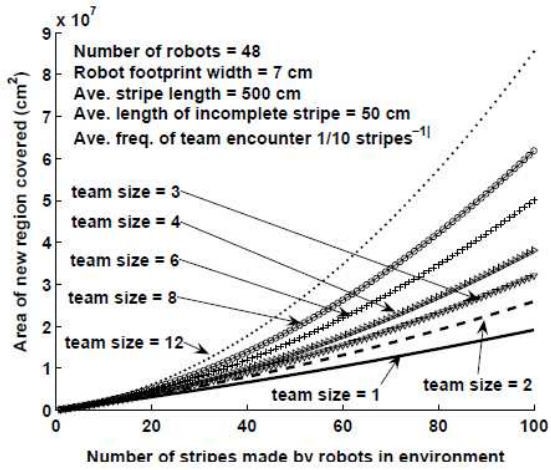
$$R^{i}_{new,team} = (i-\lfloor\frac{i}{\tau\times i_{enc}}\rfloor)\bar{\times}l\times d\times\tau\times\frac{R}{\tau} - \frac{(i-\lfloor\frac{i}{\tau\times i_{enc}}\rfloor)(i-\lfloor\frac{i}{\tau\times i_{enc}}\rfloor+1)}{4}\times(d\tau)^2\times\frac{R}{\tau} + \lfloor\frac{i}{\tau\times i_{enc}}\rfloor\bar{\times}l_{frac}\times d\times\tau\times\frac{R}{\tau} - \frac{(\lfloor\frac{i}{\tau\times i_{enc}}\rfloor)(\lfloor\frac{i}{\tau\times i_{enc}}\rfloor+1)}{4}\times\tau\times i_{enc}(d\times\tau)^2\frac{R}{\tau} \quad (5)$$

and,

$$\hat{\imath}_{team} = \frac{4(1-\frac{1}{\tau\times i_{enc}})l+\frac{4l_{frac}}{\tau\times i_{enc}}-(2-\frac{1}{\tau\times i_{enc}})\times\tau\times d}{\tau\times 2d[(1-\frac{1}{\tau\times i_{enc}})^2+\frac{1}{\tau\times i_{enc}}]} \quad (6)$$



(a)



(b)

Figure 5: Area of new region covered by robots for different team sizes. (a) With 20 robots in the environment. (b) With 48 robots in the environment.

Figures 5(a) and (b) show the improvement in coverage for different team sizes with 20 and 48 robots in the environment for different team sizes. We observe that as the team size increases but the total number of robots in the environment remains fixed, the robots are able to cover more previously uncovered region. The second derivative of $R^{i}_{new,team}$ from Equation 5 is proportional to $2\tau\left(1-\frac{1}{\tau\times i_{enc}}\right)^{2}-\frac{2}{i_{enc}}$.

Although our analyses presented in this section provide insights into the behavior of our system there are several characteristics of the system such as the effect of the frequency with which teams encounter each other on the performance of the system, the effect of team reformation delays due to physical characteristics that were not mathematically modeled such as the localization error and wheel slip noise, the effect of dynamic change in team configurations due to obstacles encountered by the teams, etc.., which are not directly amenable to theoretical analysis. To understand the behavior of our system further, we provide several empirical analyses of the performance of our system under different values of system parameters and different environment and operational constraints in the following section.

EXPERIMENTAL RESULTS

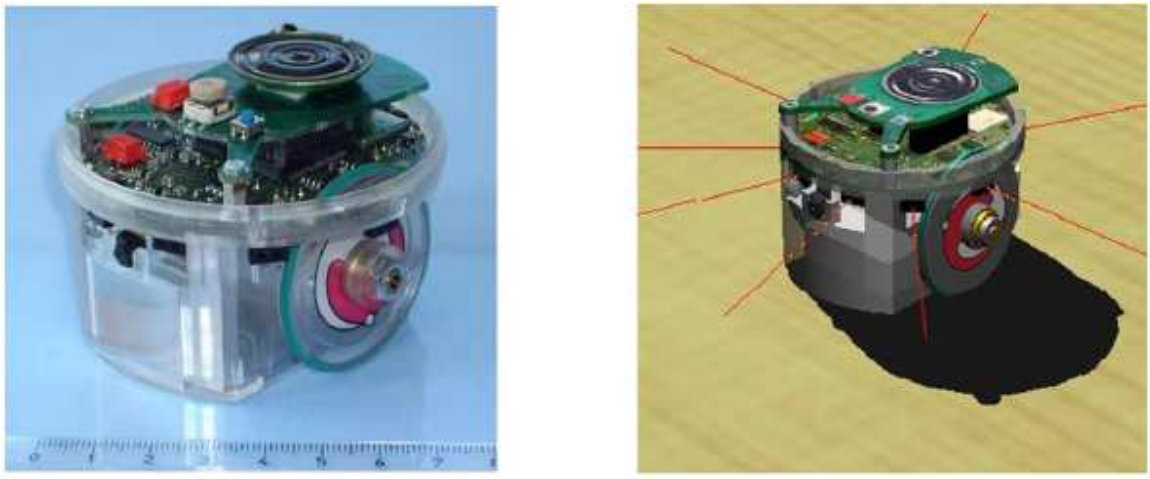


Figure 6: (a) An actual e-puck robot (Photograph courtesy: http://www.e-puck.org) (b) The model of the e-puck robot in the Webots simulator used for our simulations.

We have evaluated our team-based, multi-robot flocking and area coverage techniques through extensive experiments using simulated robots as well as on physical e-puck robots. An e-puck robot has a diameter of 7 cm and a memory capacity of 144 KB including RAM and Flash memory. Each wheel is 4.1 cm in diameter and is capable of a maximum speed of about 12cm/s. We have used the following sensors that are available on the e-puck robot: (1) Eight infra-red distance sensors measuring ambient light and proximity of obstacles in a range of 4 cm, and, (2)

Bluetooth capability for wireless communication. Each e-puck robot is also provided with a local positioning system (a GPS node in the simulator or an overhead camera-based positioning system in the physical experiments) to determine its position in the environment within a 2-D coordinate system. A photograph of the e-puck robot is shown in Figure 6(a). For all our simulations, the inter-robot separation between a pair of follower robots in a team is set to 20 cm. For multi-team coverage, the radius for the potential field-based navigation (χ_r) is set to 1.1 m.

