Camera Network Coordination for Intruder Detection

Fabio Pasqualetti, Filippo Zanella, Jeffrey R. Peters, Markus Spindler, Ruggero Carli, and Francesco Bullo

Abstract

This work 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 and the average detection time. We focus on the case of a chain of cameras, and we obtain the following results. First, we characterize a lower bound on the worst-case and on the average detection time of smart intruders. Second, we propose a team trajectory for the cameras, namely Equal-waiting trajectory, with minimum worst-case detection time and with guarantees on the average detection time. 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 simulations and experiments.

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.

**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 one dimensional, in the sense that it can be completely observed by a chain of cameras by using the panning motion only (the perimeter surveillance problem is a special case of this framework). We assume the cameras to be subject to physical constraints, such as limited field of view (f.o.v.) and panning 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. We also assume that intruders do not leave the environment once they have entered. We propose cameras trajectories and control algorithms to minimize the *worst case detection time* and the *average detection time* of smart intruders.

**Related work.** Of relevance to this work 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 *worst-
In the context of video surveillance most approaches consider the case of static cameras, where the surveillance problem reduces to an optimal sensor placement problem. In [15], this sensor placement problem is considered in a static camera network with the goal of maximizing the observability of a set of aggregate tasks that occur in a dynamic environment. In [16], a similar problem is considered in a static camera network with the goal of visual tagging. In the case of surveillance in active (PTZ) camera networks, [17] assumes the possibility to cover the entire region at all times, and it considers the problem of coordinating camera motion using a game-theoretic approach. With respect to this work, we do not make the assumption that cameras can cover the whole region at all times, and we provide specific results for the case of one-dimensional environment.

**Contributions.** The contributions of this work 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 worst-case detection time and average detection time criteria, and we characterize lower bounds on both performance criteria.
Second, we propose the **Equal-waiting** cameras trajectory, which achieves minimum worst case detection time, and constant factor optimal average detection time (under reasonable assumptions). The Equal-waiting trajectory is easy to compute given a camera network, and it is amenable to distributed implementation. In fact, 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 for the cameras to react to failures and to adapt to time-varying topologies. In particular, our reconfiguration algorithm 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.

Fourth and finally, we validate our findings through simulations and experiments. Through simulations we validate our theoretical findings and algorithms for different configurations. Analogously, our experiments validate our modeling framework and assumptions, and show that our methods are robust to cameras failure, model uncertainties, and sensor noise.

**Paper Organization.** The remainder of the 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 simulations 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 results

In this section we describe the one-dimensional 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 one dimensional open path (boundary) \( \Gamma \) of length \( L \) (see Fig 1). For the ease of notation and without affecting generality, we represent \( \Gamma \) with the segment \([0, L]\), and we label the cameras in increasing order from \( c_1 \) to \( c_n \) according to their position on \( \Gamma \). We make the following assumptions:
(A1) the f.o.v. of each camera is represented by a point on \( \Gamma \),

(A2) the motion of each f.o.v. is uniquely determined by the pan movement of the corresponding camera, and

(A3) the speed \( v_i \) of the \( i \)-th camera satisfies \( |v_i| \in \{0, v_i^{\text{max}}\} \), with \( v_i^{\text{max}} \in \mathbb{R}_{>0} \).

For assumption (A3) 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 \( v_i^{\text{max}} \). Let \( v^{\text{max}} = \max\{v_1^{\text{max}}, \ldots, v_n^{\text{max}}\} \).

Let \( x_i : \mathbb{R}_{\geq 0} \rightarrow \Gamma \) be a map, such that \( x_i(t) \) specifies the position on \( \Gamma \) of the \( i \)-th 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 segment of \( \Gamma \) containing the f.o.v. of camera \( c_i \) at all times, where \( \ell_i \) 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 in fact a partition of \( \Gamma \). Finally, let \( d_i \) be the length of \( A_i \), let \( d^{\text{max}} = \max\{d_1, \ldots, d_n\} \) and \( d^{\text{min}} = \min\{d_1, \ldots, d_n\} \), and define the longest cameras sweeping time as \( \tau^{\text{max}} = \max\{\tau_1, \ldots, \tau_n\} \), where \( \tau_i = d_i/v_i^{\text{max}} \) 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 \( \Gamma \). 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 \( c_{i+1} \).

B. Model of intruder and performance functions

We consider the problem of detecting intruders appearing at random times and moving on \( \Gamma \). We model an intruder as an arbitrarily fast point on \( \Gamma \), and we let the continuous map \( \mathcal{I}_{t_0,p_0} : \mathbb{R}_{\geq t_0} \rightarrow \Gamma \) 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

\[ \dot{\alpha}_i = v_i^{\text{max}}/(a_i \sec^2(\alpha)) \]

where \( \alpha \) denotes the panning angle, and \( a_i \) is the distance of the \( i \)-th camera from \( \Gamma \). See Section IV-B for a related example.

\[ 1 \text{For instance, the controller may set the panning velocity of the } i \text{-th camera to } \dot{\alpha}_i = v_i^{\text{max}}/(a_i \sec^2(\alpha)) \text{, where } \alpha \text{ denotes the panning angle, and } a_i \text{ is the distance of the } i \text{-th camera from } \Gamma \text{. See Section IV-B for a related example.} \]
Fig. 2. This figure shows a cameras trajectory in which cameras $c_1$ and $c_2$ are synchronized, while cameras $c_2$ and $c_3$ are not synchronized. Notice that, because of the synchronization, intruder $e_1$, and in fact any smart intruder, is detected when the f.o.v. of cameras $c_1$ and $c_2$ occupy the same position. Instead, intruder $e_2$ may avoid detection by properly choosing its trajectory.

