Proceedings of the 2012 IEEE
International Conference on Robotics and Biomimetics
December 11-14, 2012, Guangzhou, China

Multi-Robot Repeated Boundary Coverage Under Uncertainty

Pooyan Fazli and Alan K. Mackworth

***Abstract*—We address the problem of repeated coverage by a team of robots of the boundaries of a target area and the structures inside it. Events may occur on any parts of the boundaries and may have different importance weights. In addition, the boundaries of the area and the structures are heterogeneous, so that events may appear with varying probabilities on different parts of the boundary, and this probability may change over time. The goal is to maximize the reward by detecting the maximum number of events, weighted by their importance, in minimum time. The reward a robot receives for detecting an event depends on how early the event is detected. To this end, each robot autonomously and continuously *learns* the pattern of event occurrence on the boundaries over time, capturing the uncertainties in the target area. Based on the policy being learned to maximize the reward, each robot then *plans* in a decentralized manner to select the best path at that time in the target area to visit the most promising parts of the boundary. The performance of the learning algorithm is compared with a heuristic algorithm for the Travelling Salesman Problem, on the basis of the total reward collected by the team during a finite repeated boundary coverage mission.**

I. Introduction

Multi-Robot Boundary Coverage is a challenging problem with various applications including surveillance and monitoring, cleaning, intrusion detection and facility inspection. In this task, a team of robots cooperatively visits (observes or sweeps) the boundaries of the target area and the structures inside it. The goal is to build efficient paths for all the robots which jointly ensure that each point on the boundaries is visited by at least one of the robots. The *Boundary Coverage* is a variant of the *Area Coverage* [3], [8] problem, in that the aim is to cover just the boundaries, not the entire area.

There are two classes of *Boundary Coverage* problems:

- ***Single Coverage:*** The aim is to cover the boundary until all its accessible points of interest have been visited at least once, while minimizing the time, sum/maximum length of the paths/tours generated for the robots, or balancing the workload distribution among the robots [4], [14].
- ***Repeated Coverage:*** The goal is to cover all the accessible points of interest on the boundary repeatedly over time, while maximizing the frequency of visiting points, minimizing the weighted average event detection time, or detecting the maximum number of events/intruders. Visiting the points on the boundary can be accomplished with uniform or non-uniform frequency, depending on the priorities of different parts of the boundary [1], [2].

II. Problem Definition and Preliminaries

In this paper, we address the *Multi-Robot Repeated Boundary Coverage* problem with the following specifications:

- The environment is a simple polygon consisting of rectilinear or non-rectilinear polygonal structures.
- The 2D map of the environment is given *a priori*.
- An arbitrary number of robots is involved in the coverage mission.
- The robots are equipped with a panoramic visual sensor with limited range.
- The robots have limited communication range.
- The events may occur on any part of the boundary.
- The events may have different types. Each event type has its own importance weight.
- The boundary is heterogeneous, in that events of one type may occur with varying probabilities on different parts of the boundary, and this probability may change over time.
- A robot can detect an event if the event is within the visual range of the robot.
- Once a robot detects an event, the event is discarded from the boundary.
- A robot is aware of the types of the events and their importance weights once it detects the events.
- The robots are not *a priori* aware of the probability distribution of the events occurrence on the boundaries.
- The reward a robot receives for detecting an event depends on how soon the event is detected. At each time step after the event occurrence, the detection reward of the event is decreased by a multiplicative discount factor.

Definition 1. *Event Type:* m types of events may occur on the boundary. The set of all events types is $E = \{E_1, E_2, ..., E_m\}$. Similarly, an event of type E_i is denoted as e_i.

Definition 2. *Event Importance:* The importance degree of an event of type E is given by $weight(E)$. It is assumed that $weight(E) \in (0, 1]$ such that 1 is the highest degree of importance. The importance can also be referred to as the *priority*, in that an event of higher importance should have higher priority of being detected.

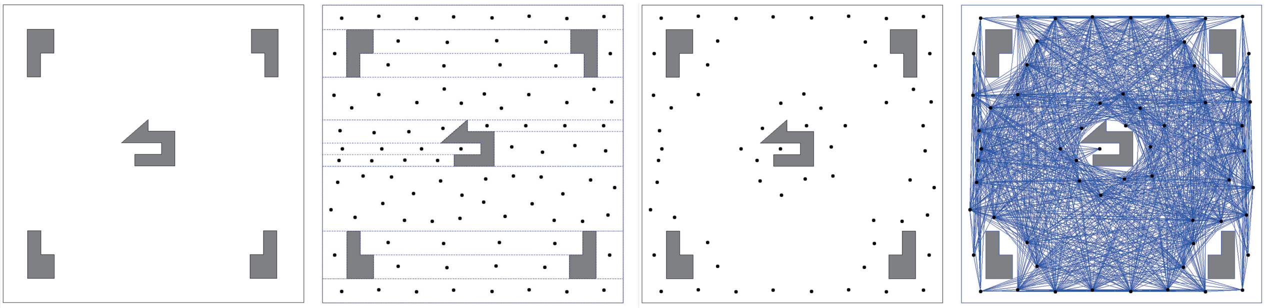
To address the problem, two classes of algorithms are proposed: (1) *Uninformed Boundary Coverage* and (2) *Informed Boundary Coverage*.