Simulations in Webots

The first objective of our experiments is to understand the behavior of a multi-robot system using the coverage techniques described in this paper, and, to quantify the performance of those techniques while varying the different system and environment related parameters. To achieve this objective we have used extensive multi-robot simulations that allows us to analyze the robots' coverage performance within different experimental settings. We have used the Webots simulation platform (Michel, 2004) for our experiments under this category. Webots is a powerful robotic simulation platform that allows realistic modeling of robots and environments including the parameters of different sensors on robots and the physics of the environment. Each robot in our simulated system is modeled as an e-puck robot with accurate models of the features and characteristics of the physical e-puck robot, as shown in Figure **Error! Reference source not found.**(b).

We have used four metrics to evaluate the performance of our multi-robot team-based area coverage techniques which are given below:

1. The percentage of the area of an environment covered during 2 hours of real time.
2. The percentage of time spent in reformations by a multi-robot team. This metric measures the direct overhead in terms of time of team-based coverage vs. covering the environment individually.
3. The competitive ratio (CR) of the distributed coverage compared to an optimal offline coverage technique. To calculate the competitive ratio(CR), we first calculate the amount of redundancy or repeated coverage of the environment given by:

   $$WR = \sum_i i \times \text{Area of the region visited } i \text{ times.}$$

   The competitive ratio (CR) is given by:

   $$CR = \frac{\text{Free area of the environment}}{WR}.$$

   A higher value of CR (near 1) indicates near-optimal coverage while smaller values of CR (approaching 0) indicate increased repeated coverage of the same region by multiple robots that degrades the performance of the system.
4. The number of obstacles (including walls) encountered by the leader and follower robots in a team. This metric measures the overhead of having larger sized teams because larger (wider) teams encounter obstacles more frequently than smaller ones.

For evaluating the efficacy of team-based coverage, we have compared each of the results obtained using multi-robot teams with an identical scenario where the same number of robots cover the environment individually without forming teams. Each scenario was allowed to run for a duration of 2 hours of real time and results were averaged over 10 runs for each scenario. We have divided our experiments into four categories to verify the performance of area coverage while using different configurations of robot teams, and to understand the effect of different types of environments and noise on the performance of the system.

No. of robots and their configurations	Average team size	% of env. covered		% of time spent in reformations		CR
15 robots		Mean	Std. Dev.	Mean	Std. Dev.	
{3× ⑤}	5	76.59	4.29	70.82	6.95	0.30
{1× ③, 1× ⑤, 1× ⑦}	5	76.13	5.49	66.14	4.85	0.31
{5× ③}	3	82.95	1.97	59.36	2.28	0.29
{15× ①}	1	89.13	0.83	39.65	1.36	0.26
27 robots						
{3× ⑨}	9	76.91	5.70	78.23	3.05	0.27
{1× ③, 2× ⑤, 2× ⑦}	5.4	87.18	2.26	70.82	3.10	0.22
{4× ③, 3× ⑤}	3.85	90.21	1.06	64.90	2.15	0.20
{27× ①}	1	93.60	0.15	43.09	1.27	0.16
48 robots						
{4× ⑤, 4× ⑨}	6	91.95	1.92	71.46	1.72	0.22
{6× ③, 6× ⑤}	4	93.56	0.62	68.83	2.63	0.22
{16× ③}	3	94.45	0.39	63.13	1.61	0.21
{48× ①}	1	94.48	0.12	40.71	1.03	0.19

Table 1: Effect of changing number of robots, number of teams and team sizes with 15, 27 and 48 robots on the different metrics used in our experiments. All results shown are for the office environment shown in Figure 7(c).

Effect of varying number of robots and robot team size

In our first set of experiments we quantify the effect of changing the number of robots, number of teams and the team sizes of the robots covering the environment on the performance of our metrics. Our simulation and physical robot experiments are done with different numbers of robot teams, where the size of each team is either 1 robot (individual), 3 robots, 5 robots, 7 robots or 9 robots[2]. With these sizes for each robot team, we tested our algorithms with three different population sizes of robots within the environment 15, 27 and 48 robots. These population sizes ensure that the robots can be divided into the desired size for each team (3, 5, 7 or 9 robots) while approximately doubling the total number of robots in the environment from one population size to the next. The numbers of different sized teams for each population size are shown in the first column of Table **Error! Reference source not found.**. The circled multiplicand (e.g.,⑤) denotes the number of robots in a team, while the multiplier denotes the number of teams. Using this convention, the notation 3× ⑤ denotes 3 teams with 5 robots in each team. The last configuration within each set denoted by *number of robots* × ① considers robots moving individually without forming teams and provides a comparison along the different metrics between forming and not forming teams while using the same number of robots. The results of varying the number of robots, number of teams teams and the sizes of the teams on the metrics

used for our experiments are shown in Table **Error! Reference source not found.**. We observe that the percentage of environment covered by the robots increases with the number of robots - ranging from an average value of 81.2% with 15 robots to 86.98% with 27 robots and finally to 93.61% with 48 robots. However, the increase in the amount area covered is sublinear in the number of robots because with more robots, robot teams encounter each other more often and spend more time in reformations to avoid colliding with each other. Further analysis of the values in Table **Error! Reference source not found.** shows that with 15 robots in the environment, when the average team size changes from 5 to 3 robots, the coverage improves by $\frac{82.95-76.13}{76.13}\times 100=8.96\%$. With 27 robots, when the average team size drops from 9 to 3.85 robots, the improvement in coverage becomes $\frac{90.21-76.91}{76.91}\times 100=17.29\%$. Finally, the improvement in coverage from a team size of 6 to a team size of 3 robots is $\frac{94.45-91.95}{91.95}\times 100=2.72\%$. These numbers indicate the smaller team sizes are able to achieve better coverage. To further validate this hypothesis, we performed a regression analysis between the average team size and the percentage of environment covered from the data reported in Table **Error! Reference source not found.**. The correlation coefficient for different numbers of robots in the environment are shown in Table 2. We observe that a strong inverse correlation exists between the average size of the robot teams and the percentage of environment covered. Overall, the results of the percentage of environment covered for different robot teams sizes indicate that smaller team sizes are able to achieve better coverage. This is in contrast to the result from Proposition 3 that the coverage performance is proportional to the robot team size. The anomaly in the analytical and experimental results can be explained by the fact that the analytical model does assumes 'instantaneous' reformation when the team leader encounters an obstacle. On the other hand, the experiments include physical characteristics of the setting such as localization errors, wheel slip noise and follower robots encountering obstacles. All of these characteristics force the team to reform and these additional reformation times degrade the coverage performance of larger robot teams.

