J. M.l Palacios-Gasos, Z. Talebpour, E. Montijano, C. Sagues and A. Martinoli
Optimal Path Planning and Coverage Control for Multi-Robot Persistent Coverage in Environments with Obstacles, 2017 IEEE International Conference on Robotics and Automation (ICRA), pages 1321-1327
(ISBN 978-1-5090-4632-4/17), Singapore, May 29 - June 3, 2017

Optimal Path Planning and Coverage Control for Multi-Robot Persistent Coverage in Environments with Obstacles

José Manuel Palacios-Gasós, Zeynab Talebpour, Eduardo Montijano, Carlos Sagüés and Alcherio Martinoli

Abstract— **Persistent coverage aims to maintain a certain coverage level over time in an environment where such level deteriorates. This level can be associated to temperature, dust or sensor information. We propose an algorithmic solution in which each robot locally finds the best paths and coverage actions to keep the desired coverage level over the whole environment. Using Fast Marching Methods, optimal paths are computed in terms of coverage quality, while keeping a safety distance to obstacles. Additionally, our solution enables a computationally efficient evaluation of a list of potential trajectories, allowing us to choose the one that mostly improves the coverage along the whole path. The combination of this algorithm with a Dynamic Window navigation makes our approach competitive in terms of flexibility and robustness in changing environments with existing solutions. Finally, we also propose a coverage action controller, locally computed and optimal, that makes the robots maintain the coverage level of the environment significantly close to the objective. Simulations and real experiments validate the whole approach.**

I. Introduction

Persistent tasks are repetitive, redundant and time consuming activities in our days. Autonomous vacuum cleaners or lawn mowers [1], either alone or as members of a team, are ready to automate such tasks. Heating buildings, watering crops or monitoring the environment can also experience a considerable increase in efficiency if they are performed by autonomous robots [2]. In addition, new applications in which the robots directly help people have arisen recently, for instance giving information in a touristic place [3] or edutainment in a pediatric hospital [4].

In this paper, we address the persistent coverage problem [5] to give the robots the autonomy to perform these tasks. The goal is to maintain a certain coverage level over a dynamic environment, in which such level deteriorates over time, using a team of robotic agents. This problem is different from that of traditional coverage [6], [7] in the sense that this deterioration requires the robots to keep moving to maintain the desired coverage level. The coverage level can be interpreted as a physical magnitude, namely, temperature of rooms in a building in a heating mission, or as uncertainty of measurements at points of interest in the case of sensing.

J.M. Palacios-Gasós, E. Montijano and C. Sagüés are with Instituto de Investigación en Ingeniería de Aragón (I3A), Universidad de Zaragoza, Zaragoza, Spain. {jmpala, emonti, csagues}@unizar.es
Zeynab Talebpour and Alcherio Martinoli are with Distributed Intelligent Systems and Algorithms Laboratory, Ècole Polytechnique Fédrale de Lausanne, Lausanne, Switzerland. {zeynab.talebpour, alcherio.martinoli}@epfl.ch

This work was partially supported by projects RTC-2014-1847-6 of Retos-Colaboración, DPI2015-69376-R from Ministerio de Economía y Competitividad/Unión Europea, DGA T04-FSE, CUD2013-05 and DGA Scholarship C076/2014, partially funded by European Social Fund.

Three different approaches to persistent coverage can be found in the literature. The first type of works assumes a known, finite number of points of interest and the trajectories between them. They calculate the order in which the points should be visited, the time spent at each one and which robot is responsible of covering them. In [2], a Mixed Integer Program maximizes the reward of a single robot covering some points under a time constraint. The problem of finding the optimal coverage times for a multi-robot system is addressed in [8]. A market-based approach is presented in [9], where the robots locally calculate the cost of covering each point of interest and bid for it. The limitations of these methods are the assumptions of known paths and an a priori discretization of the environment.

The second type of studies are focused on finding a controller that drives the robots according to the coverage level of a spatially continuous environment. In [10] a centralized system finds coverage goals for the robots and a local, gradient-based controller guides them to the respectively assigned goals. Additionally, [11] avoids collision using a bounded potential repulsion. In [12] the robots are capable of finding their own goals distributedly and the motion also gathers the reward of covering them. These methods behave well in the long term but may overcover areas locally and waste time and energy.

The third option is to plan a priori periodic paths that cover the entire continuous environment. This problem is addressed in [13] with the objective of finding straight paths and minimize the number of turns. A polynomial-time coverage algorithm is designed in [14] based on the Spanning Tree Coverage introduced in [15]. In [16], authors aim to minimize the amount of time during which points remain unvisited by planning paths through a graph. Lin et al. [17] formulate a parametric optimization to determine a set of elliptical trajectories for the robots in a two-dimensional space. Rapidly-exploring Random Trees are used in [18] to sense a dynamic gaussian random field. In changing environments these solutions are limited. If an obstacle blocks a single path or an agent fails, a replanning of the whole team is needed, typically resulting in a high computational cost and a centralized solution.

