Anisotropic Vision-Based Coverage Control for Mobile Robots

C. Franco, G. López-Nicolás, D. M. Stipanović, C. Sagüés

***Abstract*— We consider the problem of vision-based coverage control with a team of robots in the sense of dynamic coverage. Therefore, the aim is to actively cover certain domain by means of robots' visual sensors while they navigate in the workspace. Each robot is equipped with a conventional camera, an anisotropic sensor modeled as a wedge-shaped region in front of the robot. The contribution is a new algorithm for coordinated coverage control which, weights local and global information while avoiding local minima and obeying the particularities of the anisotropic vision-based sensors. The performance of the proposed technique is illustrated with simulation results.**

I. INTRODUCTION

Visual control is currently a mature field of research, nevertheless it is still a very active area as new computer vision algorithms or control techniques are being developed and more ambitious applications are envisioned. Fundamental concepts and basic approaches in visual servo control are described in the tutorial [1]. More specific is the survey on vision for mobile robot navigation presented in [2], where visual control refers to the pose control of a vehicle in a closed loop using the input of a visual sensor.

Although visual control and autonomous navigation for a single robot is still an open research area, the use of multiple robots to fulfill particular tasks has been increasingly demanding during the last decade. This is due not only to the advances in hardware devices and commercial software but also to the fact that multiple robots may carry out tasks that are difficult or unfeasible for one single robot such as exploration, surveillance, security or rescue applications. However, not many works in the literature consider the use of vision in the algorithm design for multi-robot systems. Some examples of works using vision to fulfill tasks performed by multiple mobile robots are the localization method presented in [3], the vision-based formation control in [4] or the robot coordination proposed in [5]. Other related works are [6], that aims to enable groups of mobile robots to visually maintain formations in the absence of communication, and [7], that encapsulates the multi-robot system information in a single homography so as to drive the team to a desired formation.

In this work, we focused on the problem of coverage control by using a team of robots equipped with conventional cameras. The coverage task may be found in a wide variety of applications such as demining, cleaning, lawn mowing, painting or surveillance. However, the challenges involved in the multi-robot system control are far from trivial and need to be solved in order to exploit the benefits of multi-robot systems for the efficient coordination of the resources.

The problem of coverage can be classified as static or dynamic depending on how it is addressed. If the resources or robots are static, the problem is known as allocation of resources [8]. The other approach considers mobile resources, and may also consider variable or unknown environment. This problem is often referred to as area coverage and, although multiple applications are possible, literature is mainly focused on sensing tasks. Several approaches tackle the problem by means of an optimization function to be minimized in a decentralized manner with Voronoi partitions [9], [10], by using potential fields [11], [12], or gradient-based approaches [13], [14].

The goal of this work is to explore the feasibility of an anisotropic sensor, i.e. the conventional camera, in the context of dynamic coverage with a team of robots. The problem consists in a coordinated covering of the workspace with the camera field of view. Regarding the type of camera selected, for the last years, the use of omnidirectional cameras is growing because of their effectiveness due to the panoramic view from a single image. This type of camera can be modeled as an isotropic sensor with a circle shape. However, some applications require better resolution rather than a large field of view. In particular, we consider a camera mounted onboard the robots pointing forward. The problem of motion control of a single robot with camera field-of-view constraints has been considered, for instance, in [15], [16]. There, the goal is to keep the camera field of view focused in a particular zone of the environment during the navigation rather than perform visual coverage.

To the best of our knowledge, this is the first dynamic coverage control algorithm proposed for a team of robots considering anisotropic visual sensors. Closely related works are [10] that considers anisotropic sensors modeled with elliptic shape and [17], [18] with the same wedge-shape sensor considered in our work. However, both works focus on the problem of coverage control in the sense of deployment, whereas we are interested in the problem of dynamic coverage. In this paper, we propose a new motion strategy that weights continuously local and global components avoiding local minima and providing an efficient coverage of the domain. The local strategy is based on the gradient, while a blob analysis based approach is defined for the global strategy. The main novelty of this work resides in that the

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. López-Nicolás and C. Sagüés 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. M. Stipanović is with the Department of Industrial and Enterprise Systems Engineering, and the Coordinated Science Laboratory, University of Illinois, Urbana, IL 61801, USA.