No. of robots in env.	Correl(team size, % of env. covered)
15	−0.999
27	−0.977
48	−0.914

Table 2: Table showing the correlation coefficient between the average size of robot team and the percentage of environment covered for a different numbers of robots in the environment.

The better performance of smaller robot teams due to rapid reformation times can also be explained by analyzing the 'percentage of time spent in reformations column of Table **Error! Reference source not found.**. We observe that smaller teams with a size of 3 robots spend about 60% of their total time of operation in reformations. This value increases to as much as 78.23%

for a team size of 9 robots. To further validate the correlation between team size and time spent in reformations, we performed a linear regression test between the team size as the independent variable and the percentage of time spent in reformations as the independent variable, from the data reported in Table **Error! Reference source not found.**. The correlation coefficient between the team size and the percentage of time spent in reformations is 0.92, the slope of the linear curve is 5.06 and its y-coefficient is 41.49 - confirming our hypothesis that team size affects the time spent in reformations by a team. The larger reformation times for larger robot teams follows intuitively too because when a larger team encounters an obstacle, more robots have to get into new positions before the team can regain formation and start moving in a new direction. When robots move individually, the reformation times are the lowest because a single robot only has to turn itself to avoid an obstacle without worrying about getting all follower robots into correct positions to regain team formation after avoiding an obstacle.

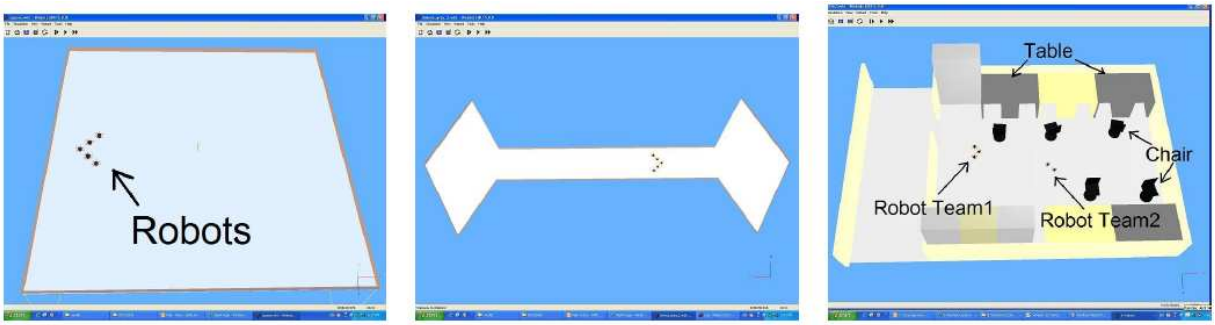


Figure 7: The three different types of environment used in our experiments. (a) A 4×4 m^2 square environment with no obstacles, (b) A corridor environment consisting of two diamond shaped regions, joined by a corridor that is 8 m long and 1 m wide, and, (c) A 4 m ×2 m office environment that is occupied by furniture.

Among the experimental results reported in Table **Error! Reference source not found.**, individually moving robots also appear to achieve better or comparable coverage than robots that move together as a team, irrespective of the team size. The inferior coverage performance of larger teams together with longer reformation times leads us to the question - are larger teams always worse for team-based area coverage? The answer to this question can be inferred from the results in the competitive ratio (CR) column of Table **Error! Reference source not found.**. The competitive ratio expresses the efficiency of the coverage performed by the robots by incorporating the amount of repeated, and hence, unnecessary coverage of previously covered regions done by the robots. The repeated coverage happens in our system because leader robots refresh their coverage histories after H steps. We observe that although robots that move individually are able to cover a marginally higher percentage of the total environment than team-based robots, the competitive ratio of robots that move individually is lower than that of robots moving in teams. This indicates that robots that move individually sacrifice a significant amount of the advantage of their lower reformation times by performing repeated coverage of previously covered regions. The lower competitive ratio for area coverage by the individually moving robots can be attributed to the fact that individual coordination between robots requires each robot to exchange and fuse coverage information from more robots, more frequently. In contrast, with team-based coverage only leader robots aggregate the team's coverage information and

exchange it with each other. This results in more efficient information exchange and judicious decision making by robot teams to avoid repeated coverage of previously covered regions.