In this work, we propose a solution that leverages advantages and overcomes limitations of the three kind of methods mentioned above. Our distributed algorithm is based on actively selecting coverage goals in a continuous environment and planning the optimal paths to such goals. The selection of goals can be compared to discretized approaches. Nevertheless, our active search of the points that require the most coverage at each time allows us to cover continuous environments. Our path planning strategy is based on Fast Marching Methods (FMM), that have proved to be successful in real domestic spaces with high complexity [19]. This on board, fast computation of paths allows the avoidance of unexpected obstacles as opposed to approaches that plan a priori, and the optimality of the paths outscores the use of local motion controllers. The paths are followed using a Dynamic Window Approach (DWA) [20]. Moreover, we introduce a coverage action controller that allows the robots to provide the optimal coverage at each point. This is essential in applications such as heating or watering, to avoid over-heating or flooding, respectively, or to save power in the case of vacuuming or cleaning.

The contribution of this approach with respect to previous works is two-fold: the novel combination of an active selection of goals with a coverage-optimal path planning, and an optimal controller of the coverage action. In particular, for the first we create two cost functions to find optimal paths to the goals and keep the robots safe from obstacles. Moreover, we provide a new method to select the best goal based on the coverage paths and introduce the notion of safe goal. For the second contribution, we formulate the controller and prove its optimality. Additionally, we show a significant improvement with respect to [12] in simulation and demonstrate the effectiveness of our approach with real experiments.

II. Problem Formulation and Solution Overview

In order to introduce our approach to the problem, we formulate the evolution of the coverage and the improvement function, and give an overview of our solution.

A. Persistent Coverage Problem

The aim of persistent coverage is to maintain a bounded environment $\mathbf{Q} \subset \mathbb{R}^2$ covered over time. The desired coverage level, or coverage objective, is denoted as $Z^*(\mathbf{q}) > 0, \forall\, \mathbf{q} \in \mathbf{Q}$. The actual coverage level is represented by a time-varying field, $Z(\mathbf{q}, k) \geq 0$, and it deteriorates over time with a constant decay rate, $d(\mathbf{q})$, with $0 < d(\mathbf{q}) < 1$.

In order to keep the coverage level as close as possible to the desired level, we deploy a team of N robots that are considered to be holonomic, $\mathbf{p}_i(k) = \mathbf{p}_i(k-1) + \mathbf{u}_i(k-1)$, where $\mathbf{p}_i(k) \in \mathbf{Q}$ is the position of robot i at time $k \geq 1$, with $i \in \{1, ..., N\}$, and $\mathbf{u}_i(k)$, the control input. It is assumed that the position of each robot is accurately provided by an internal localization system. Additionally, we assume for simplicity of the formulation that the shape of the robots is a circle of radius r. Each robot has a map of the environment that may include walls and obstacles. We define $\mathbf{Q}_f(k)$ as the points in which the robot can be located without colliding with the obstacles. This is obtained by expanding the obstacles by the radius of the robot. Although it may change over time, we omit this dependency in the rest of the paper for simplicity.

Each robot is capable of increasing the coverage level in an area $\mathbf{\Omega}_i\big(\mathbf{p}_i(k)\big)$, which we call coverage area. Although it is not necessarily circular, convex or equal for all the robots, it is bounded by a circle of radius r_i^{cov} centered at $\mathbf{p}_i(k)$. Inside this area, the coverage is increased according to a production function, $\alpha_i\big(\mathbf{q}, \mathbf{p}_i(k)\big) \geq 0, \mathbf{q} \in \mathbf{\Omega}_i\big(\mathbf{p}_i(k)\big)$. Outside this area, the production is equal to zero, i.e., $\alpha_i\big(\mathbf{q}, \mathbf{p}_i(k)\big) = 0, \forall\, \mathbf{q} \notin \mathbf{\Omega}_i\big(\mathbf{p}_i(k)\big)$. This production can be controlled by a certain gain, $0 \leq \rho_i(k) \leq \rho_i^{\max}$, that is the coverage action.

The model of the evolution of the coverage level can be expressed as the following recurrence equation:

$$Z(\mathbf{q}, k) = d(\mathbf{q}) Z(\mathbf{q}, k-1) + \alpha(k), \tag{1}$$

where $\alpha(k) \equiv \sum_{i \in \{1,...,N\}} \rho_i(k) \alpha_i\big(\mathbf{q}, \mathbf{p}_i(k)\big)$ is the total coverage action of the team at each point. For the sake of clarity, in the rest of the paper we omit the spatial dependencies when possible, i.e., $Z(\mathbf{q}, k) \equiv Z(k)$ or $Z^*(\mathbf{q}) \equiv Z^*$, and also make $\alpha_i(k) \equiv \alpha_i\big(\mathbf{q}, \mathbf{p}_i(k)\big)$ for clarity.

B. Improvement Function

The solution for persistent coverage must give the motion and coverage actions that control the robots. To calculate these actions, we use the *improvement function* from [12], similar to the locational cost in static coverage [6]:

$$M_i(\mathbf{p}, k) = \frac{\displaystyle\int_{\mathbf{\Omega}_i(\mathbf{p})} \frac{Z^* - Z_i(k)}{Z^*}\, \Phi(\mathbf{q})\, \rho_i^{\max}\, \alpha_i(\mathbf{q}, \mathbf{p})\, \mathrm{d}\mathbf{q}}{\displaystyle\int_{\mathbf{\Omega}_i(\mathbf{p})} \Phi(\mathbf{q})\, \rho_i^{\max}\, \alpha_i(\mathbf{q}, \mathbf{p})\, \mathrm{d}\mathbf{q}}, \tag{2}$$