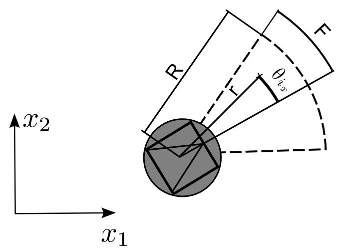

Fig. 1. Scheme of the variables of the sensing function. The dashed line represents the area covered by the camera onboard

proposed approach overcomes the challenges raised by the use of the camera sensor to achieve the coverage objective efficiently.

The paper is organized as follows. Section II introduces the problem formulation. The coverage control laws are presented in Section III. The strategy to select global objectives avoiding local minima is presented in Section IV. Simulations are given in Section V to illustrate the performance of the proposed approach. Conclusion and avenues for future research are given in Section VI.

II. Problem formulation

In this section we describe the framework for a team of nonholonomic agents performing dynamic coverage tasks with anisotropic sensors. The main objective is to reach a coverage level $\Lambda^*(x) > 0$ inside a domain $D_x \subset \mathbb{R}^2$. We assume the agents moves according to the following unicycle model:

$$\begin{aligned} \dot{p}_{i_1} &= v_i \cos(\theta_i), \\ \dot{p}_{i_2} &= v_i \sin(\theta_i), \\ \dot{\theta}_i &= \omega_i. \end{aligned} \tag{1}$$

Here, $p_i = [p_{i_1}, p_{i_2}]^T$ is the position of agent i in a convex domain $D_p \subset \mathbb{R}^2$, and $\theta_i \in [-\pi, \pi]$ is the orientation angle. The positions and the orientation angles of agents are assumed to be known, for instance, by visual localization or by a GPS system. v_i, ω_i are the linear and angular velocity inputs respectively. In this paper, we focus on sensing with vision cameras so let us define the sensing ability $\alpha_i(r, \theta_{i_x})$ as:

$$\alpha_i(r, \theta_{i_x}) = \begin{cases} \alpha_M \frac{(R-r)(F-\theta_{i_x})}{RF}, & r \le R, \theta_{i_x} \le F \\ 0, & \text{elsewhere} \end{cases} \tag{2}$$

where α_M is the maximum ability of sensing, R is the sensing range, $r = \|p_i - x\|$ is the distance from the agent to a point x, F is the half of the angle of view of the camera and $\theta_{i_x} = |\theta_i - \theta_x|$ is the angle between the camera and the point x. A graphical depiction of the variables is shown in Fig. 1. This wedge shaped function is maximum in the position of the agent and decreases with the linear and angular distance to the agent. The points that are nearer are better sensed and then it will take less time to cover those points. The coverage action of the team of agents is defined as $\alpha = \sum_{i \in \{1,\dots,N\}} \alpha_i(r, \theta_{i_x})$, with N being the number of robots. Furthermore we define $\Lambda(t, x)$ as the coverage developed by a team of agents over a point x at time t. The coverage information is updated continuously as follows:

$$\frac{\partial \Lambda(t, x)}{\partial t} = \alpha(r, \theta_{i_x}) \tag{3}$$

We assume that the points are initially uncovered $\Lambda(0, x) = 0, \quad \forall x \in D_x$. We introduce the lack of coverage $\Upsilon(t, x)$ over a point x at time t as:

$$\Upsilon(t, x) = \max\left(0, 1 - \frac{\Lambda(t, x)}{\Lambda^*(x)}\right). \tag{4}$$

At this point let us define the error function of the whole domain as:

$$e_{D_x}(t) = \frac{\int_{D_x} \Upsilon(t, x) dx}{S_{D_x}}, \tag{5}$$

where S_{D_x} is the area of the surface of the domain, and the error function of the actuator domain of each agent as:

$$e_{\Omega_i}(t) = \frac{\int_{\Omega_i} \Upsilon(t, x) dx}{S_{\Omega_i}}, \tag{6}$$

where Ω_i is the sensing domain, and S_{Ω_i} is the area of the surface of the sensing domain i.e. the area which the camera sees.

