Citation: C. Franco, D. Paesa, G. López-Nicolás, C. Sagüés, and S. Llorente. Hierarchical strategy for dynamic coverage. In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2012), pp. 5341-5346, Vilamoura, Algarve, Portugal, October 7-12, 2012

# Hierarchical strategy for dynamic coverage

C. Franco, D. Paesa, G. Lopez-Nicolas, C. Sagues and S. Llorente

*Abstract*— **This paper is focused on dynamic coverage control with a team of robots. In this framework, decentralized control algorithms have been investigated to deal with the efficient coordination of the resources. The main contribution is a novel global strategy based on a hierarchical grid decomposition of the domain. This decomposition allows an ordered coverage of the domain that combined with a gradient based control law of the local error, achieves a better performance than previous approaches of dynamic coverage. The total coverage of the domain is proven, and the good performance of the approach is supported with simulations.**

## I. INTRODUCTION

The coverage, interpreted as giving service to an area, is of interest in a wide variety of applications such as: demining, cleaning, lawn mowing, painting, etc. Due to the evolution of wireless communication, in the last decade numerous researchers have focused on coverage developed by multiple mobile robots. A group of robots working together would perform better than a single one would do. However, to exploit the benefits of multi-robot systems is necessary to solve technical challenges involving the efficient coordination of the resources. Here, we focus on problems in which a team of robots is moving continuously in a coordinated way to cover the domain.

Many topics related to coverage can be found in the literature. If the resources or robots are static, the problem is known as location-allocation of resources. It is an interesting problem that has many applications. The first paper dealing with allocation of resources dates from 1909 when Weber studied the optimal location of industries in a region [1]. Since then, location optimization problems have been studied and reviewed from different points of view [2], [3], [4].

In the last years, authors have started to consider mobile resources, and variable and unknown environment. In this way, teams of mobile robots can adapt their locations to environmental changes or robot failures. They also can move periodically to cover a bigger area achieving a better performance than static robots. This problem is referred to as area coverage and, although multiple applications are possible, literature is mainly focused on sensing. The coverage problem is usually formulated by means of an optimization function to minimize/maximize some performance index in a decentralized manner. Several approaches have been developed to face the optimization function: Voronoi partitions [5], [6] potential fields [7], [8] gradient-based approaches [9], [10], [11].

This locational optimization problem can also be considered as static [5], [9], or dynamic [10], [12]. In the static case, both, the density function and the final position of sensors are static, which is similar to the facility location problem. The dynamic case is far wider: there are path planning algorithms to cover areas bigger than the capacity of the sensors [12], patrolling algorithms to visit periodically the points of space [13], [14], and dynamic coverage algorithms that are able to adapt to an evolving environment [10], [11].

Here, we focus on the motion control of a team of robots for dynamic coverage. The objective is to cover coordinately a domain $D_x$ until a level $\Lambda^*$. The formulation is based on the ideas introduced in [10], that are applicable to multiple coverage problems. There, the authors propose a control law that switches between two actions: a covering action, which is a gradient type kinematic control strategy to develop coverage in the direction of the maximum benefit obtained at each instant; and a perturbation action, which is used when the robot is trapped in local minima, and consists in a linear feedback control law to direct the robot towards the nearest point that is not covered yet. Here, we propose a motion strategy that weights continuously local and global strategies, instead of switching, to achieve an efficient coverage of the domain. The local strategy is also based on the gradient, but we propose a new approach for the global strategy. This is the main contribution and it is based on a hierarchical grid decomposition of the domain. It is inspired by ideas that have been used in path planning algorithms for finding a free path in environments with obstacles [15] but, to the best of our knowledge, this is the first time that are proposed to deal with dynamic coverage problems. In the field of path planning algorithms, [16] also uses a hierarchical approach, but based on Voronoi diagrams of wide and narrow spaces. Our approach allows an ordered coverage of the domain in a simple way, achieving the coverage objective with a high efficiency. This proposal is decentralized, and each robot moves according to its coverage information that consists of their own coverage information and the information shared with others in case they meet during their motion. In addition, the achievement of the objective using our proposal is proven and simulation results are also included to evaluate the effectiveness of the approach.

The paper is organized as follows: Section II introduces the problem formulation and the local strategy. Section III

This work was supported by Ministerio de Ciencia e Innovación/Unión Europea, DPI2009-08126, by project IPT-2011-1158-920000 of subprogram INNPACTO from Ministerio de Economía y Competitividad, by DGA-FSE (grupo T04), and by grant B139/2010 by DGA.

