Hybrid coverage and inspection control for anisotropic mobile sensor teams

Antonio Adaldo * **Dimos V. Dimarogonas** *
Karl H. Johansson *

* *Department of Automatic Control and ACCESS Linnaeus Center, School of Electrical Engineering, KTH Royal Institute of Technology, Osquldas väg 10, 10044, Stockholm, Sweden; emails:* {`adaldo,dimos,kallej`}`@kth.se`.

Abstract: In this paper, we present an algorithm for pose control of a team of mobile sensors for coverage and inspection applications. The region to cover is abstracted into a finite set of landmarks, and each sensor is responsible to cover some of the landmarks. The sensors progressively improve their coverage by adjusting their poses and by transferring the ownership of some landmarks to each other. Inter-sensor communication is pairwise and intermittent. The sensor team is formally modeled as a multi-agent hybrid system, and an invariance argument formally shows that the team reaches an equilibrium configuration, while a global coverage measure is improving monotonically. A numerical simulation corroborates the theoretical results.

Keywords: multi-agent systems, sensor networks, hybrid and switched systems modeling

1. INTRODUCTION

A wide variety of applications involve collecting information in hazardous environments, which makes it desirable to delegate such missions to a team of autonomous agents with sensing capabilities. Therefore, in the last few decades, a lot of research interest has been devoted to the problem of autonomous deployment of robot teams in an assigned space—see Cortés et al. (2004); Martinez et al. (2007); Zhong and Cassandras (2011); Le Ny and Pappas (2013) to name just a few. Typically, the goal is to design a distributed algorithm that gradually drives the agents to a spatial configuration such that the team's collective perception of the environment is optimized. This problem is commonly known as the *coverage* problem, and it is often approached using Voronoi tessellations and the Lloyd algorithm (see Du et al. (1999)). The majority of the existing work on coverage considers agents with omnidirectional perception of the surrounding environment. Recently, agents with anisotropic sensing patterns (see Stergiopoulos and Tzes (2013); Gusrialdi et al. (2008); Laventall and Cortes (2008)) as well as vision-based sensing patterns (see Kantaros et al. (2015); Marier et al. (2012)) have been considered. Dynamic versions of the coverage problem have also been studied, where the agents do not converge to fixed positions, but keep navigating the environment in order to maintain a satisfactory coverage over time. This problem is commonly known as *effective* or *dynamic coverage* (see Hussein and Stipanovic (2007)). A vision-based version of effective coverage is studied in Panagou et al. (2015).

⋆ This work has received funding from the European Union Horizon 2020 Research and Innovation Programme under the Grant Agreement No. 644128, AEOROWORKS, from the Swedish Foundation for Strategic Research, from the Swedish Research Council, and from the Knut och Alice Wallenberg foundation

Information exchange among the sensing agents constitutes one of the major challenges in the real-world implementation of coverage algorithms. For this reason, a gossip-based communication strategy for coverage is studied in Durham et al. (2012) among others. In some coverage missions, it is convenient to abstract the environment into a finite set of points (see Durham et al. (2012)), which may either correspond to a sparse set of points of interest, or to a discretized approximation of the environment itself.

Our contribution with respect to the related work is twofold. On one hand, we generalize the notions of Voronoi cell and Voronoi tessellation, thus extending the use of this formalism to coverage problems with general, anisotropic sensing patterns. On the other hand, we address the coverage problem with a distributed hybrid control algorithm, where system flow is given by the motion of the sensing agents, while the system jumps are given by the communication instances between different agents.

Using the hybrid system formalism developed in Goebel et al. (2012), we formally show that the agents reach an equilibrium configuration, while a measure of the coverage attained by the team improves monotonically. The effectiveness of the proposed algorithm is then demonstrated by simulating a team of sensors in ROS (Quigley et al. (2009)).

2. PRELIMINARIES

We let $\mathbb{N}$ denote the set of the nonnegative integers, and $\mathbb{R}_{\geq 0}$ denote the set of the nonnegative real numbers. For $V : \mathbb{R}^n \to \mathbb{R}$, with $n \in \mathbb{N}$, $\nabla V(\cdot)$ denotes the gradient of V. All vectors in $\mathbb{R}^n$ (including gradients) are column vectors. We let 0_n (with $n \in \mathbb{N}$) be the vector of $\mathbb{R}^n$ whose entries are all zero. The set of the unity-norm vectors in $\mathbb{R}^3$ is also called the *unit sphere*, and is denoted

$\mathcal{S}^2$. The special orthogonal group in three dimensions is denoted SO(3). Each element of SO(3) is represented as a rotation matrix, and each column of such matrix belongs to $\mathcal{S}^2$; namely, if $R \in \mathrm{SO}(3)$, then $R = [x\ y\ z]$, with $x, y, z \in \mathcal{S}^2$. The special Euclidean group in three dimensions is denoted as SE(3). Each element of SE(3) is represented as a homogeneous transformation matrix; namely $T = \begin{bmatrix} R & p \\ 0_3^\top & 1 \end{bmatrix}$, where $R \in \mathrm{SO}(3)$, and $p \in \mathbb{R}^3$. The set of the skew-symmetric matrices in three dimensions is denoted as so(3). When writing the kinematics of a rigid body, we use the skew operator, $\mathrm{S} : \mathbb{R}^3 \to \mathrm{so}(3)$, which is defined as $\mathrm{S}(u)v = u \times v$, where $\times$ denotes the cross product. The trace of a square matrix A is denoted as $\mathrm{Tr}(A)$. Set closure is denoted as $\mathrm{Cl}(\cdot)$. In algorithms, $=$ denotes logical equality, while $:=$ denotes assignment.

