Camera Network Coordination for Intruder Detection

Fabio Pasqualetti, Filippo Zanella, *Member, IEEE*, Jeffrey Russel Peters, Markus Spindler, Ruggero Carli, and Francesco Bullo

***Abstract*— This paper proposes surveillance trajectories for a network of autonomous cameras to detect intruders. We consider smart intruders, which appear at arbitrary times and locations, are aware of the cameras configuration, and move to avoid detection for as long as possible. As performance criteria, we consider the worst case detection time (WDT) and the average detection time (ADT). We focus on the case of a chain of cameras, and we obtain the following results. First, we characterize a lower bound on the WDT and on the ADT of smart intruders. Second, we propose a team trajectory for the cameras, namely equal-waiting trajectory, with minimum WDT and with guarantees on the ADT. Third, we design a distributed algorithm to coordinate the cameras along an equal-waiting trajectory. Fourth, we design a distributed algorithm for cameras reconfiguration in the case of failure or network change. Finally, we illustrate the effectiveness and robustness of our algorithms via numerical studies and experiments.**

***Index Terms*— Autonomous system, camera network, distributed optimization, distributed surveillance, multi-agent system.**

I. Introduction

COORDINATED teams of mobile agents have recently been used for many tasks requiring continuous execution, including the monitoring of oil spills [1], the detection of forest fires [2], the tracking of border changes [3], and the patrolling of environments [4]. The use of mobile agents has several advantages with respect to the classic approach of deploying a large number of static sensors, such as improved situation awareness and fast reconfigurability. In this paper, we address the challenging problem of scheduling the agents trajectories to optimize the performance in persistent surveillance tasks.

A. Problem Description

We consider a network of identical pan-tilt-zoom (PTZ) cameras for video surveillance, and we focus on the development of distributed and autonomous surveillance strategies for the detection of moving intruders. We make combined assumptions on the environment to be monitored, the cameras, and the intruders. We assume the environment to be 1-D, in the sense that it can be completely observed by a chain of cameras using linear motion only (the perimeter surveillance problem is a special case of this framework). We assume the cameras to be subjected to physical constraints, such as limited field of view (f.o.v.) and speed, and to be equipped with a low-level routine to detect intruders that fall within their f.o.v. We assume intruders to be smart, in the sense that they have access to the cameras configuration at every time instant, and schedule their trajectory to avoid detection for as long as possible. Since the probability of success of an intrusion increases with the time an intruder remains undetected in the environment [5], we propose cameras trajectories and control algorithms to minimize the worst case detection time (WDT) and the average detection time (ADT) of smart intruders.

B. Related Work

Of relevance to this paper are the research areas of robotic patrolling and video surveillance. In a typical robotic patrolling setup, the environment is represented by a graph on which the agents motion is constrained, and the patrolling performance is given by the WDT of static events. In [6]–[8], performance evaluations of certain patrolling heuristics are performed. In [2] and [4], an efficient and distributed solution to the (worst case) perimeter patrolling problem for robots with zero communication range is proposed. In [9], the computational complexity of the patrolling problem is studied as a function of the environment topology, and optimal strategies as well as constant-factor approximations are proposed. With respect to these works, we consider smart intruders, as opposed to static ones, and we study also the ADT, as opposed to the WDT only. In the context of camera networks the perimeter patrolling problem is discussed in [10] and [11], where distributed algorithms are proposed for the cameras to partition a 1-D environment and to coordinate along a trajectory with minimum WDT of static intruders. Graph partitioning and intruder detection with minimum WDT for 2-D camera networks are studied in [12]. We improve the results along this direction by showing that the strategies proposed in [10] and [11] generally fail at detecting smart intruders, and by studying the ADT of smart intruders. Complementary approaches based on numerical analysis and game

Manuscript received June 3, 2013; revised October 3, 2013; accepted November 3, 2013. Manuscript received in final form November 9, 2013. Date of publication November 26, 2013; date of current version July 24, 2014. This work was supported in part by the NSF under Award CPS 1035917 and in part by the ARO under Award W911NF-11-1-0092. Recommended by Associate Editor C. N. Hadjicostis.

F. Pasqualetti is with the Department of Mechanical Engineering, University of California, Riverside, CA 92521 USA (e-mail: fabiopas@engr.ucr.edu).

J. R. Peters, F. Bullo, and M. Spindler are with the Center for Control, Dynamical Systems and Computation, University of California, Santa Barbara, CA 93106 USA (e-mail: jrpeters@engineering.ucsb.edu; bullo@engineering.ucsb.edu; spindlermarkus@gmail.com).

F. Zanella and R. Carli are with the Department of Information Engineering, University of Padova, Padova 35122, Italy (e-mail: filippo.zanella@dei.unipd.it; carlirug@dei.unipd.it).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCST.2013.2290708

theory for the surveillance of 2-D environments are discussed in [13] and [14]. Finally, preliminary versions of this paper were presented in [15] and [16].

In the context of video surveillance, most of the approaches consider the case of static cameras, where the surveillance problem reduces to an optimal sensor placement problem. In [17] and [18], sensor placement problems are considered in a static camera network with the goals of maximizing the observability of a set of aggregate tasks that occur in a dynamic environment, and of visual tagging, respectively. In [19], a resource aware scheme for coverage and task assignment is considered. In the case of surveillance in active (PTZ) camera networks, there have been many attempts to formulate feasible approaches for camera control to detect and track targets, adapt sensor coverage, and achieve high image resolution; see [20]. Lim *et al.* [21] considered an image-based control scheme in a setup containing a master camera for detection and tracking. In [22] and [23], methodologies for obtaining high-resolution images in camera networks containing both static and active cameras are developed and tested in a virtual environment. In [24], a similar problem of opportunistic visual sensing is considered. In [25], the problem of coordinating camera motion is addressed using a game-theoretic approach with the assumption that the entire environment is covered at all times. Finally, the problem of context-aware anomaly detection is studied in [26]. We depart from the aforementioned works with PTZ camera networks in the following ways. First, we focus on the problem of intruder detection, rather than tracking or scene analysis, and we define appropriate performance metrics for this problem. Second, we do not make the assumption that cameras can cover the whole environment at all times, and we develop coordination methods for the cameras to surveil the environment. Third and finally, we do not require a source of global information for our algorithms, so that our methods are fully distributed.

C. Contributions

The contributions of this paper are as follows.

First, we mathematically formalize the concepts of cameras trajectory and smart intruder, and we propose the trajectory design problem for video surveillance. We formalize the WDT and ADT criteria, and we characterize lower bounds on both performance criteria.

Second, we propose the equal-waiting cameras trajectory, which achieves minimum WDT, and constant factor optimal ADT (under reasonable assumptions). The equal-waiting trajectory is easy to compute given a camera network, and it is amenable to distributed implementation. We develop a distributed coordination algorithm to steer the cameras along an equal-waiting trajectory. Our coordination algorithm converges in finite time, which we characterize, and it requires only local communication and minimal information to be implemented.

Third, we design a distributed reconfiguration algorithm (REC) for the cameras to react to failures and to adapt to time-varying topologies. In particular, REC takes advantage of gossip communication to continuously partition the environment and, at the same time, coordinate the motion of the cameras to optimize the detection performance.

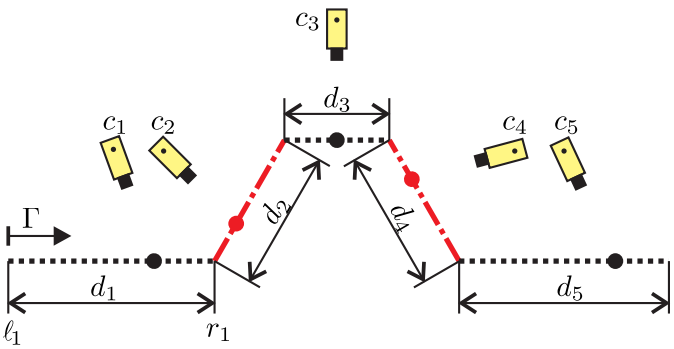


Fig. 1. This figure shows five cameras installed along a 1-D open path. The f.o.v. of each camera is a point on the path. Cameras coordinate their motion to detect smart moving intruders along the path.

Fourth, we illustrate our findings through numerical studies and experiments. Our numerical studies verify our results for different network configurations. Our experiments validate our modeling framework and assumptions, and show that our methods are robust to cameras failure, model uncertainties, and sensor noise.

We finally note that our algorithms are applicable beyond the domain of camera networks. For instance, we envision applicability to real-time scheduling for manufacturing, where tasks with spatial and temporal constraints are allocated, and robots need to complete these tasks while satisfying the given constraints [27]–[29].

D. Paper Organization

The remainder of this paper is organized as follows. Section II contains our problem setup and some preliminary results. In Section III, we present our main results, that is, we propose and characterize the equal-waiting trajectory, and we describe our distributed coordination algorithm. In Section IV, we detail our numerical studies and experiments. Section V contains our algorithm for cameras reconfiguration. Finally, our conclusion and final remarks are in Section VI.

II. Problem Setup and Preliminary Result

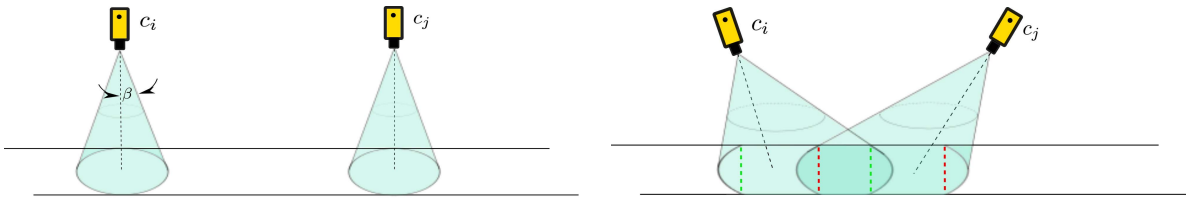
In this section, we describe the 1-D surveillance problem under consideration, and we present some useful definitions and mathematical tools for its analysis.

A. Problem Setting and Notation

Consider a set of $n \in \mathbb{N}$ identical active cameras installed along a 1-D open path (boundary) Γ of length L (Fig. 1). For the ease of notation and without affecting generality, we represent Γ with the segment $[0, L]$, and we label the cameras in an increasing order from c_1 to c_n according to their physical position on Γ. We make the following assumptions.

(A1) The f.o.v. of each camera is represented by a point on Γ.

(A2) The speed v_i of the ith camera satisfies $|v_i| \in \{0, v_i^{\max}\}$, with $v_i^{\max} \in \mathbb{R}_{>0}$.



(a) (b)

Fig. 2. (a) f.o.v.s of two cameras observing a corridor. In their resting state, the ground plane f.o.v.s are sufficiently wide to span the width of the corridor. Since the ground plane f.o.v. is formed by intersecting a cone with a plane, this condition guarantees that each f.o.v spans the width of the corridor as the camera sweeps the corridor. (b) Two neighboring f.o.v.s forming an intersection that spans the corridor width. The time instant at which this first happens is before the f.o.v. centers of the cameras are aligned. Furthermore, notice that each camera has a rectangular subsection whose width spans the width of the corridor. This rectangle is shown for camera c_i as the area between the green dotted lines, and for camera c_j as the area between the red dotted lines.

For assumption (A2) to be satisfied, we let each camera be equipped with a low-level controller that maintains the speed of its f.o.v. at 0 or $v_i^{\max}$.[1] Let $v^{\max} = \max\{v_1^{\max}, \ldots, v_n^{\max}\}$.

Let $x_i : \mathbb{R}_{\geq 0} \to \Gamma$ be a map, such that $x_i(t)$ specifies the position on Γ of the ith f.o.v. at time t. We define the patrolling window $A_i = [\ell_i, r_i] \subseteq \Gamma$ of camera c_i as the smallest segment of Γ containing the f.o.v. of camera c_i at all times, where ℓ_i and r_i denote the start and end points of the segment A_i, respectively. We assume the patrolling windows to be given and constant in time (except for our analysis in Section V). We additionally assume that $\ell_1 = 0$, $r_n = L$, and $\ell_i = r_{i-1}$, with $i = 2, \ldots, n$, so that $\{A_1, \ldots, A_n\}$ is a partition of Γ.[2] Finally, let d_i be the length of A_i, let $d^{\max} = \max\{d_1, \ldots, d_n\}$ and $d^{\min} = \min\{d_1, \ldots, d_n\}$, and define the longest cameras sweeping time as $\tau^{\max} = \max\{\tau_1, \ldots, \tau_n\}$, where $\tau_i = d_i / v_i^{\max}$ is the sweeping time of camera c_i.