where $\mathbf{p} \in \mathbf{Q}_f$, $Z_i(k)$ is the local estimation that a robot has of $Z(k)$ and $\Phi(\mathbf{q}) \in (0, 1]$ is a weighting function to adjust the importance of covering each point. Note that $M_i(\mathbf{p}, k) \leq 1$ by definition. This function is an interesting metric to evaluate how profitable it is in terms of coverage to locate a robot at a specific position $\mathbf{p}$. It means that the coverage area around the points where $M_i(\mathbf{p}, k) < 0$ is over-covered and, where $0 < M_i(\mathbf{p}, k) < 1$, it is only slightly covered. In fact, in the points with the highest positive values of $M_i(\mathbf{p}, k)$ is where the robot can improve the coverage the most. These points can be seen as candidate goals that the robots can follow to provide the best coverage of the environment. An additional advantage of using the improvement function is that the value for each point already includes the coverage of the entire coverage area of a robot located there. Therefore, for planning optimal paths in terms of coverage, the robot can be considered as a point.

C. Solution Overview

The intuitive idea of our distributed approach to address the persistent coverage problem is the following. In the first place, the robots divide the environment in particular regions, one for each robot, similarly as done in static coverage [6]. For this step, the robots only need to know the position of the others that are close to them. After that, all the processing is local and does not require any additional communication with the others. Each robot also creates a list of candidate goals that includes the points of its region in which the coverage level can be improved the most. At each iteration the partition and the list are updated, the latter using the improvement function $M_i(k)$. The goals that are no longer feasible or safe are removed and new goals are included. If the current goal of a robot stops being valid, a new one must be selected. To do so, it first calculates the paths to all the candidates from the list and then selects the one whose path optimizes the coverage. In both cases, the path to the current goal is followed calculating the optimal coverage action and controlling the movement of the robot.

An overview of this procedure that each robot locally performs can be seen in Alg. 1.

Algorithm 1 *Solution Overview*

```
 1: Create initial partition and goal list.
 2: while Robot in operation do
 3:     Update $Z(k)$, (1), and $M_i(k)$, (4).
 4:     Update partition and goal list (Sec. IV-A).
 5:     if Current goal not valid (Sec. IV-C) then
 6:         Plan paths to goals (Sec. III).
 7:         Select new current goal (Sec. IV-B).
 8:     end if
 9:     Calculate coverage action (Sec. V-A).
10:     Control movement (Sec. V-B).
11: end while
```

III. Optimal Coverage Path Planning

In this section we introduce the method to plan the paths to the coverage goals. These paths take into account the coverage and the obstacles that may be present.

A. FMM-based Planning

In order to reach the coverage goals the robots have to plan the paths from their location. These paths not only must avoid all the obstacles but also must go through the lowest coverage areas. The idea is that the robots go to the goals providing the best improvement of the coverage along their paths. To do so, we use an FMM-based planning method that allows us to take into account these two variables: obstacles and improvement of the coverage. This method is suitable for online computation and with the same calculation permit us to find the paths to several points. This represent an interesting advantage, since each robot can analize several goals at the same time and compare them based on the quality of their associated paths.

Given a robot location, FMM-based path planning methods [19] calculate a potential field of the environment, $v_i(\mathbf{q}), \forall\, \mathbf{q} \in \mathbf{Q}_f$, that provides the optimal path if the gradient descent of the field is followed from any goal. This potential field represents the minimal time that a wave front needs to propagate from the robot location to any other feasible point of the environment. To obtain this field, we have to provide the FMM planner with the initial point of the path, i.e., the position of the robot, and the speed of propagation of the wave front as a function of the points of the environment, $F_i(\mathbf{q}, k) > 0$. This speed, which is not directly related with the speed of the robot, can be seen as the inverse of a cost of the robot visiting each point of the environment. When the FMM calculates the potential field, that is smooth and free of local minima, the optimal path $\gamma_i(\tau)$ to any goal is calculated by following the direction of the gradient descent of the field from the goal position:

$$\gamma_i(\tau) = \gamma_i(\tau - \mathrm{d}\tau) - \mathrm{d}\tau \nabla v_i(\gamma_i(\tau - \mathrm{d}\tau)), \tag{3}$$

where the path is parametrized by $\tau \in [0, L]$ and L is the length of the path. Although this path is calculated regardless of the dynamics of the robot, it does not have abrupt turns thanks to the smoothness of the FMM potential field. On the other hand, it is optimal in the sense that it minimizes

$$\int_{\gamma_i(0)}^{\gamma_i(L)} \left[F_i\Big(\gamma_i(\tau), k\Big) \right]^{-1} \mathrm{d}\tau.$$

We consider as speed function

$$F_i(\mathbf{q}, k) = F_i^o(\mathbf{q}, k) + F_i^M(\mathbf{q}, k),$$

that is the result of merging together a function associated to obstacles and another associated to the improvement of the coverage. The idea is that the path optimizes the coverage from the robot position to the goal and avoids the obstacles.

B. Coverage Improvement Speed Function

In order to find the path towards the goal that optimizes the coverage, we introduce a sigmoid speed function that includes the improvement of the coverage as follows:

$$F_i^M(\mathbf{q}, k) = \frac{1}{1 + e^{-\beta_M M_i(\mathbf{q}, k)}}.$$

where β_M is a parameter to modify the shape of the sigmoid. This function assigns a high speed for the wave front to the points in which the improvement function is greater than zero, i.e., where the robot can improve the coverage, and a low speed to the points with negative improvement. The effect of this function is that the FMM planner calculates the path of maximum accumulated improvement to the goal.