In this paper, we use the hybrid system formalism from Goebel et al. (2012) to prove the convergence properties of our coverage algorithm. Here, we recall some basic definitions and properties related to this formalism. A *hybrid system* is a tuple $\mathscr{H} = (\mathscr{C}, F, \mathscr{D}, G)$, where $\mathscr{C}, \mathscr{D} \subseteq \mathbb{R}^n$ (with $n \in \mathbb{N}$) are called respectively the *flow set* and the *jump set*, and $F : \mathscr{C} \to \mathbb{R}^n$, $G : \mathscr{D} \to \mathbb{R}^n$ are called respectively the *flow map* and the *jump map*. In a hybrid system, the state is subject to both continuous flow and instantaneous jumps, according to the differential inclusions

$$\begin{cases} \dot{x} \in F(x)\ \forall x \in \mathscr{C}, \\ x^+ \in G(x)\ \forall x \in \mathscr{D}. \end{cases} \tag{1}$$

A state trajectory of a hybrid system is parametrized by the time elapsed and the number of jumps occurred; formally, a state trajectory is a function $x : E \to \mathbb{R}^n$, where $E \subset \mathbb{R}_{\geq 0} \times \mathbb{N}$. A *solution* of $\mathscr{H}$ is a state trajectory that satisfies the following properties: (i) $x(0,0) \in \overline{\mathscr{C}} \cup \mathscr{D}$; (ii) for all $k \in \mathbb{N}$ such that $I^k = \{t : (t,k) \in E\}$ has nonempty interior, $x(t,k) \in \mathscr{C}$ for all $t \in \mathrm{int}(I^k)$ and $\dot{x}(t,k) \in F(x(t,k))$ for almost all $t \in I^k$; (iii) for all $(t,k) \in E$ such that $(t,k+1) \in E$, $x(t,k) \in \mathscr{D}$ and $x(t,k+1) \in G(x(t,k))$. A solution is *complete* if its domain E is unbounded. A solution $x : E \to \mathbb{R}^n$ is *maximal* if there does not exist another solution $x' : E' \to \mathbb{R}^n$ such that $E \subseteq E'$ and $x'(t,k) = x(t,k)$ for all $(t,k) \in E$. A set $\mathscr{I} \subseteq \mathbb{R}^n$ is *weakly invariant* for $\mathscr{H}$ if, for each $\xi \in \mathscr{I}$: (i) there exists at least one complete solution $x(\cdot,\cdot)$ with $x(0,0) = \xi$ that stays forever in $\mathscr{I}$; (ii) for each $\tau > 0$, there exists at least one maximal solution such that $x(t^*, k^*) = \xi$ for some $t^* + k^* \geq \tau$ and $x(t,k) \in \mathscr{I}$ for all $t + k < t^* + k^*$. The following lemma is used to prove our main result.

Lemma 2.1. (From Corollary 8.4 in Goebel et al. (2012)). Consider a hybrid system $\mathscr{H} = (\mathscr{C}, F, \mathscr{D}, G)$, and let $V : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable in a neighborhood of $\mathscr{C}$. Suppose that $\nabla V(x)^\top \phi \leq 0\ \forall x \in \mathscr{C}, \phi \in F(x)$ and $V(x) \leq V(g)\ \forall x \in \mathscr{D}, g \in G(x)$. Let $Z_c = \{x \in \mathscr{C} : \nabla V(x)^\top \phi = 0\ \forall \phi \in F(x)\}$ and $Z_d = \{x \in \mathscr{D} : V(g) = V(x)\ \forall g \in G(x)\}$. Then, there exists $r \geq 0$ such that every complete and bounded solution of $\mathscr{H}$ converges to the nonempty set which is the largest weakly invariant subset of $V^{-1}(r) \cap (\mathrm{Cl}(Z_c) \cup (Z_d \cap G(Z_d)))$.

3. PROBLEM FORMULATION

In this paper, we use the abstract concepts of *landmark* and *sensor* to model a coverage problem. A *landmark* represents a body, a point, or a small area of interest, which must be kept under observation. A landmark ℓ is characterized by its pose $T_\ell \in \mathrm{SE}(3)$. In this work landmarks are considered fixed objects; therefore, the pose of a landmark is constant. A *sensor* represents a mobile sensing device, which is deployed in the environment in order to monitor the landmarks. A sensor $\jmath$ is characterized by a time-varying pose $T_\jmath \in \mathrm{SE}(3)$, and by a continuously differentiable function $f_\jmath : \mathrm{SE}(3) \to \mathbb{R}_{\geq 0}$, which is called the *footprint* of the sensor. When we want to refer to a sensor with the particular pose that it assumes at time t, we use the notation $\jmath(t) = (T_\jmath(t), f_\jmath)$; otherwise, we use the notation $\jmath = (T_\jmath, f_\jmath)$. The footprint of a sensor is a mathematical model of how well the sensor can perceive the landmarks in its surrounding environment. Namely, $f_\jmath(T_\jmath^{-1} T_\ell)$ is a measure of how well the sensor $\jmath$ perceives the landmark ℓ. (Note that ${T_\jmath}^{-1} T_\ell$ represents the pose of the landmark from the point of view of the sensor.) This concept is formalized in the following definition.

Definition 3.1. The *perception* of a landmark ℓ attained by a sensor $\jmath$ is defined as

$$\mathrm{per}(\jmath, \ell) := f_\jmath(T_\jmath^{-1} T_\ell). \tag{2}$$

We adopt the convention that a smaller value of the perception corresponds to a better perception, with $\mathrm{per}(\jmath, \ell) = 0$ being the best possible perception. Some sensing models used in the literature can be seen as particular cases of this model, as it is illustrated in the following examples.