III. Dynamic coverage control laws

We divide our control strategy in two control laws that are weighted during the coverage process. One part of the control law depends on the local error, that is the error of the points that are in the coverage domain of the agent. The other part of the control law depends on the coverage error of the whole domain.

A. Local control law

The speed of the local control law is controlled with the amount of local error:

$$u_i^{loc} = 1 - e_{\Omega_i}. \tag{7}$$

In this way, when the local error is high, the agents slow down to cover the domain, and when the local error is low, they speed up to escape from covered areas. To obtain the angular velocity we start by computing:

$$\tilde{e} = \int_{D_x} \frac{\partial \Upsilon}{\partial t} dx = -\frac{1}{\Lambda^*(x)} \int_{\Omega - \Upsilon_0} \alpha(r, \theta_{i_x}) dx \tag{8}$$

We take out from the integral the points of D_x that are not in the coverage domain of the agents, i.e. the points that are not in $\Omega = \bigcup_{i=1}^N \Omega_i$, and the points whose lack of coverage is 0, $\Upsilon_0 = \{x : \Upsilon(t, x) = 0\}$, because both do not contribute to the integral. To optimize the orientation of each robot with respect to the variation of the error we take the partial derivative as:

$$\frac{\partial \tilde{e}}{\partial \theta_i} = \frac{1}{\Lambda^*(x)} \int_{\Omega - \Upsilon_0} \frac{(R - r)(\theta_i - \theta_x)}{RF\theta_{i_x}} dx \tag{9}$$

With this expression we get the direction $\theta_i^{loc^*}$ to obtain the maximum benefit for developing local coverage . Therefore, the contribution of the local control law to the control strategy is given by $(u_i^{loc}, \theta_i^{loc^*})$.

B. Global control law

When an agent falls into an area where the error is constant or symmetric from the point of view of the camera, expression in equation (9) equals zero, and the agent needs another input to reach uncovered areas until the domain is fully covered. In this section we propose a control law by defining $(u_i^{glo}, \theta_i^{glo^*})$ to reach an uncovered point p_i^{obj}, and in section IV we will explain how to choose these points between all uncovered points. The speed of the global control law is controlled with the distance from the agent to the objective, $\|p_i - p_i^{obj}\|$, minus a distance from where the agent can sense the objective, for example $R/2$ as:

$$u_i^{glo} = \frac{2}{\pi}\arctan(\|p_i - p_i^{obj}\| - R/2). \tag{10}$$

u_i^{glo} is close to one until the agent approach the surrounding area of the objective from where can cover it. Then, it decreases to 0 at a distance equal to $R/2$, and is negative inside a circle whose center is p_i and radius $R/2$. Then, when an agent is too near to the objective, it moves away avoiding to drive in circles around the point to see. The orientation to reach global objectives is obtained as:

$$\theta_i^{glo^*} = \arctan 2(p_i - p_i^{obj}). \tag{11}$$

C. Coverage control law

In order to combine both global and local control laws let us introduce a local weight W_i^{loc} and a global weight W_i^{glo} as follows:

$$W_i^{loc}(t) = e_{\Omega_i}^{\beta}(t) \tag{12}$$

$$W_i^{glo}(t) = 1 - e_{\Omega_i}^{\beta}(t) \tag{13}$$

where $\beta \in \mathbb{R}^+$ is a parameter which allows tuning the weights depending on the amount of error. We define the angular error $e_{\theta_i} \in (-\pi, \pi]$ as:

$$e_{\theta_i} = (\theta_i^{loc*} - \theta_i)W_i^{loc} + (\theta_i^{glo*} - \theta_i)W_i^{glo}, \tag{14}$$

and the linear and angular velocities are finally obtained with:

$$v_i = k_v(u_i^{loc}W_i^{loc} + u_i^{glo}W_i^{glo})(1 - \frac{2}{\pi}e_{\theta_i}), \tag{15}$$