Since the speed function is based on the improvement function (2) that changes over time, we introduce the temporal evolution of the improvement to simplify its computation:

$$\begin{aligned} M_i(\mathbf{p}, k) &= dM_i(\mathbf{p}, k-1) + A_i(\mathbf{p}) \\ &\quad - \frac{1}{B_i(\mathbf{p})} \int_{\mathbf{\Omega}_i(\mathbf{p})} \alpha(k)\, \Phi(\mathbf{q})\, \rho_i^{\max}\, \alpha_i(\mathbf{q}, \mathbf{p})\, \mathrm{d}\mathbf{q}, \end{aligned} \tag{4}$$

where

$$A_i(\mathbf{p}) = \frac{1}{B_i(\mathbf{p})} \int_{\mathbf{\Omega}_i(\mathbf{p})} \big(1 - d(\mathbf{q})\big)\, \Phi(\mathbf{q})\, \rho_i^{\max}\, \alpha_i(\mathbf{q}, \mathbf{p})\, \mathrm{d}\mathbf{q},$$

$$B_i(\mathbf{p}) = \int_{\mathbf{\Omega}_i(\mathbf{p})} \Phi(\mathbf{q})\, \rho_i^{\max}\, \alpha_i(\mathbf{q}, \mathbf{p})\, \mathrm{d}\mathbf{q}$$

are constants that can be calculated a priori. This temporary evolution calculates directly the improvement at each time as a function of its previous value and the coverage of the robots. It is interesting to highlight that the evolution of the improvement is the opposite of the evolution of the coverage. The improvement increases constantly with time due to the decay in $A_i(\mathbf{p})$ and decreases when the robots are producing an increase of the coverage level around each point.

C. Obstacle Speed Function

The FMM-based planning already avoids collision with obstacles since the potential field is only calculated for $\mathbf{q} \in \mathbf{Q}_f$, i.e., the collision-free positions. Nevertheless, it is also desirable to keep a safety distance to the obstacles. To do so, we assign a higher cost to the points around the obstacles than to points in free areas so that the wave front propagates faster in free areas than near obstacles. Therefore, the resulting path of the robot goes through free spaces rather than close to obstacles. The speed function that we propose, adapted from [21], is

$$F_i^o(\mathbf{q},k) = 1 - \max\Big(-\frac{\delta(\mathbf{q},k)^2}{C^2} + 2\frac{\delta(\mathbf{q},k)}{C}, 0\Big)$$

when $\delta(\mathbf{q},k) \geq C$ and $F_i^o(\mathbf{q},k) = 0$, otherwise, with C, the safety distance and $\delta(\mathbf{q},k)$, the euclidean distance between the point $\mathbf{q}$ and the closest point of the obstacles.

We do not develop further the avoidance of moving obstacles because of the difficulty of predicting their future positions. However, the navigation algorithm based on DWA handles the obstacles that may appear in the path of the robots and avoids collisions with them.

IV. Goal Selection

In this section, we present a method to obtain the goals based on the improvement function in two steps: first, a search for potential goals; and second, the selection of the best according to the improvement of their entire paths.

A. Distributed Partition and Goal Search

The method to search for candidate goals that we use in this work is similar to the one presented in our previous work [12]. We only present here the main features and the modifications that we have introduced for this work.

Firstly, the environment is partitioned in several regions, one for each robot, to avoid long shifts, oscillating behaviors and conflicts with the others due to close goals or crossing paths. We use a distributed Voronoi partition since it has already been used for coverage problems [6] and prevents collisions, although any other partition could be used.

The search for goals inside the region is done by using a list of possible goals, $\mathbf{L}_i(k)$, that includes the points in which the improvement function of each robot reaches its positive maxima. These points must also be safe and separated from the robot and the other goals of the list. We introduce here for the first time the notion of safe goal as a goal that is not close to any other robot: $\|\mathbf{g}_i - \mathbf{p}_j(k)\| > \mathcal{E}'$, with $\mathbf{g}_i \in \mathbf{L}_i(k)$, $j \neq i, j \in \{1,\dots,N\}$ and $\mathcal{E}'$, the safety distance.

The partition and the list are updated iteratively at a certain rate and also when the current goal is reached or becomes covered or unsafe. In this update only the feasible and profitable goals are kept and new goals are included.

B. Selection of the Current Goal

The point of the list with the best improvement may not be the best choice for the coverage of the entire environment. It is the one that needs the most coverage locally but the path that the robot has to follow to reach it may not be profitable at all. For this reason, we propose a new strategy to select the goal from the list.

The first step is to plan the paths to the goals of the list, $\gamma_{\mathbf{g}_i}(\tau)$, $\forall\, \mathbf{g}_i \in \mathbf{L}_i(k)$, according to (3). To do so, the potential field $v_i(\mathbf{q})$ only has to be calculated once from the current position of the robot for all the goals.

The total improvement of the path per length unit is

$$M_{\gamma_i}(\mathbf{g}_i,k) = \frac{1}{L(\mathbf{g}_i)} \int_{\gamma_{\mathbf{g}_i(0)}}^{\gamma_{\mathbf{g}_i(L(\mathbf{g}_i))}} M_i\big(\gamma_{\mathbf{g}_i}(\tau),k\big)\mathrm{d}\tau,$$

where $L(\mathbf{g}_i) \equiv L(\gamma_{\mathbf{g}_i}(\tau))$ is the length of $\gamma_{\mathbf{g}_i}(\tau)$.