$t_0 \in \mathbb{R}_{\geq 0}$, an initial point $p_0 \in \Gamma$, and a cameras trajectory $X$, the trajectory $I^*_t_{0,p_0}$ of a smart intruder satisfies

$$T_{\text{det}}(I^*_{t_0,p_0}, X) = \max_{I_t_{0,p_0}} T_{\text{det}}(I_{t_0,p_0}, X) - t_0,$$

where $T_{\text{det}}(I_t_{0,p_0}, X)$ is the time at which the intruder is detected by the cameras, that is,

$$T_{\text{det}}(I_{t_0,p_0}, X) = \min\{\{t : I_{t_0,p_0}(t) \in X(t)\}, \infty\}.$$

Notice that the trajectory $I^*_t_{0,p_0}$ is in general not unique.

We consider two criteria for the detection performance of a $T$-periodic cameras trajectory, namely the worst-case detection time (WDT), and the average detection time (ADT). These two criteria are formally defined as

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

and

$$\text{ADT}(X) = \frac{1}{TL} \int_0^T \int_\Gamma T_{\text{det}}(I^*_{t,\gamma}, X) - t \ d\gamma \ dt.$$

In other words, the WDT criterion measures the longest detection time of a smart intruder, while the ADT criterion measures the average detection time of a smart intruder over the boundary $\Gamma$ and the periodicity $T$. 
The worst case detection time criterion for static intruders, namely WDTs, is defined in [18] as

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

and it corresponds to the longest time for the cameras to detect a static intruder, or simply event, along \( \Gamma \). We next informally discuss the relation between WDT and WDTs, and we refer the reader to [7], [8], [18] for a proof of these results. Let

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

Clearly, \( WDT(X) \geq WDTs(X) \) for every cameras trajectory \( X \). For instance, as shown by the example in Fig. 2, if a cameras trajectory \( X \) is not synchronized and yet it covers every location of \( \Gamma \), then \( WDTs(X) < \infty \) and \( WDT(X) = \infty \). Additionally, because the patrolling windows define a partition of \( \Gamma \), it can be easily verified that the static worst case detection time satisfies

$$WDTs^* = 2\tau_{\text{max}},$$

and that any \( 2\tau_{\text{max}} \)-periodic cameras trajectory achieves minimum static worst-case detection time. Similarly, any synchronized \( 2\tau_{\text{max}} \)-periodic cameras trajectory \( X \) satisfies (see Fig. 2)

$$WDT(X) = 2\tau_{\text{max}}. (4)$$

Thus, any synchronized \( 2\tau_{\text{max}} \)-periodic cameras trajectory achieves minimum worst-case detection time (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 \( \tau_{\text{max}} \) be the longest cameras sweeping time. Design a cameras trajectory \( X^* \) satisfying

$$\text{ADT}(X^*) = \text{ADT}^* = \min_{X \in \Omega} \text{ADT}(X),$$

where \( \Omega \) is the set of all synchronized \( 2\tau_{\text{max}} \)-periodic cameras trajectories.

**Remark 1 (Optimal patrolling windows)** We assume that the patrolling windows are given and form a partition of the path \( \Gamma \). With these assumptions, the worst-case detection time satisfies
WDT\(^*\) ≥ WDTs\(^*\) ≥ 2\(\tau\)\(^{\text{max}}\), and any synchronized 2\(\tau\)\(^{\text{max}}\)-periodic cameras trajectory achieves the lower bound.

If the patrolling windows are not given but are still required to be a partition of \(\Gamma\), then the longest cameras sweeping time, and hence the worst-case detection performance, can be minimized by solving a min-max partitioning problem [7], [18], [14]. We will discuss this partitioning problem in Section V, where we develop an algorithm to simultaneously partition the boundary and coordinate the motion of the cameras.

If the patrolling windows are not required to be a partition of \(\Gamma\), then WDTs\(^*\) may be smaller than 2\(\tau\)\(^*\). We refer the interested reader to [19, Conjecture 1] and [20].

A second focus of this paper is the design of distributed coordination algorithms for the cameras to converge to 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 algorithm.