C. Franco, G. Lopez-Nicolas and C. Sagues are with the Instituto de Investigación en Ingeniería de Aragón (I3A), Universidad de Zaragoza, Mariano Esquillor s/n, 50018, Zaragoza, España. cfranco@unizar.es

D. Paesa and S. Llorente are with the Research and Development Department, Induction Technology, Product Division Cookers, BSH Home Appliances Group, 50016, Zaragoza, España.

presents the strategy to select global objectives based on a hierarchical grid decomposition. Section IV presents and discusses simulation results.

## II. Problem formulation

In this section, we propose a new control law for dynamic coverage tasks developed by a team of robots. The objective is to reach a desired coverage level $\Lambda^*(x) > 0$ of all the points $x \in D_x$ over a bounded domain $D_x \subset \mathbb{R}^n$. We assume that the motion of the robots is holonomic, and then $\dot{p}_i(t) = u_i$ for each robot $A_i$ of the team $A$, where $i = 1, ..., N$. $p_i(t)$ is the position of each robot in a convex domain $D_p \subset \mathbb{R}^n$ and $u_i$ is the input to the robots. Note that $D_p$ can be different to $D_x$: robots can pass through points that do not need to be covered, and due to the range of the actuator, it is possible to give service to points not reachable by the robot. In any case, the points of $D_x$ must keep a distance lower than the actuator range with the points of $D_p$ to ensure the coverage of the domain.

Let us introduce the coverage developed by a robot $\Lambda_i(t, x)$, which is obtained by integrating the covering action $\alpha_i(r)$ over time $t$:

$$\Lambda_i(t, x) = \int_0^t \alpha_i(r) dt. \quad (1)$$

Here, $r = \|x - p_i(t)\|$. In this work, we restrict the actuators to those with positive covering action in its domain $\Omega_i$ and null outside, i.e., $\alpha_i(r) > 0 \quad \forall x \in \Omega_i$, and $\alpha_i(r) = 0 \quad \forall x \notin \Omega_i$. We assign $\Lambda_i(0, x) = 0 \quad \forall x \in D_x$, which means that at the beginning, points are not serviced at all. The total coverage of the team is computed as the sum of the coverage action of each robot $\Lambda(t, x) = \sum_{i \in A} \Lambda_i(t, x)$, although other consensus schemes could be used instead. Here, we introduce the coverage lack $\Upsilon_{i_x}(t)$ of each robot over a point $x$ at time $t$ as:

$$\Upsilon_{i_x}(t) = 1 - \frac{\Lambda_i(t, x)}{\Lambda^*(x)}. \quad (2)$$

We also introduce the global coverage lack:

$$\Upsilon_x(t) = 1 - \frac{\Lambda(t, x)}{\Lambda^*(x)}. \quad (3)$$

We will focus on covering problems where the excess of coverage is not harmful. Therefore, we introduce a positive semidefinite penalization function of the coverage lack: $0 < \mathcal{P}(\Upsilon_x(t)) \leq 1 \ \forall \Upsilon_x(t) > 0$ and $\mathcal{P}(\Upsilon_x(t)) = 0 \quad \forall \Upsilon_x(t) \leq 0$, being $\partial \mathcal{P}(\Upsilon_x(t)) / \partial \Upsilon_x(t) \geq 0$. Moreover, we introduce $\Phi(x) \in (0, 1] \quad \forall x \in D_x$ as the priority to cover each point $x$ at time $t$. $\Phi(x)$ is a map that weights the interest of the points in the domain to give more priority to determined zones.

We consider the error function of the whole domain:

$$e_{D_x}(t) = \frac{\int_{D_x} \mathcal{P}(\Upsilon_x(t)) \Phi(x) dx}{\int_{D_x} \Phi(x) dx}, \quad (4)$$

and the error function of the actuator domain of each robot:

$$e_{\Omega_i}(t) = \frac{\int_{\Omega_i} \mathcal{P}(\Upsilon_{i_x}(t)) \Phi(x) dx}{\int_{\Omega_i} \Phi(x) dx}. \quad (5)$$

The objective of the control law is to drive the error $e_{D_x}(t)$ to 0. We propose a control law with two components: one local, $u_i^L(t)$, that depends on the coverage error of servicing range of the robot, and $u_i^G(t)$, that depends on the level of coverage of the whole domain. The local component is computed with:

$$u_i^L(t) = \int_{\Omega_i} \mathcal{P}(\Upsilon_{i_x}(t)) \Phi(x) \frac{\partial \alpha_i(r)}{\partial r} (p_i(t) - x) dx. \quad (6)$$