A cameras trajectory is an array $X = \{x_1, \ldots, x_n\}$ of n continuous functions describing the motions of the cameras f.o.v. on Γ. We focus on periodic cameras trajectories, for which there exists a duration $T \in \mathbb{R}_{\geq 0}$ such that $X(t + T) = X(t)$ or, equivalently, $x_i(t + T) = x_i(t)$ for all $i \in \{1, \ldots, n\}$. We say that a cameras trajectory is synchronized if there exists a time $t_i \in [0, T]$ such that $x_i(t_i) = r_i = l_{i+1} = x_{i+1}(t_i)$ for each pair of neighboring cameras c_i and c_{i+1}.

Remark 1 (2-D Environment and f.o.v.): Our assumptions of 1-D environment and point-wise f.o.v. do not prevent applicability of our results to practical cases. To observe this, consider a surveillance task for a corridor of uniform width, and assume that the cameras f.o.v.s are 2-D surfaces on the ground plane. In addition, assume that the f.o.v. of each camera is sufficiently wide to span the width of the corridor when the camera is in its resting state, that is, when it is viewing the area directly below its physical position. Finally, assume that cameras sweep the environment, and that f.o.v.s of neighboring cameras form an intersection that spans the width of the corridor at some locations (Fig. 2). Partition the environment into patrolling windows, in a way that each camera can entirely sweep its assigned region. Let τ_i be the time needed by camera c_i to sweep its region, and let $d_i = \tau_i v_i$, where v_i is the speed of camera c_i. It is now clear that: 1) synchronization of the cameras depend only upon τ_i, v_i, and d_i, as captured by our simplified framework; 2) our synchronization algorithm is applicable for the realistic case of cameras with 2-D f.o.v.s, since it only requires the parameters τ_i, d_i, and v_i to be implemented; and 3) the detection performance obtained on our simplified camera model is a conservative bound for the performance with cameras with 2-D f.o.v.s. At each time, a camera with point-wise f.o.v. can only detect intruders contained in a region of zero area. Furthermore, if we assume that the intruders only appear at times and locations for which they are not detected immediately, then the performance will be the same as in our simplified camera model. A related example is in Section IV-A. □

B. Model of Intruder and Performance Functions

We consider the problem of detecting intruders appearing at random times and moving on Γ. We model an intruder as an arbitrarily fast point on Γ, and we let the continuous map $\mathcal{I}_{t_0,p_0} : \mathbb{R}_{\geq t_0} \to \Gamma$ be defined such that $\mathcal{I}_{t_0,p_0}(t)$ describes the position of the intruder at a time t, where t_0 and $p_0 = \mathcal{I}_{t_0,p_0}(t_0)$ are the time and location at which the intruder appears, respectively. We focus on smart intruders, which have full knowledge of the cameras trajectory and select their motion to avoid detection for as long as possible. More formally, given an initial time $t_0 \in \mathbb{R}_{\geq 0}$, an initial point $p_0 \in \Gamma$, and a cameras trajectory X, the trajectory $\mathcal{I}^*_{t_0,p_0}$ of a smart intruder satisfies

$$T_{\text{det}}(\mathcal{I}^*_{t_0,p_0}, X) = \max_{\mathcal{I}_{t_0,p_0}} T_{\text{det}}(\mathcal{I}_{t_0,p_0}, X) - t_0$$

where $T_{\text{det}}(\mathcal{I}_{t_0,p_0}, X)$ is the time at which the intruder is detected by the cameras, that is

$$t_d^*(p, t_0) = \min\{\{t \ : \ p(t) \in X(t), t \geq t_0\} \cup \{\infty\}\}$$

Notice that the trajectory $\mathcal{I}^*_{t_0,p_0}$ is in general not unique.

We consider two criteria for the detection performance of a T-periodic cameras trajectory, namely the WDT, and the ADT. These two criteria are formally defined as

$$\text{WDT}(X) = \sup_{t_0,p_0} T_{\text{det}}(\mathcal{I}^*_{t_0,p_0}, X) - t_0 \qquad (1)$$

[1] For instance, to move the camera f.o.v. along its panning direction, the controller may set the panning velocity of the ith camera to $\dot{\alpha}_i = v_i^{\max}/(a_i \sec^2(\alpha))$, where α denotes the panning angle, and a_i is the distance of the ith camera from Γ. See Section IV-B for a related example.

[2] As discussed below, this assumption ensures detectability of intruders (4).

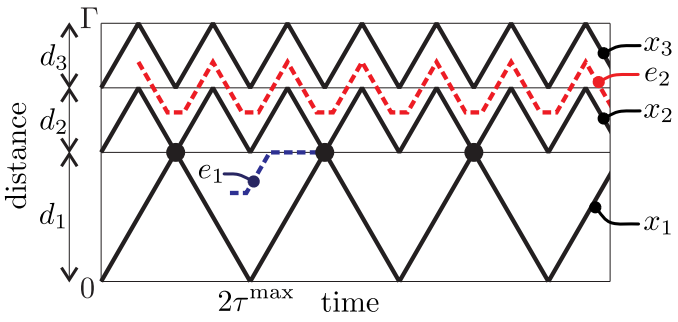


Fig. 3. This figure shows a $2\tau^{\max}$-periodic cameras trajectory in which cameras c_1 and c_2 are synchronized ($x_1(t) = x_2(t)$ for $t = k\tau^{\max}$ with $k \in \mathbb{N}_{>0}$; see Section II-A), while cameras c_2 and c_3 are not synchronized. Notice that, because of the synchronization among cameras, intruder e_1, and in fact any smart intruder appearing between cameras c_1 and c_2, is detected at time $k\tau^{\max}$, for some $k \in \mathbb{N}_{>0}$. Therefore, the WDT for intruders appearing between cameras c_1 and c_2 is $2\tau^{\max}$. Intruder e_2, and any smart intruder appearing between cameras c_2 and c_3, may avoid detection by properly choosing its trajectory.

and

$$\mathrm{ADT}(X) = \frac{1}{TL}\int_0^T \int_\Gamma T_{\det}(\mathcal{I}^*_{t,p}, X) - t \; dpdt. \tag{2}$$

In other words, the WDT criterion measures the longest time that a smart intruder may remain in the environment without detection, while the ADT criterion measures the average time that a smart intruder may remain in the environment without detection, over the boundary Γ and the periodicity T.

The WDT criterion for static intruders, namely WDTs, is defined in [30] as

$$\mathrm{WDTs}(X) = \sup_{t_0, p_0}\{t - t_0 \; : \; t \geq t_0 p_0 \in X(t)\} \tag{3}$$

and it corresponds to the longest time for the cameras to detect a static intruder, or simply event, along Γ. We next informally discuss the relation between WDT and WDTs, and we refer the reader to [9], [10], and [30] for a proof of these results. Let

$$\mathrm{WDT}^* = \min_X \; \mathrm{WDT}(X), \text{ and } \mathrm{WDTs}^* = \min_X \; \mathrm{WDTs}(X).$$

Clearly, $\mathrm{WDT}(X) \geq \mathrm{WDTs}(X)$ for every cameras trajectory X, as static intruders do not move to avoid camera f.o.v.s. For instance, as shown by the example in Fig. 3, if a cameras trajectory X is not synchronized but covers every location of Γ, then $\mathrm{WDTs}(X) < \infty$ and $\mathrm{WDT}(X) = \infty$. In addition, because the patrolling windows define a partition of Γ, it can be easily verified that the static WDT satisfies

$$\mathrm{WDTs}^* = 2\tau^{\max}$$

and that any $2\tau^{\max}$-periodic cameras trajectory achieves minimum static WDT. Similarly, any synchronized $2\tau^{\max}$-periodic cameras trajectory X satisfies (Fig. 3)

$$\mathrm{WDT}(X) = 2\tau^{\max}. \tag{4}$$

Since the f.o.v.s of neighboring cameras intersect at least once in any interval of length $2\tau^{\max}$, intruders cannot avoid detection for more than $2\tau^{\max}$. Thus, any synchronized $2\tau^{\max}$-periodic cameras trajectory achieves minimum WDT and WDTs). This discussion motivates us to restrict our attention to periodic and synchronized cameras trajectories.

Problem 1 (Design of Cameras Trajectories): Consider an open path partitioned among a set of n cameras, and let $\tau^{\max}$ be the longest cameras sweeping time. Design a cameras trajectory X^* satisfying

$$\mathrm{ADT}(X^*) = \mathrm{ADT}^* = \min_{X \in \Omega} \; \mathrm{ADT}(X)$$

where Ω is the set of all synchronized $2\tau^{\max}$-periodic cameras trajectories.

Remark 2 (Optimal Patrolling Windows): We assume that the patrolling windows are given and form a partition of the path Γ. With these assumptions, the WDT satisfies $\mathrm{WDT}^* \geq \mathrm{WDTs}^* \geq 2\tau^{\max}$, and any synchronized $2\tau^{\max}$-periodic cameras trajectory achieves the lower bound (see the above discussion).

If the patrolling windows are not given but are still required to be a partition of Γ, then the longest cameras sweeping time, and hence the worst case detection performance, can be minimized by solving a min-max graph partitioning problem [9], [16], [30]. We will discuss this aspect in Section V, where we develop an algorithm to simultaneously partition the environment and coordinate the motion of the cameras to optimize the detection of intruders.

If the patrolling windows are not required to be a partition of Γ, then the bound WDTs^* may be smaller than $2\tau^*$. We refer the reader interested in this case to [5, Conjecture 1] and [31]. □

A second focus of this paper is the design of distributed algorithms to coordinate the cameras along a desired trajectory. We consider a distributed scenario in which cameras c_i and c_j are allowed to communicate at time t only if $|j - i| = 1$ (neighboring cameras) and $x_i(t) = x_j(t)$. Although conservative, this assumption allows us to design algorithms implementable with many low-cost communication devices; see Section IV-B. Notice that additional communications cannot decrease the performance of our algorithms.

Problem 2 (Distributed Coordination): For a set of n cameras on a 1-D open path, design a distributed algorithm to coordinate the cameras along a trajectory with minimum ADT of smart intruders.

III. Equal-Waiting Cameras Trajectory and Coordination Algorithm

In this section, we present our results for Problems 1 and 2. In particular, we propose a cameras trajectory with performance guarantees for the ADT, and a distributed algorithm for the cameras to converge to such a trajectory. We remark that, in general, cameras trajectories with minimum ADT can be designed via standard, yet computationally intensive, optimization techniques. Such an approach is adopted for instance in [32], where the problem of designing robots' strategies is cast into an optimal control framework and a gradient-based algorithm is used to compute a locally optimal solution, and in [33, Ch. 7], where optimal cameras trajectories are explicitly computed for environments satisfying $d^{\max} < 2d^{\min}$ and for cameras moving at unit speed. The approach taken in this

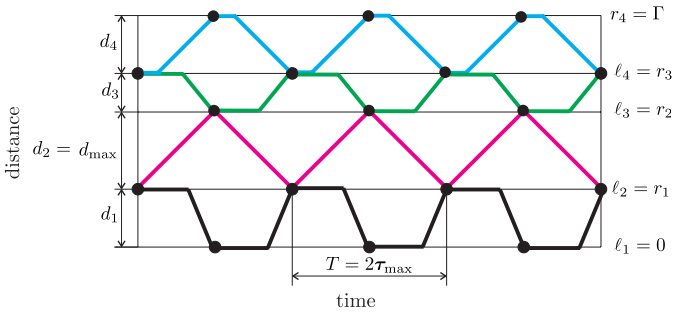


Fig. 4. Equal-waiting trajectory for cameras. Notice that: 1) the cameras are synchronized; 2) the trajectory is $2\tau_{\max}$-periodic; and 3) the waiting time of each camera is the same at both its boundaries.