**Problem 2 (Distributed coordination)** For a set of \(n\) cameras on a one-dimensional open path, design a distributed algorithm to coordinate the cameras along a trajectory with minimum average detection time 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 average detection time, and a distributed algorithm for the cameras to converge to such a trajectory. We remark that in some cases an exact solution to Problem 1 can be computed through standard optimization techniques [21]. Such computation, however, is not scalable with the number of cameras, and it is not amenable to distributed implementation. Instead, our approximate solution is valid for every number of cameras and environment configuration, is extremely simple and efficient to compute, and its performance is shown to be within a bound of the optimum. The cameras trajectory that we propose can informally be described as follows:
Fig. 3. This figure shows the Equal-waiting trajectory for 4 cameras. Notice that (i) the cameras are synchronized, (ii) the trajectory is $2d_{\text{max}}$-periodic, and (iii) the waiting time of each camera is the same at both its boundaries.

(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. Additionally, the waiting times of each camera are chosen so that the resulting cameras trajectory is synchronized and periodic.

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. 3, and a formal description is in Trajectory 1.

As discussed in Section II, the Equal-waiting cameras trajectory is optimal with respect to the worst-case detection time criterion. Indeed, the Equal-waiting cameras trajectory is synchronized and $2\tau_{\text{max}}$-periodic. We now characterize the average detection time performance of the Equal-waiting trajectory. A proof of this result is postponed to the Appendix.

**Theorem III.1 (Performance of Equal-waiting trajectories)** Let $X^{\text{eq}}$ be the Equal-waiting trajectory defined in Trajectory 1. Then,

(i) The average detection time of a smart intruder satisfies the lower bound:

$$\text{ADT}^* \geq \frac{1}{L} \sum_{i=1}^{n} v_{i}^{\text{max}} \tau_{i}^{2},$$

(ii) The Equal-waiting trajectory $X^{\text{eq}}$ satisfies

$$\text{ADT}(X^{\text{eq}}) = \frac{1}{2} \left( \tau_{\text{max}} + \frac{1}{L} \sum_{i=1}^{n} v_{i}^{\text{max}} \tau_{i}^{2} \right),$$
### Trajectory 1: Equal-waiting trajectory (camera $c_i$)

**Input**: $\tau_{\text{max}}, r_i, \ell_i, \tau_i, v_i^{\text{max}}$.

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

If $i$ is odd then
- $x_i(t) = r_i$ for $\omega_i(k - 1) + \tau_i \leq t \leq \omega_i(k)$
- $x_i(t) = -v_i^{\text{max}}(t - \omega_i(k)) + r_i$ for $\omega_i(k) \leq t \leq \omega_i(k) + \tau_i$
- $x_i(t) = \ell_i$ for $\omega_i(k) + \tau_i \leq t \leq \omega_i(k + 1)$
- $x_i(t) = v_i^{\text{max}}(t - \omega_i(k + 1)) + \ell_i$ for $\omega_i(k + 1) \leq t \leq \omega_i(k + 1) + \tau_i$

Else if $i$ is even then
- $x_i(t) = \ell_i$ for $\omega_i(k - 1) + \tau_i \leq t \leq \omega_i(k)$
- $x_i(t) = v_i^{\text{max}}(t - \omega_i(k + 1)) + \ell_i$ for $\omega_i(k) \leq t \leq \omega_i(k) + \tau_i$
- $x_i(t) = r_i$ for $\omega_i(k) + \tau_i \leq t \leq \omega_i(k + 1)$
- $x_i(t) = -v_i^{\text{max}}(t - \omega_i(k)) + r_i$ for $\omega_i(k + 1) \leq t \leq \omega_i(k + 1) + \tau_i$

(iii) The Equal-waiting trajectory $X^{eq}$ satisfies

$$\frac{\text{ADT}(X^{eq})}{\text{ADT}^*} \leq \min\left\{ \frac{\tau_{\text{max}} + \tau_{\text{min}}}{2\tau_{\text{min}}}, \frac{(n + 1)d_{\text{max}}}{2d_{\text{min}}} \right\},$$

(iv) If $v_i^{\text{max}} = 1$ for all $i \in \{1, \ldots, n\}$, then the Equal-waiting trajectory $X^{eq}$ satisfies

$$\frac{\text{ADT}(X^{eq})}{\text{ADT}^*} \leq \min\left\{ \frac{d_{\text{max}} + d_{\text{min}}}{2d_{\text{min}}}, \frac{3 + \sqrt{n}}{4} \right\}.$$

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

(i) all cameras move at unitary speed,
(ii) there exists an index $h \in \{1, \ldots, n\}$ such that $d_h > d_i$ for all $i \in \{1, \ldots, n\} \setminus \{h\}$, and
(iii) for all $i \in \{1, \ldots, n\} \setminus \{h\}$ the patrolling windows satisfy

$$d_i = \frac{d_{\text{max}}}{1 + \sqrt{n}},$$

(9)
then (see the proof of Theorem III.1 and Fig. 4(b))
\[
\frac{\text{ADT}(X^{\text{eq}})}{\text{ADT}^*} = \frac{3 + \sqrt{n}}{4}.
\]
(10)