Example 3.1. (Temperature sensors). Typically, a temperature sensor can measure the temperature of the surrounding environment with a degree of accuracy that degrades with the distance from the sensor itself. Therefore, a reasonable footprint for a temperature sensor is

$$f_\jmath(T) = \|p\|^2, \tag{3}$$

where $p = T_{\{1:3,4\}}$. where $p_\jmath = T_{\jmath\{1:3,4\}}$ is the position of the sensor and $p_\ell = T_{\ell\{1:3,4\}}$ is the position where we want to measure the temperature. Most of the existing research on coverage focuses on this particular sensing pattern—see for example Cortés et al. (2004).

Example 3.2. (Inspection with aerial robots). Consider the task of inspecting a wind turbine with a team of aerial robots endowed with cameras. The perception that a vehicle has of a point on the surface of the turbine depends on the distance between the vehicle and the point, but also on the direction that the camera is pointing to: perception is best if the point lies on the line of sight of the camera, and it is worse if the point approaches the boundaries of the camera's cone of view. This type of perception can be described by the following footprint:

$$f_\jmath(T) = k_1 \|\beta e_1 - p\|^2 + k_2 \|\beta e_1 - p\| (\beta e_1 - p)^\top e_1, \tag{4}$$

where $0 < k_1 < k_2$, $p = T_{\{1:3,4\}}$, $e_1 = [1,0,0]^\top$, and $\beta > 0$ is the optimal distance between the camera and the observed point. Figure 1 illustrates a contour plot of footprint (4).

Next, we define the *coverage* of a set of landmarks attained by a sensor as the sum of the perceptions of the landmarks. This concept is formalized in the following definition.

Definition 3.2. The *coverage* of a finite set of landmarks $\mathscr{L} = \{\ell_1, \ldots, \ell_M\}$, attained by a sensor $\jmath$ is defined as

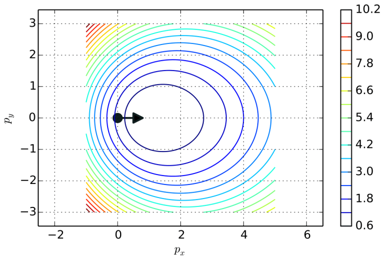

Fig. 1. Contour plot of footprint (4) with $\beta = 1$, $k_f = 0.6$, and $k_r = 0.4$, as a function of p_x and p_y, where $p = [p_x, p_y, 0]$.

$$\mathrm{cov}(\mathcal{s}, \mathscr{L}) := \sum_{\ell \in \mathscr{L}} \mathrm{per}(\mathcal{s}, \ell). \tag{5}$$

In this paper, our aim is to use a team of sensors in order to optimize the coverage of an environment. Therefore, we shall now formally define the coverage of a set of landmarks attained by a team of sensors.

Definition 3.3. Consider a team of sensors $\mathscr{S} = (\mathcal{s}_1, \ldots, \mathcal{s}_N)$ and a set of landmarks $\mathscr{L} = \{1, \ldots, M\}$. Let $\mathscr{L}$ be partitioned in N subsets $\mathscr{L}_1, \ldots, \mathscr{L}_N$. Each sensor $\mathcal{s}_i$ in the team is assigned the landmarks $\mathscr{L}_i$, thus defining a partition $\mathscr{P} = (\mathscr{L}_1, \ldots, \mathscr{L}_N)$. Then, we define the coverage of the set $\mathscr{L}$ attained by the sensor team $\mathscr{S}$ with the partition $\mathscr{P}$ as

$$\mathrm{Cov}(\mathscr{S}, \mathscr{P}) := \sum_{i=1}^{N} \mathrm{cov}(\mathcal{s}_i, \mathscr{L}_i). \tag{6}$$

In other words, the coverage attained by the team is simply the sum of the coverages attained by the single sensors, each over its own subset of landmarks.

Our objective is to control the poses of the sensors and the partition of the landmarks in such a way that the team coverage improves over time (i.e., $\mathrm{Cov}(\mathscr{S}, \mathscr{P})$ decreases with time). This objective can be formalized as an optimization problem. To this aim, introduce the binary variables $\sigma_{1,1}, \ldots, \sigma_{N,M}$, such that $\sigma_{i,j} = 1$ if $\ell_j \in \mathscr{L}_i$ (i.e., if the sensor $\mathcal{s}_i$ owns the landmark ℓ_j), and $\sigma_{i,j} = 0$ otherwise. Imposing that $\sigma_{i,1} + \cdots \sigma_{i,N} = 1$ ensures that each landmark is owned by exactly one sensor. Then, the optimization of the team coverage can be formulated as the following problem:

$$\begin{aligned} \underset{\substack{T_1, \ldots, T_N \in \mathrm{SE}(3) \\ \sigma_{1,1}, \ldots, \sigma_{N,M} \in \{0,1\}}}{\text{minimize}} \quad & \sum_{i=1}^{N} \sum_{j=1}^{M} \sigma_{i,j} f_i(T_i^{-1} T_{\ell_j}), \\ \text{subject to} \quad & \sigma_{i,1} + \cdots + \sigma_{i,M} = 1, \ i \in \{1, \ldots, N\}, \end{aligned} \tag{7}$$