Algorithm 1: Equal-Waiting Trajectory (Camera c_i)

Input : $\tau^{\max}, r_i, \ell_i, \tau_i, v_i^{\max}$;

Set : $\omega_i(k) = (k+1)\tau^{\max} - \tau_i$, $k = -1, 0, \ldots$;

if *i is odd* **then**

$x_i(t) = r_i$ for $\omega_i(k-1) + \tau_i \le t \le \omega_i(k)$

$x_i(t) = -v_i^{\max}(t - \omega_i(k)) + r_i$ for $\omega_i(k) \le t \le \omega_i(k) + \tau_i$

$x_i(t) = \ell_i$ for $\omega_i(k) + \tau_i \le t \le \omega_i(k+1)$

$x_i(t) = v_i^{\max}(t - \omega_i(k+1)) + \ell_i$ for $\omega_i(k+1) \le t \le \omega_i(k+1) + \tau_i$

else if *i is even* **then**

$x_i(t) = \ell_i$ for $\omega_i(k-1) + \tau_i \le t \le \omega_i(k)$

$x_i(t) = v_i^{\max}(t - \omega_i(k+1)) + \ell_i$ for $\omega_i(k) \le t \le \omega_i(k) + \tau_i$

$x_i(t) = r_i$ for $\omega_i(k) + \tau_i \le t \le \omega_i(k+1)$

$x_i(t) = -v_i^{\max}(t - \omega_i(k)) + r_i$ for $\omega_i(k+1) \le t \le \omega_i(k+1) + \tau_i$

section is different, as our objective is to gain a comprehensive insight into the cameras surveillance problem, and to design surveillance strategies that are easily implementable and reconfigurable.

The cameras trajectory that we propose can informally be described as follows.

Equal-Waiting Trajectory: Each camera continuously sweeps its patrolling window at maximum speed, and it stops for some waiting time when its f.o.v. reaches an extreme of its patrolling window. For each camera, the waiting times at its two extremes are equal to each other. In addition, the waiting times of each camera are chosen so that the resulting cameras trajectory is synchronized and periodic. See Fig. 4 for an illustrative explanation. □

Because we let each camera wait for the same amount of time at the two extremes of its patrolling window, we call this cameras trajectory equal-waiting trajectory. An example of equal-waiting trajectory is in Fig. 4, and a formal description is in Algorithm 1.

As discussed in Section II, the equal-waiting cameras trajectory is optimal with respect to the WDT criterion. Indeed, the equal-waiting cameras trajectory is synchronized and $2\tau^{\max}$-periodic. We now characterize the ADT performance of the equal-waiting trajectory. A proof of this result is postponed to the Appendix.

Theorem 1 (Performance of Equal-Waiting Trajectories): Let X^{eq} be the equal-waiting trajectory defined in Algorithm 1.

1) The ADT of a smart intruder satisfies the lower bound
$$\text{ADT}^* \ge \frac{1}{L}\sum_{i=1}^{n} v_i^{\max}\tau_i^2. \tag{5}$$
2) The equal-waiting trajectory X^{eq} satisfies
$$\text{ADT}(X^{\text{eq}}) = \frac{1}{2}\left(\tau^{\max} + \frac{1}{L}\sum_{i=1}^{n} v_i^{\max}\tau_i^2\right). \tag{6}$$
3) The equal-waiting trajectory X^{eq} satisfies
$$\frac{\text{ADT}(X^{\text{eq}})}{\text{ADT}^*} \le \min\left\{\frac{\tau^{\max}+\tau^{\min}}{2\tau^{\min}}, \frac{(n+1)d^{\max}}{2d^{\min}}\right\}. \tag{7}$$
4) If $v_i^{\max} = 1$ for all $i \in \{1, \ldots, n\}$, then the equal-waiting trajectory X^{eq} satisfies
$$\frac{\text{ADT}(X^{\text{eq}})}{\text{ADT}^*} \le \min\left\{\frac{d^{\max}+d^{\min}}{2d^{\min}}, \frac{3+\sqrt{n}}{4}\right\}. \tag{8}$$

The following comments are in order. First, the ADT of the equal-waiting trajectory is within a constant factor of the optimal if either $\tau^{\max}/\tau^{\min}$ or n are constant. Second, if all patrolling windows have the same sweeping time, that is $\tau^{\max} = \tau^{\min}$, then our equal-waiting trajectory is an optimal solution to Problem 1 (it achieves minimum WDTs and ADT). Moreover, our lower bound (5) is tight and holds with equality if $\tau^{\max} = \tau^{\min}$. Third, the lower bounds in Theorem 1 are independent in the ordering of the patrolling windows. Fourth, if:

1) all cameras move at unit speed;
2) there exists an index $h \in \{1, \ldots, n\}$ such that $d_h > d_i$ for all $i \in \{1, \ldots, n\} \setminus \{h\}$;
3) for all $i \in \{1, \ldots, n\} \setminus \{h\}$ the patrolling windows satisfy
$$d_i = \frac{d^{\max}}{1+\sqrt{n}} \tag{9}$$
then [see the proof of Theorem 1 and Fig. 5(b)]
$$\frac{\text{ADT}(X^{\text{eq}})}{\text{ADT}^*} = \frac{3+\sqrt{n}}{4}. \tag{10}$$

Fifth and finally, different cameras speeds can be considered in our bound (8). If $v_i^{\max}/v_j^{\max} \le C$ for all $i, j \in \{1, \ldots, n\}$ and for some $C \in \mathbb{R}$, then (see the proof of Theorem 1)
$$\frac{\text{ADT}(X^{\text{eq}})}{\text{ADT}^*} \le \min\left\{\frac{C\left(d^{\max}+d^{\min}\right)}{2d^{\min}}, \frac{2+C\left(1+\sqrt{n}\right)}{4}\right\}.$$

We now design a distributed feedback algorithm that steers the cameras toward an equal-waiting trajectory. Our algorithm to coordinate the cameras along an equal-waiting trajectory is described in Algorithm 2, where for convenience we set $x_0(t) = \ell_1$ and $x_{n+1}(t) = r_n$ at all times.

An informal description is shown in Algorithm 2.

Distributed Coordination: Camera c_i moves to ℓ_i (line 1) and, if $i > 1$, it waits until the f.o.v. of its left neighboring camera c_{i-1} occupies the same position (line 2). Then, camera

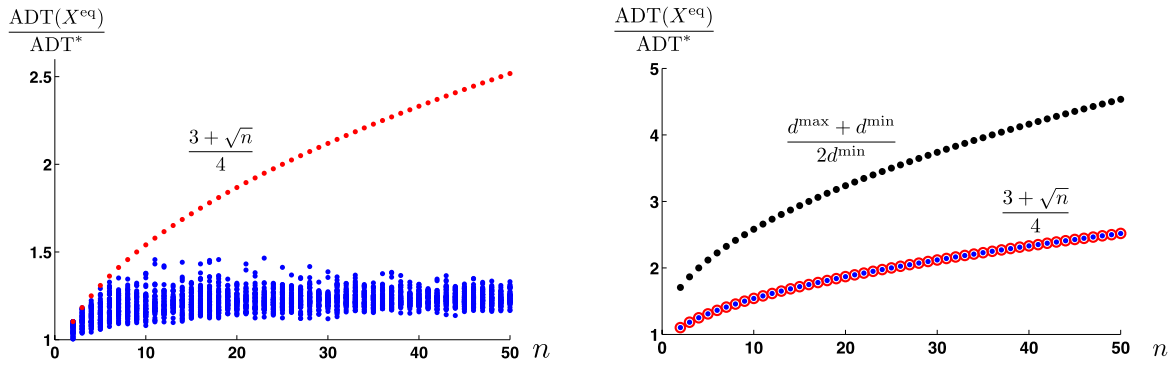


Fig. 5. (a) Ratio $\mathrm{ADT}(X^{\mathrm{eq}})/\mathrm{ADT}^*$ as a function of the number of cameras n (blue dots), and the bound $(3+\sqrt{n})/4$ in Theorem 1 (red dots). For the considered configurations, the bound $(d^{\max}+d^{\min})/(2d^{\min})$ is much larger than the experimental data, and it is not considered here. The lengths d_i of the patrolling windows are uniformly distributed in the interval $(0, 1]$, with $d_1 = d^{\max} = 1$. We assume that cameras have unit speed. In Fig. 5(b), we report the ratio $\mathrm{ADT}(X^{\mathrm{eq}})/\mathrm{ADT}^*$ as a function of the number of cameras n (blue dots), and the bounds in Theorem 1 (black dots and red circles). The lengths d_i of the patrolling windows are chosen as $d_1 = d^{\max} = 1$ and $d_i = (1+\sqrt{n})^{-1}$ for all $i \in \{2, \ldots, n\}$. As predicted by (10), the performance bound in (8) is tightly achieved.

Algorithm 2 Distributed Camera Coordination Along an Equal-Waiting Trajectory (Camera c_i)

Input : $\tau^{\max}, r_i, \ell_i, \tau_i, v_i^{\max}$;

Set : $\omega_i = \tau^{\max} - \tau_i$, $x_0(t) = \ell_1$ and $x_{n+1}(t) = r_n$ $\forall t$;

1 Move towards ℓ_i with $|v_i(t)| = v_i^{\max}$;

while *True* **do**

2 **if** $x_i(t) = x_{i-1}(t)$ *or* $x_i(t) = x_{i+1}(t)$ **then**

3 Wait until time $t + \omega_i$;

4 Move towards the opposite boundary with $|v_i(t)| = v_i^{\max}$;

c_i stops as specified in Algorithm 1 (line 3), and finally move to r_i (line 4). Camera c_1 (resp. c_n) moves to r_2 (resp. ℓ_{n-1}) as soon as its f.o.v. arrives at ℓ_1 (resp. r_n). □

It should be observed that, by construction, cameras sweep their patrolling windows (lines 1 and 4), and that the cameras trajectory obtained via Algorithm 2 is synchronized and equal-waiting. Moreover, since cameras wait until the f.o.v. of a neighboring camera occupies the same position (line 3), our coordination algorithm is robust to cameras failures and motion uncertainties. A related example is in Section IV-A. Regarding the implementation of Algorithm 2, notice that each camera is required to know the endpoints of its patrolling window, its sweeping time and the maximum sweeping time in the network, and to be able to communicate with neighboring cameras. The following theorem characterizes the convergence properties of Algorithm 2, where we write $X(t \geq \bar{t})$ to denote the restriction of the trajectory $X(t)$ to the interval $t \in [\bar{t}, \infty)$.

Theorem 2 (Convergence of Algorithm 2): For a set of n cameras with sweeping times $\tau_1, \ldots, \tau_n$, let $X(t)$ be the cameras trajectory generated by Algorithm 2. Then, $X(t \geq n\tau^{\max})$ is an equal-waiting trajectory.

Proof: Notice that the f.o.v. of camera c_1 coincides with the f.o.v. of camera c_2 within time $\max\{2\tau_1, \tau_2\} \leq 2\tau^{\max}$. Then, the f.o.v. of camera c_i coincides with the f.o.v. of camera c_{i+1} within time $(i+1)\tau^{\max}$. Hence, within time $n\tau^{\max}$ the cameras trajectory coincides with an equal-waiting trajectory in Algorithm 1. The claimed statement follows. ■

Notice that our camera's trajectory and coordination algorithm are easy to compute, valid for every number of cameras and environment configuration, and their performance are guaranteed to be within a bound of the optimum.