Effect of different environments

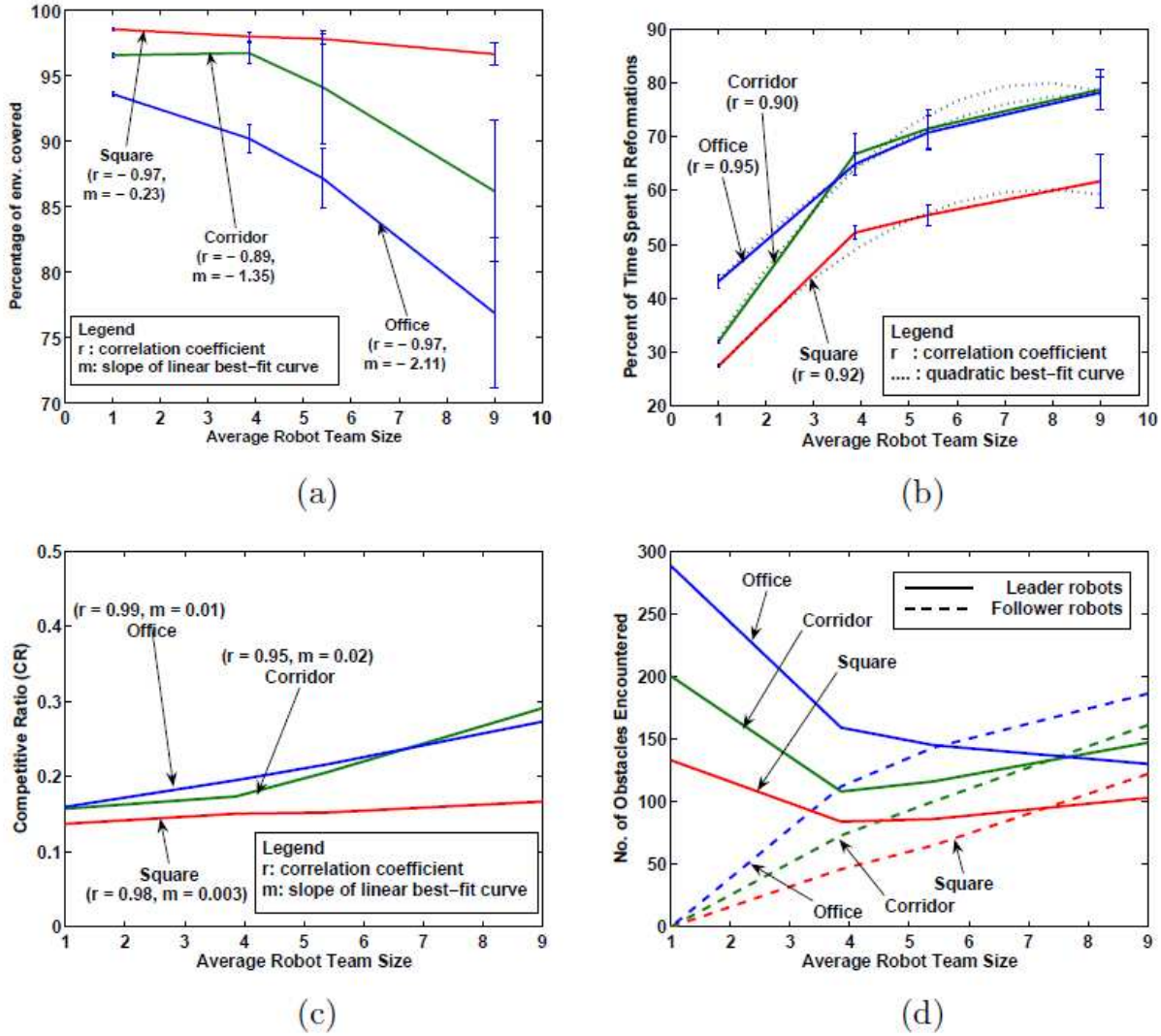


Figure 8: Effect of navigating in three types of environment on the different metrics used for our experiments. The results reported are for different configurations of 27 robots.

For our next set of experiments we vary the environment in which the robots operate and observe the effect on the performance of the system. We consider three different environments with different geometric features and different numbers of obstacles in the environment as shown in Figure 7. The results of this experiment for different team sizes while using 27 robots are shown in Figure 8(a)-(d). We observe that for the square environment with no obstacles, the percentage of the environment covered by the robots shown in Figure 8(a) and the competitive ratio shown in Figure 8(c) are not significantly affected by changing the team size. This is substantiated by the high correlation coefficient of 0.97 and 0.98 respectively, but a very small slope of the linear

best-fit curve, 0.23 and 0.003 respectively, for these cases obtained by a linear regression analysis of the data. These results are in contrast to the analytical results mentioned in Proposition 3 which state that the area of the previously uncovered region increases proportionally with the robot team size. However, the mathematical model of the robots did not account for physical characteristics such as the localization error and wheel slip noise, which cause the performance of the area coverage to get adversely affected in the experimental results. For the more complex environments of the corridor and the office, the robot team size adversely affects the percentage of the area of the environment that gets covered as shown in Figure 8(a). This relationship between the average team size and coverage performance is confirmed by the high negative value of the correlation coefficient −0.89 and −0.97 respectively, coupled with a considerable slope of the linear best-fit curve at −1.35 and −2.11 respectively, as shown in Figure 8(a). The decrease in coverage can be attributed to the longer reformation times of larger teams as shown in Figure 8(b) - larger robot teams spend longer times to reconfigure after encountering an obstacle, and therefore, have lesser time to perform coverage of the environment. The competitive ratio of the robots in the corridor and office environments increases marginally with larger sized robot teams as shown by the positive slope of the linear best-fit curve of 0.02 and 0.01 respectively in Figure 8(c). But this improvement comes at the expense of lower coverage in the environment. Finally, Figure 8(d) shows the number of obstacles encountered by the leader and follower robots in robot teams for different team sizes. We observe in this graph that as the team size gets larger, follower robots evidently encounter more obstacles because of the larger physical width of the team. This further aggravates the reformation time for larger teams as evidenced in Figure 8(b), and reduces the coverage efficiency for large sized teams in environments with a considerable number of obstacles like the corridor and office environments.