where f_i denotes the footprint of sensor $\mathcal{s}_i$ and T_{ℓ_j} denotes the pose of landmark ℓ_j. Using (2), (5) and (6), we can see that the cost function in Problem (7) is indeed equal to the team coverage $\mathrm{Cov}(\mathscr{S}, \mathscr{P})$. Finding a globally optimal solution of Problem (7) may require computationally demanding optimization algorithms, especially because of the implicit constraint $T_1, \ldots, T_N \in \mathrm{SE}(3)$ (which essentially corresponds to the orthonormality of the vectors that describe the orientation of each sensor). In the rest of this paper, we describe a distributed algorithm to find a locally optimal solution of Problem (7). In the proposed algorithm, most of the computation is performed locally and asynchronously by each individual sensor; communication among the sensors is only intermittent and pairwise. The algorithm can be performed online (i.e., while the sensor team is performing a coverage mission), and guarantees convergence to an equilibium configuration, with a monotonically decreasing (i.e., improving) value of the team coverage.

4. SINGLE-SENSOR CONTROL FOR COVERAGE IMPROVEMENT

The first step in defining our coverage algorithm is to define how we control the motion of each individual sensor to improve the coverage attained by that sensor. Let $T_i(t) \in \mathrm{SE}(3)$ be the pose of sensor $\mathcal{s}_i$ at time t, let f_i be its footprint, and let $v_i(t), \omega_i(t) \in \mathbb{R}^3$ be respectively its linear velocity and angular velocity. From the laws of the kinematics of rigid bodies, we have that the pose of the sensor evolves according to the differential equation

$$\dot{T}_i(t) = \begin{bmatrix} \mathrm{S}(\omega_i(t)) R_i(t) & v_i(t) \\ 0_3^\top & 0 \end{bmatrix}, \tag{8}$$

where $R_i = T_{i\{1:3,1:3\}}$. We know that the coverage of a set of landmarks $\mathscr{L}_i$ attained by $\mathcal{s}_i$ is given by (5), which, using (2), can be rewritten as

$$\mathrm{cov}(\mathcal{s}_i(t), \mathscr{L}_i) = \sum_{\ell \in \mathscr{L}_i} f_i(T_i(t)^{-1} T_\ell). \tag{9}$$

Taking the time derivative of both sides of (9), and applying the chain rule for the derivative of a scalar with respect to a matrix (see Petersen and Pedersen (2012)), we have

$$\frac{\mathrm{d}}{\mathrm{d}t} \mathrm{cov}(\mathcal{s}_i(t), \mathscr{L}_i) = \sum_{\ell \in \mathscr{L}_i} \mathrm{Tr}\left(\frac{\partial f_i(T_i(t)^{-1} T_\ell)}{\partial T_i}^\top \dot{T}_i(t) \right). \tag{10}$$

Substituting (8) into (10), expanding the trace, and using the property $\mathrm{S}(u) w = -\mathrm{S}(w) u \ \forall u, w \in \mathbb{R}^3$, we have

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} \mathrm{cov}(\mathcal{s}_i(t), \mathscr{L}_i) = \sum_{\ell \in \mathscr{L}_i} \Bigg(& \frac{\partial f_i(T_i(t)^{-1} T_\ell)}{\partial p_i}^\top v_i(t) \\ - \sum_{\xi \in \{x_i(t), y_i(t), z_i(t)\}} & \frac{\partial f_i(T_i(t)^{-1} T_\ell)}{\partial \xi}^\top \mathrm{S}(\xi(t)) \omega_i(t) \Bigg), \end{aligned} \tag{11}$$

where $[x_i \ y_i \ z_i \ p_i] = T_{i\{1:3,1:4\}}$. From (11), we can see immediately that $\mathrm{cov}(\mathcal{s}_i(t), \mathscr{L}_i)$ can be made nonincreasing by choosing

$$v_i(t) = -\sum_{\ell \in \mathscr{L}_i} \frac{\partial f_i(T_i(t)^{-1} T_\ell)}{\partial p_i}, \tag{12}$$

$$\omega_i(t) = -\sum_{\ell \in \mathscr{L}_i} \sum_{\xi \in \{x_i(t), y_i(t), z_i(t)\}} \mathrm{S}(\xi) \frac{\partial f_i(T_i(t)^{-1} T_\ell)}{\partial \xi}. \tag{13}$$

5. MULTI-AGENT COVERAGE ALGORITHM

Before we start illustrating the proposed multi-agent coverage algorithm, we characterize the class of locally optimal solutions of Problem (7) that constitute our control objective. A feasible solution of Problem (7) is defined by specifying the pose and the landmarks of each sensor, with the constraint that the each landmark must belong exactly to one sensor. We are looking for a feasible solution such that the value of the team coverage cannot be improved by only moving one sensor in a neighborhood of its position (without changing the ownership of any landmark), or by only changing the ownership of some landmarks (without moving any sensor). We call this class of feasible solution *Voronoi solutions*, because of their analogy with Voronoi tessellations (see Du et al. (1999)). The formal definition of a Voronoi solution is given next.

Definition 5.1. A feasible solution $(T_1, \dots, T_N, \sigma_{1,1}, \dots, \sigma_{N,M})$ of Problem (7) is called a *Voronoi solution* if it satisfies the following properties for all $i \in \{1, \dots, N\}$:

$$\sum_{\ell \in \mathscr{L}_i} \frac{\partial f_i(T_i^{-1} T_\ell)}{\partial p_i} = 0_3; \tag{14}$$

$$\sum_{\ell \in \mathscr{L}_i} \sum_{\xi \in \{x_i, y_i, z_i\}} \mathrm{S}(\xi) \frac{\partial f_i(T_i^{-1} T_\ell)}{\partial \xi} = 0_3; \tag{15}$$

$$\sigma_{\nu,j} = 1 \implies \mathrm{per}((T_\nu, f_\nu), \ell_j) \leq \mathrm{per}((T_i, f_i), \ell_j). \tag{16}$$

Obviously, any globally optimal solution of Problem (7) is a Voronoi solution, but there may exist Voronoi solutions that are not globally optimal. Our goal is to define a distributed algorithm that drives the sensors to a Voronoi solution.