Uninformed Boundary Coverage uses a heuristic algorithm for the *Travelling Salesman Problem* to patrol the boundaries. On the other hand, *Informed Boundary Coverage* is primarily based on an algorithm in which each robot autonomously and continuously *learns* the pattern of event occurrence on the boundaries over time, capturing the uncertainties in the target area. Based on the policy being learned to maximize the reward, each robot then *plans* in a distributed manner to select the best possible path at the time in the target area to visit the most promising parts of the boundary. *Informed Boundary Coverage* is an online learning algorithm.

The performance of the proposed approaches will be evaluated on the basis of the total reward received by the team during a finite repeated boundary coverage mission.

Pooyan Fazli and Alan K. Mackworth are with the Department of Computer Science, University of British Columbia, Vancouver, BC, Canada {pooyanf,mack}@cs.ubc.ca

978-1-4673-2127-3/12/$31.00 © 2012 IEEE



(a) *Original Map* (b) *Trapezoidation + Area Guards* (c) *Boundary Guards* (d) *Boundary Graph*

Fig. 1: *Sequential Stages of Building the Boundary Graph*

III. Background and Review

Elmaliach *et al.* [7] addressed the problem of frequency-based patrolling (*i.e.* maximizing the minimum, maximum, or uniform point-visit frequency) of open polylines (*e.g.* as in open-ended fences), where the two endpoints of the polylines are not connected. They also investigated the velocity uncertainties and accumulating motion errors of the robots during the mission. Jensen *et al.* [9] extended Elmaliach *et al.*'s work on patrolling open polylines, with a focus on maintaining the patrol over the long-term. They accomplished this task by replacing the robots having power level below a threshold with some reserve robots. Marino *et al.* [12] proposed a decentralized multi-robot approach to patrol both open and closed polylines.

The patrolling problem, as studied in the above papers, is investigated in terms of optimizing point-visit frequency. Some other work considered the existence of adversary agents in the workspace. For instance, Agmon *et al.* [1], [2] studied patrolling a cyclic border, in which the robots' goal was to maximize their rewards by detecting an adversary agent, attempting to penetrate through a point on the boundary, unknown to the robots. The intruder needed some time interval of length t to accomplish the intrusion. Jurek Czyzowicz *et al.* [6] addressed the same problem using a team of variable-speed robots.

As far as the authors are aware, there is no work using the framework studied in this paper. In our work, instead of patrolling a single open or closed polyline, the robots patrol the boundaries of a full environment and the structures inside it, and instead of optimizing frequency criteria, it is assumed that different parts of the boundary may have different priorities depending on the probability distribution of events occurrence on the boundaries. Finally, our robots aim to detect multiple events/intruders simultaneously, as opposed to single intruder scenarios studied in the previous work.

IV. Environment Modeling

Uninformed Boundary Coverage and *Informed Boundary Coverage* both require that a roadmap is built within the target area, capturing the connectivity of the free space close to the boundaries. To this end, a graph-based representation called the *Boundary Graph* is constructed on the target area. The *Boundary Graph* enables the robots to move throughout the environment to monitor the boundaries of the area and the structures inside. Since the environment is known to the robots, each robot can independently build the *Boundary Graph* in the target area. In order to construct the *Boundary Graph*, a sufficient number of control points, called the *boundary guards*, are placed within the environment, considering the limited visual range of the robots.

A. *Locating Guards with Limited Visual Range*

In our problem definition, we presume the robots are equipped with panoramic cameras with a 360° field of view. However, the cameras' visual range is limited. The proposed approach initially locates a set of *area guards* required to visually cover an entire area. The term *guard* is taken from the *Art Gallery Problem* [13]. These static area guards are control points that can jointly cover the whole environment while satisfying the limited visual range constraint of the robots. In other words, if we had as many robots as the number of guards, and each robot was stationed on a guard, the entire area would be covered visually by the robots.

To locate the guards, the algorithm decomposes the initial target area (Figure 1a), a 2D simple polygon with static structures, into a collection of convex polygons using a *Trapezoidal Decomposition* method, and then applies a post-processing approach to eliminate as many trapezoids as possible [15] (Figure 1b). The post-processing step is more effective in cluttered areas, and since the number of guards located by the algorithm is directly related to the number of trapezoids, fewer trapezoids will result in fewer guards.

At the next step, a divide-and-conquer method [10] is used to successively subdivide each of the resulting convex polygons (trapezoids) into smaller convex sub-polygons until each of them can be covered visually by one guard (Figure 1b).

Since, in the current problem, we are interested in monitoring only the boundaries, not all the computed area guards are necessary. So, each guard such that the visual area of a robot does not intersect the boundaries when it is located on that guard, is removed from the set of area guards. Figure 1c illustrates the *boundary guards (BG)* computed on the sample environment.

$$BG = \{g_1, g_2, ..., g_k\}. \quad (1)$$

B. *Boundary Graph*

Once the boundary guards are located in the target area, a graph called the *Visibility Graph (VG)* is constructed on

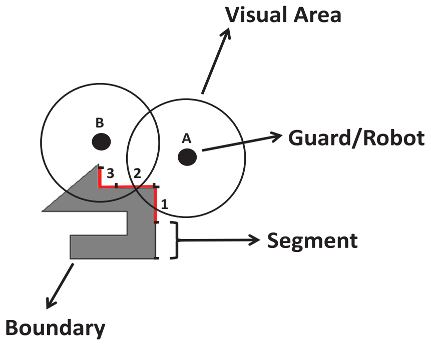


Fig. 2: *Boundary Segmentation*

the guards (Figure 1d). In order to build the *Visibility Graph*, any pair of boundary guards which are mutually visible are connected by an edge. Two guards are mutually visible if the edge connecting them does not intersect any structures in the environment.