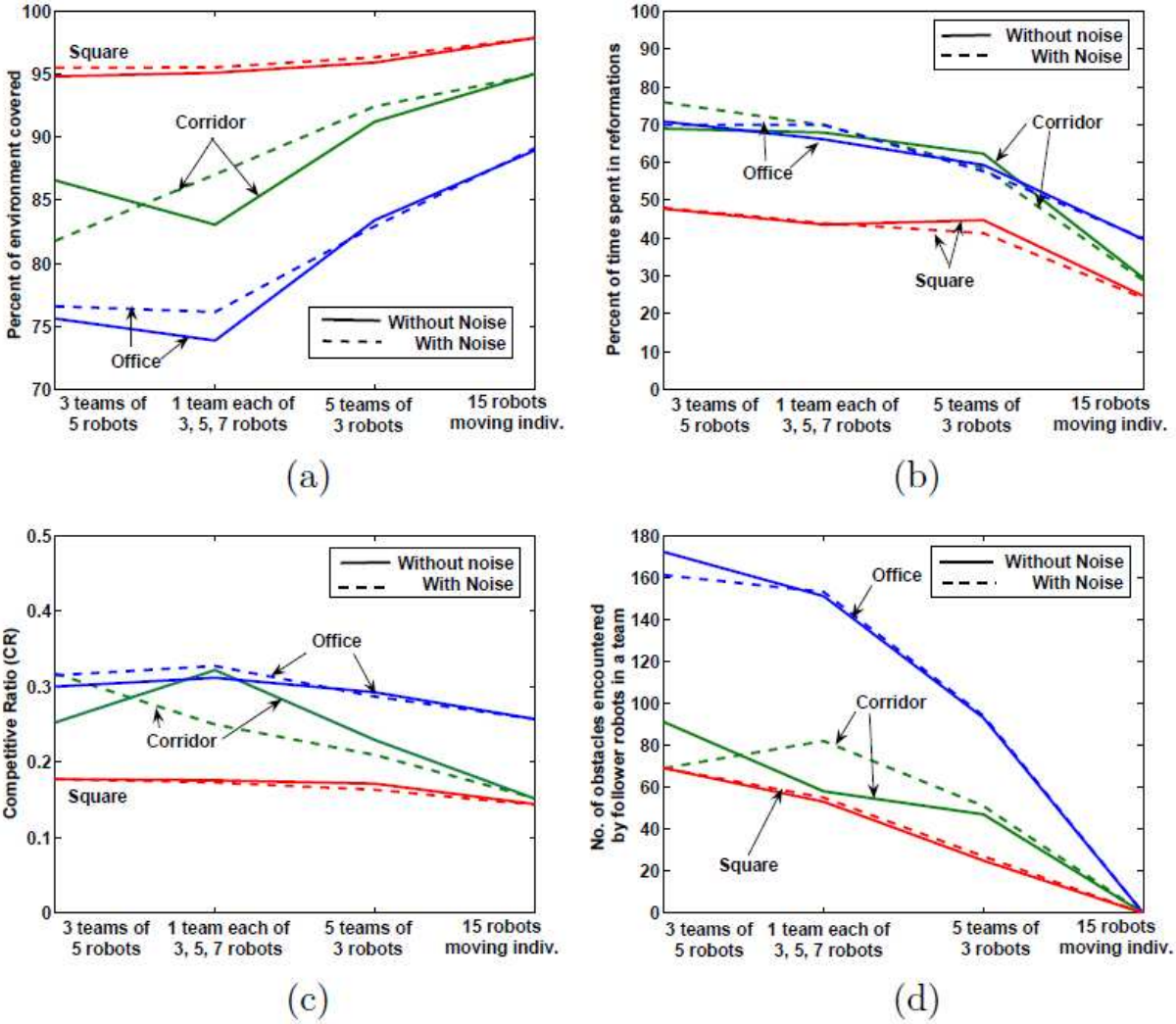


Figure 9: Effect of incorporating wheel slip noise within the three types of environments considered on the different metrics used for our experiments. The results reported are for different configurations of 15 robots.

	The epuck with a red hat		
Parameter	X (cm)	Y (cm)	Θ (Degrees)
Mean Error	0.25	0.088	-3.49
Standard Deviation	0.33	0.15	2.38
	The epuck with a blue hat		
Mean Error	0.68	0.62	3.05
Standard Deviation	0.15	0.148	0.3
	The epuck with a green hat		
Mean Error	-0.07	-0.18	-1.43
Standard Deviation	0.12	0.147	5.72
	The epuck with a pink hat		
Mean Error	-0.02	-0.089	4.85
Standard Deviation	0.19	0.14	1.75
	The epuck with a purple hat		
Mean Error	-0.29	-0.398	5.69
Standard Deviation	0.18	0.137	2.17

Table 3: Localization error due to image processing errors for five experiments.

Effect of noise on the system

In our final set of experiments we quantify the effect of noise on the coverage performance of the robots. We consider two sources of noise: a) wheel slip noise that depends on the friction between the robot's wheels and floor of the environment, and, b) localization noise that is introduced due to the local positioning mechanism used by the robots in our system.

For determining the wheel slip noise, we performed 5 sets of trials by moving a physical e-puck robot from a fixed start location to a target location. Each trial set consisted of 10 individual runs and in each trial set, the robot was moved through a distance of 3.85 m at different angles (0, 30, 45, 60 and 90 degrees) relative to the local coordinate system in the environment. For each trial, we measured the difference in distance between the actual location reached by the robot and its target location. The error due to wheel slip noise, obtained by averaging the results of these trials, was calculated as 0.1339 in the x-axis and 0.1261 in the y-axis of the environment's coordinate system. We averaged these two values to set a wheel slip noise of 0.13 for each simulated e-puck robot inside Webots. The effects of the wheel slip noise of the coverage metrics is shown in Figure 9(a)-(d) with different configurations of 15 robots moving in the three environments shown in Figure 7. We observe that the effect of the wheel slip noise on the different metrics used for our experiments is nominal in the case of the square and the office environments. However, the wheel slip noise results in a more pronounced effect on these metrics when the robots move in the corridor environment. Specifically, for the configurations {3× ⑤ } and {1× ③ , 1× ⑤ , 1× ⑦ } in the corridor environment, we observe that the wheel sleep noise adversely affects the coverage performance, the competitive ratio and the number of obstacles encountered by follower robots. The reason for this behavior can be understood by analyzing the effect of wheel slip noise and the space of the corridor. The wheel slip noise we observed on the physical robots causes each wheel to turn at a different speed than was set by the wheel encoders, due to friction with the floor's surface. This causes the robots to drift intermittently from their planned paths instead of moving in a straight line. In the corridor, that is

1 m wide, the wheel slip noise causes the robots, especially the follower robots to drift from their planned path through the corridor and encounter the walls of the narrow corridor. This results in the higher number of obstacles encountered by the follower robots with the {1× ③ , 1× ⑤ , 1× ⑦ } configuration in the corridor environment as shown in Figure 9(d).