$$\omega_i = k_\omega e_{\theta_i}(1 - u_i^{glo}). \tag{16}$$

k_v and k_ω are the control gains of the linear and angular velocity inputs, respectively. When the local error is high, i.e. e_{Ω_i} is close to 1, $W_i^{loc}(t)$ is also close to 1, and the agents obey the local control law which is based on the gradient of the coverage error. Thus, the agents move to get the most coverage benefit. However, when the local error is low, the agents do not get benefit covering its neighborhood. $W_i^{loc}(t)$ is close to 0, and the agents obey the global control law, which direct them to new areas with higher error. These control laws are both bounded by definition, with $v_i \in [-k_v, k_v]$ and $\omega_i \in [-k_\omega\pi, k_\omega\pi]$, allowing a straightforward implementation of the algorithm in real robots by adjusting the gains to the maximum speed of the robots.

Algorithm 1 Blob-based algorithm for the selection of global objectives

Require: D_x, $\Upsilon(t,x)$, Ψ, π_j;
Ensure: Ψ, π_j;

for $j = 1,..,M$ **do**
  if $\Upsilon(t, \psi_j) \leq 0$ **then**
    $\Pi = \Pi - \pi_j$; $\Psi = \Psi - \psi_j$;
  end if
end for
for $j = 1,..,M$ **do**
  $d_j^{min} = min(\|\psi_j - \psi_r\|), r = 1..M/j$
end for
for $j = 1,..,M$ **do**
  if $d_j^{min} < R$ **then**
    $\Pi = \Pi - \pi_j$; $\Psi = \Psi - \psi_j$;
  end if
end for
$D_{blob} = D_x - \Pi - \pi^\emptyset$;
while $D_{blob} \neq \emptyset$ **do**
  $(\psi^1, ..., \psi^K, \pi^1, .., \pi^K) = blob(D_{blob})$;
  for k=1,..,K **do**
    if $\psi^k \cup \pi^k \neq \emptyset$ **then**
      $\Psi_{M+1} = \psi^k$; $\pi_{M+1} = \pi^k$;
      $D_{blob} = D_{blob} - \pi^k$;
    end if
  end for
  $D_{blob} = erode(D_{blob})$;
end while
Assign eroded points to the nearest blob;

IV. Selection of global objectives

In this section, we propose a strategy to find areas with large error and to provide the agents with inputs to reach them (i.e. we now describe the procedure to define the values of p_i^{obj} used in section III.B). It is based on blob detection of the uncovered information. We use this image processing technique to find islands of uncovered information in the map $\Lambda(t,x)$, and then we compute their sizes and their centroids. With this information we propose a criterium to select a centroid as a global objective based on the uncertainty and the proximity of the blobs. Hence, a global objective is the centroid of an uncovered area which is close to the agent.

Let us define $\Psi = \{\psi_1, \psi_2, ..., \psi_j, ..., \psi_M\}$ as the collection of M global objectives, $\psi_j \in D_x$. We refer to ψ's as objectives and they represent points that belongs to a set of uncovered points. Let us also define π_j as the collection of points of the domain composing each blob and whose global objective is ψ_j, $\Pi = \bigcup_{j=1}^{M} \pi_j$ as the collection of points of the domain assigned to objectives ψ_j, and $\pi^\emptyset = \{x \in D_x | \Upsilon(t,x) \leq 0\}$ as the collection of points that are covered. The method to select the global objectives is described in Algorithm 1.

The algorithm starts by checking if some of the M global objectives ψ_j have been covered. Those covered are erased from the list of global objectives Ψ and the points π_j

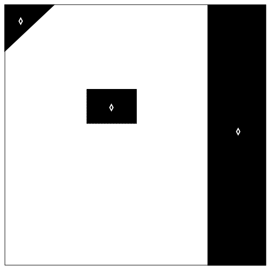

Fig. 2. Example of an uncertainty map that causes the centroid to fall inside of the blob. Black area represents the uncovered zone and white the covered zone whose points belongs to $\pi^{\emptyset}$. The centroids are represented by rhombi. The uncertainty map has 3 blobs and then 3 global objectives are generated. The points π_j in black areas are assigned to their respective centroids ψ_j.