Visibility Graph is used to build the roadmap, because it provides the robots with more paths and more freedom of movement to traverse the environment, compared to other approaches like *Constrained Delaunay Triangulation* or *Voronoi Diagram*.

C. Boundary Segmentation

The boundaries of the area and the structures are divided into identical length *segments*, each of which is small enough, such that if an event occurs in a *segment*, the event is visible from any part of that *segment*. In other words, if a robot's visual range covers just part of the *segment* on the boundary, the robot is still capable of detecting all the events occurred in any part of the *segment*.

$$Segments = \{seg_1, seg_2, ..., seg_n\}. \tag{2}$$

Definition 3. *Visual Area of a Guard (VA):* The *visual area* of a guard, $VA(g)$, is the set of all the *segments* visible to the robot when it is located on the guard g.

$$VA(g) = \{seg_i, seg_j, ..., seg_s\}. \tag{3}$$

Definition 4. *Shared Segment:* A *shared segment* is common to the *visual area* of two or more guards.

Definition 5. *Segment Parent Guards (SPG): Parent guards* of a *shared segment* are guards whose *visual area* contain that *segment*.

$$SPG(seg) = \{g_i, g_j, ..., g_p\}. \tag{4}$$

The notion of *segment parent guards* implies that an event occurred in a *segment* can be detected from more than one guard, but once it is detected from a guard, the event is no longer visible from the other guards.

Assumption 1. *Events are only detected when a robot is located on a guard.*

In Figure 2, the *visual area* of guard A covers *segments* 1 and 2, and the *visual area* of guard B covers *segments* 2 and 3. *Segment* 2 is a *shared segment* between guards A and B, and subsequently, guards A and B are the *parent guards* of *segment* 2. A robot located on guard A can detect the events occurred in any part of *segments* 1 and 2, and a robot located on guard B can detect the events occurred in any part of *segments* 2 and 3.

V. Uninformed Boundary Coverage

In *Uninformed Boundary Coverage*, a tour is constructed on the *Boundary Graph* using the *Chained Lin-Kernighan* algorithm.

Chained Lin-Kernighan, a modification of the *Lin-Kernighan* algorithm [11], is generally considered to be one of the best heuristic methods for generating *optimal* or *near-optimal* solutions for the *Euclidean Traveling Salesman Problem* [5]. Given the distance between each pair of a finite number of nodes in a *complete* graph, the *Travelling Salesman Problem* is to find the shortest tour passing through all the nodes exactly once and returning to the starting node.

The input of the *Chained Lin-Kernighan* algorithm is the distance matrix of the *Boundary Graph*. The matrix consists of the shortest path distances between all pairs of guards in the *Boundary Graph*, and is consequently indicative of a complete graph, even though the *Boundary Graph* itself is not complete. Having built the shortest tour passing through all the guards of the *Boundary graph*, the robots are then distributed equidistantly along the tour and move repeatedly around it in the same direction.

VI. Informed Boundary Coverage

For *Informed Boundary Coverage*, the robots try to maximize the reward by detecting the maximum number of events, weighted by their importance, in minimum time. To this end, each robot independently *learns* the pattern of event occurrence on the boundaries over time and based on that, estimates the expected reward of visiting a state in the target area at each time step. Each robot then *plans* in a decentralized manner to select the best possible path to visit the most promising states at the time in the target area. The initial locations of the robots are chosen randomly in the target area.

The *Multi-Robot Repeated Boundary Coverage* problem is formulated for each robot as a tuple *(BG,A,ST,STR)* where:

- *BG* is the set of states or boundary guards, representing the position of the robot in the target area.
- *A* is the set of actions available for a robot in each state. An action is defined as moving from one guard to any other guard in the *Boundary Graph*. At the beginning, each robot calculates the shortest path between each pair of guards in the *Boundary Graph*, using the *Floyd-Warshall* algorithm, hence the robots will not need to repeatedly compute the shortest paths in the graph during the planning stage of the coverage mission.
- *ST* is the state transition function which is deterministic, such that it guarantees reaching the target state chosen by the robot from the current state, when the action is performed.
- *STR* is the state reward, which is equal to the sum of the discounted importance of the detected events at the state (*i.e.* guard).

$$STR(g,t) = \sum_{seg \in VA(g)} \sum_{E_i \in E} \sum_{e_i \in E_i} weight(E_i) \times \gamma^{t-st(e_i)}, \quad (5)$$

where $t - st(e)$ is the time interval between starting event e and the time of the visit to g, *i.e.* the detection time of the event e.
Once a robot arrives at a guard g, it can detect all the events occurred within the $VA(g)$, the *visual area* of the guard g. It is assumed that the reward a robot receives for an event depends on how early the event is detected. At each time step after the event occurrence, the detection reward of the event is multiplied by a discount factor of $\gamma = 0.95$.

Definition 6. *Time of Last Visit (TLV):* Each robot separately keeps track of the times of the last visit to the guards. If $BG = \{g_1, g_2, ..., g_k\}$ is the set of boundary guards, then for each guard $g \in BG$, $TLV(g)$ represents the last time the guard g was visited by the robot or any other robot managed to communicate the visit to the guard with the robot. Therefore, the times of the last visit to the guards are not globally shared by the robot team, rather each robot, at each time step, may have a different knowledge from the rest of the robot team of the times of the last visit to the guards.