We further performed 2-way and 1-way ANOVA (analysis of variance) tests at 95% confidence interval to validate our conclusions for each of the data sets shown in Figures 9 (a) - (d). The results show that, when considered together with the team size variable (in column 2 of Table 4), slip noise does not have a significant effect on the first three metrics - percentage of environment covered by the robots, percentage of time spent in reformations, and competitive ratio. However, for the last metric - number of obstacles encountered by follower robots, the wheel slip noise does make a significant impact. The marginal impact of the wheel slip noise is in accordance with the results shown in Figures 9(a)-(c), where the graphs for the metrics with and without slip noise are almost identical. To further analyze our results we also performed 1-way ANOVA tests with the variables team size and wheel slip noise respectively. As noted previously in this section, we once again observe from the significance results that the team size variable affects the performance of the first three metrics. The wheel slip noise is a significant variable only for the percentage of environment covered and the number of obstacles encountered by follower robots metrics. These results can be explained by the fact that when the follower robots are closer to walls of obstacles, especially for the narrow channel connecting the two rooms in the corridor environment, the wheel slip noise causes the robots to deviate from their desired locations within the team and encounter the walls more often. Without wheel slip noise added to their motion, the robots deviate from their positions less frequently, and, therefore, they encounter obstacles less frequently. More obstacles should result in more reformation time. However, the percentage of reformation times shown in Figure 9(b) is not affected significantly by the wheel slip noise. An increased number of obstacles encountered, but unchanged number of reformations indicate that robot teams are able to avoid obstacles encountered by followers without reforming. In other words, teams are able to continue coverage despite follower robots encountering obstacles. This behavior in turn translates to a slight improvement in the amount of coverage achieved by the robots, while considering wheel slip noise, as shown in Figure 9(a).

Metric	sig. value with 2-way ANOVA (team size * slip noise)	sig. value with 1-way ANOVA (team size)	sig. value with 1-way ANOVA (slip noise)
% of env. covered	0.974 (not sig.)	0 (sig.)	0.05 (sig.)
% of time in reform	0.992 (not sig.)	0 (sig.)	0.974 (not sig.)
CR	0.956 (not sig.)	0 (sig.)	0.969 (not sig.)
No. of obst. (followers)	0.05 (sig.)	0.679 (not sig.)	0.05 (sig.)

Table 4: Analysis of variance of the results with slip noise for different metrics used in our experiments shown in Figures 9(a)-(d). The significance level α is set to α=0.05 (95% confidence intervals). The abbreviation "sig." stands for significant.

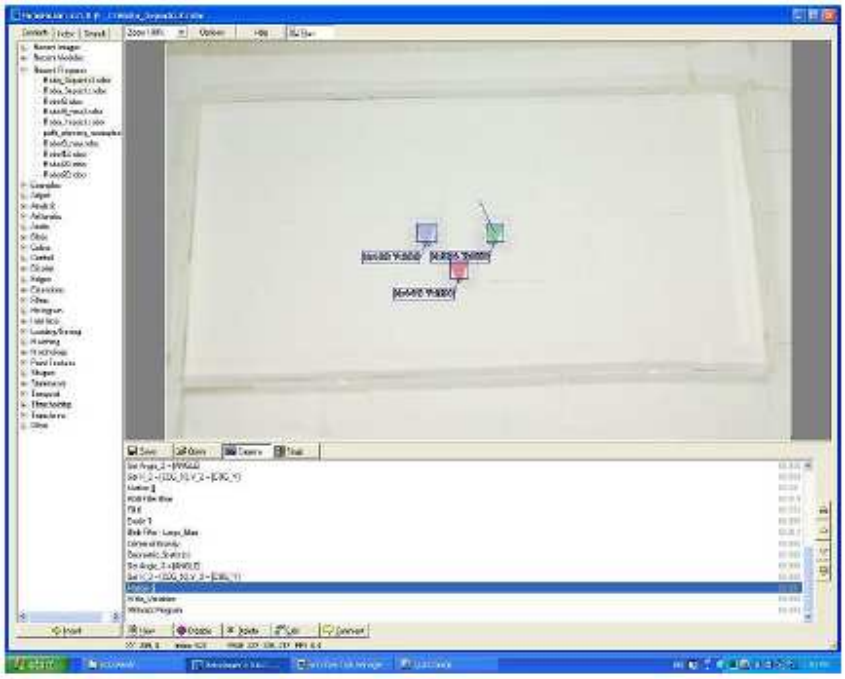
Environment	% of env. covered		% of time spent in reformations		CR
	Mean	Std. Dev.	Mean	Std. Dev.	
	1 team of 5 robots WITHOUT localization noise				
Square	75.13	3.90	26.56	4.56	0.43
Corridor	70.57	5.67	45.23	4.66	0.46
Office	45.42	7.35	62.09	7.77	0.79
	1 team of 5 robots WITH localization noise				
Square	78.56	3.64	38.33	4.78	0.44
Corridor	59.81	10.52	60.42	10.31	0.60
Office	43.44	8.82	67.19	9.86	0.82

Table 5: Effect of localization noise on the different metrics used for our experiments. The results are reported for navigating a team of 5 robots in the square environment.