Finally, we select as current goal for robot i the one with the highest value of $M_{\gamma_i}(\mathbf{g}_i,k)$, that is, $\mathbf{g}_i^* = \arg\max_{\mathbf{g}_i \in \mathbf{L}_i(k)} M_{\gamma_i}(\mathbf{g}_i,k)$. In the case that various goals satisfy this equation, we select as current goal the one with the longest L, since it provides the longest plan in time for the robot and therefore the greatest overall improvement. Note that, for every goal, the path is optimal to reach it. However, since the selection is made from a finite set of goals, the path to the selected goal may not correspond to the global optimum among all the possible paths in the partition.

C. Goal Maintenance

For a goal to remain valid throughout the execution of a plan, the robots need to make sure that it is feasible, profitable, not covered and safe. Each robot can easily check every time that the goal satisfies these conditions using $Z(k)$ and $M_i(k)$ and, if not, it selects a new goal (see [12] for more details for more details).

If an obstacle appears on the path to the goal while the robot is following it, the navigation system that we introduce in the next section is in charge of avoiding it. In the case that the deviation from the path is important, a new goal can be selected and a new path planned.

V. Coverage Action Control and Navigation

The last step to complete our persistent coverage approach is the control of the coverage action of the robots and the navigation towards the goals following the planned paths.

A. Coverage Action Control

We propose the following law to control the coverage action. It is based on the difference between the coverage level of the environment and the coverage objective, both in the coverage area of the robot:

$$\rho_i(k) = \begin{cases} \rho_i^{\max}, & \text{if } \rho_i^*(k) \geq \rho_i^{\max}, \\ \rho_i^*(k), & \text{if } 0 < \rho_i^*(k) < \rho_i^{\max}, \\ 0, & \text{if } \rho_i^*(k) \leq 0, \end{cases} \tag{5}$$

with

$$\rho_i^*(k) = \frac{\displaystyle\int_{\mathbf{\Omega}_i(\mathbf{p}_i(k))} \Phi\big(Z^* - dZ(k-1)\big)\alpha_i(k)\,\mathrm{d}\mathbf{q}}{\displaystyle\int_{\mathbf{\Omega}_i(\mathbf{p}_i(k))} \Phi\,\alpha_i(k)^2\,\mathrm{d}\mathbf{q}}.$$

Proposition V.1. *The coverage controller from* (5) *is optimal in terms of the quadratic coverage error,*

$$\tilde{\varepsilon}(k) = \int_{\mathbf{Q}} \Phi\big(Z^* - Z(k)\big)^2 \,\mathrm{d}\mathbf{q}.$$

Proof: We want to find the coverage action that optimizes the quadratic coverage error, i.e., $\rho_i^*(k) = \arg\min_{\rho_i(k)} \tilde{\varepsilon}(k)$. Introducing (1) and recalling that $\alpha_i(k) = 0,\ \forall\, \mathbf{q} \notin \mathbf{\Omega}_i$, this error can be formulated as

$$\tilde{\varepsilon}(k) = \int_{\mathbf{\Omega}_i} \Phi\big(Z^* - dZ(k-1) - \rho_i(k)\alpha_i(k)\big)^2 \mathrm{d}\mathbf{q}.$$

For the rest of the proof we use $\mathbf{\Omega}_i \equiv \mathbf{\Omega}_i(\mathbf{p}_i(k))$ for simplicity. Now, we derive it with respect to $\rho_i(k)$ to find its optima:

$$\frac{\partial \tilde{\varepsilon}(k)}{\partial \rho_i(k)} = \int\limits_{\mathbf{\Omega}_i} -2\,\Phi\big(Z^* - dZ(k-1) - \rho_i(k)\alpha_i(k)\big)\alpha_i(k)\mathrm{d}\mathbf{q}.$$

If we make this derivative equal to zero, we obtain

$$\frac{\partial \tilde{\varepsilon}(k)}{\partial \rho_i(k)} = 0 \Rightarrow \rho_i^*(k) = \frac{\displaystyle\int_{\mathbf{\Omega}_i} \Phi\big(Z^* - dZ(k-1)\big)\alpha_i(k)\mathrm{d}\mathbf{q}}{\displaystyle\int_{\mathbf{\Omega}_i} \Phi\,\alpha_i(k)^2 \mathrm{d}\mathbf{q}},$$

that is the optimum value of $\rho_i(k)$ since

$$\frac{\partial^2 \tilde{\varepsilon}(k)}{\partial \rho_i(k)^2} = \int_{\mathbf{\Omega}_i} \Phi\,\alpha_i(k)^2 \mathrm{d}\mathbf{q} > 0.$$

Finally, we can directly apply the restrictions on the coverage action $0 \leq \rho_i(k) \leq \rho_i^{\max}$. Since the value of $\rho_i^*(k)$ is a minimum and the function $f\big(\rho_i(k), \mathbf{q}\big)$ is a sum of quadratic functions, the value of $\rho_i(k)$ in the restricted interval that gives the minimum $\tilde{\varepsilon}(k)$ is always the closest to $\rho_i^*(k)$.■

The main feature of the coverage action controller is that it allows the robots to maintain the coverage level as close as possible to the objective without overcovering it unboundedly. In comparison to our previous approach this is a significant advantage, as we will show the simulation.