IV. Numerical Studies and Experiment

In this section, we report the results of our numerical studies and experiments. Besides validating our theory, these results show that our models are accurate, that our algorithms can be implemented on real hardware, and that our algorithms are robust to sensor noise and model uncertainties.[3]

A. Numerical Studies

Five numerical studies are presented in this section. For our first numerical study, we let the number of cameras n vary from 2 to 50. For each value of n, we generate 50 sets of patrolling window with lengths $\{d_1, \ldots, d_n\}$, where $d_1 = d^{\max} = 1$ m, and d_i is uniformly distributed within the interval $(0, 1]$ m, for all $i \in \{2, \ldots, n\}$. For each configuration, we let $v_i^{\max} = 1$ m/s for all cameras, we design the equal-waiting trajectory X^{eq}, and we evaluate the cost $\mathrm{ADT}(X^{\mathrm{eq}})$ and the lower bound ADT^* from (5). We report the result of this paper in Fig. 5(a). Notice that, when the number of cameras is large and the lengths of the patrolling windows are uniformly distributed, the bound in (8) is conservative. On the other hand, if the lengths of the patrolling windows are chosen as in (9), then the bound in (8) is tightly achieved (Fig. 5(b)).

[3]A video of our experiments can be found at http://www.fabiopas.it/CameraNetworkCoordinationForIntruderDetection.avi.

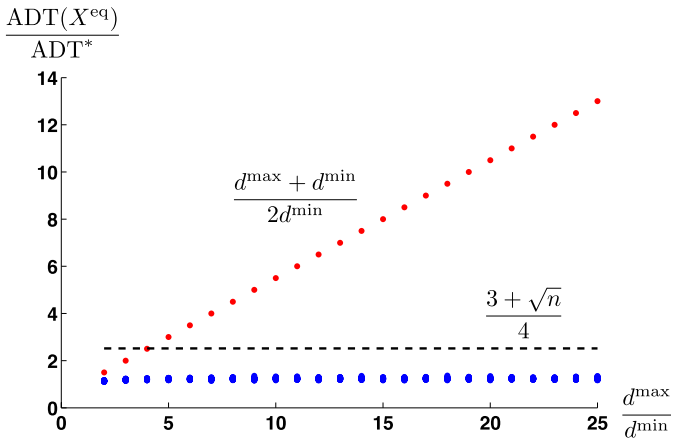


Fig. 6. Ratio $\mathrm{ADT}(X^{\mathrm{eq}})/\mathrm{ADT}^*$ as a function of $d^{\max}/d^{\min}$ (blue dots), and the bounds in Theorem 1. For each value of $d^{\max}/d^{\min}$, the lengths of the patrolling windows are uniformly distributed in the interval $[d^{\min}/d^{\max}, 1]$, with $d_1 = d^{\max} = 1$. Notice that the theoretical bounds are compatible with the experimental data.

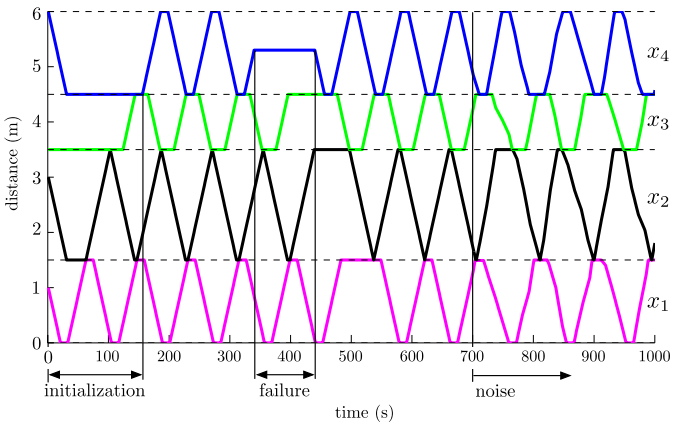


Fig. 7. Validation of Algorithm 2 for a set of four cameras with unit speed. Cameras start at random positions inside their patrolling window and achieve coordination at time 150. Notice that the algorithm recovers from the temporary failure of camera c_4 between time 340 and 440. Moreover, the coordination performance of the algorithm degrade gracefully in the presence of noise affecting the cameras motion of the cameras (time 700). In this numerical study, the cameras motion noise is assumed to be normally distributed with mean 0.2 and unit standard deviation.

For our second numerical study, we let the number of cameras be fixed (50 cameras), and we vary the value $d^{\max}/d^{\min}$ between 2 and 25. Specifically, we let $d_1 = d^{\max} = 1$ m, and d_i, with $i = 2, \ldots, 50$ be uniformly distributed within the interval $[d^{\min}/d^{\max}, 1]$ m. For each value of $d^{\max}/d^{\min}$ we generate 50 sets of patrolling windows with lengths $\{d_1, \ldots, d_n\}$, compute the equal-waiting trajectory X^{eq}, evaluate the cost $\mathrm{ADT}(X^{\mathrm{eq}})$, and compute the lower bound in (5). The results of this numerical study are shown in Fig. 6, where we observe that the theoretical bounds derived in Theorem 6 are compatible with the experimental data.

In our third numerical study, we validate the effectiveness of our coordination algorithm. We consider a set of four cameras with prespecified patrolling windows and unit speed. The cameras trajectory generated by Algorithm 2 is shown in Fig. 7. Observe that our coordination algorithm drives the cameras toward an equal-waiting trajectory, and it is robust to failures and motion uncertainty. In particular: 1) coordination is achieved for cameras starting at random initial positions; 2) the algorithm is robust to temporary cameras failure; and 3) the ADT degrades gracefully in the presence of motion uncertainties.

In our fourth numerical study, we adopt our algorithms for the surveillance of a 2-D corridor using cameras with 2-D f.o.v.s. We consider a corridor of uniform width 1 m and length 100 m, that is, $\Gamma_{2\mathrm{D}} = [0, 100] \times [0, 1]$. We place cameras $c_1, \ldots, c_6$ along the corridor at physical positions $(15, 0.5)$, $(35, 0.5)$, $(45, 0.5)$, $(70, 0.5)$, $(80, 0.5)$, $(95, 0.5)$, respectively, and at a height of 5 m. The cameras patrolling windows are as follows:

$$
\begin{aligned}
(A_1)_{2\mathrm{D}} &= [(\ell_1)_{2\mathrm{D}}, (r_1)_{2\mathrm{D}}] \times [0, 1] = [0, 20] \times [0, 1] \\
(A_2)_{2\mathrm{D}} &= [(\ell_2)_{2\mathrm{D}}, (r_2)_{2\mathrm{D}}] \times [0, 1] = [20, 40] \times [0, 1] \\
(A_3)_{2\mathrm{D}} &= [(\ell_3)_{2\mathrm{D}}, (r_3)_{2\mathrm{D}}] \times [0, 1] = [40, 55] \times [0, 1] \\
(A_4)_{2\mathrm{D}} &= [(\ell_4)_{2\mathrm{D}}, (r_4)_{2\mathrm{D}}] \times [0, 1] = [55, 75] \times [0, 1] \\
(A_5)_{2\mathrm{D}} &= [(\ell_5)_{2\mathrm{D}}, (r_5)_{2\mathrm{D}}] \times [0, 1] = [75, 90] \times [0, 1] \\
(A_6)_{2\mathrm{D}} &= [(\ell_6)_{2\mathrm{D}}, (r_6)_{2\mathrm{D}}] \times [0, 1] = [90, 100] \times [0, 1].
\end{aligned}
$$

We consider the 2-D ground plane f.o.v. of each camera, which is formed by the intersection of the ground plane with a 3-D cone originating from the lens of the camera and spreading out uniformly with angle β (Fig. 2). Let (x_i, y_i) be the coordinates of the ith f.o.v., where $y_i(t) = 0.5$ at all times. Finally, an intruder is detected at a certain time if its position belongs to the f.o.v. of some cameras.

We implement Algorithm 2 as follows. We let each camera sweep its patrolling window. At each time t, let $x_i^{\ell}(t), x_i^{r}(t)$ define the largest rectangle contained in the camera f.o.v., that is, the smallest and largest values, respectively, such that the set $[x_i^{\ell}(t), x_i^{r}(t)] \times [0, 1]$ is contained in the ith f.o.v. at time t. Let t_1, t_2 satisfy $x_i^{\ell}(t_1) = (\ell_i)_{2\mathrm{D}}$ and $x_i^{r}(t_2) = (r_i)_{2\mathrm{D}}$. Finally, define the 1-D patrolling window of camera c_i to be $A_i = [\ell_i, r_i] = [x_i(t_1), x_i(t_2)]$: this ensures that the ith camera sweeps its 2-D patrolling window by moving its f.o.v. from $(\ell_i, 0.5)$ to $(r_i, 0.5)$, and it allows to use Algorithm 2 for 2-D patrolling problems.

Let intruders to appear at arbitrary locations and times, in a way that they are not immediately detected by the cameras. For some values of the angle β, we calculate the ADTs over 100 simulations, and compare them with the bounds in Theorem 1 for the corresponding 1-D setups. In particular, in Fig. 8, we show the detection times for the 2-D problems, and those predicted by the corresponding 1-D setups for different values of β. Notice that the WDT and ADT for the 2-D cases satisfy the bounds in Theorem 1 for the corresponding 1-D problems.

Finally, in our fifth numerical study, we consider a particular configuration of cameras, namely partially homogeneous f.o.v., and we compare the equal-waiting trajectory with a cameras trajectory with minimum ADT. In particular, consider a set of four cameras moving at unit speed. Let the lengths of their patrolling windows be $d_1 = 4$, $d_2 = 3$, $d_3 = 3$, and $d_4 = 4$, respectively. As shown in [33], since $d^{\max} < 2d^{\min}$, a cameras trajectory with minimum ADT can be computed in

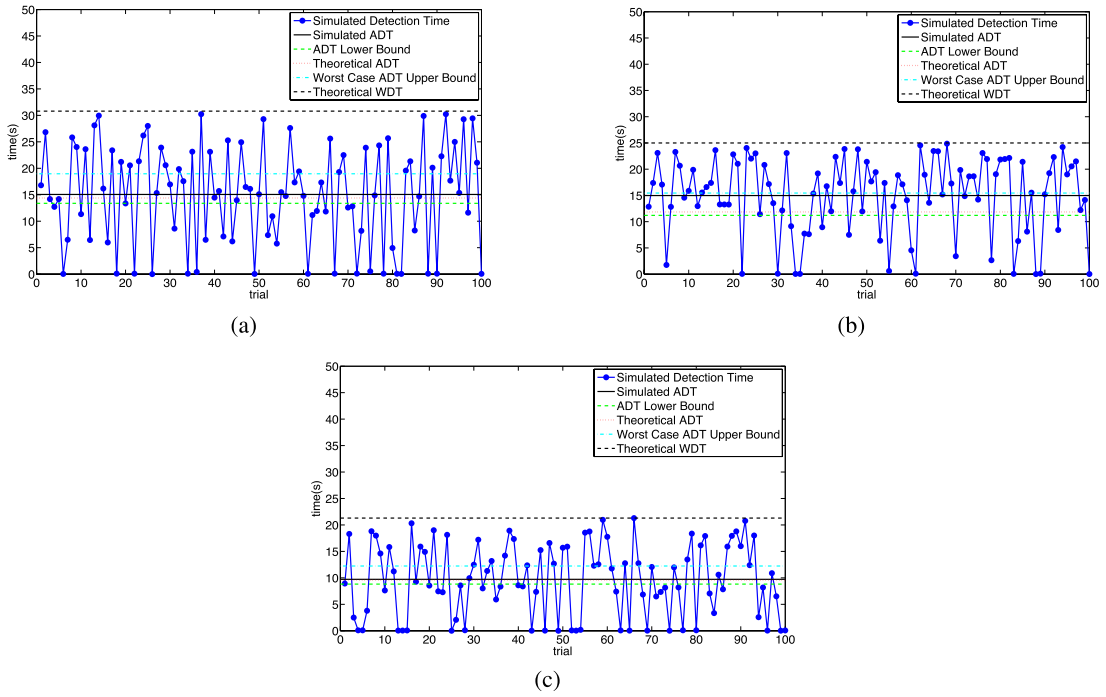


Fig. 8. Results from our numerical example using 2-D f.o.v.s. In this numerical study, we compute the ADTs over 100 simulations for spread angles $\beta = 6, 10, 15$ degrees, and compare the results with the bounds predicted by Theorem 2. (a)–(c) Results for $\beta = 6$, $\beta = 10$, and $\beta = 15$, respectively. The detection times of this simulation are depicted by a solid blue line. Notice that the detection times are smaller than the upper bound predicted in Section II (dashed black line). The average of the simulated detection times are depicted by a solid black line. For the considered configuration of cameras, the lower bound ADT* in (5) (dashed green line), ADT(X^{eq}) as in (6) (dotted red line), and the worst case upper bound (dashed light blue line), which is calculated by multiplying the lower bound on ADT* by the quantity $(\tau^{\max} + \tau^{\min})/2\tau^{\min}$ from (7) are reported for the corresponding 1-D configurations.