Its value depends on the points in the actuator domain, and is a gradient type function that directs the robots towards the points less serviced. Such a control law is efficient to cover the neighborhood of the robot since it drives the robots towards the direction of maximum coverage benefit. However, an important drawback is that gradient strategies fall in local minima and stop the robots if there are symmetries (e.g. null coverage or total coverage). To ensure the total coverage of the domain, a local control law that depends only on the actuator domain is not enough. To avoid blockages we propose to add a control component that allows continuing the coverage of the domain by leaving the symmetrically covered zone. We suggest a global component that depends on the whole coverage map and that directs the robots towards a position $p_i^{obj}(t) \in D_p$ from where it can cover non covered points. The strategy to select global objectives $p_i^{obj}(t)$, and the control law to reach the objectives are developed in the following sections. We present now $u_i^G(t)$ as a general function with the requirement of driving the robots to a position from where they can cover non covered points.

$$u_i^G(t) = f(p_i(t) - p_i^{obj}(t)), \quad (7)$$

Another important drawback of gradient strategies in this kind of problems is the value of its module when the neighborhood of the robot is almost covered. In this case, the gradient is very low and the robot tends to slow down until it stops, being trapped in an almost covered zone as if it were quicksand. To overcome this drawback, and given that the strong point of the gradient is its direction, we propose to combine local actions with global actions with an scheme that extracts their directions and weights them continuously. The weights depend on the coverage error over the coverage domain of each robot $e_{\Omega_i}(t)$ and, once they are obtained, give more importance to the local coverage control law if the error over the coverage domain is high, and more importance to global coverage control law when the benefit of developing coverage in the surrounding space of the robot is small. In symmetrically covered zones, as the gradient is null, the local direction is $\overrightarrow{0}$ and the robot is governed by the global action. The local weight $W_i^L$ and the global weight $W_i^G$ are obtained with:

$$W_i^L(t) = e_{\Omega_i}^{\beta}(t) \quad (8)$$

$$W_i^G(t) = 1 - e_{\Omega_i}^{\beta}(t) \quad (9)$$

where $\beta \in \mathbb{R}^+$ is a parameter that emphasizes the global strategy if $\beta > 1$ or the local one if $\beta < 1$. Furthermore, the global term of the equation is multiplied by a gain $k_i^G(d_i^{obj}(t)) \in [0,1]$. This gain depends on the distance of the robot to the objective $d_i^{obj}(t) = \|p_i(t) - p_i^{obj}(t)\|$, with $k_i^G(d_i^{obj}(t)) = 0$ if $d_i^{obj}(t) = 0$ and $\partial k_i^G(d_i^{obj}(t))/\partial d_i^{obj}(t) > 0$. The global term vanishes when the robot reaches the objective and it grows as the robot goes away from the objective. Consequently we propose:

$$\hat{u}_i(t) = W_i^L(t)\hat{u}_i^L(t) + k_i^G(d_i^{obj}(t))W_i^G(t)\hat{u}_i^G(t), \quad (10)$$

Finally, this action, whose module is less or equal 1 is multiplied by a constant gain $k_i$ that represents the maximum velocity of each robot and by $(1-e_{\Omega_i}(t))$. In this way, when the local error is close to 1, the robot slows down to achieve the coverage objective of $\Omega_i$ in the first try. As a result, the path length and the time to completion is reduced because it is not needed to cover twice the same points. Otherwise, when a robot falls into an almost covered zone, the robot speeds up to leave it and to arrive rapidly to an uncovered zone reducing also the time to completion. The proposed control law $u_i(t)$ is then:

$$u_i(t) = k_i(1 - e_{\Omega_i}(t))\hat{u}_i(t). \quad (11)$$

In our proposal, each robot $A_i$ has a collection of coverage maps $\{M_{c_i}(t_i) | i \in A\}$ of all the robots of the team with a time label $t_i$. This coverage map contains the amount of coverage developed by each robot in each point of the domain. Each robot $A_i$ updates continuously its own map $M_{c_i}(t_i)$. Besides, when it meets with other robots their information is shared. Two robots meet if their distance is lower than a determined communication radius $r_c$. The shared information between two robots that have met is not only their respective maps, but also the last updated maps of other robots. Finally, each robot computes a coverage action with the merging of information of all the maps in its memory. In our proposal, the merging of information is simply made by the sum of the amount of coverage of each robot as explained before.

Next, we demonstrate that the coverage task is fulfilled by using the algorithm proposed:

*Proposition 2.1:* The control strategy from (6) to (11) drives the coverage error of the domain $e_{D_x}(t) \to 0$ as $t \to \infty$.