B. Robot Navigation

The navigation and guidance algorithm has to calculate the velocity commands for the robot to follow the path respecting its admissible actions. At the same time, it has to avoid the unmapped obstacles that may appear during the motion and can be detected by laser range sensors on board the robots. To this end, we make use of DWA [20], that computes the velocity command as a function of the previous command, the desired path, the motion and acceleration restrictions of the robot and its current pose and sensor readings.

The implications of this algorithm are the following. When there are no unmapped obstacles in the surroundings of the robot, it follows the planned path at its maximum allowed speed and provides the optimal coverage to the environment. If an obstacle appears, it is avoided while trying to minimize the deviation from the path. The same happens when the path can not be precisely followed due to the robot dynamics. In both cases the coverage along the path is not optimal, due to such deviation. However, thanks to the coverage action controller, the coverage provided at each point is still optimal.

VI. Simulations and Real Experiments

A. Robotic Platform and Simulations Settings

For the validation of this work we used a team of robots (Fig. 1) developed within the MOnarCH Project[1] [4]. The maximum linear velocity of the robots is $u_i^{\max} = 2.5m/s$; the maximum acceleration, $\dot{u}_i^{\max} = 1m/s^2$; and the maximum angular velocity, $\omega_i^{\max} = 600°/s$. Among other sensors, the robots are equipped with two laser range finders that are used for mapping and localization with off-the-shelf ROS packages.

Fig. 1. Team of MOnarCH robots used for the experiments.

Since these robots were not specifically built to develop coverage and they are not equipped with actuators such as vacuuming or heating systems, we simulate the coverage as follows. The coverage area of each robot, $\mathbf{\Omega}_i\big(\mathbf{p}_i(k)\big)$, is a circle of radius equal to the actual radius of the robot, $r_i^{cov} = 0.325m$. The production is defined as constant inside this area with a value of $\alpha_i\big(\mathbf{q}, \mathbf{p}_i(k)\big) = 10$. The coverage of the environments decreases over time with a decay rate $d(\mathbf{q}) = 0.9995$ and the coverage objective is $Z^*(\mathbf{q}) = 100$. Recall that the aim of the team is to keep the coverage of all the points as close as possible to this value.

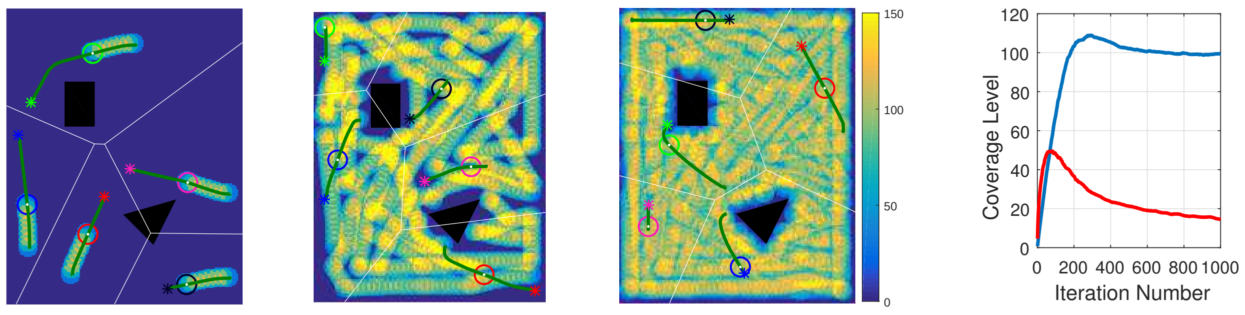
B. Simulation Results

In the first place we introduce simulation results to evaluate the effectiveness of the approach and the scalability of the team. Moreover, we compare the current approach with our previous work [12].

In Fig. 2 we show an example of the persistent coverage of a 10 x 8 m^2, non-convex environment with five robots. At the beginning (Fig. 2a), the robots start covering the environment following almost straight paths to the goals and only avoiding obstacles. When the coverage of the environment increases (Fig. 2b), the planned paths go through the least covered areas to the goals. In the end, when all the point of the environment are well covered (Fig. 2c), the robots keep moving to maintain the coverage level as close to the objective as possible.

The mean coverage level of the environment increases quickly when the robots start moving and reaches a steady

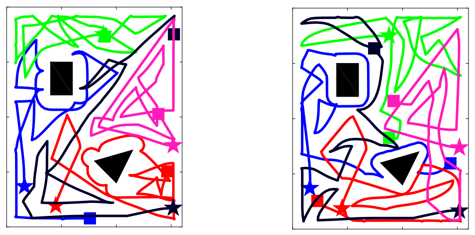
[1]Multi-Robot Cognitive Systems Operating in Hospitals. FP7-ICT-2011-9-601033. Website: http://monarch-fp7.eu/



(a) k = 10. (b) k = 150. (c) k = 500. (d)

Fig. 2. Example of simulation. (a)-(c) Coverage level of the environment with the coverage areas of the robots in different colors for three different time instants. Colored asterisks represent the goals of the robots; green lines, the planned paths of the robots; and white lines, the Voronoi partition of the environment. Obstacles are mapped as black areas. (d) Evolution of the mean coverage level (blue) and its standard deviation (red).

state in around 200 iterations as shown in Fig. 2d. The mean coverage level in the steady-state value is less than 4% away from the objective. Additionally, the standard deviation is around 15 coverage units while it only reaches a maximum value of 48 in the transient.