assigned to the objective are released. It also checks if there are objectives that are closer than a distance R, which is the coverage radius of the agents. Those objectives are also erased and their points released to try to merge them to get a bigger blob in the blob searching procedure. Afterwards, the domain to obtain the new blobs of the scene is computed by subtracting the covered points $\pi^{\emptyset}$ and the assigned points Π from the domain to cover D_x. With $blob(D_{blob})$ the centroids ψ^k and the points π^k of the K regions of the space to be covered are obtained (as shown in Fig. 2). Then, the centroids ψ^k are checked to see if they belong to the points π^k of the blob. If the centroids ψ^k are inside the blob, they and the points of the blobs π^k are saved, whereas the blob domain is reduced by the points of the new found blob.

Once the checking is complete, it is possible that some centroids fall outside of their respective blobs. This is not a desired situation because due to the coverage range of the agent, it is possible that once the agent has arrived to the global objective, it cannot reach uncovered points causing a blockage. In this case, the image is eroded (as depicted in Fig. 3). This results in the elimination of the points in the domain that are in contact with covered or assigned points in such a way that the irregularities of the blob that cause the centroid to fall outside the blob are eliminated. Afterwards, blob analysis is repeated while the blob domain is not null. Finally, the eroded points are assigned to the nearest blob. It is possible but unlikely that, due to symmetries, no global objectives are found. In that case a nearest uncovered point is the only global objective until the symmetry breaks.

The choice of the objective p_i^{obj} for each robot i is done according to proximity. First, the distance between the i-th agent and the j-th centroid is calculated to create a matrix D as follows:

$$D(i,j) = \left(\frac{\|p_i - \psi_j\|}{\max(\|p_i - \psi_j\|)} \right). \qquad (17)$$

It is divided by the maximum distance in such a way that the elements of D are in interval $(0,1]$. Then, with this matrix, the global objectives are assigned using Algorithm 2. It is repeated until all the agents have a global objective. First, the algorithm finds the minimum distance between an agent and a centroid. Then, the agent takes that centroid as the global

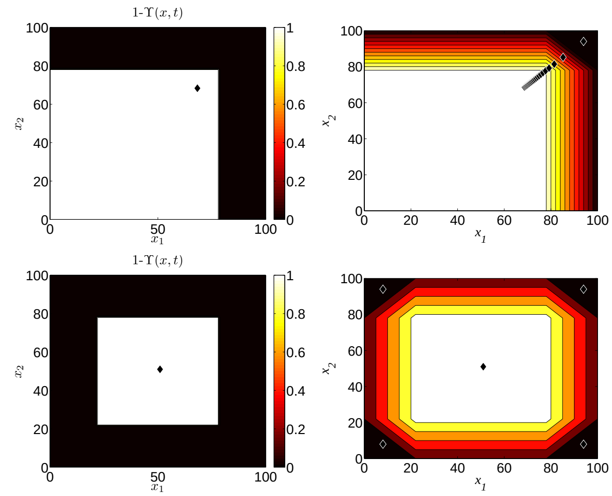

Fig. 3. Two examples of uncertainty maps that cause the centroid to fall outside of the blob. In the left column black areas represent the uncovered zones and white the covered zones. The centroids are represented by rhombi. A blob analysis of these scenes produces in each case only one connected blob with one centroid. Due to the shapes of the blobs, the centroids fall outside their blobs and according to Algorithm 1 an erosion process is needed. In the right column the iterative erosion process is shown until the centroid falls into the blob. After this process is completed, eroded points are assigned to the last centroids obtained according to proximity.

objective. Next, the algorithm adds 1 to the column where the distances of the centroid to other has been calculated to avoid that the centroid is assigned to other agent, distributing the team between all the available centroids. Also, the row of the agent is increased by N units to avoid the assignation of other centroid to the agent. If there are more agents than centroids, when all the centroids have been assigned once, all the distances between centroids and non assigned agents are increased by 1 and then, the centroids can be assigned again with the distance criterium. This process is repeated $\lfloor N/M \rfloor$ times distributing at most $\lceil N/M \rceil$ to each centroid and at least $\lfloor N/M \rfloor$.