Fifth and finally, different cameras speeds can be taken into account in our bound (8). In fact, if \(v_i^{\max}/v_j^{\max} \leq C\) for all \(i, j \in \{1, \ldots, n\}\) and for some \(C \in \mathbb{R}\), then (see the proof of Theorem III.1)
\[
\frac{\text{ADT}(X^{\text{eq}})}{\text{ADT}^*} \leq \min \left\{ \frac{C (d_{\text{max}} + d_{\text{min}})}{2d_{\text{min}}}, \frac{2 + C (1 + \sqrt{n})}{4} \right\}.
\]

We now design a distributed feedback algorithm that steers the cameras towards an Equal-waiting trajectory. Our coordination algorithm is informally described as follows:

(Distributed coordination) Camera \(c_i\) moves to \(\ell_i\) and, if \(i > 1\), it waits until the f.o.v. of its left neighboring camera \(c_{i-1}\) occupies the same position. Then, camera \(c_i\) stops as specified in Trajectory 1, and finally move to \(r_i\). Camera \(c_1\) (resp. \(c_n\)) moves to \(r_2\) (resp. \(\ell_{n-1}\)) as soon as its f.o.v. arrives at \(\ell_1\) (resp. \(r_n\)). \(\square\)

Our coordination algorithm is formally described in Algorithm 2, where for convenience we set \(x_0(t) = \ell_1\) and \(x_{n+1}(t) = r_n\) at all times.

**Algorithm 2: Distributed camera coordination along an Equal-waiting trajectory (camera \(c_i\))**

**Input**
\[ t^{\max}, r_i, \ell_i, \tau_i, v_i^{\max}; \]

**Set**
\[ \omega_i = t^{\max} - \tau_i, \quad x_0(t) = \ell_1 \quad \text{and} \quad x_{n+1}(t) = r_n \quad \forall t; \]

**Move towards \(\ell_i\) with \(|v_i(t)| = v_i^{\max}|;**

**while True do**

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

\[ \text{Wait until time } t + \omega_i; \]

\[ \text{Move towards the opposite boundary with } |v_i(t)| = v_i^{\max}; \]

It should be observed that, by construction, 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, 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 III.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, $X(t \geq n\tau_{\text{max}})$ 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 $\max\{2\tau_1, \tau_2\} \leq 2\tau_{\text{max}}$. Then, the f.o.v. of camera $c_i$ coincides with the f.o.v. of camera $c_{i+1}$ within time $(i+1)\tau_{\text{max}}$. Hence, within time $n\tau_{\text{max}}$ the cameras trajectory coincides with an Equal-waiting trajectory in Trajectory 1. The claimed statement follows. ■

**IV. SIMULATIONS AND EXPERIMENTS**

In this section we report the results of our simulation 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.

**A. Simulations**

Three simulation studies are presented in this section. For our first simulation 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 $d_1 = d_{\text{max}} = 1$ 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 $v_{\text{max}} = 1$ m/s for all cameras, we design the Equal-waiting trajectory $X^{\text{eq}}$, and we evaluate the cost $\text{ADT}(X^{\text{eq}})$ and the lower bound $\text{ADT}^*$ from equation (5). We report the result of this study in Fig. 4(a). Notice that, when the number of cameras is 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. 4(b)).
Fig. 4. In Fig. 4(a) we report the ratio $\frac{\text{ADT}(X^{eq})}{\text{ADT}^*}$ as a function of the number of cameras $n$ (blue dots), and the bound $(3 + \sqrt{n})/4$ in Theorem III.1 (red dots). For the considered configurations, the bound $(d_{\text{max}} + d_{\text{min}})/(2d_{\text{min}})$ 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 $d_1 = d_{\text{max}} = 1$. We assume that cameras have unitary speed. In Fig. 5 we report the ratio $\frac{\text{ADT}(X^{eq})}{\text{ADT}^*}$ as a function of the number of cameras $n$ (blue dots), and the bounds in Theorem III.1 (black dots and red circles). The lengths $d_i$ of the patrolling windows are chosen as $d_1 = d_{\text{max}} = 1$ and $d_i = (1 + \sqrt{n})^{-1}$ for all $i \in \{2, \ldots, n\}$. As predicted by equation (10), the performance bound in equation (8) is tightly achieved.

Fig. 5. In this figure we report the ratio $\frac{\text{ADT}(X^{eq})}{\text{ADT}^*}$ as a function of $d_{\text{max}}/d_{\text{min}}$ (blue dots), and the bounds in Theorem III.1. For each value of $d_{\text{max}}/d_{\text{min}}$, the lengths of the patrolling windows are uniformly distributed in the interval $[d_{\text{min}}/d_{\text{max}}, 1]$, with $d_1 = d_{\text{max}} = 1$. 
Fig. 6. In this figure we validate Algorithm 2 for a set of 4 cameras with unitary 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 simulation the noise is assumed to be normally distributed with mean $0.2$ and unitary standard deviation.

For our second simulation study, we let the number of cameras be fixed (50 cameras), and we vary the value $d^{\text{max}}/d^{\text{min}}$ between 2 and 25. Specifically, we let $d_1 = d^{\text{max}} = 1$ m, and $d_i$, with $i = 2, \ldots, 50$ be uniformly distributed within the interval $[d^{\text{min}}/d^{\text{max}}, 1]$ m. For each value of $d^{\text{max}}/d^{\text{min}}$ we generate 50 sets of patrolling windows with lengths $\{d_1, \ldots, d_n\}$, compute the Equal-waiting trajectory $X^{\text{eq}}$, evaluate the cost $\text{ADT}(X^{\text{eq}})$, and compute the lower bound in equation (5). The results of this simulation study are reported in Fig. 5.