We are now ready to present the proposed multi-agent coverage algorithm. The input to the algorithm is a team of sensors $\mathcal{s}_1, \dots, \mathcal{s}_N$ in their initial poses $T_1^0, \dots, T_N^0$, and a set of landmarks $\mathscr{L} = \{\ell_1, \dots, \ell_M\}$. The landmarks constitute an abstract representation of an object or an area of interest that we want to keep under observation by using our sensor team.

The algorithm is initialized by assigning each landmark to one of the sensors. This initial assignment must not be considered a centralized operation: in fact, it can be done at random or by assigning all the landmarks to the same sensor. The pose of each sensor is continuously controlled by (12) and (13), while occasionally a sensor interacts with another sensor to transfer the ownership of some landmarks. A landmark is transferred if the recipient sensor has a better perception of that landmark than the original owner. Following these rules, moving a sensor or transferring a landmark always produces an improvement in the team coverage $\mathrm{Cov}(\mathscr{S}, \mathscr{P})$. The time instants when a sensor $\mathcal{s}_i$ contacts another sensor to give it some landmarks are denoted as $t_{i,k}$, with $k \in \mathbb{N}$. These time instants may be periodic, event-triggered, or triggered by user requests. We only require that there exist a minimum and a maximum dwell time between two consecutive communication instances; namely,

$$\exists \tau_{\min}, \tau_{\max} > 0 : \ \tau_{\min} \leq t_{i,k+1} - t_{i,k} \leq \tau_{\max}. \tag{17}$$

For the sake of generality, consider the threshold functions

$$\varsigma_i(\tau_i, T_i, \mathscr{L}_i) \in \mathbb{R}_{\leq 0}, \ i \in \{1, \dots, N\}, \tag{18}$$

where τ_i is a clock, and let the communication times $t_{i,k}$ be triggered by the condition $\varsigma_i(\tau_i, T_i, \mathscr{L}_i) \geq 0$. Note that this formalism encompasses both time-triggered and event-triggered communication. If $\mathcal{s}_i$ contacts another sensor $\mathcal{s}_{i'}$ but cannot give it any of its landmarks (i.e., if $\mathrm{per}(\mathcal{s}_j, \ell) \geq \mathrm{per}(\mathcal{s}_i, \ell)$ for all $\ell \in \mathscr{L}_i$), then $\mathcal{s}_i$ contacts another sensor until it manages to transfer at least one landmark, or it has contacted all the other sensors in the team. The described program can be formalized as the following algorithm.

Algorithm 5.1. (executed by each sensor $\mathcal{s}_i$ at each $t \geq 0$).

```
compute v_i by (12)
compute ω_i by (13)
update pose T_i as by (8)
update the clock as by τ̇_i = 1
if ς_i(τ_i, T_i, L_i) = 0 then
    X_i := {S} \ s_i
    while X_i ≠ ∅ do
        choose another sensor s_j from X_i
        for ℓ ∈ L_i do
            if per(s_j, ℓ) < per(s_i, ℓ) then
                transfer ℓ to s_j
                break while
            else remove s_j from X_i
            end if
        end for
    end while
    τ_i := 0
end if
```

6. MAIN RESULT

Our main result is to show that Algorithm 5.1 drives the sensors to a Voronoi solution of Problem (7). To make this result into a formal statement, first we model the sensor team and their landmarks as a hybrid system according to the formalism defined in Section 2.

The state of the system is given by the values of the clocks, the poses of the sensors, and the ownership of the landmarks. The ownership of the landmarks is captured by the variables $\sigma_{1,1}, \dots, \sigma_{N,M}$, introduced in Section 3. Hence, we let

$$X = (\tau_1, \dots, \tau_N, \mathrm{vec}(T_1)^\top, \dots, \mathrm{vec}(T_N)^\top, \sigma_{1,1}, \dots, \sigma_{N,M})^\top \in \mathbb{R}^{N+16N+NM}, \tag{19}$$

where $\mathrm{vec}(\cdot)$ denotes the vectorized form of a matrix.Since X is the state of a hybrid system, it is a function of both the time elapsed and the number of jumps occurred, and we should write $X(t, k)$; the same applies to all the state variables. However, we omit these dependencies whenever possible to avoid clutter in the notation.

The state flow is given by the movement of the sensors, described by (8), (12), and (13). State flow can happen from any state; therefore, we set $\mathscr{C} = \mathbb{R}^{N+16N+NM}$. The flow map captures the clocks' progression and the dynamics of the sensors' motion. Namely, we set $F(X) = \{\phi(X)\}$, with $\phi(X) = (1, \dots, 1, \mathrm{vec}(\dot{T}_1)^\top, \dots, \mathrm{vec}(\dot{T}_N)^\top, 0_{NM}^\top)^\top$, where $\dot{T}_i$ denotes here the closed-loop derivative of T_i, obtained by using (12) and (13) in (8).