*Proof:* The coverage error of the domain is a bounded positive semidefinite function by definition since $0 < \mathcal{P}(\Upsilon_x(t)) \leq 1$, $\Phi(x) \in (0,1]$ and therefore:

$$0 \leq \frac{\int_{D_x} \mathcal{P}(\Upsilon_x(t))\Phi(x)dx}{\int_{D_x} \Phi(x)dx} \leq 1 \quad (12)$$

The time derivative of the penalization inside the error is:

$$\tilde{e}_{D_x}(t) = \int_{D_x} \frac{\partial \mathcal{P}(\Upsilon_x(t))}{\partial t}\Phi(x)dx$$

$$= \int_{\Omega} \frac{\partial \mathcal{P}(\Upsilon_x(t))}{\partial \Upsilon_x(t)} \frac{\partial \Upsilon_x(t)}{\partial \Lambda(t,x)} \frac{\partial \Lambda(t,x)}{\partial t}\Phi(x)dx \quad (13)$$

As the robot can cover only the domain $\Omega_i$, the error of the whole domain $e_{D_x}$ varies only in $\Omega = \bigcup_{i=1}^N \Omega_i$. Analyzing each term of (13) separately we have:

$$\frac{\partial \mathcal{P}(\Upsilon_x(t))}{\partial \Upsilon_x(t)} \geq 0, \quad (14)$$

$$\frac{\partial \Upsilon_x(t)}{\partial \Lambda(t,x)} = -\frac{1}{\Lambda^*(x)} < 0, \quad (15)$$

$$\frac{\partial \Lambda(t,x)}{\partial t} = \sum_{i=1}^N \alpha_i(r) > 0, \quad (16)$$

$$0 < \Phi(x) \leq 1. \quad (17)$$

Where $r$ depends on the position of the robot $p_i(t)$, which is computed with the control law (11) with:

$$p_i(t) = p_i(0) + \int_0^t u_i(t)dt. \quad (18)$$

As (14) is positive semidefinite, (15) is negative definite, and (16), (17) are positive definite, $\tilde{e}_{D_x}(t) \leq 0$ with $\tilde{e}_{D_x}(t) = 0$ only if the actuator domain is totally covered. This happens if the whole domain $D_x$ has been covered, and then $e_{D_x}(t) = 0$. And also at some time $t_b$ if $\Upsilon_x(t_b) = 0 \; \forall x \in \Omega$ but $\Upsilon_x(t_b) > 0$ for some $x \in D_x$, i.e., if the domain of the actuator of the robots is covered but some point of the domain is not covered yet. Consequently, there will be global objectives and

$$u_i(t_b) = k_i\hat{u}_i^G(t_b)k_i^G(d_i^{obj}(t_b)). \quad (19)$$

This is a linear feedback control law that, by requirement of (7), directs the robot towards global objectives from where the robot can cover not totally covered regions. This implies that, in a finite time, some robot will leave the totally covered zone that causes $\dot{e}_{D_x} = 0$, and as a consequence $\dot{e}_{D_x} < 0$ in $t = t_b + \epsilon$ with $\epsilon < \infty$. ■

## III. Hierarchical Grid Strategy

In this section, we propose a strategy to select global objectives. It is based on the division of the domain in a hierarchical grid of $J$ levels. The procedure consists in dividing the space into $(2^n)^J$ equal cells, being $n$ the dimension of the domain, and covering them hierarchically from the lowest level to the highest. Let us define $\Psi_j = \{\psi_{j1}, \psi_{j2}, ..., \psi_{jl}\}$ as the collection of cells of level $j = 1, ..., J$ with $l = (2^n)^j$, and $\pi_{ijl}$ as the centroid of the non covered points of each cell $\psi_{jl}$. The number of levels $J$ is determined by iteratively dividing the space, until the length of the diagonal of the cell of the last level $D_J$ is smaller than two times the actuator range of the robot ($R$), which depends on the actuator domain $\Omega_i$ of the sensor:

$$D_j = \sqrt{\sum_{i=1}^n \left(\frac{L_i}{2j}\right)^2} < 2R \quad (20)$$

with $L_i$ the size of each dimension of the domain. In this way, we ensure that from any $\pi_{ijl}$ the robot can cover the points of $\psi_{jl}$ that are not covered and does not get blocked. Each $\psi_{jl}$ is then divided into $2^n$ cells until the last level is reached. An example of the division procedure of a square

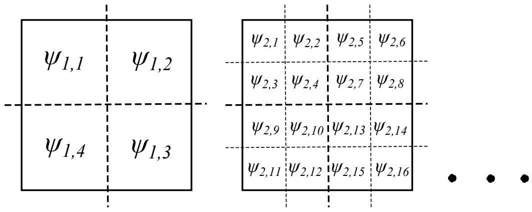


Fig. 1. First and second level of division of a square space.