Based on this, we can calculate the time of the last visit to each *segment* of the boundary:

$$TLV(seg) = max\{TLV(g) | g \in SPG(seg)\}. \quad (6)$$

Intuitively, the time of the last visit to a *segment* is the most recent visit of the robot to one of the *segment*'s *parent guards*.

Definition 7. *Policy:* A policy $\pi : BG \to A$ at each state determines which action should be performed next by the robot.

Note that the learning procedure described below is performed by each robot independently of the rest of the team.

A. Learning

If the robot had knowledge of the probability of occurrence of the different events in each state as well as the starting time of the events, it would be able to calculate the *STR* to find a policy, maximizing the total reward of the boundary coverage mission, but since this information is not available to the robot, it estimates the *STR*, as the sum of the *Expected Segment Reward (ESR)* of the *segments* comprising a state:

$$STR(g,t) \simeq \sum_{seg \in VA(g)} (ESR(seg,t)). \quad (7)$$

Expected Segment Reward (ESR) is defined to represent the expected reward of a *segment*, *seg*, at the time t. The *ESR* can be calculated using the sum of the discounted importance of the events occurred between the last visit, $TLV(seg)$, and the current visit time, t, to the *segment*:

$$\begin{aligned} ESR(seg,t) = \sum_{E_i \in E} \sum_{e_i \in E_i} (1 + \gamma^1 + \gamma^2 + ... + \gamma^{t-TLV(seg)}) \times \\ PSE(E_i, seg) \times weight(E_i), \end{aligned} \quad (8)$$

where γ is the reward discount factor. We assume that for every time step after an event occurs without being detected, the event detection reward is discounted by γ. Furthermore, the *Probability of Segment Event (PSE)* is defined for each event type $E_i \in E$ and each *segment*, *seg*, to indicate the probability of events of type E_i occur within the *segment* at each time step.

In formula (8), $\sum_{E_i \in E} \sum_{e_i \in E_i} PSE(E_i, seg) \times weight(E_i)$ is the *Segment Reward Accumulation Rate* of all the events at the *segment*, *seg*, and is represented by *SRAR*(seg). If a robot knows the *SRAR* of the events at each *segment*, it can calculate the *ESR* for any arbitrary time t.

To this end, a learning procedure for estimating the *SRAR* gradually updates its initial value. In the initialization step, the robot assumes that all the events have the same probability of occurrence at each *segment*. Therefore, all the *SRAR*s are initialized to 1. When the robot arrives at a guard g, it can detect whether or not an event has occurred at the *segments* belonging to $VA(g)$. The *SRAR* of the guard's *segments* is then updated using the following formula:

$$\begin{aligned} &\forall seg \in VA(g),\ SRAR(seg) = \\ &(1-\alpha) \times SRAR(seg) + \alpha \times \frac{\sum_{E_i \in E} \sum_{e_i \in E_i} (weight(E_i))}{t - TLV(seg)}, \end{aligned} \quad (9)$$

where α is the learning rate set to 0.9 and t is the time of the visit to g. This formula gives more weight to the new information than the old information. The robot performs the updating process for all the event types and all the *segments* of the guards.

In summary, the *SRAR* of a *segment* is updated once a robot visits one of its *parent guards*. As already mentioned, the *SRAR* represents the reward accumulation rate on the *segment*. Now we can use the *SRAR* to calculate the *ESR* of the *segments* using the following procedure:

At the beginning, the *ESR* of all the *segments* are initialized to zero. Then, at each time step, if the robot has yet to arrive at a guard, the *ESR* of all the *segments* of the boundary is updated using the following equation:

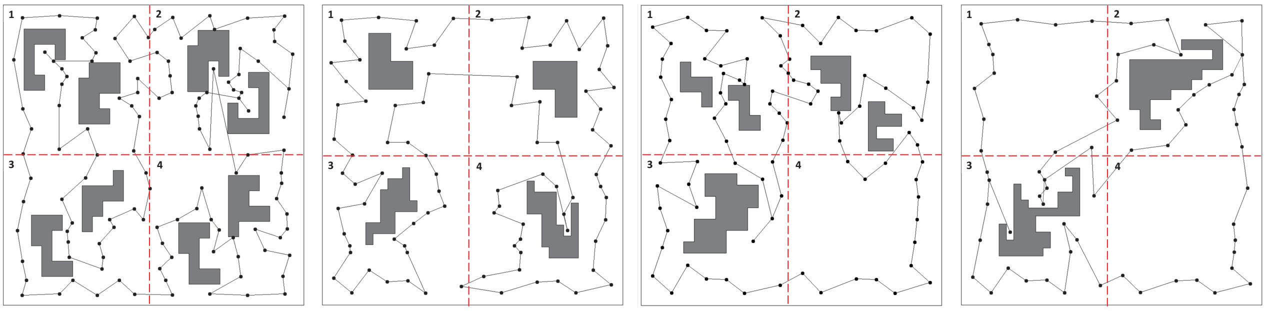
$$\begin{aligned} &\forall seg \in Segments, \\ &ESR(seg,t) = \gamma \times ESR(seg, t-1) + SRAR(seg). \end{aligned} \quad (10)$$

If the robot arrives at a guard g, it detects all the events that have occurred in its *segments* and communicates the guard ID to the robots located within its communication range. Since all the events occurred in the *segments* of g have been detected, the expected reward of the *segments* at the time t (*i.e.* the time of the visit to the guard g) becomes zero. Consequently, for the robot and all the communicated robots:

$$\forall seg \in VA(g),\ ESR(seg,t) = 0. \quad (11)$$

Note that each robot has its own knowledge of the *ESR* of the *segments*, so the other robots (except the ones who received the communication) may still assume that there are some undetected events in the *segments* of g at the time, and subsequently their *ESR*s of the *segments* of g are not zero.