In our third simulation study we validate the effectiveness of our coordination algorithm. We consider a set of 4 cameras with pre-specified patrolling windows and unitary speed. The cameras trajectory generated by Algorithm 2 is reported in Fig. 6. Observe that our coordination algorithm drives the cameras towards an Equal-waiting trajectory, and it is robust to failures and motion uncertainty. In particular, (i) coordination is achieved for cameras starting at random initial positions, (ii) the algorithm is robust to temporary cameras failure, and (iii) the average detection time degrades gracefully in the presence of motion uncertainties.
B. Experiments

In this section we detail the experiments we have conducted to validate our theoretical findings and simulation results. For our experiments we use a network of six AXIS 213 PTZ (Pan-Tilt-Zoom) network cameras mounted along a square perimeter. In order to simulate a 1-dimensional 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 in order to alert the user when an intruder enters its f.o.v.. All programming of the cameras is performed in Python, utilizing 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. 7 and Table I.

In our first experiment, we validate our distributed coordination algorithm to control the motion of the cameras. Fig. 8 shows the results of our experiment. Notice that the algorithm steers the cameras into an equal waiting trajectory within time $6\tau_{\text{max}}$ as predicted by Theorem III.2. In fact, since the cameras are all starting the experiment at their left boundary, we see that the system reaches an equal waiting trajectory in only slightly longer than $5\tau_{\text{max}} = 150$ s. This is consistent with Theorem III.2, since delays in communication and network bandwidth limits
TABLE I
RELEVANT EXPERIMENTAL PARAMETERS.

<table>
<thead>
<tr>
<th>Camera</th>
<th>$d_i$ (cm)</th>
<th>$v_i^{max}$ (cm/s)</th>
<th>$\tau_i$ (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c_1$</td>
<td>624.3</td>
<td>20.8</td>
<td>30.0</td>
</tr>
<tr>
<td>$c_2$</td>
<td>290.3</td>
<td>18.0</td>
<td>16.1</td>
</tr>
<tr>
<td>$c_3$</td>
<td>291.0</td>
<td>20.6</td>
<td>14.1</td>
</tr>
<tr>
<td>$c_4$</td>
<td>619.3</td>
<td>21.1</td>
<td>29.0</td>
</tr>
<tr>
<td>$c_5$</td>
<td>331.5</td>
<td>19.0</td>
<td>17.4</td>
</tr>
<tr>
<td>$c_6$</td>
<td>232.7</td>
<td>17.3</td>
<td>13.5</td>
</tr>
</tbody>
</table>

Fig. 8. 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.

In our second experiment we focus on the worst-case detection time of intruders. We utilize an Erratic mobile robot from Videre Design to simulate a smart intruder. The robot is equipped...
with an on-board computer with Ubuntu Linux and uses Player/Stage in order 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. 9(a), where we notice that the theoretical worst-case detection time is a relatively tight bound for the experimental worst-case detection time.

In our third experiment we focus on the average detection time 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 driven to avoid detection for as long as possible. We report the results of our second experiment in Fig. 9(b), where we notice that the theoretical bounds in Theorem III.1 are compatible with the experimental data (the slight difference is due to the fact that 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 see from Fig. 8, 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. We now informally describe our reconfiguration algorithm (REC).

(Cameras reconfiguration) Camera $c_i$ sweeps back and forth at the maximum speed its patrolling
In Fig. 9(a) we show the 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. In Fig. 9(b) we show the 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 $\frac{\tau_{\text{max}} + \tau_{\text{min}}}{2\tau_{\text{min}}}$ from (7) are reported.

windows $A_i$, and it updates $A_i$ upon communication with neighboring cameras. 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. Cameras stop for a certain waiting time when their f.o.v. reaches an extreme of their patrolling window. These waiting times ensure that (i) communication among neighboring cameras is maintained over time, and (ii) 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, in order to achieve motion synchronization, the (time varying) maximum sweeping time $\tau_{\text{max}}$ is be propagated across cameras during the execution of the algorithm. In order to do so, the auxiliary variable $q_i$ is used by the $i$-th camera to store the information about the camera associated with $\tau_{\text{max}}$. □

The REC algorithm is formally described in Algorithm 3, where $D_i$ represents cameras visibility constraints ($A_i \subseteq D_i$).
Algorithm 3: Cameras Reconfiguration (Camera \( c_i \))

Input: \( A_i = [\ell_i, r_i], D_i = [\gamma_i, \tau_i], v_i^{\text{max}} \);

Require: \( \{A_1, \ldots, A_n\} \) is a partition of \( \Gamma \);

Set \( r_i^{\text{max}} = \tau_i \) and \( q_i = c_i \);