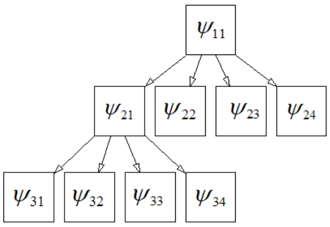


Fig. 2. Hierarchical objectives tree.

domain is shown in Fig. 1. Let us define the coverage error $e_{i_{\psi_{jl}}}(t)$ of each cell $\psi_{jl}$ for each robot $i$ as:

$$e_{i_{\psi_{jl}}}(t) = \frac{\int_{\psi_{jl}} \mathcal{P}(\Upsilon_{i_x}(t))\Phi(x)dx}{\int_{\psi_{jl}} \Phi(x)dx}. \quad (21)$$

For a given robot $i$ and level $j$, the choice of the objective $p_i^{obj}(t)$ is done with a criterion that weights distance and coverage error of the cells.

$$p_i^{obj}(t) = \left\{\pi_{ijl} : max \left\{1 - \frac{\|p_i(t) - \pi_{ijl}\|}{r_{max}} + e_{i_{\psi_{jl}}}(t)\right\}\right\}, \quad (22)$$

The global objective selection is now explained by means of an example for better understanding. Let us suppose that robot $i = 4$ starts the covering process in a hierarchical grid of 3 levels with the scheme in Fig. 2, at a position belonging to cells $\psi_{11}, \psi_{21}, \psi_{32}$. Then, it selects its global objective as $\pi_{432}$. Once that cell is covered, the robot will check whether its parent cell ($\psi_{21}$ in this case) is covered. If not, the next objective will be one of their remaining children cells ($\psi_{31}, \psi_{33}, \psi_{34}$) according to (22). In this example $j = 3$, $l = 1, 3, 4$. When all the children cells of $\psi_{21}$ have been covered, $\psi_{21}$ will be totally covered and then, the robot will check whether cell $\psi_{11}$ is covered. If not, the selection process will be repeated for level two between the children of $\psi_{11}$, and once level two is assigned, for their children of level three. However, if cell $\psi_{11}$ is totally covered, the selection process will start for the rest of cells of level 1, $\psi_{12}, \psi_{13}, \psi_{14}$, and will continue for the children of the selected cell. In this way, a robot will perform the coverage cell by cell hierarchically, covering all the children cells before selecting another parent. The main advantage of this approach is that it saves energy, given that the dynamic covering process is hierarchically ordered by zones.

## IV. Simulation Results

In this section, we present simulation results using the control algorithm proposed. For the following example, we

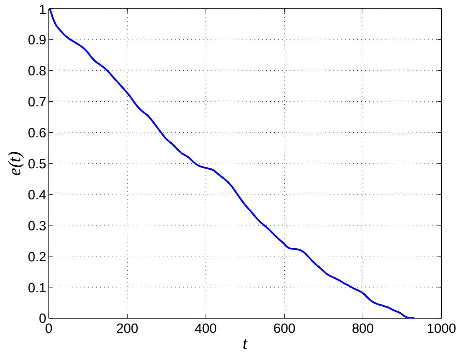


Fig. 3. Normalized coverage error evolution of the simulation. The total coverage is achieved at t=896.

will focus on $D_x = D_p \subset \mathbb{R}^2$. The domain is 100x100 units square area with a coverage objective $\Lambda^*(x) = 100$ and $\Phi(x) = 1 \ \forall x \in D_x$, and a team of 4 robots. We select $\beta = 1/2$, $k_i = 1$, and

$$k_i^G(d_i^{obj}(t)) = tanh\left(\frac{2d_i^{obj}(t)}{R}\right). \quad (23)$$

We also define,

$$\mathcal{P}(\Upsilon_x(t)) = \begin{cases} 0 & \Upsilon_x(t) < 0 \\ \frac{1}{2}(1 - \cos(\pi\Upsilon_x(t))) & 0 \le \Upsilon_x(t) \le 1 \\ 1 & \Upsilon_x(t) > 1 \end{cases} \quad (24)$$