This updating process continues during the boundary coverage operation.



(a) *Map 1*

(b) *Map 2*

(c) *Map 3*

(d) *Map 4*

Fig. 3: *Maps Used in the Experiments*

B. Planning

Once a robot arrives at a guard and detects all the events which may have occurred in the *segments* of the guard, the robot selects the next action to perform. As already mentioned, an action in the boundary coverage operation is defined as moving from one guard to another guard in the *Boundary Graph*. At each state, the robot considers all the precomputed shortest paths to all the other guards it can move to. For each path, $path(g_c, g_d)$, where g_c is the current guard and g_d is the destination guard, *Path Reward (PR)* is defined as the reward the robot receives when moving from the guard g_c to the guard g_d. The path from g_c to g_d includes zero or more intermediate guards and can be represented as:

$$path(g_c, g_d) = [g_c, g_i, g_j, ..., g_r, g_d]\,. \tag{12}$$

Given the speed of the robot, the arrival time at each of the guards on the path can be estimated. Hence, the robot can have an estimate of the $ESR(seg, t(g))$ for each *segment* of the guard $g \in path(g_c, g_d)$, in which $t(g)$ is the arrival time to the guard g. For such a path, the *PR* is calculated as below:

$$PR(path(g_c, g_d)) = \sum_{g \in path(g_c, g_d)} \sum_{seg \in VA(g)} ESR(seg, t(g)). \tag{13}$$

When calculating the *PR*, the robot should take into account the *segments* shared by some *parent guards* as well, namely the robot in its calculations initializes the *shared segment*'s *ESR* to zero when it is going to visit one of its *parent guards* along the path.

Next, for each path, the *Average Path Reward (APR)* is calculated using the following formula:

$$APR(path(g_c, g_d)) = \frac{PR(path(g_c, g_d))}{t(g_d) - TLV(g_c)}, \tag{14}$$

where g_d is the destination guard on the path, and $t(g_d)$ is the arrival time to the guard g_d. The robot will select a path with the maximum *Average Path Reward* to traverse next.

VII. Experiments and Results

We wish to compare *Informed Boundary Coverage (IBC)* with *Uninformed Boundary Coverage (UBC)* in terms of the total reward being received by the team for detecting the events in a finite simulation time. We have developed a simulator to test the algorithms in different scenarios. The simulator can support different numbers of robots in the target area, different visual ranges for the robots, and varying degrees of clutter in the environment.

The experiments are conducted using $1, 2, 5$ and 10 robots, with a visual range of $0.5m$ and a communication range of $1m$, on the sample environments of Figure 3. The size of the environments is $10m \times 10m$, and the boundaries are divided into *segments* of length $0.5m$.

We divide the environment maps into 4 disjoint sub-regions. In maps 1 and 2, there are structures in all sub-regions of the environment. In map 3, there are structures in 3 of the sub-regions, and in map 4, there are structures in 2 of the sub-regions. Four experiments were designed, in each, the pattern of event occurrence varies in the sub-regions. Figures 4, 5, 6 and 7 show the total reward being received by the team $(1, 2, 5$ and 10 robots) in each experiment during 15000 cycles of the simulation run on map 1. The other maps show a similar trend to that for map 1.

Figure 3 shows the cyclic tours built on the maps using *Uninformed Boundary Coverage*, assuming that the robots' visual range is $0.5m$.

Figure 8 also shows the percentage of time a team of 10 robots spends in each sub-region during the 15000 cycle simulation of the 4 experiments.

A. Experiment 1: Uniform Event Occurrence

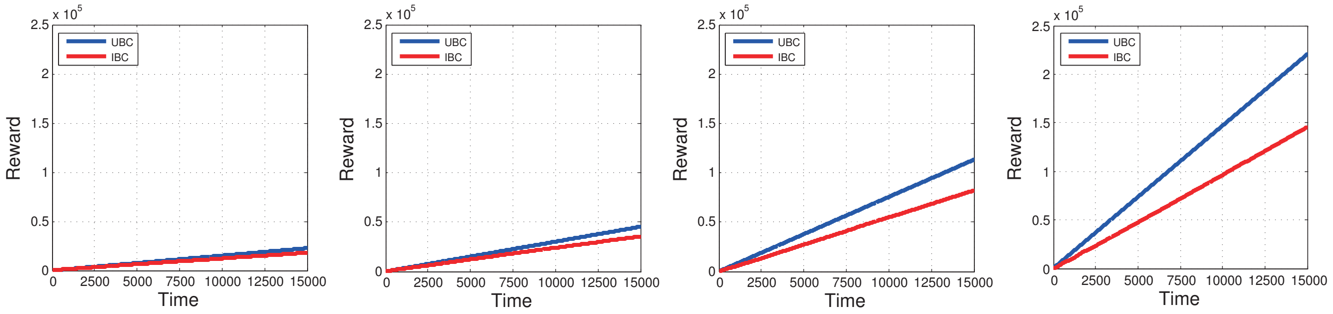
In this experiment, the events occur in all the *segments* of the boundaries, and in all the sub-regions of the environment uniformly with an equal probability of 0.5, meaning that at each cycle there is a 0.5 chance that an event occurs in a *segment*. Each event has a weight of 1.