The state jumps are given by the landmark transfers, and they are triggered by the conditions $\varsigma_i(\tau_i, T_i, \mathscr{L}_i) = 0$. Hence, state jumps are only possible from states such that $\varsigma_i(\tau_i, T_i, \mathscr{L}_i) = 0$ for some sensor. Therefore, we let

$$\mathscr{D} = \{X \in \mathscr{C} : \max_{i=1,\dots,N} \{\varsigma_i(\tau_i, T_i, \mathscr{L}_i)\} = 0\}. \tag{20}$$

The jump map captures the rules by which the sensors transfer landmarks to each other. Let $g_{i,i'}(X)$ be the new state after $\mathcal{s}_i$ has transferred landmarks to $\mathcal{s}_{i'}$. Then, we have

$$g_{i,i'}(X) = (\tau_1^+, \dots, \tau_N^+, 0_{16N}^\top, \sigma_{1,1}^+, \dots, \sigma_{N,M}^+), \tag{21}$$

where: $\tau_i^+ = 0$ and $\tau_q^+ = \tau_q$ for all $q \neq i$; $\sigma_{i,j}^+ = 0$ and $\sigma_{i',j}^+ = 1$ if $\sigma_{i,j} = 1$ and $\mathrm{per}(\mathscr{s}_{i'}, \ell_j) < \mathrm{per}(\mathscr{s}_i, \ell_j)$; $\sigma_{\xi,\zeta}^+ = \sigma_{\xi,\zeta}$ otherwise. Let $g_i(X)$ be the state after $\mathscr{s}_i$ has failed to transfer landmarks to all the other sensors. Then, we have

$$g_i(X) = (\tau_1^+, \ldots, \tau_N^+, 0_{16N}^\top, \sigma_{1,1}, \ldots, \sigma_{N,M}), \tag{22}$$

with $\tau_i^+ = 0$ and $\tau_q^+ = \tau_q$ for all $q \neq i$. With this notation, we can write the jump map as

$$G(X) = \begin{cases} \{g_{i,i'}(X) : g_{i,i'}(X) \neq X\} & \text{if } g_{i,i'}(X) \neq \emptyset, \\ \{g_i(X)\} & \text{otherwise.} \end{cases} \tag{23}$$

Our hybrid system is now defined as $\mathscr{H} = (\mathscr{C}, F, \mathscr{D}, G)$.

Now we need to define Voronoi solutions in terms of $\mathscr{H}$; in other words, we have to recast (14), (15), (16) in terms of X. To this aim, first note (using (8), (12) and (13)) that (14) and (15) are equivalent to

$$\phi(X) = (1, \ldots, 1, 0_{16N}^\top, 0_{NM}^\top)^\top. \tag{24}$$

Then, note that (16) means that it is not possible to transfer any landmark, which, using (23), can also be written as

$$G(X) = \{g_i(X)\}. \tag{25}$$

Therefore, the set of the states X corresponding to the Voronoi solutions of Problem (7) is

$$\begin{aligned} \mathscr{V} := \{X \in \mathbb{R}^{16N+NM+N} : X \in \mathscr{C} \implies \\ \phi(X) = (1, \ldots, 1, 0_{16N}, 0_{NM}^\top)^\top, \\ X \in \mathscr{D} \implies G(X) = \{g_i(X)\}\}. \end{aligned} \tag{26}$$

Our main result can now be formalized as the following theorem.

Theorem 6.1. (Main theorem). Consider a team of sensors $\mathscr{S} = (\mathscr{s}_1, \ldots, \mathscr{s}_N)$, with initial poses $T_1^0, \ldots, T_N^0 \in \mathrm{SE}(3)$, and a set of landmarks $\mathscr{L} = \{\ell_1, \ldots, \ell_M\}$. Let $\mathscr{L}_i$ be the subsets of the landmarks owned by sensor $\mathscr{s}_i$. Let $\mathscr{L}_i^0$ be the set of landmarks initially assigned to sensor $\mathscr{s}_i$, so that $(\mathscr{L}_1^0, \ldots, \mathscr{L}_N^0)$ is a partition of $\mathscr{L}$. Let $\mathscr{H} = (\mathscr{C}, F, \mathscr{D}, G)$ be the corresponding hybrid system defined by (19)–(23). If each sensor executes Algorithm 5.1, and the threshold functions ς_i are such that (17) is satisfied, then each complete solution of $\mathscr{H}$ converges to the set $\mathscr{V}$ of the Voronoi solutions of Problem (7).

Proof. Consider the function $\widetilde{\mathrm{Cov}}(X) := \mathrm{Cov}(\mathscr{S}, \mathscr{P})$. We are going to show that this function is nonincreasing both along the system flow and upon the system jumps, so that we can then apply Lemma 2.1.

Let us first study the evolution of this function along the system flow. Differentiating $\widetilde{\mathrm{Cov}}(X)$ with respect to X, we have

$$\nabla\widetilde{\mathrm{Cov}}(X) = \left(0, \mathrm{vec}\left(\frac{\partial\,\mathrm{cov}(\mathscr{s}_1, \mathscr{L}_1)}{\partial T_1}\right)^\top, \ldots, \mathrm{vec}\left(\frac{\partial\,\mathrm{cov}(\mathscr{s}_N, \mathscr{L}_N)}{\partial T_N}\right)^\top, 0_{NM}^\top\right)^\top. \tag{27}$$

Trasposing both sides, and right-multiplying by $\phi(X)$, we have

$$\nabla\widetilde{\mathrm{Cov}}(X)^\top \phi(X) = \sum_{i=1}^N \mathrm{vec}\left(\frac{\partial\,\mathrm{cov}(\mathscr{s}_i, \mathscr{L}_i)}{\partial T_i}\right)^\top \mathrm{vec}(\dot{T}_i). \tag{28}$$