The algorithm has been tested with several actuator models and here, we present simulations with the following model:

$$\alpha(\hat{r}) = \begin{cases} \alpha_M & if \quad \hat{r} < r_m, \\ \alpha_M \left(1 + 2\left(\frac{\hat{r}-r_m}{1-r_m}\right)^3 - 3\left(\frac{\hat{r}-r_m}{1-r_m}\right)^2\right) & \\ & if \quad r_m \le \hat{r} \le 1, \\ 0 & if \quad \hat{r} > 1, \end{cases} \quad (25)$$

where $\alpha_M$ is the maximum action, $\hat{r} = \|x - p_i(t)\|/R$, being $R$ the total actuator range, and $r_m$ is the actuator percentage of the range where the action is maximum. This function may model the behavior of laser sensors, demining, lawn mowing or cleaning robots and also, for instance, an aerial vehicle with a camera onboard pointing downwards for exploration or surveillance. In the simulations we use $\alpha_M = 5$ and $r_m = 0.5$ and $R = 10$. In the experiment, when two robots meet and have the same global objective cell $\psi_{jl}$, the farther starts a new global objective selection from level 1 to increase the coverage efficiency. Here, we have considered a communication radius $r_c = 20$ where robots are able to share information. We have carried out extensive simulations with robots starting at random positions. Fig. 3 shows the evolution of the error and Fig. 4 the action of each robot in one of the simulations.

In Fig. 5, the coverage map of each robot at $t = 1, 200, 600$ is shown. Robots 2 and 3 start together and share their coverage information. Afterwards, around $t = 200$, robots 2 and 4 meet, share their maps and therefore have a similar coverage map. Around $t = 600$ robots 1 and 3 have also

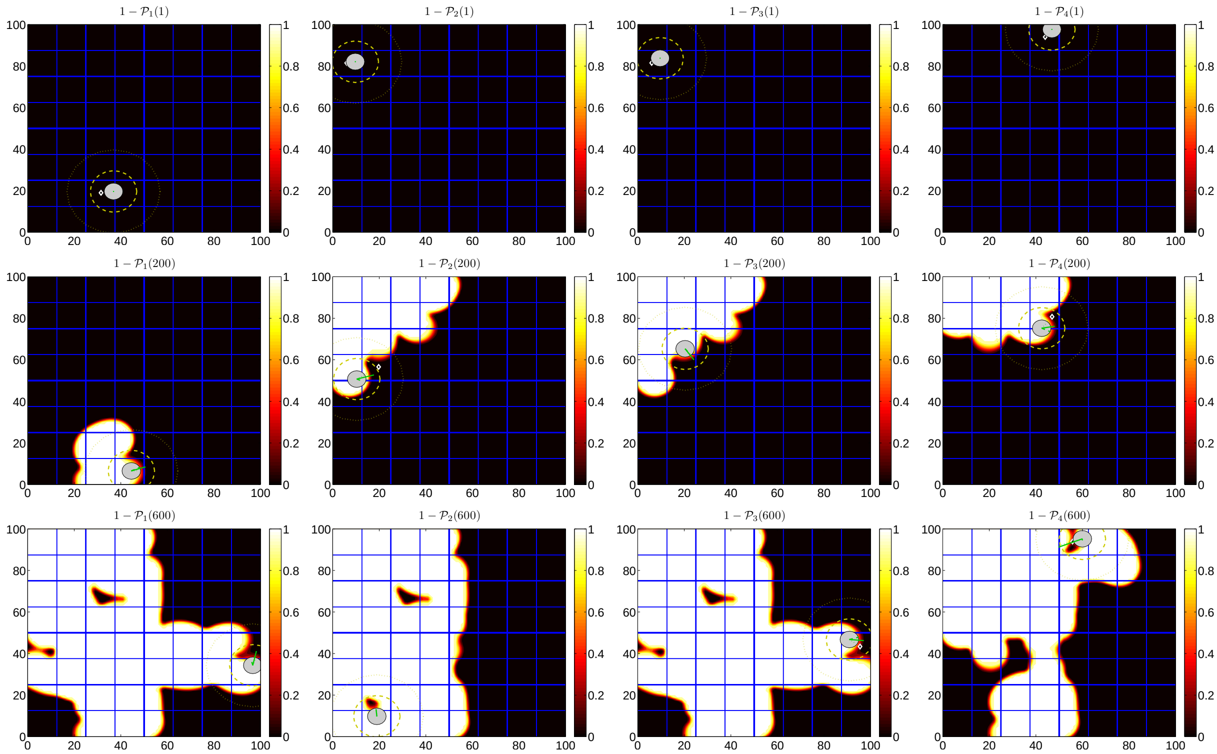


Fig. 5. Penalization map of each of the four robots of the team in different times. The penalization map at t=1, t=200, t=600 of each robot is shown in the first, second and third row respectively. Small circles represent the position of the robots, the coverage domain is represented by a thick dashed line and the communication range by a thin yellow dotted line. Continuous straight line represents the total action. The global objectives are represented by a rhombus.