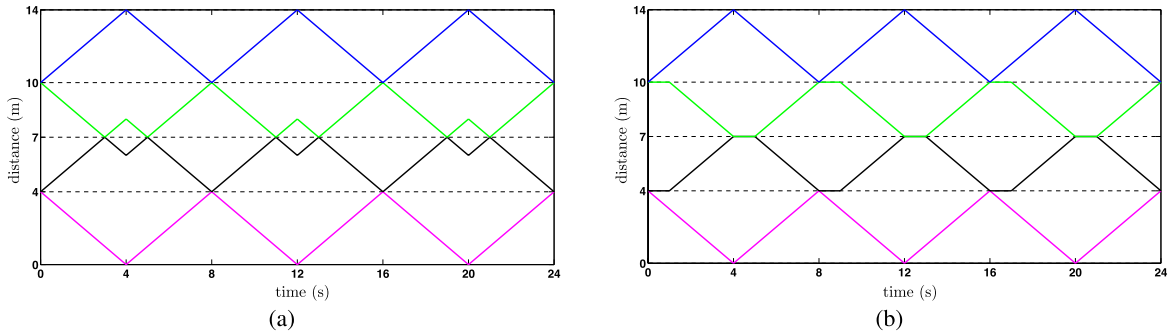


Fig. 9. For a set of four cameras with unit velocity, let the lengths of the patrolling windows be $d_1 = 4$, $d_2 = 3$, $d_3 = 3$, and $d_4 = 4$. (a) Camera's trajectory X^* with minimum ADT. The trajectory is computed as in [33] for the partially homogeneous f.o.v. In Fig. 9(b), we report the equal-waiting trajectory X^{eq}. Notice that the two trajectories differ in the second and third patrolling windows. It can be verified using (1) and Theorem 1 that ADT(X^*) ≈ 3.52 while ADT(X^{eq}) ≈ 3.79.

closed form. In Fig. 9, we compare a cameras trajectory with minimum ADT with our equal-waiting trajectory.

B. Experiment

In this section, we detail the experiments we have conducted to validate our theoretical findings and numerical results. For our experiments, we use a network of six AXIS 213 PTZ network cameras mounted along a square perimeter. To simulate a 1-D environment, we assign each camera responsibility for surveilling a segment of the perimeter and assume that camera c_1 and camera c_6 have no left and right neighbors, respectively. Movement of the cameras is restricted to a panning motion and is controlled in such a way as to keep the center of the f.o.v. moving at constant speed. Each camera is equipped with a low-level detection algorithm to alert the user when an intruder enters its f.o.v. All programming of the cameras is performed in Python, using the OpenCV computer vision package for image processing. A diagram of our camera network and a table with our experimental parameters are shown in Fig. 10 and Table I.

In our first experiment, we validate our distributed coordination algorithm to control the motion of the cameras. Fig. 11 shows the results of our experiment. Notice that the algorithm steers the cameras into an equal-waiting trajectory within time $6\tau^{\max}$, as predicted by Theorem 2. Since the cameras are all

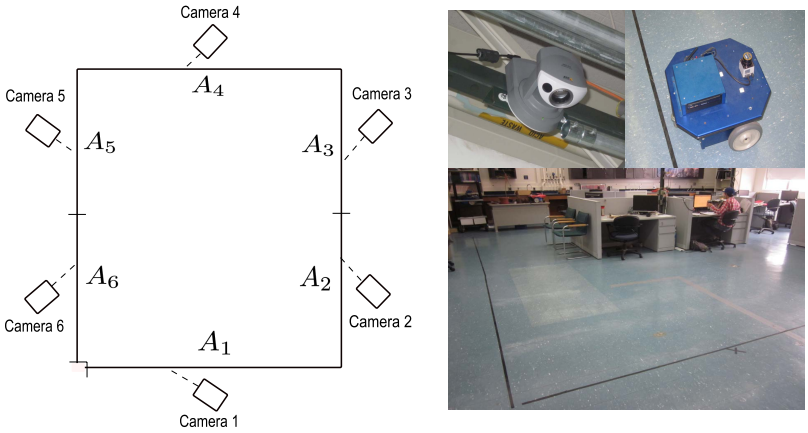


Fig. 10. Illustration of the experimental setup and photos of the hardware.

TABLE I
RELEVANT EXPERIMENTAL PARAMETERS

Camera	d_i(cm)	$v_i^{\max}$(cm/s)	τ_i (s)
c_1	624.3	20.8	30.0
c_2	290.3	18.0	16.1
c_3	291.0	20.6	14.1
c_4	619.3	21.1	29.0
c_5	331.5	19.0	17.4
c_6	232.7	17.3	13.5

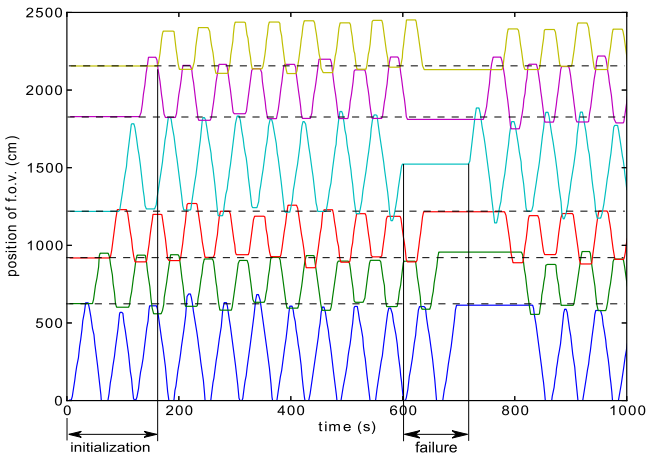


Fig. 11. Cameras trajectories as obtained from our experimental implementation of Algorithm 2. See Table I for the cameras parameters. Notice that the trajectory is robust to noise, as well as small overshoots and undershoots introduced by hardware and network uncertainty. These inaccuracies in the individual camera trajectories do not significantly affect coordination. The cameras trajectory is also robust to momentary failures, as shown at time $t \approx 600$ s.

starting the experiment at their left boundary, we observe that the system reaches an equal-waiting trajectory in only slightly longer than $5\tau^{\max} = 150$ s. This is consistent with Theorem 2, since delays in communication and network bandwidth limits cause some lagging in our experimental implementation. To demonstrate the behavior of the algorithm under a camera failure, camera c_4 is stopped at time $t = 600$ s. Notice that the algorithm continues to function despite this temporary hardware failure.

In our second experiment, we focus on the WDT of intruders. We use an Erratic mobile robot from Videre Design to simulate a smart intruder. The robot is equipped with an onboard computer with Ubuntu Linux and uses Player/Stage to interface with the user and allow for manual steering. We assume that the cameras motion is controlled by Algorithm 2, and we run 40 trials where the Erratic robot enters the environment at specific times and locations (we let the Erratic robot move only along the first segment, that is, the segment with longest sweeping time), and it is manually driven to avoid detection for as long as possible. We report the results of our second experiment in Fig. 12(a), where we notice that the theoretical WDT is a relatively tight bound for the experimental WDT.

In our third experiment, we focus on the ADT of intruders. As in our second experiment, we let the cameras motion be controlled by Algorithm 2, and we use an Erratic robot as an intruder. We run 40 trials where the Erratic robot enters the environment at random times and locations, and it is manually

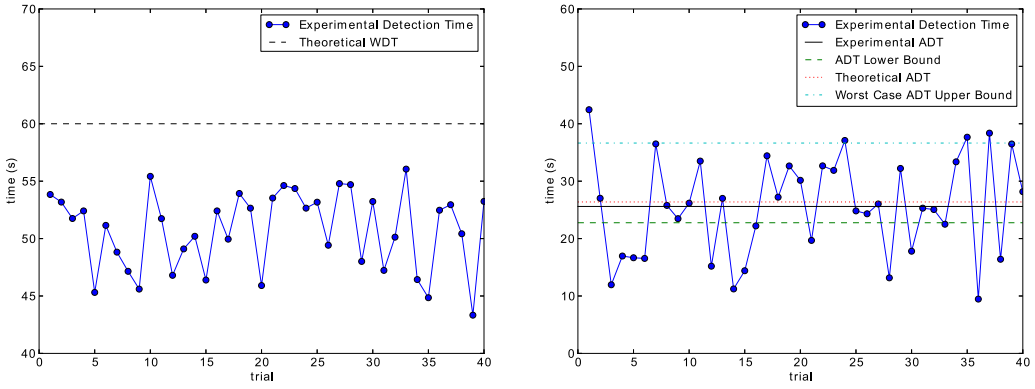


Fig. 12. (a) Detection times for our second experiment, in which smart intruders appear at worst case times and locations. The detection times of this experiment are depicted by a solid blue line. Notice that the detection times are smaller than the upper bound predicted in Section II. (b) Detection times for each trial of our third experiment, in which smart intruders appear at random times and locations. The detection times of this experiment are depicted by a solid blue line. The solid black line corresponds to the average of the experimental detection times. For the considered configuration of cameras, the lower bound ADT* in (5) (dashed green line), ADT(X^{eq}) as in (6) (dotted red line), and the worst case upper bound (dashed light blue line), which is calculated by multiplying the lower bound on ADT* by the quantity $(\tau^{\max}+\tau^{\min})/2\tau^{\min}$ from (7) are reported.

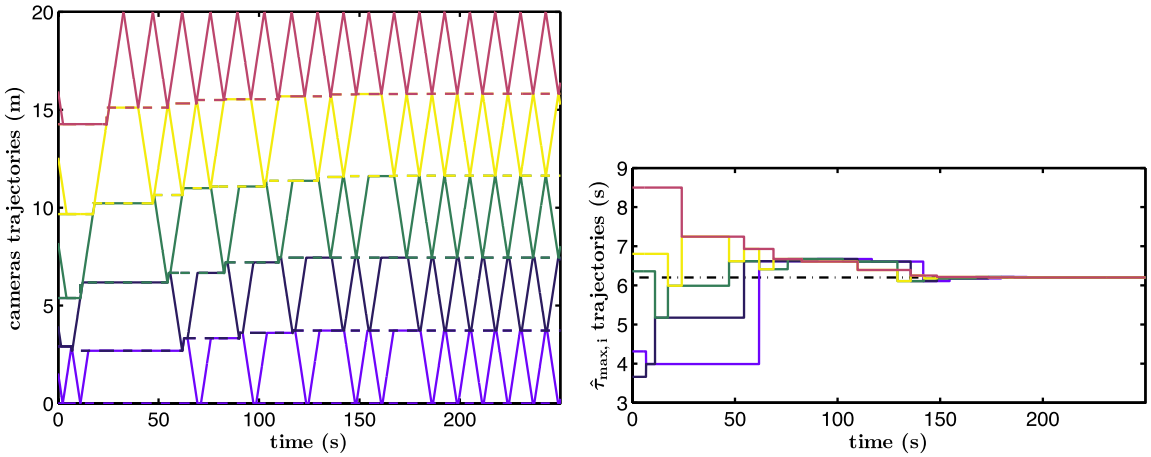


Fig. 13. Numerical study of REC with $n = 5$ cameras with maximum speed $v^{\max} = 0.67$ m/s and patrolling windows constraints. (a) Cameras trajectories starting from random positions. The dashed lines refer to the trajectories of the active boundaries. (b) Dynamics of the longest patrolling time $\hat{\tau}_i^{\max}$. Notice that $\hat{\tau}_i^{\max}$ converges to the optimal value $\tau^* = 6.2023$ s (dashed-dotted line).

driven to avoid detection for as long as possible. We report the results of our second experiment in Fig. 12(b), where we notice that the theoretical bounds in Theorem 1 are compatible with the experimental data (the slight difference is because of the theoretical value is calculated by considering all possible intruder initial locations and times).