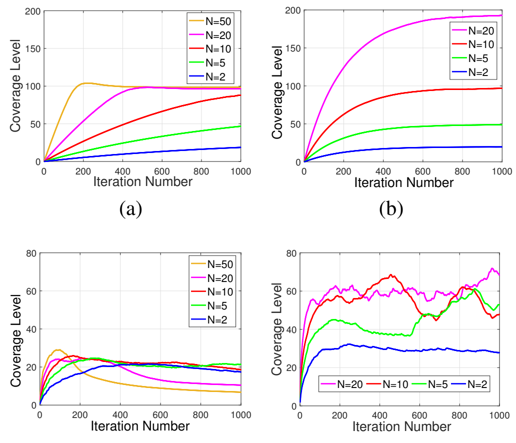


(a) (b)

Fig. 3. Paths followed by the robots during the first 200 iterations (a) using the obstacle speed function and (b) with no obstacle speed function. Stars represent the initial points and squares, the final ones.

Fig. 3a shows the paths followed by the robots in the first 200 iterations with an iteration period of 125 ms. The green and magenta robots cover specific regions while the others exchange regions due to the presence of obstacles in the environment. The paths keep a safety distance to these obstacles to avoid collisions. In contrast, they drive the robots much closer to the obstacles if the obstacle speed function is not included in the planning, as can be seen in Fig. 3b. The average minimum distance to an obstacle in these two cases is 0.94 and 0.72 m, respectively, and the minimum distance is lower than 0.5 m in 21 and 86 out of 200 iterations, respectively. This supports the use of the obstacle speed function to reduce the risk of collisions due to noise in mapping and/or localization.

As mentioned before, in Fig. 4 we test the scalability of our approach and compare it with our previous approach. These simulations have been carried out in a 15 $\times$ 15 m^2 environment to be able to increase the number of robots. Fig. 4a shows that a team of 2 or 5 robots is not capable of reaching on average the objective of $Z^*(\mathbf{q}) = 100$ and produces a higher standard deviation (Fig. 4c) than a team of 20 or 50 robots. These teams reach the objective rapidly with a low standard deviation. One of the main differences with our previous approach is that the coverage action was not controlled as it is in this paper, i.e., it was fixed.



(a) (b)

(c) (d)

Fig. 4. Evolution of the coverage with the number of robots. (a) Mean coverage level of this approach. (b) Mean coverage level with our pevious approach. (c) Standard deviation of the coverage level with this approach. (d) Standard deviation with our previous approach.

Consequently, the mean coverage level increased with the number of robots (Fig. 4a) independently from the objective and only a team of 10 robots maintain the objective level over time. Additionally, the deviation of the coverage level with the previous approach is considerably higher.

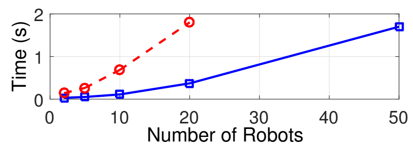
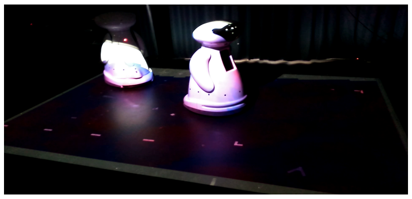
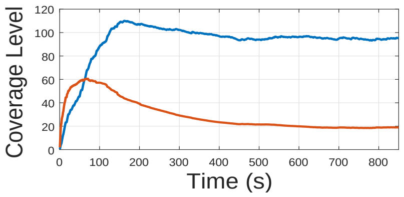


Fig. 5. Evolution of the iteration time with the number of robots. Blue square markers represent the times with this approach and red circular markers, with our previous approach.

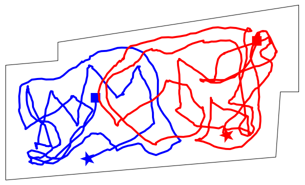
The total computational cost of the simulation has been measured and represented in Fig. 5. Although this time increases more than linearly with the number of robots, with 50 of them it takes less than 35 ms per robot to calculate their control actions. This shows that our approach is scalable and can be applied distributedly. Additionally, we show that this new approach is faster than the previous one, since once the path is planned, the calculation of the motion action is fast compared with the gradient calculated every time in [12].



(a)



(b)



(c)

Fig. 6. (a) Snapshot of one of the experiments. The coverage level of a part of the environment is projected on the ground. Shades of blue represent undercovered points and shades of red, points whose level is around the objective. (b) Evolution of the mean coverage level (blue) and its standard deviation (red). (c) Paths followed by the robots during the first 200 iterations. Stars represent the initial points and squares, the final ones.

C. Experimental Results

The experimental validation of this work was developed in the robotics lab of the Distributed Intelligent Systems and Algorithms Laboratory at EPFL. The environment was almost rectangular region of approximately 15 m^2 where $N = 2$ robots were deployed. In Fig. 6a we show a snapshot of one of the experiments and the complete experiment can be seen in the video included as Supplementary Material.[2] In Fig. 6b the mean coverage level of the environment and its standard deviation are depicted. The evolution is similar to the simulated one and it proves the validity of the approach with real robots. Finally, we show the paths followed by the robots. These paths are more overlapped than the simulated ones due to the limited accuracy of the localization and motion of the robots. Nevertheless, they cover the entire environment with almost no overlapping of the regions covered by each robot.