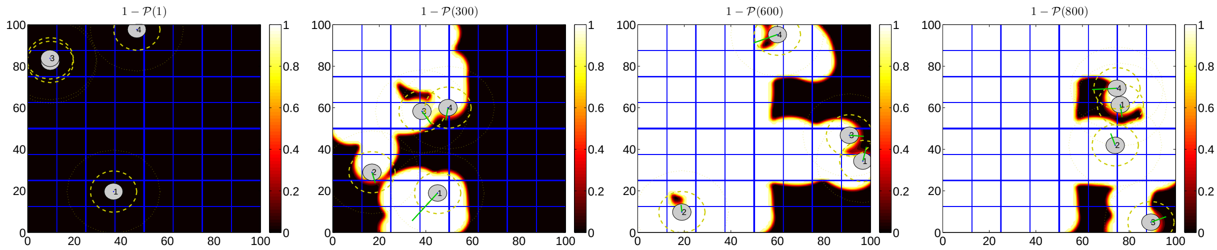


Fig. 6. Evolution of the global coverage map throughout the coverage process. Small circles represent the position of the robots, the coverage domain is represented by a thick dashed line and the communication range by a thin dotted line. Continuous straight line represents the total action.

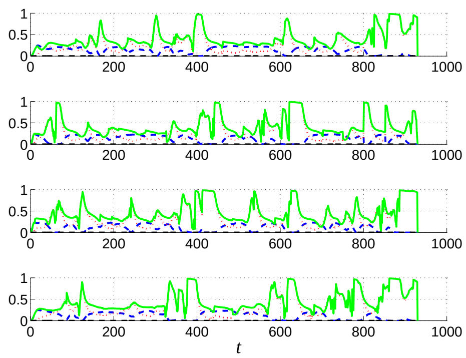


Fig. 4. Evolution of the action of each one of the 4 robots. Continuous line represents $u_i(t)$, dotted line represents the global component of $u_i(t)$, $\hat{u}_i^G(t)k_i^G(d_i^{obj}(t))W_i^G(t)(1-e_{\Omega_i}(t))k_i$, and dashed line represents its local component $\hat{u}_i^L(t)W_i^L(t)(1-e_{\Omega_i}(t))k_i$. These components are obtained introducing (10) into (11).

met, and therefore have also similar coverage maps. The attached video shows the evolution of the coverage map of each agent during the whole process. In Fig. 6, the global coverage map evolution of the team of robots is shown. The domain is rather covered at t=800 and it is totally covered at t=896. Finally, we carried out some simulations to compare the hierarchical strategy shown and a nearest centroid strategy with $\beta = 1/10$. The nearest centroid strategy selected consists in selecting as global objective the nearest non covered cell. If two robots meet and share the same global objective, the farthest one searches another different global objective. The value $1/10$ of the parameter $\beta$, makes the global action negligible until the domain of the actuator of the robot is almost covered. As a result, with these two changes we achieved a strategy to compute direction of motion very similar to [17]. There, a control law based on the gradient is used until the robot is blocked and then,

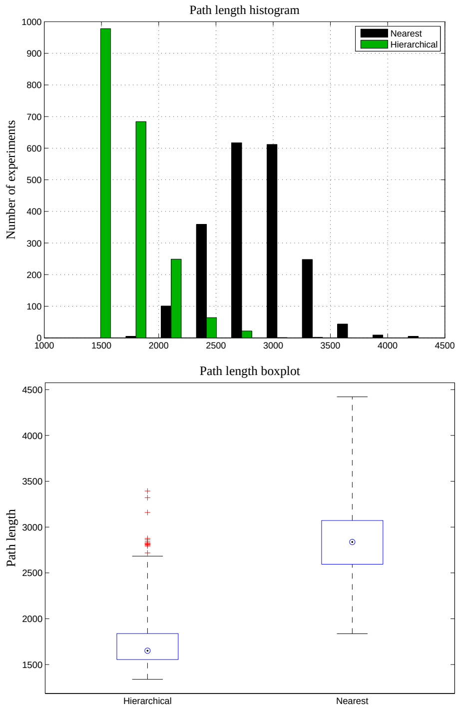


Fig. 7. Histogram and box plot of 2000 experiments comparing our hierarchical strategy with nearest strategy. In the diagram, the x-axis represents different path lengths while the y-axis represents the number of executions resulting with the corresponding path length. The box plot shows the average of the data with a dot inside a circle, the limits of the box are the 25th and the 75th percentiles, the whiskers are 3 interquartile length and the crosses are the outliers.

a perturbation control law that directs towards the nearest non covered point is used. We carry out 2000 experiments starting at random positions and we compared the sum of the path lengths of the team of robots as a power consumption measurement. Fig. 7 shows the histogram and the box plot of the data. The average of the hierarchical approach is 1730 units of distance, whereas the average of the nearest approach is 2838, a 64% higher. Moreover, near the 75th percentile of the path lengths of our hierarchical approach are below the path length of the nearest approach. Due to the reduction in the path length, there is also a reduction in the time to completion. The media of the 2000 experiments for our hierarchical strategy is 989 units of time whereas the average of the nearest approach is 1278, a 30% higher.