Move according to Algorithm 2;

if Communication between cameras \( c_i \) and \( c_{i+1} \) then

Transmit \( \ell_i, r_i, r_i^{\text{max}}, q_i \) to \( c_{i+1} \);

Receive \( \ell_{i+1}, r_{i+1}, r_{i+1}^{\text{max}}, q_{i+1} \) from \( c_{i+1} \);

Compute \( m = (\ell_i v_i^{\text{max}} + r_i + 1 v_i^{\text{max}}) / (v_i^{\text{max}} + v_{i+1}^{\text{max}}) \);

Update \( r_i, \ell_i+1 \) as:

\[
    r_i = \ell_i+1 = \begin{cases} 
        m & \text{if } m \in [\gamma_{i+1}, \tau_i], \\
        \pi_i & \text{if } m > \tau_i, \\
        \gamma_{i+1} & \text{if } m < \gamma_{i+1},
    \end{cases}
\]

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

case \( q_i \geq c_i \)

if \( q_{i+1} > c_{i+1} \) then

\[
    \hat{r}_i^{\text{max}} = r_i^{\text{max}} = \max\{\tau_i, \tau_{i+1}, \hat{r}_i^{\text{max}}\};
\]

\[
    q_i = q_{i+1} = \arg \max_{c_i, c_{i+1}, q_{i+1}} \{\tau_i, \tau_{i+1}, \hat{r}_i^{\text{max}}\};
\]

else if \( q_{i+1} \leq c_{i+1} \) then

\[
    \hat{r}_i^{\text{max}} = r_i^{\text{max}} = \max\{\tau_i, \tau_{i+1}\};
\]

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

else if \( q_{i+1} < c_{i+1} \) then

\[
    \hat{r}_i^{\text{max}} = r_i^{\text{max}} = \max\{\tau_i, \tau_{i+1}, \hat{r}_i^{\text{max}}\};
\]

\[
    q_i = q_{i+1} = \arg \max_{c_i, c_{i+1}, q_{i+1}} \{\tau_i, \tau_{i+1}, \hat{r}_i^{\text{max}}\};
\]

else if \( q_{i+1} > c_{i+1} \) then

\[
    \hat{r}_i^{\text{max}} = r_i^{\text{max}} = \max\{\tau_i, \tau_{i+1}, \hat{r}_i^{\text{max}}\};
\]

\[
    q_i = q_{i+1} = \arg \max_{c_i, c_{i+1}, q_{i+1}} \{\tau_i, \tau_{i+1}, \hat{r}_i^{\text{max}}\};
\]

\[
    \hat{r}_{i+1}^{\text{max}} = r_{i+1}^{\text{max}} = \max\{\tau_i, \tau_{i+1}, \hat{r}_i^{\text{max}}\};
\]

\[
    q_i = q_{i+1} = \arg \max_{c_i, c_{i+1}, q_i, q_{i+1}} \{\tau_i, \tau_{i+1}, \hat{r}_i^{\text{max}}, \hat{r}_{i+1}^{\text{max}}\};
\]

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 \( i \)-th 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 V.1 (Convergence of REC)** Consider a set of \( n \) cameras installed along a one-dimensional open path \( \Gamma \). 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,

(i) for all iterations \( k \in \mathbb{N} \) and for all \( i \in \{1, \ldots, n\} \) the patrolling window \( A_i(k) \) satisfies \( A_i(k) \subseteq D_i \),

(ii) for all iterations \( k \in \mathbb{N} \) the set \( \{A_1(k), \ldots, A_n(k)\} \) is a partition of \( \Gamma \), and
\[
\tau^* = \lim_{k \to \infty} \max_{i \in \{1, \ldots, n\}} \frac{d_i(k)}{\tau^\text{max}_i} = \min \mathcal{P} \max_{i \in \{1, \ldots, n\}} \frac{d_i}{\tau^\text{max}_i},
\]
where \( \mathcal{P} \) is the set of partitions \( \{A_1, \ldots, A_n\} \) of \( \Gamma \) satisfying \( A_i \subseteq D_i \) for all \( i \in \{1, \ldots, n\} \) and \( d_i \) is the length of \( A_i \), and

(iii) 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 \( \ell_i \) is such that \( A_i \) belongs to the constraint set \( D_i \), so that statement (i) follows. Second notice that cameras persistently communicate over time. Indeed, (i) each camera sweeps back and forth its assigned segment, (ii) cameras wait at their boundaries until communication with a neighboring camera takes place, and (iii) cameras 1 and \( n \) do not stop at \( \ell_1 \) and \( r_n \), respectively. In particular, it can be shown that any two neighboring cameras communicate within an interval of finite length. Then, statement (ii) follows from [18, Theorem IV.1]. Third, because of the persistence of communication among cameras, the value \( \tau^\text{max}(k) \), which is decreasing in \( k \), propagates in some time \( T^\text{prop} \) to every camera, for every iteration \( k \). Let \( \bar{t} \) be such that \( \tau^\text{max}(\bar{t}) = \tau^* + \varepsilon \), for some \( \varepsilon \in \mathbb{R}_{>0} \). Then, after time \( \bar{t} + T^\text{prop} \), the period \( T_i \) of \( c_i \) is within \( 2\varepsilon \) of \( 2\tau^* \), for all \( i \in \{1, \ldots, n\} \). Statement (iii) follows by letting \( \varepsilon \) tend to zero. \( \blacksquare \)