As shown in Figure 4, *Uninformed Boundary Coverage* collects more rewards than *Informed Boundary Coverage* regardless of the size of the team, and as the number of robots increases in the environment, the difference between the two algorithms grows. When the events occur uniformly along the boundaries, following the shortest tour on the *Boundary Graph* is a reasonable approach for the robots.

Figure 8a also shows the robots using *Informed Boundary Coverage* spend an almost equal amount of time in each sub-region during the simulation.

B. Experiment 2: Non-uniform Event Occurrence

In this experiment, no events occur on the boundaries of sub-region 1. Within sub-region 2 the events occur with a



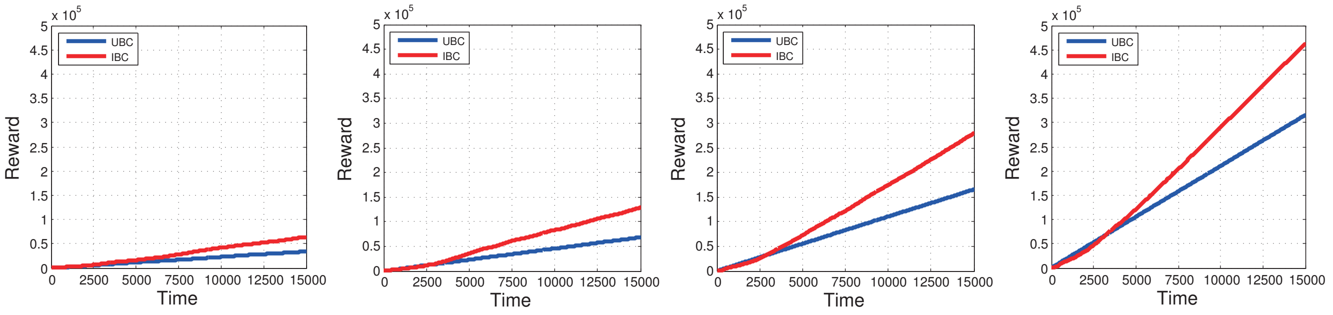
(a) *Number of Robots = 1*

(b) *Number of Robots = 2*

(c) *Number of Robots = 5*

(d) *Number of Robots = 10*

Fig. 4: *Experiment 1: Uniform Event Occurrence*



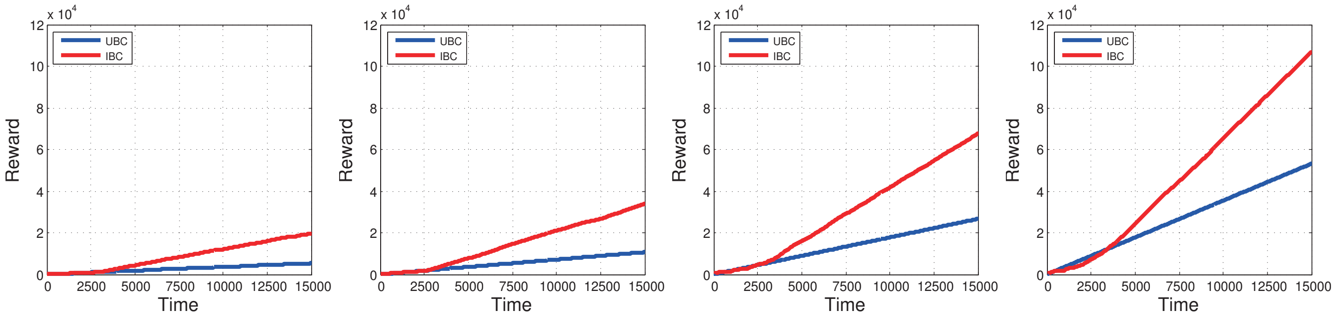
(a) *Number of Robots = 1*

(b) *Number of Robots = 2*

(c) *Number of Robots = 5*

(d) *Number of Robots = 10*

Fig. 5: *Experiment 2: Non-uniform Event Occurrence*



(a) *Number of Robots = 1*

(b) *Number of Robots = 2*

(c) *Number of Robots = 5*

(d) *Number of Robots = 10*

Fig. 6: *Experiment 3: All Events Occur in One Sub-region*

probability of 0.1 in each *segment* of the boundaries. Within sub-region 3, the events occur with a probability of 0.9, and within sub-region 4, the events occur according to a Poisson distribution with a mean of $\lambda = 4$. This implies that more than one event can occur in each *segment* at every cycle. Each event occurring in sub-regions 2 and 3 is weighted 1, and 0.5 in sub-region 4.

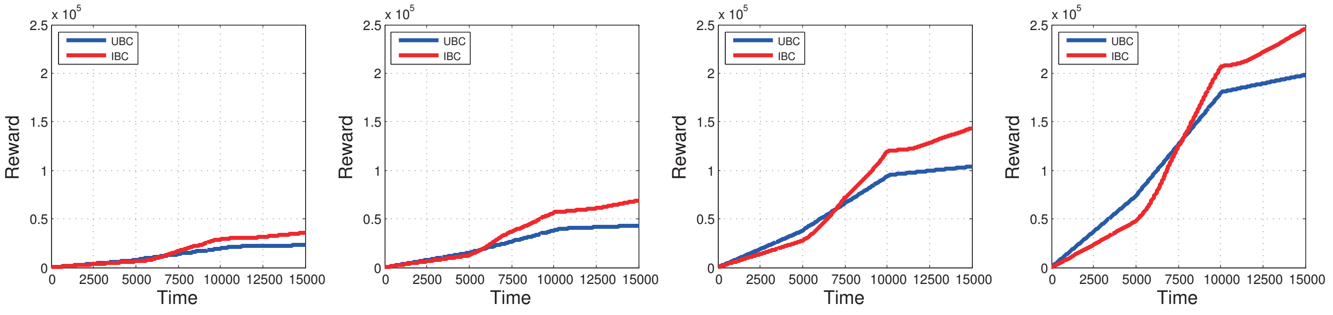
As shown in Figure 5, *Informed Boundary Coverage* outperforms *Uninformed Boundary Coverage* regardless of the size of the team, and as the number of robots increases in the environment, the difference between the two algorithms grows.