Using (5) and (10) in (28), we can rewrite the right-hand side of (28) as

$$\nabla\widetilde{\mathrm{Cov}}(X)^\top \phi(X) = \sum_{i=1}^N \frac{\mathrm{d}}{\mathrm{d}t}\,\mathrm{cov}(\mathscr{s}_i, \mathscr{L}_i) \leq 0. \tag{29}$$

Let us now study the evolution of $\widetilde{\mathrm{Cov}}(X)$ upon the system jumps. Consider the generic jump time $t_{i,k}$. If $\mathscr{s}_i$ transfers some landmarks to another sensor $\mathscr{s}_{i'}$, then the change in the value of $\widetilde{\mathrm{Cov}}(\cdot)$ is

$$\widetilde{\mathrm{Cov}}(g_{i,i'}(X)) - \widetilde{\mathrm{Cov}}(X) = = \sum_{\substack{\{\ell \in \mathscr{L}_i: \\ \mathrm{per}(\mathscr{s}_{i'}, \ell) \\ < \mathrm{per}(\mathscr{s}_i, \ell)\}}} \mathrm{per}(\mathscr{s}_{i'}, \ell) - \mathrm{per}(\mathscr{s}_i, \ell) < 0. \tag{30}$$

If it is not possible to transfer landmarks to any other sensor, then $G(X) = \{g_i(X)\}$. Since $g_i(X)$ has the same values of the $\sigma_{i,j}$ variables as X, we have

$$\widetilde{\mathrm{Cov}}(g_i(X)) - \widetilde{\mathrm{Cov}}(X) = 0, \tag{31}$$

which means that the value of $\widetilde{\mathrm{Cov}}(\cdot)$ does not change upon this jump. Since (28), (30) and (31) hold, we can invoke Lemma 2.1 to conclude that the largest weakly invariant set $\mathscr{I}$ contained in $\mathscr{A} := \mathrm{Cl}\{X \in \mathscr{C} : \nabla\widetilde{\mathrm{Cov}}(X)^\top \phi(X) = 0\} \cup \{X \in \mathscr{D} : \widetilde{\mathrm{Cov}}(g) = \widetilde{\mathrm{Cov}}(X)\ \forall g \in G(X)\}$ is not empty, and any complete and bounded solution converges to it. Moreover, recalling that sensor footprints are radially unbounded, while $\widetilde{\mathrm{Cov}}(X)$ is nonincreasing, we can conclude that all solutions of $\mathscr{H}$ are bounded. Therefore, any complete solution of $\mathscr{H}$ converges to $\mathscr{I}$, and we can conclude the proof by showing that $\mathscr{I} = \mathscr{V}$. Obviosuly, $\mathscr{V}$ is invariant, and it is a subset of $\mathscr{A}$, so $\mathscr{V} \subseteq \mathscr{I}$. Suppose by contradiction that $\mathscr{V} \subsetneq \mathscr{I}$. Let $\xi = \arg\max_{X \in \mathscr{I} \setminus \mathscr{V}} \widetilde{\mathrm{Cov}}(X)$. Since $\mathscr{I}$ is weakly invariant, for any $\tau > 0$ there exists at least one maximal solution $\Xi(t,j)$ such that $\Xi(t^*, j^*) = \xi$ for some $t^* + j^* \geq \tau$. Since $\widetilde{\mathrm{Cov}}(\Xi)$ is nondecreasing, it must be $\widetilde{\mathrm{Cov}}(\Xi(t,j)) = \widetilde{\mathrm{Cov}}(\xi)$ for all $(t,j) \leq (t^*, j^*)$. However, since (17) holds, if we choose τ large enough, the solution Ξ must comprise both flow and jumps between $(0,0)$ and (t^*, j^*). Hence, the whole solution must lie in the subset of $\mathscr{A}$ such that $\widetilde{\mathrm{Cov}}(\Xi)$ does not change with either flow or jumps. From (26), we can see that this set is $\mathscr{V}$. But this is a contradiction, because we have supposed that $\xi \in \mathscr{I} \setminus \mathscr{V}$. Hence, we can conclude that $\mathscr{I} = \mathscr{V}$.

7. SIMULATION

In this section, we present a simulation of Algorithm 5.1 built upon the ROS middleware, with each sensor being implemented as a single ROS node.

We consider a team of $N = 4$ sensors, which are required to monitor a square room with side of length 6. The room is abstracted into a set of $M = 625$ landmarks, which are equally spaced about the room. Initially, all the landmarks are assigned to the same sensor. The initial positions of sensor $\mathscr{s}_i$ is chosen as $p_i = (0, 1.25 - 0.25i, 0)$, while the initial orientation is chosen as $R_i = I_3$ for all the sensors. The threshold functions are chosen as

$$\varsigma_i(\tau_i, T_i, \mathscr{L}_i) = \min\{\tau_i - \tau_d, \epsilon - \|v_i\|, \epsilon - \|\omega_i\|\}, \tag{32}$$