As stated in Theorem V.1, Algorithm 3 drives the cameras towards an Equal-waiting trajectory. Then, the detection performance of the cameras trajectory generated by our reconfiguration algorithm are as in Theorem III.1 with \( \tau^\text{max} = \tau^* \).
We now validate our reconfiguration algorithm via simulation. We consider two scenarios with 5 cameras. All cameras start their trajectory from some initial point in their patrolling window. In the first scenario (Fig. 10) cameras have the same maximum speed $v^{\text{max}} = 0.67$ m/s, and they are not subject to patrolling windows constraints. Relevant parameters for this simulation study are reported in Table II. Observe from Fig. 10 that the cameras trajectory converges to an Equal-waiting trajectory, and that the length of the largest patrolling window $\tau^{\text{max}}$ is decreasing and converges to $\tau^* = 6.2023$ s.

In the second scenario, cameras have different maximum speeds ($v_1 = 0.61$, $v_2 = 0.57$, $v_3 = 0.47$, $v_4 = 0.68$, $v_5 = 0.68$ m/s), and they are subject to patrolling windows constraints. Relevant parameters for this simulation study are reported in Table III. As shown in Fig. 11, the cameras trajectory converges to an Equal-waiting trajectory, and that the length of the largest patrolling window $\tau^{\text{max}}$ is decreasing and converges to $\tau^* = 6.65$ s.

VI. CONCLUSION

This work studies the problem of coordinating a team of autonomous cameras along a one-dimensional open path to detect moving intruders. We propose mathematical models of cameras
TABLE II
PARAMETERS AND RESULTS FOR UNIFORM CAMERAS SPEED.

<table>
<thead>
<tr>
<th></th>
<th>$c_1$</th>
<th>$c_2$</th>
<th>$c_3$</th>
<th>$c_4$</th>
<th>$c_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D_1$</td>
<td>[0.468]</td>
<td>[1.14 7.45]</td>
<td>[3.32 12.09]</td>
<td>[7.26 18.41]</td>
<td>[10.12 20]</td>
</tr>
<tr>
<td>$A_i(0)$</td>
<td>[0 2.91]</td>
<td>[2.91 5.38]</td>
<td>[5.38 9.67]</td>
<td>[9.67 14.26]</td>
<td>[14.26 20]</td>
</tr>
<tr>
<td>$A_i(\infty)$</td>
<td>[0 3.72]</td>
<td>[3.72 7.45]</td>
<td>[7.45 11.63]</td>
<td>[11.63 15.82]</td>
<td>[15.82 20]</td>
</tr>
</tbody>
</table>

Fig. 11. Simulation of REC with $n = 5$ cameras with non-uniform maximum speeds $v_{i\text{max}} \sim \mathcal{U}[0.45, 0.75]$ m/s and no patrolling windows constraints. In Fig. 11(a) we show the cameras trajectories starting from random positions. The dashed lines refer to the trajectories of the active boundaries. In Fig. 11(b) we report the dynamics of $\hat{\tau}_{i\text{max}}$. Notice that $\hat{\tau}_{i\text{max}}$ converges to the optimal value $\tau^* = 6.65$ s (dash-dot line).

and intruders, and we define the worst-case and average detection times 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 simulations and experiments.
TABLE III
PARAMETERS AND RESULTS FOR NON-UNIFORM CAMERAS SPEED.

<table>
<thead>
<tr>
<th>c1</th>
<th>c2</th>
<th>c3</th>
<th>c4</th>
<th>c5</th>
</tr>
</thead>
<tbody>
<tr>
<td>[0 20]</td>
<td>[0 20]</td>
<td>[0 20]</td>
<td>[0 20]</td>
<td>[0 20]</td>
</tr>
<tr>
<td>[0 4]</td>
<td>[4 8]</td>
<td>[8 12]</td>
<td>[12 16]</td>
<td>[16 20]</td>
</tr>
<tr>
<td>[0 4.04]</td>
<td>[4.04 7.81]</td>
<td>[7.81 10.94]</td>
<td>[10.94 15.48]</td>
<td>[15.48 20]</td>
</tr>
</tbody>
</table>

APPENDIX

This section contains a proof of Theorem III.1. We start with a lower bound for the average detection time.