VII. Conclusions and Future Work

In this paper we have introduced a distributed solution that allows a group of mobile robots to persistently cover an environment in which the coverage decays over time. We have formulated the coverage problem and elaborated on the improvement function to be able to consider the robots as points in the path planning step, where we have introduced a method based on FMM. For this method, we have proposed two speed functions: one to optimize coverage and another one to keep a safety distance to obstacles. We have also presented a method to find and select the coverage goals which the robots have to reach based on the total improvement of the path. To keep the robot following their paths we use a DWA. Additionally, we have proposed a law to control the coverage action of the robots at every time. Finally, we have presented simulation results and real experiments that validate our approach and compare it with our previous one. In fact, the coverage controller provides a significant improvement to reach the coverage objective and the path planning strategy deals with obstacles while optimizing the coverage of the paths.

References

[1] H. Sahin and L. Guvenc, “Household robotics: autonomous devices for vacuuming and lawn mowing,” *IEEE Control Syst. Mag.*, vol. 27, no. 2, pp. 20–96, 2007.

[2] J. Yu, J. Aslam, S. Karaman, and D. Rus, “Anytime planning of optimal schedules for a mobile sensing robot,” in *IEEE Int. Conf. Intell. Robots Syst.*, 2015, pp. 5279–5286.

[3] A. Al-Wazzan, R. Al-Farhan, F. Al-Ali, and M. El-Abd, “Tour-guide robot,” in *Int. Conf. Ind. Informatics Comput. Syst.*, 2016, pp. 1–5.

[4] J. Messias, R. Ventura, P. Lima, J. Sequeira, P. Alvito, C. Marques, and P. Carri, “A robotic platform for edutainment activities in a pediatric hospital,” in *Int. Conf. Auton. Robot Syst. Compet.*, 2014, pp. 193–198.

[5] S. L. Smith, M. Schwager, and D. Rus, “Persistent robotic tasks: monitoring and sweeping in changing environments,” *IEEE Trans. Robot.*, vol. 28, no. 2, pp. 410–426, 2012.

[6] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, “Coverage control for mobile sensing networks,” *IEEE Trans. Robot. Autom.*, vol. 20, no. 2, pp. 243–255, 2004.

[7] Y. Diaz-Mercado, S. G. Lee, and M. Egerstedt, “Distributed dynamic density coverage for human-swarm interactions,” in *Amer. Control Conf.*, 2015, pp. 353–358.

[8] J. M. Palacios-Gasós, E. Montijano, C. Sagüés, and S. Llorente, “Multi-robot persistent coverage with optimal times,” in *Conf. Decision and Control*, 2016, pp. 3511–3517.

[9] P. Amstutz, N. Correll, and A. Martinoli, “Distributed boundary coverage with a team of networked miniature robots using a robust market-based algorithm,” *Ann. Math. Artificial Intell.*, vol. 52, no. 2-4, pp. 307–333, 2008.

[10] C. Franco, G. López-Nicolás, C. Sagüés, and S. Llorente, “Adaptive action for multi-agent persistent coverage,” *Asian J. Control*, vol. 18, no. 2, pp. 419–432, 2016.

[11] C. Franco, D. M. Stipanović, G. López-Nicolás, C. Sagüés, and S. Llorente, “Persistent coverage control for a team of agents with collision avoidance,” *European J. Control*, vol. 22, pp. 30–45, 2015.

[12] J. M. Palacios-Gasós, E. Montijano, C. Sagüés, and S. Llorente, “Distributed coverage estimation and control for multi-robot persistent tasks,” *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1444–1460, 2016.

[13] S. Bochkarev and S. L. Smith, “On minimizing turns in robot coverage path planning,” in *IEEE Conf. Autom. Sci. Eng.*, 2016, pp. 1237–1242.

[14] X. Zheng, S. Koenig, D. Kempe, and S. Jain, “Multirobot forest coverage for weighted and unweighted terrain,” *IEEE Trans. Robot.*, vol. 26, no. 6, pp. 1018–1031, 2010.

[15] Y. Gabriely and E. Rimon, “Spanning-tree based coverage of continuous areas by a mobile robot,” *Ann. Math. Artificial Intell.*, vol. 31, no. 1/4, pp. 77–98, 2001.

[16] D. Portugal, C. Pippin, R. P. Rocha, and H. Christensen, “Finding optimal routes for multi-robot patrolling in generic graphs,” in *IEEE Int. Conf. Intell. Robots Syst.*, 2014, pp. 363–369.

[17] X. Lin and C. G. Cassandras, “An optimal control approach to the multi-agent persistent monitoring problem in two-dimensional spaces,” in *IEEE Conf. Decision and Control*, 2013, pp. 6886–6891.

[18] X. Lan and M. Schwager, “Planning periodic persistent monitoring trajectories for sensing robots in gaussian random fields,” in *IEEE Int. Conf. Robot. Autom.*, 2013, pp. 2407–2412.

[19] J. A. Sethian, “Fast Marching Methods,” *SIAM Review*, vol. 41, no. 2, pp. 199–235, 1999.

[20] W. B. D. Fox and S. Thrun, “The dynamic window approach to collision avoidance,” *IEEE Trans. Robot. Autom.*, vol. 4, p. 1, 1997.

[21] R. Ventura and A. Ahmad, “Towards Optimal Robot Navigation in Domestic Spaces,” in *RoboCup 2014: Robot World Cup XVIII*. Springer, 2015, pp. 318–331.

[2]The video and more information about the project can be found at http://disal.epfl.ch/research/SocialRoboticsNavigation