Figure 8b also shows that when *Informed Boundary Coverage* is used, the percentage of time the robots spend in sub-region 4 increases during the progress of the simulation because of the higher number of events occurring in that area compared to the other sub-regions. Sub-region 3 is the second most promising area for the robots, and finally are sub-regions 2 and 1 subsequently.

C. *Experiment 3: All Events Occur in One Sub-region*

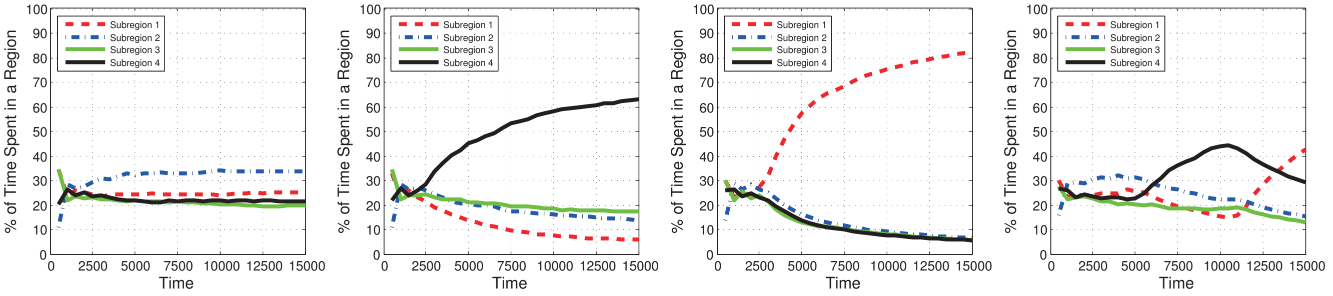
In this experiment, in sub-region 1, the events occur with a probability of 0.9 in the *segments* of the boundary. No events occur in sub-regions 2, 3 and 4. Each event is weighted 1.

As shown in Figure 6, *Informed Boundary Coverage* outperforms *Uninformed Boundary Coverage* regardless of the size



(a) *Number of Robots = 1* (b) *Number of Robots = 2* (c) *Number of Robots = 5* (d) *Number of Robots = 10*

Fig. 7: *Experiment 4: Dynamic Event Occurrence*



(a) *Experiment 1* (b) *Experiment 2* (c) *Experiment 3* (d) *Experiment 4*

Fig. 8: *Percentage of Time a Team of 10 Robots Using IBC Spends in Each Region on Different Experiments*

of the robot team, and as the number of robots increases in the environment, the difference between the two algorithms grows.

Figure 8c also shows that when *Informed Boundary Coverage* is used, the percentage of time the robots spend in sub-region 1 increases dramatically over time because of its higher expected reward for the robot team compared to the other areas. On the other hand, the robots' presence in sub-regions 2, 3 and 4 declines.

D. *Experiment 4: Dynamic Event Occurrence*

In this experiment, the pattern of event occurrence changes at some times unknown to the robots during the simulation run. We presume that during the first 5000 cycles, the events occur in the environment according to the pattern discussed in Experiment 1 (Uniform Event Occurrence), between cycles $5000 - 10000$, the events occur according to the pattern discussed in Experiment 2 (Non-uniform Event Occurrence), and between cycles $10000 - 15000$, the events occur according to the pattern mentioned in Experiment 3 (All Events Occur in One Sub-region).

As shown in Figure 7, *Informed Boundary Coverage* outperforms *Uninformed Boundary Coverage* regardless of the size of the robot team, and as the number of robots increases in the environment, the difference between the two algorithms grows. In this experiment, the robots adapted themselves to the changes in the pattern of event occurrence on the boundaries, and updated their policies based on these changes. Although, as we expected, *Informed Boundary Coverage* cannot outperform *Uninformed Boundary Coverage* during the first 5000 cycles, when the events occur uniformly along the boundaries; however, it ends up collecting more rewards in the rest of the simulation when the events occur according to the patterns discussed in Experiments 2 and 3.

Figure 8d also shows that in the first 5000 cycles of the simulation, the robots spends an almost equal amount of time in each sub-region. In the second 5000 cycles, the robots presence in sub-region 4 increases and in the third 5000 cycles of the simulation, the robots' presence increases in sub-region 1 and lessens in sub-region 4.

VIII. Discussion

Table I shows the total reward received by the team using the *Informed Boundary Coverage (IBC)* and the *Uninformed Boundary Coverage (UBC)* algorithms on the four maps of Figure 3 during a 15000 cycle simulation. The results on maps 2, 3 and 4 are consistent with the results discussed in detail for map 1. As shown in the table, *Uninformed Boundary Coverage* collected more rewards on all the maps when the events occur uniformly along the boundaries (Experiment 1). However, *Informed Boundary Coverage* outperforms *Uninformed Boundary Coverage* on all the maps when the events occur non-uniformly, on just parts of the boundaries, or when the pattern of event occurrence changes over time.