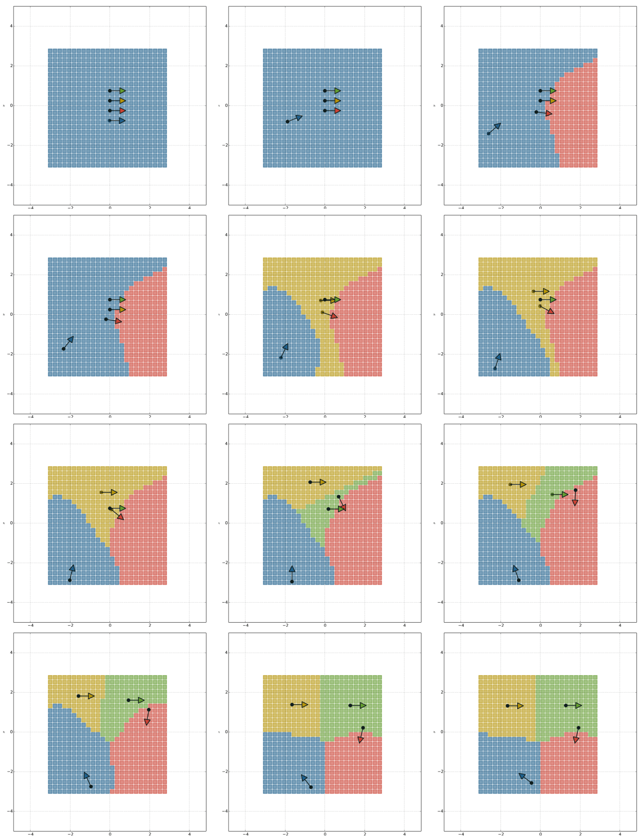

Fig. 2. Results of the simulation. Seconds elapsed in each snapshot, in lexicographical order: 0, 0.2, 1.0, 2.0, 5.0, 10.0, 15.0, 20.0, 30.0, 40.0, 75.0, 105.0. The blue sensor and the red sensor have footprint (4).

where $\epsilon = 2 \cdot 10^{-2}$, and v_i and ω_i are computed as (12) and (13) respectively. It is easy to verify that the thresholds (32) satisfy (17).

The footprint of the first two sensors is chosen as (4), with $k_f = 0.6$ and $k_r = 0.4$, while $\beta = 1$ for the first sensor and $\beta = 2$ for the second sensor. The footprint of the other two sensors is chosen as (3). This case corresponds to the scenario where some agents have normal cameras with different focal distance, while some other agents have omnidirectional cameras. The results of the simulation are shown in Figure 2, where we can see how the sensors are progressively deployed about the room, and how the landmark distribution assumes patterns corresponding to better coverage costs. Note that the final landmark distribution pattern reflects the shape of the different sensor footprints.

8. CONCLUSIONS AND FUTURE DEVELOPMENTS

We have presented a distributed algorithm for coverage and inspection tasks with multi-agent sensor teams. The proposed algorithm is based on the abstraction of the environment into a finite set of landmarks, and can handle sensors with anisotropic and heterogeneous sensing patterns. The sensor team is modeled as a hybrid multi-agent system, and the algorithm is formally shown to drive the agents to an equilibrium configuration, while a global cost function representing the coverage attained by the team is nonincreasing. Future work will extend the proposed algorithm to explicitly account for practical issues such as collision avoidance.

REFERENCES

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

Du, Q., Faber, V., and Gunzburger, M. (1999). Centroidal Voronoi Tessellations: Applications and Algorithms. *SIAM Review*, 41(4), 637–676.

Durham, J.W., Carli, R., Frasca, P., and Bullo, F. (2012). Discrete Partitioning and Coverage Control for Gossiping Robots. *IEEE Transactions on Robotics*, 28(2), 364–378.

Goebel, R., Sanfelice, R.G., and Teel, A.R. (2012). *Hybrid Dynamical Systems: Modeling, Stability, and Robustness*. Princeton Univerisity Press.

Gusrialdi, A., Hirche, S., Hatanaka, T., and Fujita, M. (2008). Voronoi based coverage control with anisotropic sensors. In *American Control Conference*, 736–741.

Hussein, I.I. and Stipanovic, D.M. (2007). Effective Coverage Control for Mobile Sensor Networks With Guaranteed Collision Avoidance. *IEEE Transactions on Control Systems Technology*, 15(4), 642–657.

Kantaros, Y., Thanou, M., and Tzes, A. (2015). Distributed coverage control for concave areas by a heterogeneous Robot-Swarm with visibility sensing constraints. *Automatica*, 53, 195–207.

Laventall, K. and Cortes, J. (2008). Coverage control by multi-robot networks with limited-range anisotropic sensory. In *American Control Conference*.

Le Ny, J. and Pappas, G.J. (2013). Adaptive Deployment of Mobile Robotic Networks. *IEEE Transactions on Automatic Control*, 58(3), 654–666.

Marier, J.S., Rabbath, C.A., and Lechevin, N. (2012). Visibility-limited Coverage Control Using Nonsmooth Optimization. In *American Control Conference*.

Martinez, S., Cortes, J., and Bullo, F. (2007). Motion Coordination with Distributed Information. *IEEE Control Systems Magazine*, 27(4), 75–88.

Panagou, D., Stipanovic, D.M., and Voulgaris, P.G. (2015). Dynamic coverage control in unicycle multi-robot networks under anisotropic sensing. *Frontiers in Robotics and AI*, 2, 1–17.

Petersen, K.B. and Pedersen, M.S. (2012). *The Matrix Cookbook*. Technical University of Denmark.

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Berger, E., Wheeler, R., and Mg, A. (2009). ROS: an open-source Robot Operating System. In *IEEE International Conference on Robotics and Automation*.

Stergiopoulos, Y. and Tzes, A. (2013). Spatially distributed area coverage optimisation in mobile robotic networks with arbitrary convex anisotropic patterns. *Automatica*, 49(1), 232–237.

Zhong, M. and Cassandras, C.G. (2011). Distributed Coverage Control and Data Collection with Mobile Sensor Networks. *IEEE Transactions on Automatic Control*, 56(10), 2445–2455.