We remark that there is a small amount of uncertainty in the execution of the algorithm by the cameras, resulting in small overshoots and undershoots in the individual camera trajectories. As we observe from Fig. 11, these small gaps, which are to be expected in practical applications, do not have a significant effect on the performance of the algorithm. We conclude that our experimental results validate our theory, our camera models, and our assumptions.

V. Distributed Cameras Reconfiguration

In this section, we describe an algorithm to reconfigure the cameras patrolling windows to improve the detection performance, and to allow the camera network to recover from a permanent camera failure and autonomously adapt to the addition and removal of cameras. We consider a symmetric gossip communication protocol among cameras, where communication is allowed only among neighboring cameras, and where each camera updates its patrolling window only after communication with a neighboring camera. Our REC is described in Algorithm 3, where D_i represents cameras visibility constraints ($A_i \subseteq D_i$).

An informal description of Algorithm 3 follows.

Cameras Reconfiguration: Camera c_i sweeps back and forth at maximum speed its patrolling windows A_i (line 1), and it updates A_i upon communication with neighboring cameras (lines 2–5). The update of A_i is performed so that, as time progresses, the cameras patrolling windows form a partition of the boundary that minimizes the longest sweeping time (line 5). Following Algorithm 2, cameras stop for a certain waiting time when their f.o.v. reaches an extreme of their patrolling window. These waiting times ensure that: 1) communication among

Algorithm 3 Cameras Reconfiguration (Camera c_i)

Input : $A_i = [\ell_i, r_i]$, $D_i = [\underline{\gamma}_i, \overline{\gamma}_i]$, $v_i^{\max}$;

Require : $\{A_1, \ldots, A_n\}$ is a partition of Γ;

Set $\hat{\tau}_i^{\max} = \tau_i$ and $q_i = c_i$;

1 Move according to Algorithm 2;

if *Communication between cameras* c_i *and* c_{i+1} **then**

2 Transmit ℓ_i, r_i, $\hat{\tau}_i^{\max}$, q_i to c_{i+1};

3 Receive ℓ_{i+1}, r_{i+1}, $\hat{\tau}_{i-1}^{\max}$, q_{i+1} from c_{i+1};

4 Compute $m = (\ell_i v_{i+1}^{\max} + r_{i+1} v_i^{\max}) \,/\, (v_i^{\max} + v_{i+1}^{\max})$;

5 Update r_i and ℓ_{i+1} as:

$$r_i = \ell_{i+1} = \begin{cases} m, & \text{if} \quad m \in [\underline{\gamma}_{i+1}, \overline{\gamma}_i], \\ \overline{\gamma}_i, & \text{if} \quad m > \overline{\gamma}_i, \\ \underline{\gamma}_{i+1}, & \text{if} \quad m < \underline{\gamma}_{i+1}; \end{cases}$$

6 Compute new value for $\tau_i = (r_i - \ell_i)/v_i^{\max}$;

7 **case** $q_i \geq c_i$

8 **if** $q_{i+1} > c_{i+1}$ **then**

9 $\hat{\tau}_i^{\max} = \hat{\tau}_{i+1}^{\max} = \max\{\tau_i, \tau_{i+1}, \hat{\tau}_{i+1}^{\max}\}$;

10 $q_i = q_{i+1} = \arg \max_{c_i, c_{i+1}, q_{i+1}} \{\tau_i, \tau_{i+1}, \hat{\tau}_{i+1}^{\max}\}$;

11 **else if** $q_{i+1} \leq c_{i+1}$ **then**

12 $\hat{\tau}_i^{\max} = \hat{\tau}_{i+1}^{\max} = \max\{\tau_i, \tau_{i+1}\}$;

13 $q_i = q_{i+1} = \arg \max_{c_i, c_{i+1}} \{\tau_i, \tau_{i+1}\}$;

14 **case** $q_i < c_i$

15 **if** $q_{i+1} \leq c_{i+1}$ **then**

16 $\hat{\tau}_i^{\max} = \hat{\tau}_{i+1}^{\max} = \max\{\tau_i, \tau_{i+1}, \hat{\tau}_i^{\max}\}$;

17 $q_i = q_{i+1} = \arg \max_{c_i, c_{i+1}, q_i} \{\tau_i, \tau_{i+1}, \hat{\tau}_i^{\max}\}$;

18 **else if** $q_{i+1} > c_{i+1}$ **then**

19 $\hat{\tau}_i^{\max} = \hat{\tau}_{i+1}^{\max} = \max\{\tau_i, \tau_{i+1}, \hat{\tau}_i^{\max}, \hat{\tau}_{i+1}^{\max}\}$;

20 $q_i = q_{i+1} = \arg \max_{c_i, c_{i+1}, q_i, q_{i+1}} \{\tau_i, \tau_{i+1}, \hat{\tau}_i^{\max}, \hat{\tau}_{i+1}^{\max}\}$;

neighboring cameras is maintained over time and 2) cameras trajectories are synchronized along an equal-waiting trajectory. As previously mentioned, in an equal-waiting trajectory the waiting times at the two extremes of the patrolling window are equally long. Finally, to achieve motion synchronization, the (time-varying) maximum sweeping time $\tau^{\max}$ is propagated across cameras during the execution of the algorithm. To do so, the auxiliary variable q_i is used by the ith camera to store the information about the camera associated with $\tau^{\max}$ (lines 7–20). □

For the analysis of Algorithm 3, notice that the patrolling window A_i is updated every time camera c_i communicates with a neighboring camera. Let $A_i(k)$ denote the ith patrolling window after k communications of camera c_i, and let $d_i(k)$ be the length of $A_i(k)$. We say that a cameras trajectory X is asymptotically T-periodic if there exists a duration $T \in \mathbb{R}_{>0}$ satisfying

$$\lim_{t \to \infty} X(t + T) - X(t) = 0.$$

Theorem 3 (Convergence of REC): Consider a set of n cameras installed along a 1-D open path Γ. Let $A_1, \ldots, A_n$ be the initial patrolling windows, with $A_i \subseteq D_i$ for all $i \in \{1, \ldots, n\}$. Let the cameras implement the Algorithm 3. Then:

1) for all iterations $k \in \mathbb{N}$ and for all $i \in \{1, \ldots, n\}$ the patrolling window $A_i(k)$ satisfies $A_i(k) \subseteq D_i$;
2) for all iterations $k \in \mathbb{N}$ the set $\{A_1(k), \ldots, A_n(k)\}$ is a partition of Γ, and
$$\tau^* = \lim_{k \to \infty} \max_{i \in \{1, \ldots, n\}} \frac{d_i(k)}{v_i^{\max}} = \min_{\mathcal{P}} \max_{i \in \{1, \ldots, n\}} \frac{d_i}{v_i^{\max}}$$
where $\mathcal{P}$ is the set of partitions $\{A_1, \ldots, A_n\}$ of Γ satisfying $A_i \subseteq D_i$ for all $i \in \{1, \ldots, n\}$ and d_i is the length of A_i;
3) the cameras trajectory generated by the REC algorithm is asymptotically $2\tau^*$-periodic, and it converges to an equal-waiting trajectory.

Proof: In the interest of space, we only sketch the proof. First, notice that the update of r_i and ℓ_i is such that A_i belongs to the constraint set D_i, so that statement 1) follows. Second notice that cameras persistently communicate over time. Indeed: 1) each camera sweeps back and forth its assigned segment; 2) cameras wait at their boundaries until communication with a neighboring camera takes place; and 3) cameras 1 and n do not stop at ℓ_1 and r_n, respectively. In particular, it can be shown that any two neighboring cameras communicate within an interval of finite length. Then, statement 2) follows from [30, Th. IV.1]. Third, because of the persistence of communication among cameras, the value $\tau^{\max}(k)$, which is decreasing in k, propagates in some time T_{prop} to every camera, for every iteration k. Let $\bar{t}$ be such that $\tau^{\max}(\bar{t}) = \tau^* + \varepsilon$, for some $\varepsilon \in \mathbb{R}_{>0}$. Then, after time $\bar{t} + T_{\text{prop}}$, the period T_i of c_i is within 2ε of $2\tau^*$, for all $i \in \{1, \ldots, n\}$. Statement 3) follows by letting ε tend to zero. ■

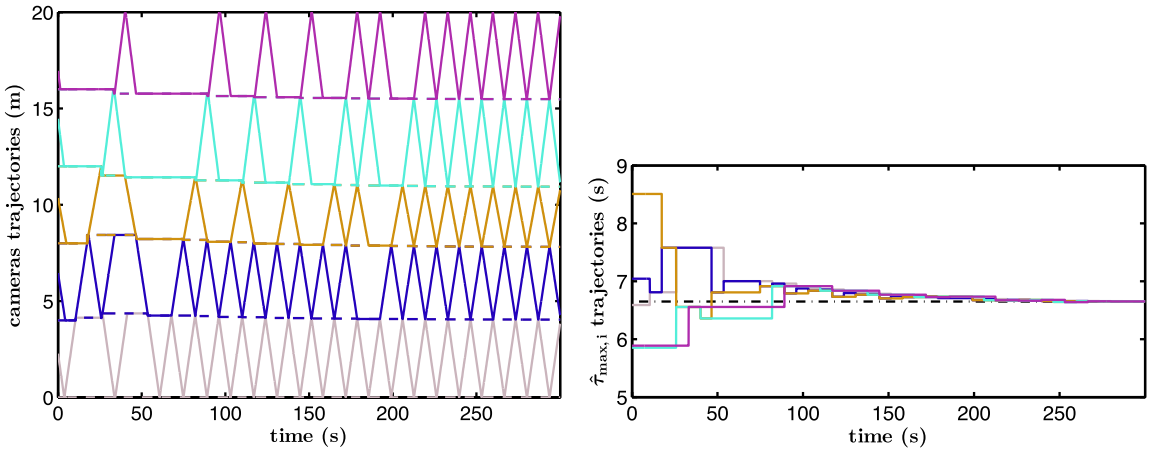
As stated in Theorem 3, Algorithm 3 drives the cameras toward an equal-waiting trajectory. Then, the detection performance of the cameras trajectory generated by our REC are as in Theorem 1 with $\tau^{\max} = \tau^*$.

We now validate our REC via a numerical study. We consider two scenarios with five cameras. All cameras start their trajectory from some initial point in their patrolling window. In the first scenario, (Fig. 13) cameras have the same maximum speed $v^{\max} = 0.67$ m/s, and they are not subjected to patrolling windows constraints. Relevant parameters for this numerical study are shown in Table II. Observe from Fig. 13 that the cameras trajectory converges to an equal-waiting trajectory, and that the length of the largest patrolling window $\tau^{\max}$ is decreasing and converges to $\tau^* = 6.2023$ s.

In the second scenario, cameras have different maximum speeds ($v_1 = 0.61$, $v_2 = 0.57$, $v_3 = 0.47$, $v_4 = 0.68$, and

TABLE II
PARAMETERS AND RESULTS FOR UNIFORM CAMERAS SPEED

	c_1	c_2	c_3	c_4	c_5
D_i	[0 4.68]	[1.14 7.45]	[3.32 12.09]	[7.26 18.41]	[10.12 20]
$A_i(0)$	[0 2.91]	[2.91 5.38]	[5.38 9.67]	[9.67 14.26]	[14.26 20]
$A_i(\infty)$	[0 3.72]	[3.72 7.45]	[7.45 11.63]	[11.63 15.82]	[15.82 20]



(a) (b)

Fig. 14. Numerical study of REC with $n = 5$ cameras with nonuniform maximum speeds $v_i^{\max} \sim \mathcal{U}[0.45, 0.75]$ m/s and no patrolling windows constraints. (a) Cameras trajectories starting from random positions. The dashed lines refer to the trajectories of the active boundaries. (b) Dynamics of $\hat{\tau}_i^{\max}$. Notice that $\hat{\tau}_i^{\max}$ converges to the optimal value $\tau^* = 6.65$ s (dashed-dotted line).