Algorithm 2 Assignation of objectives

Require: D, Ψ

Ensure: p_i^{obj}

repeat
 $[i,j] = \{i,j : D(i,j) = \min(D)\}$;
 $p_i^{obj} = \psi_j$;
 $D(i,:) = D(i,:) + N$;
 $D(:,j) = D(:,j) + 1$;
until <All agents have an objective>

V. Simulation Results

In this section we present some simulation results that illustrate the behavior of the proposed algorithms. The domain to cover D_x is a square of 100x100 units, whereas $D_p = \mathbb{R}^2$. The coverage objective is $\Lambda^* = 100$. There are four agents with: $R = 20$, $F = 54^o$, $\alpha_M = 50$, $\beta = 1$, $k_v = 1$, and $k_\omega = 1$. Fig. 4 shows the evolution of the normalized error throughout the coverage process. In 592 units of time, the

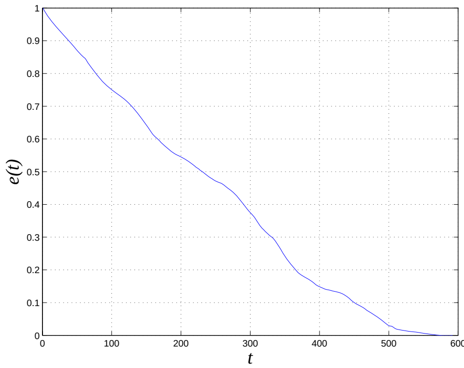

Fig. 4. Evolution of the normalized coverage error

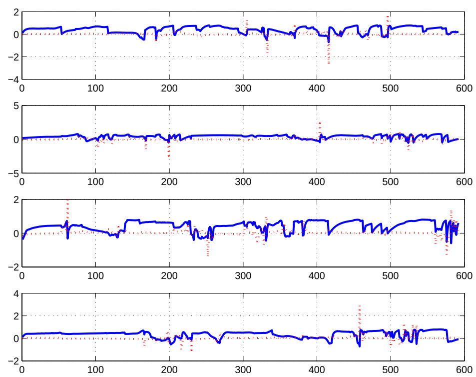

Fig. 5. Motion action of the agents. Solid line represents the linear action and dotted line the angular velocity.

objective has been fully accomplished at all the points of the domain. The motion actions to develop the coverage are shown in Fig. 5, and the map of the lack of coverage in each point at different times is shown in Fig. 6.

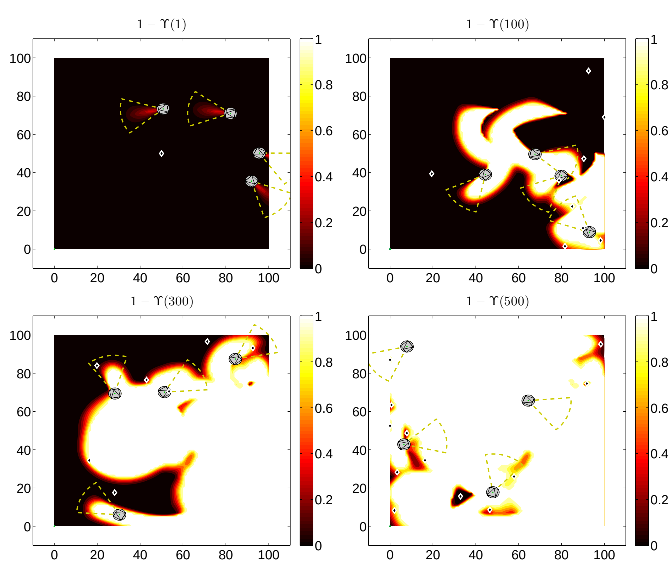

Fig. 6. Evolution of the global coverage map throughout the coverage process. Small circles represent the position of the robots and the coverage domain is represented by a dashed line. The domain is rather covered at t=500 and it is totally covered at t=592.