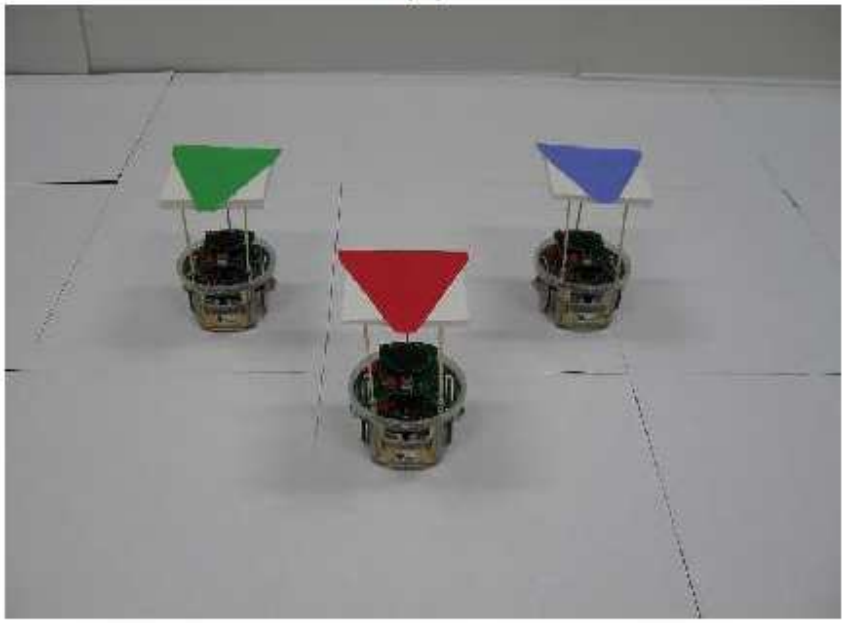
Localization noise in our system is caused mainly by the image processing algorithm that processes the video stream of the robots' movement captured by the overhead camera overlooking the experiment arena and calculates each robot's coordinates in the local coordinate system. For calculating the localization noise, we used five physical e-puck robots and tracked their paths over a distance of 3.85 m for 5 trials. Each robot was fitted with uniquely colored marker on its top that showed its heading relative to the local coordinate system. The error in the location and heading of the different robots is shown in Table 3. We added these noise values to the "accurate" GPS readings provided by Webots (using the GPSNode class) to simulate the effect of localization noise in our simulations. The results are reported in Table 5 for a team of 5 robots moving in the square environment with and without localization noise added to its coordinates. We observe that localization noise marginally affects the coverage and competitive ratio metrics. To validate this observation, we performed the Kruksal-Wallis analysis of variance test to determine if localization noise had a significant effect on the metrics used in our experiments. The results of the tests are reported in Table 6. For each of the metrics, the p-value of the data set is found to be greater than the confidence level α=0.05, indicating that the medians of the distributions with and without localization noise are identical at 95% confidence interval. This implies that the localization noise only has a marginal effect on percentage of the environment covered, the time spent in reformations and the competitive ratio.

Metric	p-value
% of env. covered	0.873
% of time in reform	0.513
CR	0.513

Table 6: Different metrics and corresponding p-values from Kruksal-Wallis Test on the data with and without localization noise reported in Table 5



(a)



(b)

Figure 10: (a) Screen shot of three e-pucks within our experiment arena in the image processing software's user console. (b) A photo of three e-pucks in a V-shape formation within the experiment arena.

	Percentage of coverage (%)			
	15 min.	30 min.	45 min.	60 min.
Ind. no obs.	20.1	32.5	48.3	53.4
Team, no obs.	13.4	18.6	29.6	42.9
Ind. 10 % obs.	20.39	33.17	49.6	55.7
Team, 10 % obs.	16.44	21.78	30.67	45.33
	No. of times obstacle encountered			
Ind., no obs.	16	44	69	99
Team, no obs.	2	5	7	8
Ind., 10 % obs.	21	46	78	107
Team, 10 % obs.	2	6	7	10
	Avg. reformation time			
Team, no obs.	34.7	39.15	39.33	38.08
Team, 10 % obs.	33.94	37.13	36.57	37.55

Table 7: Experimental results showing the different metrics for coverage with 3 e-puck robots.

Experiments on E-puck Robots

To test the performance of our team-based coverage algorithm on physical e-puck robots, we performed coverage within a 2.31×2.31 m^2 environment using 3 e-puck robots that moved together as a team. We compared the metrics obtained from this scenario with a scenario where the robots are coordinated to move individually. The environment either had no obstacles, or had 10% of its area covered by obstacles. Each scenario was run for 60 minutes and five runs were conducted for each scenario. A snapshot of the 3 e-puck robots within the image processing software's user interface and a photograph of the robots within our experiment arena is shown in Figure 10. A video of a simulation run from the e-puck robot experiments is available at http://www.youtube.com/watch?v=jmyhURYq5Uc. We observed that in the individually coordinated case robots obtain 7% to 20% more coverage than the team-based coverage at the different running times. Due to the localization error and wheel slip noise, the robot team needs to reform its shape approximately every 2 minutes, and spends about 35-40 seconds to get the team reconfigured. This results in about 2/3 of its total runtime being spent by the robot team to perform coverage while the remaining 1/3 is spent in reconfigurations. However, we observed that when robots are individually coordinated, they encounter 10 times more obstacles than the team-based coverage. The team based coverage performs especially better than the individually coordinated strategy when the robots are at a corner. The individual robot gets "stuck" for some time oscillating between the two walls of the corner, while the robot team, because of its wider coverage "swathe" can navigate out of the corner more efficiently.

CONCLUSION AND FUTURE WORK

In this paper, we have described a technique for multi-team flocking for distributed area coverage. We have shown that with many small sized teams, team-based coverage performs comparably with coverage when the robots are individually coordinated. Our techniques hold promise in scenarios where a suite of robots have to maneuver themselves as a cohesive team to provide an array of sensors located on the different robots of the team or to provide redundancy in the sensor measurements. We are currently enhancing the team-based coverage techniques to record and compress local coverage maps within the memory of each robot so that they can

reduce redundant coverage. We are also investigating techniques to improve the basic formation control mechanisms described in this paper so that robot teams can dynamically adapt their formation as well as move across teams depending on their performance.

ACKNOWLEDGEMENTS

The research reported in this paper has been supported as part of the COMRADES project, sponsored by the U.S. Office of Naval Research, grant no. N000140911174.

We are grateful to Prof. Lotfollah Najjar of the Information Systems and Quantitative Analysis Department, University of Nebraska, Omaha for his help with statistical analysis of our test results.

A preliminary version of this paper appeared as a conference paper: Cheng, K., Wang, Y., & Dasgupta, P. (2009) Distributed Area Coverage using Robot Flocks. World Congress on Nature and Biology Inspired Computing (NABIC): (pp. 678-683).

REFERENCES

Anderson D., Carter, S., Hazzard, M., Josey, T., Pearson, J., Scoggins, J., & Soltmann, L. (2008) Journal paper for the 2008 AUVSI student UAV competition. Resource document Association for Unmanned Vehicle Systems International. Retrieved on March 15, 2010 from http://www.navair.navy.mil/pma263/seafarers/ journal/journal2008/AUVSI08-NCSU.pdf

Bahceci, E., Soysal, O., & Sahin, E. (2003). Review: Pattern formation and adaptation in multi-robot systems: CMU Tech. Report no. CMU-RI-TR-03-43, Carnegie Mellon University, USA.