TABLE III
PARAMETERS AND RESULTS FOR NONUNIFORM CAMERAS SPEED

	c_1	c_2	c_3	c_4	c_5
D_i	[0 20]	[0 20]	[0 20]	[0 20]	[0 20]
$A_i(0)$	[0 4]	[4 8]	[8 12]	[12 16]	[16 20]
$A_i(\infty)$	[0 4.04]	[4.04 7.81]	[7.81 10.94]	[10.94 15.48]	[15.48 20]

$v_5 = 0.68$ m/s), and they are subjected to patrolling windows constraints. Relevant parameters for this numerical study are shown in Table III. As shown in Fig. 14, the cameras trajectory converges to an equal-waiting trajectory, and that the length of the largest patrolling window $\tau^{\max}$ is decreasing and converges to $\tau^* = 6.65$ s.

VI. CONCLUSION

This paper studies the problem of coordinating a team of autonomous cameras along a 1-D open path to detect moving intruders. We propose mathematical models of cameras and intruders, and we define the WDT and ADT as performance criteria. We propose cameras trajectories with performance guarantees, and distributed algorithms to coordinate the motion of the cameras. Finally, we validate our theoretical findings and show effectiveness of our algorithms via numerical studies and experiments.

Several extensions to this paper are of interest. First, we envision extension to more general situations, such as tree-like and cyclic environments. Second, additional performance metrics can be defined to compare different strategies, capturing robustness of the coordination algorithm and predictability of the surveillance strategy. Third, the possibility of having cameras f.o.v.s with different and nonconstant velocity profiles. Finally, the problem of jointly estimating the cameras positions

and developing a sweeping strategy to prevent intruders from entering and exiting the environment at strategic locations identified by suitable probability density functions.

APPENDIX

This section contains a proof of Theorem 1. We start with a lower bound for the ADT.

Lemma 1 (Lower Bound on the ADT): For a set of n cameras with maximum velocities $v_1^{\max}, \dots, v_n^{\max}$ and sweeping times $\tau_1, \dots, \tau_n$, the ADT of a $2\tau^{\max}$-periodic cameras trajectory X satisfies the lower bound

$$\mathrm{ADT}(X) \geq \frac{1}{L}\sum_{i=1}^{n} v_i^{\max}\tau_i^2.$$

Proof: Since a smart intruder moves away from the camera f.o.v., the detection time of a smart intruder appearing at time t and at location $p \in [\ell_i, r_i]$ satisfies the lower bound

$$\int_{\ell_i}^{r_i} T_{\det}(\mathcal{I}_{t,p}) - t \; dp = \int_{0}^{d_i} T_{\det}(\mathcal{I}_{t,p}) - t \; dp \geq T_{\mathrm{up}}$$

where $d_i = r_i - \ell_i$ and T_{up} equals

$$\int_0^{x_i(t)} \frac{x_i(t) + 2(d_i - x_i(t))}{v_i^{\max}}\, dp + \int_{x_i(t)}^{d_i} \frac{d_i - x_i(t)}{v_i^{\max}}\, dp$$

if the camera c_i first detects intruders appearing at locations $p \geq x_i(t)$. Analogously, if the camera c_i first detects intruders appearing at locations $p \leq x_i(t)$

$$\int_0^{d_i} T_{\det}(\mathcal{I}_{t,p}) - t \; dp \geq T_{\mathrm{down}}$$

where T_{down} equals

$$\int_0^{x_i(t)} \frac{x_i(t)}{v_i^{\max}}\, dp + \int_{x_i(t)}^{d_i} \frac{2x_i(t) + (d_i - x_i(t))}{v_i^{\max}}\, dp.$$

To observe this, consider the first case. The detection time of every intruder appearing at location $p > x_i(t)$ is at least $(d_i - x_i(t))/v^{\max}$ (time needed by camera c_i to reach r_i starting from $x_i(t)$). Likewise, the detection time of every intruder appearing at location $p < x_i(t)$ is at least $(x_i(t) + 2(d_i - x_i(t)))/v^{\max}$ (time needed by c_i to reach r_i and ℓ_i starting from $x_i(t)$). The other case follows similarly.

It can be verified by simple manipulation that $T_{\mathrm{up}} = T_{\mathrm{down}} = d_i^2/v^{\max}$. Finally, it follows from (2) and $\tau_i = d_i/v_i^{\max}$ that

$$\begin{aligned}\mathrm{ADT}(X) &= \frac{1}{TL}\int_0^T \int_\Gamma T_{\det}(\mathcal{I}^*_{t,p}) - t \; dpdt \\ &\geq \frac{1}{TL}\int_0^T \sum_{i=1}^{n} \frac{d_i^2}{v_i^{\max}}\, dt = \frac{1}{L}\sum_{i=1}^{n} v_i^{\max}\tau_i^2.\end{aligned}$$

■

It should be observed that the bound in Lemma 1 is tight for the case of a single camera, and it is conservative otherwise [Figs. 5(a) and 6]. We now characterize the ADT of the equal-waiting trajectory.

Lemma 2 (Equal-Waiting Trajectory Performance): For a set of n cameras with sweeping times $\tau_1, \dots, \tau_n$, let X^{eq} be the equal-waiting trajectory defined in Algorithm 1. Then

$$\mathrm{ADT}(X^{\mathrm{eq}}) = \frac{1}{2}\left(\tau^{\max} + \frac{1}{L}\sum_{i=1}^{n} v_i^{\max}\tau_i^2\right). \tag{A-1}$$

Proof: Observe that the function $\mathrm{ADT}(X^{\mathrm{eq}})$ can be written as

$$\mathrm{ADT}(X^{\mathrm{eq}}) = \frac{1}{2\tau^{\max}L}\sum_{i=1}^{n}\int_0^{2\tau^{\max}}\int_{\ell_i}^{r_i} \left(T_{\det}(\mathcal{I}_{t,p}) - t\right)\, dpdt.$$

Let i be odd and recall the description of $x_i(t)$ given in Algorithm 1. Due to symmetry, it can be verified that

$$\begin{aligned}&\int_0^{2\tau^{\max}}\int_{\ell_i}^{r_i} \left(T_{\det}(\mathcal{I}_{t,p}) - t\right)\, dpdt \\ &\qquad = 2\int_0^{\tau^{\max}}\int_{\ell_i}^{r_i} \left(T_{\det}(\mathcal{I}_{t,p}) - t\right)\, dpdt.\end{aligned}$$

Let $0 \leq t \leq \tau^{\max} - \tau_i$, and notice that $x_i(t) = r_i$. Observe that $T_{\det}(\mathcal{I}_{t,p}) = \tau^{\max}$ for all $p \in [\ell_i, r_i)$, and $T_{\det}(\mathcal{I}_{t,p}) = 0$ for $p = r_i$. Then

$$\int_0^{\tau^{\max}-\tau_i}\int_{\ell_i}^{r_i} \left(T_{\det}(\mathcal{I}_{t,p}) - t\right)\, dpdt = \frac{(\tau^{\max})^2 - \tau_i^2}{2}\, d_i. \tag{A-2}$$

Let $\tau^{\max} - \tau_i \leq t \leq \tau^{\max}$. Observe that $T_{\det}(\mathcal{I}_{t,p}) = \tau^{\max}$ for $p \in [\ell_i, x_i(t))$, $T_{\det}(\mathcal{I}_{t,p}) = 0$ for $p = x(t)$, and $T_{\det}(\mathcal{I}_{t,p}) = 2\tau^{\max}$ for $p \in (x_i(t), r_i]$. Thus

$$\begin{aligned}&\int_{\tau^{\max}-\tau_i}^{\tau^{\max}}\int_{\ell_i}^{r_i} \left(T_{\det}(\mathcal{I}_{t,p}) - t\right)\, dpdt \\ &= \int_{\tau^{\max}-\tau_i}^{\tau^{\max}}\int_{\ell_i}^{x_i(t)} \left(\tau^{\max} - t\right)\, dp + \int_{x_i(t)}^{r_i} \left(2\tau^{\max} - t\right)\, dpdt.\end{aligned}$$

Since $x_i = r_i + (t - (\tau^{\max} - \tau_i)v_i^{\max})$ (Algorithm 1), it follows from the above expression that

$$\begin{aligned}&\int_{\tau^{\max}-\tau_i}^{\tau^{\max}}\int_{\ell_i}^{r_i} \left(T_{\det}(\mathcal{I}_{t,p}) - t\right)\, dpdt \\ &\quad = \int_{\tau^{\max}-\tau_i}^{\tau^{\max}} (2\tau^{\max} - t)d_i - \tau^{\max}(\tau^{\max} - t)v_i^{\max}\, dt \\ &\quad = \frac{1}{2}\tau^{\max}\tau_i^2 v_i^{\max} + \frac{1}{2}d_i\tau_i^2.\end{aligned} \tag{A-3}$$

The statement follows by combining (A-2) and (A-3). ■

We now conclude with a proof of Theorem 1.

Proof of Theorem 1: From Lemmas 1 and 2, we have

$$\begin{aligned}\frac{\mathrm{ADT}(X^{\mathrm{eq}})}{\mathrm{ADT}^*} &= \frac{1}{2} + \frac{L\tau^{\max}}{2\sum_{i=1}^{n} v_i^{\max}\tau_i^2} \\ &= \frac{1}{2} + \frac{L\tau^{\max}}{2\sum_{i=1}^{n} d_i\tau_i} \\ &\leq \frac{1}{2} + \frac{L\tau^{\max}}{\tau^{\min}2\sum_{i=1}^{n} d_i} = \frac{\tau^{\max} + \tau^{\min}}{2\tau^{\min}}\end{aligned}$$

where we have used $L = \sum_{i=1}^{n} d_i$ and $\tau^{\min} \leq \tau_i$ for all $i \in \{1, \ldots, n\}$. To show the second bound notice that

$$\frac{\mathrm{ADT}(X^{\mathrm{eq}})}{\mathrm{ADT}^*} = \frac{1}{2} + \frac{L\tau^{\max}}{2\sum_{i=1}^{n} d_i \tau_i} = \frac{1}{2} + \frac{L}{2\sum_{i=1}^{n} d_i \frac{\tau_i}{\tau^{\max}}} \leq \frac{1}{2} + \frac{L}{2d^{\min}}$$

where the last inequality is obtained by letting $\tau_i/\tau^{\max} \to 0$ for all i except for one segment ($\tau_i/\tau^{\max} = 1$ for some i, and $d_i \geq d^{\min}$). Since $L \leq nd^{\max}$ we conclude that

$$\frac{\mathrm{ADT}(X^{\mathrm{eq}})}{\mathrm{ADT}^*} \leq \tfrac{1}{2} + \tfrac{nd^{\max}}{2d^{\min}} \leq \tfrac{(n+1)d^{\max}}{2d^{\min}}.$$

We now show the last part of the Theorem. Assume that all cameras move at unit speed and, without affecting generality, that $d_1 = d^{\max}$. Notice that

$$\frac{\mathrm{ADT}(X^{\mathrm{eq}})}{\mathrm{ADT}^*} = \frac{1}{2} + \frac{L\tau^{\max}}{2\sum_{i=1}^{n} d_i \tau_i} = \frac{1}{2} + \frac{\sum_{i=1}^{n} d_1 d_i}{2\sum_{i=1}^{n} d_i^2} = \frac{1}{2}\left(1 + \frac{1 + \sum_{i=2}^{n} y_i}{1 + \sum_{i=2}^{n} y_i^2}\right)$$

where $y_i = d_i/d_1$. Consider the minimization problem

$$\min_{\{K, x_2, \ldots, x_n\}} K \quad \text{subject to } 1 + \sum_{i=2}^{n} y_i \leq K\left(1 + \sum_{i=2}^{n} y_i^2\right) \qquad \text{(A-4)}$$

and the associated Lagrangian function [34]

$$\mathcal{L} = K + \lambda\left(1 - K + \sum_{i=2}^{n} y_i - K y_i^2\right).$$

Following standard optimization theory, necessity optimality conditions for the minimization problem (A-4) are

$$\frac{\partial \mathcal{L}}{\partial K} = 0 \implies 1 - \lambda\left(1 + \sum_{i=2}^{n} y_i^2\right) = 0$$

$$\frac{\partial \mathcal{L}}{\partial y_i} = 0 \implies \lambda(1 - 2Ky_i) = 0, \quad \text{for } i \in \{2, \ldots, n\}$$

$$\text{Complementary slackness: } \lambda\left(1 - K + \sum_{i=2}^{n} y_i - K y_i^2\right) = 0.$$

From the first and second equations, we obtain $\lambda \neq 0$ and $y_i = 1/(2K)$. Then, the third equation yields $K = (1 \pm \sqrt{n})/2$. Since $y_i > 0$, the statement follows. ∎

REFERENCES

[1] J. Clark and R. Fierro, "Mobile robotic sensors for perimeter detection and tracking," *ISA Trans.*, vol. 46, no. 1, pp. 3–13, Feb. 2007.