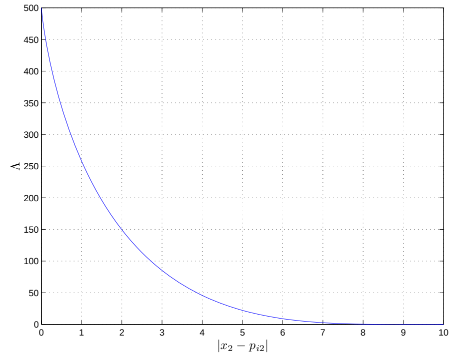

Fig. 7. Amount of coverage developed perpendicular to v_i when the agent moves along x_1.

We also develop simulations to check the efficiency in the usage of the robots. First, we start by computing the total ability of sensing $\mathcal{A}$:

$$\mathcal{A} = 2\int_0^F \int_0^R \alpha(r, \theta_{i_x}) r dr d\theta_{i_x} \tag{18}$$

With this value, we can compute the minimum time to completion for N agents t_N^*:

$$t_N^* = \frac{\int\int_{D_x} \Lambda^*(x_1, x_2) dx_1 dx_2}{\mathcal{A} \cdot N}, \tag{19}$$

Lastly, let us compare our algorithm with a typical path planning coverage trajectory carried out in zig-zag with maximum speed. The amount of coverage $\Lambda(x, t)$ developed by one agent in the perpendicular direction to v_i with $v_i = 1$ and $\omega_i = 0$ is shown in Fig. 7. The agent reaches the coverage objective over the points placed at $d_{100} = 2.72$ units away from the agent and half of the coverage objective over the points placed at $d_{50} = 3.865$ units away from the agent. The trajectory is developed taking into account these parameters and with a length L=140 units as shown in Fig. 8. The time to completion for one agent is $t_1^{zz} = 2080.6$ units of time with a path length of $PL_1^{zz} = 2054.6$ units. Assuming perfect coordination between teams of agents, we have $t_N^{zz} = 2080.6/N$ and we also assume $PL_N^{zz} = 2054.6$. Thus, we compute $E_N^{zz} = t_N^*/t_N^{zz}$.

In Fig. 9 we show the relation between the optimum and our algorithm, and the relation between the optimum and the zig-zag algorithm from 1 to 30 agents. The time to completion t_N of teams of robots varying from 1 unit to 30 units have been computed running 100 simulations in each case and taking the average value. For small teams our proposal takes 4 times the time to completion of the optimum, and as the team grows it tends to 5, compared with the zig-zag path that is 3 times slower. Furthermore, we also compare the sum of the path length of the robots to develop the coverage. We compute the path length of 100 simulations and show the average in Fig.10. Since the speed is regulated with the local error, our algorithm make the most of the traveled path. It takes some more time to cover the domain, but the path length needed in our algorithm is around a 55% of the path length needed with zig-zag strategy.

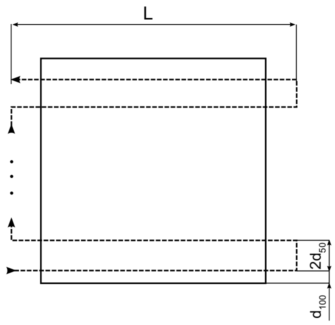

Fig. 8. Zig-zag path to cover the domain with one agent.

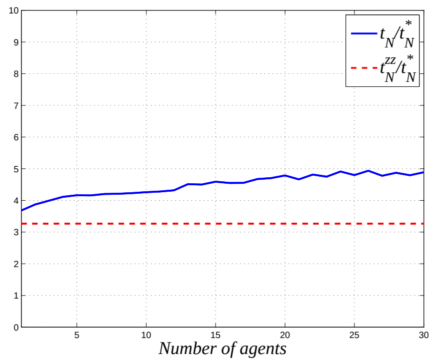

Fig. 9. Comparison of the time to completion of several teams of agents developing optimum coverage with our algorithm and with a zig-zag algorithm.

VI. Conclusion

In this paper we have proposed a new control algorithm for the dynamic coverage of a domain developed by a team of agents with anisotropic vision based sensors. The control law weights local and global actions, depending on local coverage error and global error map, respectively, to give 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 an agent is small. We also propose, a new strategy to select global objectives based on blob analysis of the whole map. Additionally, we impose bounds on both linear and angular velocities. Finally, simulations are provided to illustrate the approach. Current work focuses on the collision avoidance problem and the coverage with decentralized information. In this work, we assume that the robots do not produce occlusions between them. Even if the convergence is not compromised, future work would study this issue regarding the performance of the proposed algorithm.