Balch, T. & Arkin, R. (1998). Behavior-based formation control of multi-robot teams. IEEE Transactions on Robotics and Automation, 14(6), 926-939.

Bloch, I., Milisavljevc N., & Acheroy, M. (2007) Multisensor Data Fusion for Spaceborne and Airborne Reduction of Mine Suspected Areas. International Journal of Advanced Robotic Systems, 4(2), 173-186.

Burgard, W., Moors, M., Fox, D., Simmons, R. & Thrun, S. (2005) Collaborative multi-robot exploration. IEEE Transactions on Robotics, 21,(3), 376- 386.

Cassinis, R. (2000). Multiple single sensor robots rather than a single multi-sensor platforms: a reasonable alternative. Paper presented at International Conference on Explosives and Drug Detection Techniques.

Chen, Q. & Luh, J. (1994). Coordination and control of a group of small mobile robots. International Conference on Robotics and Automation. (pp. 2315-2320).

Cheng, K. & Dasgupta, P. (2007). Dynamic Area Coverage using Faulty Multi-agent Swarms. IEEE/WIC/ACM International Conference on Intelligent Agent Technology. (pp. 17-24).

Choset, H. (2001). Coverage for robotics: A survey of recent results," Annals of Math and AI, 31, 113-126.

Chien, S., Sherwood, R., Tran, D., Cichy, B., Rabideau, G., Castano, R., Davies, A., et al. (2005). Using Autonomy Flight Software to Improve Science Return on Earth Observing One. Journal Of Aerospace Computing, Information, And Communication, 2, 196-216.

Dudenhoefer, D. & Jones, M. (2000). A formation behavior for large-scale micro-robot force. 32nd Winter Simulation Conference. (pp. 972-982).

Fredslund, J. & Mataric, M. (2002). A general algorithm for robot formations using local sensing and minimal communication. IEEE Transactions on Robotics and Automation, 18(5), 837-846.

Gabriely, Y. & Rimon, E. (2001). Spanning-tree based coverage of continuous areas by a mobile robot. Annals of Math and AI, 31(1-4), 77-98.

Gokce, F. & Sahin, E., (2009). To flock or not to flock: the pros and cons of flocking in long-range migration of mobile robot swarms. 8th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS): (pp. 65-72).

Kaminka, G., Schechter, R., & Sadov, V. (2008). Using Sensor Morphology for Multirobot Formations. IEEE Transactions on Robotics, 24(2), 271-282.

Koenig, S., Szymanski B., & Liu, Y. (2001). Efficient and Inefficient Ant Coverage Methods. Annals of Mathematics and Artificial Intelligence, 31(1-4), 41-76.

Mastellone, S., Stipanovic, D., Graunke, C., Intlekofer, K., & Spong, M. (2008). Formation Control and Collision Avoidance for Multi-agent Non-holonomic Systems: Theory and Experiments. International Journal of Robotics Research, 27(1), 107-126.

Michel O. (2004). Webots: Professional Mobile Robot Simulation. International Journal of Advanced Robotic Systems, 1(1), 39-42.

De Mot, J. (2005). Optimal Agent Cooperation with Local Information. Ph.D. Thesis, Massachusetts Institute of Technology.

Olfati Saber, R. (2006). Flocking for Multi-Agent Dynamic Systems: Algorithms and Theory. IEEE Transactions on Automatic Control, 51(3), 401-420.

Rekleitis, I., New, A., Rankin, E. & Choset, H. (2008). Efficient Boustrophedon Multi-Robot Coverage: an algorithmic approach. Annals of Mathematics and Artificial Intelligence, 52(2-4), 109-142.

Reynolds, C. (1987). Flocks, herds and schools: A distributed behavioral model. Computer Graphics, 21(4), 25-34.

Rutishauser, S., Correll, N., & Martinoli, A. (2009). Collaborative Coverage using a Swarm of Networked Miniature Robots. Robotics and Autonomous Systems, 57(5), 517-525.

Sahin T., & Zergeroglu, E. (2008). Mobile Dynamically Reformable Formations for Efficient Flocking Behavior in Complex Environments. IEEE International Conference on Robotics and Automation. (pp. 1910-1915).

Smith, B., Egerstedt, M., & Howard, A. (2009). Automatic Generation of Persistent Formations for Multi-Agent Networks Under Range Constraints. Mobile Networks and Applications Journal, 14, 322-335.

Spears, D., Kerr, W., & Spears, W. (2006). Physics-based Robot Swarms for Coverage Problems. International Journal on Intelligent Control and Systems, 11(3), 124-140.

Stachniss, C., Mozos O., & Burgard W. (2008). Efficient exploration of unknown indoor environments using a team of mobile robots. Annals of Math and AI, 52(2-4), 205-227.

Tache F., Fischer, W., Caprari, G., Siegwart, R., Moser, R., & Mondada, F. (2009). Magnebike: A magnetic wheeled robot with high mobility for inspecting complex-shaped structures. Journal of Field Robotics, 26(5), 431-452.

Turgut, A., Celikkanat, H., Gokce F., & Sahin E. (2008). Self-organized flocking with a mobile robot swarm. International Conference on Autonomous Agents and MultiAgent Systems. (pp. 39-46).

V. Tzanov, "Distributed Area Search with a Team of Robots," Master's Thesis, Massachusetts Institute of Technology, 2006.

Wang, P. (1989). Navigation Strategies for Multiple Autonomous Mobile Robots. IEEE/RSJ International Workshop on Intelligent Robots & Systems, (pp. 486-493).

Wurm, K., Stachniss. C., and Burgard, W. (2008). Coordinated multi-robot exploration using a segmentation of the environment. IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS), (pp. 1160-1165).

Wagner, I., Altshuler, Y., Yanovski, V. & Bruckstein, A. (2008). Cooperative Cleaners: A Study in Ant Robotics. International Jour[3]nal of Robotics Research, 27, 127-151.

[1] Wheel slip noise is the error in the wheel's encoder readings caused by the slippage of the wheels on the floor.

[2] Controlling teams larger than 9 robots is difficult with the current hardware capabilities including Bluetooth communication available on the robots.