## V. Conclusion

In this paper, we have proposed a new control algorithm for the dynamic coverage of a domain developed by a team of robots. The proposed approach has been tested through simulations and compared with the nearest uncovered point strategy implemented in previous dynamic coverage algorithms. The experimental results show the better performance of our approach, which is based on a new strategy to select global objectives that consist in a hierarchical grid decomposition of the map. We have proposed the combination of the hierarchical grid decomposition with a gradient based control law through an scheme that weights them depending on the local coverage error. It gives more importance to local objectives when the local error is high, and to global objectives when the benefit of developing the coverage in the neighborhood of the robot is small, solving the problem of local minima. Our proposal is applicable to most coverage problems straightforwardly because is based on normalized error and action functions. The normalized action function is regulated up to the maximum velocity of a robot introducing in a natural way the saturation of real systems. Furthermore, we achieve an efficient coverage slowing down the robot when the error is high to ensure total coverage of the neighborhood, and speeding up the robot when the error is low to leave covered zones and to arrive rapidly to an uncovered zone.

## References


[1] A. Weber [translated by Carl J. Friedrich from Weber's 1909 book]. *Theory of the Location of Industries*. The University of Chicago Press, Chicago, 1929.

[2] L. Cooper. Location-allocation problems. *Operation Research*, 11:331–343, 1963.

[3] A. Ghosh and G. Rushton. *Spatial Analysis and Location-Allocation Models*. Van Nostrand Reinhold, New york, 1987.

[4] Z. Drezner. *Facility location: A survey of Applications and Methods*. Springer-Verlag, New york, 1995.

[5] J. Cortés, S. Martínez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, 20(2):243–255, 2004.

[6] McLurkin J. Rus D. Schwager, M. Distributed coverage control with sensory feedback for networked robots. *In Proceedings of Robotics: Science and Systems*, 2006. cited By (since 1996) 1.

[7] D. O. Popa, C. Helm, H. E. Stephanou, and A. C. Sanderson. Robotic deployment of sensor networks using potential fields. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, pages 642–647, 2004.

[8] M. J. Mataric, A. Howard, and G. S. Sukhatme. Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem. In *Proceedings of the 6th International Symposium on Distributed Autonomous Robotics Systems*, 2002.

[9] W. Li and C. G. Cassandras. Distributed cooperative coverage control of sensor networks. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 2542–2547, 2005.

[10] I. I. Hussein and D. M. Stipanović. Effective coverage control for mobile sensor networks with guaranteed collision avoidance. *IEEE Transactions on Control Systems Technology*, 15(4):642–657, 2007.

[11] S. K. Gan and S. Sukkarieh. Multi-UAV target search using explicit decentralized gradient-based negotiation. In *2011 IEEE International Conference on Robotics and Automation*, pages 751 – 756, may 2011.

[12] H. Choset. Coverage for robotics - a survey of recent results. *Annals of Mathematic and Artificial Intelligence*, 31(1-4):113–126, 2001.

[13] Y. Elmaliach, N. Agmon, and G.A. Kaminka. Multi-robot area patrol under frequency constraints. In *IEEE International Conference on Robotics and Automation,*, pages 385 –390, april 2007.

[14] S. L. Smith and D. Rus. Multi-robot monitoring in dynamic environments with guaranteed currency of observations. In *49th IEEE Conference on Decision and Control (CDC)*, pages 514 –521, december 2010.

[15] J. C. Latombe. *Robot Motion Planning*. Kluwer Academic Publications, Boston (MA), 1991.

[16] E. U. Acar, H. Choset, and P. N. Atkar. Complete sensor-based coverage with extended-range detectors: a hierarchical decomposition in terms of critical points and voronoi diagrams. In *International Conference on Intelligent Robots and Systems*, volume 3, pages 1305 –1311, 2001.

[17] I. I. Hussein and D. M. Stipanovic. Effective coverage control using dynamic sensor networks. In *45th IEEE Conference on Decision and Control*, pages 2747 –2752, dec. 2006.