TABLE I: *Total Reward Collected by the Team on Different Maps Based on Different Experiments and Various Number of Robots*

	# of Robots	Experiment 1		Experiment 2		Experiment 3		Experiment 4	
		UBC	IBC	UBC	IBC	UBC	IBC	UBC	IBC
Map #1	1	22626	18122	34316	63822	5350	19662	22951	35732
	2	45254	35422	68763	124580	10670	33879	43232	68895
	5	113020	81906	165000	278750	26629	67578	103960	143590
	10	220190	145280	314470	458370	53128	106940	198240	245720
Map #2	1	19452	16482	33085	71412	4172	13222	20618	37157
	2	38964	30403	65285	130670	8472	26003	39383	66564
	5	97228	68535	158410	279230	21170	50193	93523	143450
	10	185980	121390	304720	457830	42156	79808	179040	225240
Map #3	1	21428	17915	25314	53183	7141	19925	19292	28485
	2	42825	35308	50940	104680	14100	34909	37165	59804
	5	106770	74160	119390	208840	35277	70126	88069	121610
	10	203010	129780	222530	332110	70154	114680	165940	199410
Map #4	1	17396	16321	22851	70146	3106	11824	15144	17356
	2	34789	29572	45161	102930	6226	24011	29620	37247
	5	86652	62621	105110	194930	15585	49351	70116	96063
	10	163970	106210	196170	296840	30987	66821	130940	157440

IX. Conclusions and Future Work

We have addressed the problem of repeated coverage by a team of robots of the boundaries of a target area and the structures inside it. The robots have limited circular visual and communication range. Events may occur on any parts of the boundaries and may have different importance weights. The robots are not *a priori* aware of the probability distribution of occurrence of events on the boundaries. As a result, each robot autonomously and continuously learns the pattern of event occurrence on the boundaries over time, capturing the uncertainties in the target area, and then plans in a decentralized manner to select the best path at that time in the target area to visit the most promising parts of the boundary. The performance of the learning algorithm was compared with a heuristic algorithm for the *Travelling Salesman Problem*, on the basis of the total reward collected by the team during a finite repeated boundary coverage mission.

For future work, we plan to address issues related to noisy sensors of the robots, action uncertainty, and unknown structures. In the case of noisy sensors, the accuracy of the information achieved by a robot could vary with the distance of the boundary from the robot. Some events might also require multiple robots to form a coalition in order to be handled. Another future direction is the case that the robot team would have the ability to change its behavior over time in response to a changing environment with dynamic structures, either to improve performance or to prevent unnecessary degradation in performance.

References

[1] N. Agmon, S. Kraus, and G. Kaminka, "Multi-robot perimeter patrol in adversarial settings," in *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA*, 2008, pp. 2339–2345.

[2] N. Agmon, S. Kraus, G. A. Kaminka, and V. Sadov, "Adversarial uncertainty in multi-robot patrol," in *Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI*, 2009, pp. 1811–1817.

[3] M. Ahmadi and P. Stone, "A multi-robot system for continuous area sweeping tasks," in *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA*, 2006, pp. 1724–1729.

[4] P. Amstutz, N. Correll, and A. Martinoli, "Distributed boundary coverage with a team of networked miniature robots using a robust market-based algorithm," *Annals Mathematics Artificial Intelligence*, vol. 52, no. 2-4, pp. 307–333, 2008.

[5] D. Applegate, W. Cook, and A. Rohe, "Chained Lin-Kernighan for large traveling salesman problems," *INFORMS Journal on Computing*, vol. 15, pp. 82–92, 2003.

[6] J. Czyzowicz, L. Gasieniec, A. Kosowski, and E. Kranakis, "Boundary patrolling by mobile agents with distinct maximal speeds," in *Proceedings of the 19th European conference on Algorithms, ESA*, 2011, pp. 701–712.

[7] Y. Elmaliach, A. Shiloni, and G. A. Kaminka, "A realistic model of frequency-based multi-robot polyline patrolling," in *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, 2008, pp. 63–70.

[8] P. Fazli, A. Davoodi, P. Pasquier, and A. K. Mackworth, "Complete and robust cooperative robot area coverage with limited range," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, 2010, pp. 5577–5582.

[9] E. Jensen, M. Franklin, S. Lahr, and M. Gini, "Sustainable multi-robot patrol of an open polyline," in *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA*, 2011, pp. 4792 –4797.

[10] G. D. Kazazakis and A. A. Argyros, "Fast positioning of limited visibility guards for inspection of 2D workspaces," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, 2002, pp. 2843–2848.

[11] S. Lin and B. Kernighan, "An effective heuristic algorithm for the traveling-salesman problem," *Operations Research*, vol. 21, no. 2, pp. 498–516, 1973.

[12] A. Marino, L. Parker, G. Antonelli, and F. Caccavale, "Behavioral control for multi-robot perimeter patrol: a finite state automata approach," in *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA*, 2009, pp. 3350–3355.

[13] J. O'Rourke, *Art gallery theorems and algorithms*. New York, NY, USA: Oxford University Press, Inc., 1987.

[14] K. Williams and J. Burdick, "Multi-robot boundary coverage with plan revision," in *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA*, 2006, pp. 1716–1723.

[15] B. Zalik and G. J. Clapworthy, "A universal trapezoidation algorithm for planar polygons," *Computers & Graphics*, vol. 23, no. 3, pp. 353 – 363, 1999.