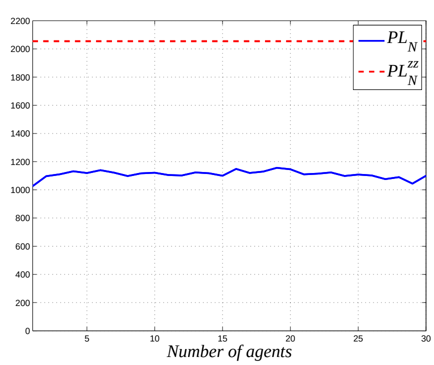

Fig. 10. Comparison of the path length of several teams of agents developing coverage with our algorithm and with a zig-zag algorithm.

References

[1] F. Chaumette and S. Hutchinson, "Visual servo control, part II: Advanced approaches," *IEEE Robotics and Automation Magazine*, vol. 14, pp. 109–118, Mar. 2007.

[2] G. N. DeSouza and A. C. Kak, "Vision for mobile robot navigation: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 2, pp. 237–267, 2002.

[3] H. Chen, D. Sun, and J. Yang, "Global localization of multirobot formations using ceiling vision SLAM strategy," *Mechatronics*, vol. 19, no. 5, pp. 617 – 628, 2009.

[4] A. K. Das, R. Fierro, V. Kumar, J. P. Ostrowski, J. Spletzer, and C. J. Taylor, "A vision-based formation control framework," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 813–825, 2002.

[5] N. Moshtagh, N. Michael, A. Jadbabaie, and K. Daniilidis, "Vision-based, distributed control laws for motion coordination of nonholonomic robots," *IEEE Transactions on Robotics*, vol. 25, pp. 851–860, Aug. 2009.

[6] R. Vidal, O. Shakernia, and S. Sastry, "Following the flock: Distributed formation control with omnidirectional vision-based motion segmentation and visual servoing," *Robotics and Autonomous Magazine*, vol. 11, no. 4, pp. 14–20, 2004.

[7] G. López-Nicolás, M. Aranda, Y. Mezouar, and C. Sagüés, "Visual control for multi-robot organized rendezvous," *IEEE Transactions on Systems, Man, and Cybernetics: Part B*, 2012.

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

[9] J. Cortés, S. Martínez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.

[10] A. Gusrialdi, S. Hirche, T. Hatanaka, and M. Fujita, "Voronoi based coverage control with anisotropic sensors," in *Proceedings of the American Control Conference*, pp. 736–741, 2008.

[11] 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*, pp. 642–647, 2004.

[12] 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.

[13] I. I. Hussein and D. M. Stipanović, "Effective coverage control for mobile sensor networks with guaranteed collision avoidance," *IEEE Transactions on Control Systems Technology*, vol. 15, no. 4, pp. 642–657, 2007.

[14] 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*, pp. 751 – 756, may 2011.

[15] G. López-Nicolás, N. R. Gans, S. Bhattacharya, J. J. Guerrero, C. Sagüés, and S. Hutchinson, "Homography-based control scheme for mobile robots with nonholonomic and field-of-view constraints," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 40, no. 4, pp. 1115–1127, 2010.

[16] P. Salaris, L. Pallottino, and A. Bicchi, "Shortest paths for finned, winged, legged, and wheeled vehicles with side-looking sensors," *International Journal of Robotics Research*, vol. 31, no. 8, pp. 997–1017, 2012.

[17] K. Laventall and J. Cortes, "Coverage control by robotic networks with limited-range anisotropic sensory," in *American Control Conference, 2008*, pp. 2666 –2671, june 2008.

[18] A. Gusrialdi, T. Hatanaka, and M. Fujita, "Coverage control for mobile networks with limited-range anisotropic sensors," in *47th IEEE Conference on Decision and Control*, pp. 4263 –4268, dec. 2008.