[2] D. B. Kingston, R. W. Beard, and R. S. Holt, "Decentralized perimeter surveillance using a team of UAVs," *IEEE Trans. Robot.*, vol. 24, no. 6, pp. 1394–1404, Dec. 2008.

[3] S. Susca, S. Martínez, and F. Bullo, "Monitoring environmental boundaries with a robotic sensor network," *IEEE Trans. Control Syst. Technol.*, vol. 16, no. 2, pp. 288–296, Feb. 2008.

[4] Y. Elmaliach, A. Shiloni, and G. A. Kaminka, "A realistic model of frequency-based multi-robot polyline patrolling," in *Proc. Int. Conf. Autonom. Agents*, Estoril, Portugal, May 2008, pp. 63–70.

[5] J. Czyzowicz, L. Gąsieniec, A. Kosowski, and E. Kranakis, "Boundary patrolling by mobile agents with distinct maximal speeds," in *Proc. Eur. Symp. Algorithms*, Saarbrücken, Germany, Sep. 2011, pp. 701–712.

[6] A. Machado, G. Ramalho, J. D. Zucker, and A. Drogoul, "Multi-agent patrolling: An empirical analysis of alternative architectures," in *Multi-Agent-Based Simulation II* (Lecture Notes in Computer Science). New York, NY, USA: Springer-Verlag, 2003, pp. 155–170.

[7] Y. Chevaleyre, "Theoretical analysis of the multi-agent patrolling problem," in *Proc. IEEE/WIC/ACM Int. Conf. Intell. Agent Technol.*, Beijing, China, Sep. 2004, pp. 302–308.

[8] A. Gusrialdi, R. Dirza, T. Hatanaka, and M. Fujita, "Improved distributed coverage control for robotic visual sensor network under limited energy storage," *Int. J. Imaging Robot.*, vol. 10, no. 2, pp. 58–74, 2013.

[9] F. Pasqualetti, A. Franchi, and F. Bullo, "On cooperative patrolling: Optimal trajectories, complexity analysis and approximation algorithms," *IEEE Trans. Robot.*, vol. 28, no. 3, pp. 592–606, Jun. 2012.

[10] M. Baseggio, A. Cenedese, P. Merlo, M. Pozzi, and L. Schenato, "Distributed perimeter patrolling and tracking for camera networks," in *Proc. 49th IEEE CDC*, Atlanta, GA, USA, Dec. 2010, pp. 2093–2098.

[11] R. Carli, A. Cenedese, and L. Schenato, "Distributed partitioning strategies for perimeter patrolling," in *Proc. ACC*, San Francisco, CA, USA, Jul. 2011, pp. 4026–4031.

[12] D. Borra, F. Pasqualetti, and F. Bullo, "Continuous graph partitioning for camera network surveillance," *Automatica*, Jul. 2012, submitted.

[13] S. M. Huck, N. Kariotoglou, S. Summers, D. M. Raimondo, and J. Lygeros, "Design of importance-map based randomized patrolling strategies," in *Proc. Complex. Eng.*, Jun. 2012, pp. 1–6.

[14] D. M. Raimondo, N. Kariotoglou, S. Summers, and J. Lygeros, "Probabilistic certification of pan-tilt-zoom camera surveillance systems," in *Proc. 50th IEEE CDC ECC*, Orlando, FL, USA, Dec. 2011, pp. 2064–2069.

[15] M. Spindler, F. Pasqualetti, and F. Bullo, "Distributed multi-camera synchronization for smart-intruder detection," in *Proc. ACC*, Montréal, Canada, Jun. 2012, pp. 5120–5125.

[16] F. Zanella, F. Pasqualetti, R. Carli, and F. Bullo, "Simultaneous boundary partitioning and cameras synchronization for optimal video surveillance," in *Proc. IFAC Workshop Distrib. Estimation Control Netw. Syst.*, Santa Barbara, CA, USA, Sep. 2012, pp. 1–6.

[17] R. Bodor, A. Drenner, P. Schrater, and N. Papanikolopoulos, "Optimal camera placement for automated surveillance tasks," *J. Intell. Robot. Syst.*, vol. 50, no. 3, pp. 257–295, Nov. 2007.

[18] J. Zhao, S.-C. Cheung, and T. Nguyen, "Optimal camera network configurations for visual tagging," *IEEE J. Sel. Topics Signal Process.*, vol. 2, no. 4, pp. 464–479, Aug. 2008.

[19] B. Dieber, C. Micheloni, and B. Rinner, "Resource-aware coverage and task assignment in visual sensor networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 10, pp. 1424–1437, Oct. 2011.

[20] C. Micheloni, B. Rinner, and G. L. Foresti, "Video analysis in pan-tilt-zoom camera networks," *IEEE Signal Process. Mag.*, vol. 27, no. 5, pp. 78–90, Sep. 2010.

[21] S.-N. Lim, A. Elgammal, and L. S. Davis, "Image-based pan-tilt camera control in a multi-camera surveillance environment," in *Proc. Int. Conf. Multimedia Expo*, vol. 1. Baltimore, MD, USA, Jul. 2003, pp. 645–648.

[22] F. Z. Qureshi and D. Terzopoulos, "Proactive PTZ camera control," in *Distributed Video Sensor Networks*. New York, NY, USA: Springer-Verlag, 2011, pp. 273–287.

[23] F. Z. Qureshi and D. Terzopoulos, "Surveillance camera scheduling: A virtual vision approach," *Multimedia Syst.*, vol. 12, no. 3, pp. 269–283, 2006.

[24] C. Ding, B. Song, A. Morye, J. A. Farrell, and A. K. Roy-Chowdhury, "Collaborative sensing in a distributed PTZ camera network," *IEEE Trans. Image Process.*, vol. 21, no. 7, pp. 3282–3295, Jul. 2012.

[25] C. Soto, B. Song, and A. K. Roy-Chowdhury, "Distributed multi-target tracking in a self-configuring camera network," in *Proc. IEEE Conf. CVPR*, Miami, FL, USA, Jun. 2009, pp. 1486–1493.

[26] Y. Zhu, N. M. Nayak, and A. K. Roy Chowdhury, "Context-aware activity recognition and anomaly detection in video," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 1, pp. 91–101, Feb. 2013.

[27] R. D. Quinn, G. C. Causey, F. L. Merat, D. M. Sargent, N. A. Barendt, W. S. Newman, *et al.*, "An agile manufacturing workcell design," *IIE Trans. Design Manuf.*, vol. 29, no. 10, pp. 901–909, Oct. 1997.

[28] T. Borangiu, P. Gilbert, N.-A. Ivanescu, and A. Rosu, "An implementing framework for holonic manufacturing control with multiple robot-vision stations," *Eng. Appl. Artif. Intell.*, vol. 22, nos. 4–5, pp. 505–521, Jun. 2009.

[29] S. S. Nestinger, B. Chen, and H. H. Cheng, "A mobile agent-based framework for flexible automation systems," *IEEE/ASME Trans. Mechatron.*, vol. 15, no. 6, pp. 942–951, Dec. 2010.

[30] R. Alberton, R. Carli, A. Cenedese, and L. Schenato, "Multi-agent perimeter patrolling subject to mobility constraints," in *Proc. ACC*, Montréal, Canada, Jun. 2012, pp. 4498–4503.

[31] A. Kawamura and Y. Kobayashi, "Fence patrolling by mobile agents with distinct speeds," in *Algorithms and Computation* (Lecture Notes in Computer Science). New York, NY, USA: Springer-Verlag, 2012, pp. 598–608.

[32] C. G. Cassandras, X. Lin, and X. C. Ding, "An optimal control approach to the multi-agent persistent monitoring problem," *IEEE Trans. Autom. Control*, vol. 58, no. 4, pp. 947–961, Apr. 2013.

[33] M. Spindler, "Distributed multi-camera synchronization for smart-intruder detection," M.S. thesis, Dept. Mechanical Eng., Univ. Stuttgart, Sep. 2011.

[34] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

Fabio Pasqualetti received the Laurea degree in computer engineering and the Laurea Magistrale degree in automation engineering from the University of Pisa, Pisa, Italy, in 2004 and 2007, respectively, and the D.Phil. degree in mechanical engineering from the University of California, Santa Barbara, CA, USA, in 2012.

He is an Assistant Professor with the Department of Mechanical Engineering, University of California, Riverside, CA, USA. His current research interests include secure control systems, with application to multiagent networks, distributed computing, and power networks, and vehicle routing and combinatorial optimization, with application to distributed area patrolling and persistent surveillance.

Filippo Zanella (M'06) was born in Treviso, Italy, in 1983. He received the M.S. degree in automation engineering and the Ph.D. degree in information engineering from the University of Padova, Padova, Italy, in 2008 and 2013, respectively.

He was a Visiting Student Researcher with UC Berkeley, Berkeley, CA, USA, in 2011, and UC Santa Barbara, Santa Barbara, CA, USA, in 2012. His current research interests include wireless cameras/sensors networks and mobile networks with emphasis on distributed control, estimation and optimization.

Dr. Zanella was a Staff Member of the IEEE Student Branch from the University of Padova from 2006 to 2008.

Jeffrey Russel Peters received the B.S. degree in mechanical engineering from the University of Illinois at Urbana-Champaign, Urbana, IL, USA, in 2011. He is currently pursuing the Ph.D. degree with the University of California, Santa Barbara, CA, USA.

His current research interests include intelligent camera networks, robotic coordination for patrolling and surveillance, analysis and estimation in sensor networks, and nonlinear optimization.

Markus Spindler received the Degree in mechanical engineering with a focus on control engineering at the University of Stuttgart, and attended the University of California, Santa Barbara, CA, USA, for his master thesis.

He conducted research at ETH Zürich, and is now with Alstom Ltd., Switzerland, where he develops inspection robots.

Ruggero Carli received the Laurea degree in computer engineering and the Ph.D. degree in information engineering from the University of Padova, Padova, Italy, in 2004 and 2007, respectively.

He was a Post-Doctoral Fellow with the Department of Mechanical Engineering, University of California at Santa Barbara, Santa Barbara, CA, USA, from 2008 to 2010. He is currently an Assistant Professor with the Department of Information Engineering, University of Padova. He has published more than 60 journal and conference papers. His current research interests include control theory and, in particular, control under communication constraints, cooperative control, and distributed estimation.

Francesco Bullo is a Professor with the Mechanical Engineering Department and the Center for Control, Dynamical Systems and Computation, University of California, Santa Barbara, CA, USA. He was with the University of Padova, Padova, Italy, California Institute of Technology, Pasadena, CA, USA, and University of Illinois at Urbana-Champaign, Urbana, IL, USA. His current research interests include network systems and distributed control with application to robotic coordination, power grids, and social networks. He is the co-author of *Geometric Control of Mechanical Systems* (Springer, 2004) and *Distributed Control of Robotic Networks* (Princeton, 2009).

He has served on the editorial boards of the IEEE TRANSACTIONS ON AUTOMATIC CONTROL, *ESAIM: Control, Optimization, and the Calculus of Variations, SIAM Journal of Control and Optimization*, and *Mathematics of Control, Signals, and Systems*.