**Lemma VI.1 (Lower bound on the average detection time)** For a set of n cameras with maximum velocities \(v_1^{\text{max}}, \ldots, v_n^{\text{max}}\) and sweeping times \(\tau_1, \ldots, \tau_n\), the average detection time of a \(2\tau^{\text{max}}\)-periodic cameras trajectory \(X\) satisfies the lower bound

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

**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 \(\gamma \in [\ell_i, r_i]\) satisfies the lower bound

\[
\int_{\ell_i}^{r_i} T_{\text{det}}(I_{t, \gamma}) - t \, d\gamma = \int_{0}^{d_i} T_{\text{det}}(I_{t, \gamma}) - t \, d\gamma \geq T_{\text{up}},
\]

where \(d_i = r_i - \ell_i\) and \(T_{\text{up}}\) equals

\[
\int_{0}^{x_i(t)} \frac{x_i(t) + 2(d_i - x_i(t))}{v_i^{\text{max}}} \, d\gamma + \int_{x_i(t)}^{d_i} \frac{d_i - x_i(t)}{v_i^{\text{max}}} \, d\gamma,
\]

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

\[
\int_{0}^{d_i} T_{\text{det}}(I_{t, \gamma}) - t \, d\gamma \geq T_{\text{down}},
\]

where \(T_{\text{down}}\) equals

\[
\int_{0}^{x_i(t)} \frac{x_i(t)}{v_i^{\text{max}}} \, d\gamma + \int_{x_i(t)}^{d_i} \frac{2x_i(t) + (d_i - x_i(t))}{v_i^{\text{max}}} \, d\gamma.
\]
To see this, consider the first case. The detection time of every intruder appearing at location \( \gamma > x_i(t) \) is at least \((d_i - x_i(t))/v_{\text{max}}\) (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 \( \gamma < x_i(t) \) is at least \((x_i(t) + 2(d_i - x_i(t)))/v_{\text{max}}\) (time needed by \( c_i \) to reach \( r_i \) and \( \ell_i \) starting from \( x_i(t) \)). The other case follows similarly.

It can be verified by simple manipulation that \( T_{\text{up}} = T_{\text{down}} = d_i^2/v_{\text{max}} \). Finally, it follows from (2) and \( \tau_i = d_i/v_{\text{max}} \) that
\[
ADT(X) = \frac{1}{TL} \int_0^T \int_\Gamma T_{\text{det}}(I_{t,\gamma}) - t \, d\gamma dt \geq \frac{1}{TL} \int_0^T \sum_{i=1}^n \frac{d_i^2}{v_{\text{max}}} \, dt = \frac{1}{L} \sum_{i=1}^n v_{\text{max}} \tau_i^2.
\]

It should be observed that the bound in Lemma VI.1 is tight for the case of a single camera, and it is conservative otherwise (see Fig. 4(a) and 5). We now characterize the average detection time of the Equal-waiting trajectory.

**Lemma VI.2 (Equal-waiting trajectory performance)** For a set of \( n \) cameras with sweeping times \( \tau_1, \ldots, \tau_n \), let \( X^{\text{eq}} \) be the Equal-waiting trajectory defined in Trajectory 1. Then
\[
ADT(X^{\text{eq}}) = \frac{1}{2} \left( \tau_{\text{max}} \sum_{i=1}^n \frac{v_{\text{max}} \tau_i^2}{2} \right).
\]

*Proof:* Observe that the function \( ADT(X^{\text{eq}}) \) can be written as
\[
ADT(X^{\text{eq}}) = \frac{1}{2 \tau_{\text{max}} L} \sum_{i=1}^n \int_0^{2 \tau_{\text{max}}} \int_{\ell_i}^{r_i} (T_{\text{det}}(I_{t,\gamma}) - t) \, d\gamma dt.
\]
Let \( i \) be odd and recall the description of \( x_i(t) \) given in Trajectory 1. Due to symmetry, it can be verified that
\[
\int_0^{2 \tau_{\text{max}}} \int_{\ell_i}^{r_i} (T_{\text{det}}(I_{t,\gamma}) - t) \, d\gamma dt = 2 \int_0^{\tau_{\text{max}}} \int_{\ell_i}^{r_i} (T_{\text{det}}(I_{t,\gamma}) - t) \, d\gamma dt
\]
Let \( 0 \leq t \leq \tau_{\text{max}} - \tau_i \), and notice that \( x_i(t) = r_i \). Observe that \( T_{\text{det}}(I_{t,\gamma}) = \tau_{\text{max}} \) for all \( \gamma \in [\ell_i, r_i] \), and \( T_{\text{det}}(I_{t,\gamma}) = 0 \) for \( \gamma = r_i \). Then
\[
\int_0^{\tau_{\text{max}} - \tau_i} \int_{\ell_i}^{r_i} (T_{\text{det}}(I_{t,\gamma}) - t) \, d\gamma dt = \frac{\tau_{\text{max}}^2 - \tau_i^2}{2} \frac{d_i}{\tau_{\text{max}} - \tau_i}.
\]
Let \( \tau_{\text{max}} - \tau_i \leq t \leq \tau_{\text{max}} \). Observe that \( T_{\text{det}}(I_{t,\gamma}) = \tau_{\text{max}} \) for \( \gamma \in [\ell_i, x_i(t)] \), \( T_{\text{det}}(I_{t,\gamma}) = 0 \) for \( \gamma = x(t) \), and \( T_{\text{det}}(I_{t,\gamma}) = 2 \tau_{\text{max}} \) for \( \gamma \in (x_i(t), r_i] \). Thus,
\[
\int_0^{\tau_{\text{max}} - \tau_i} \int_{\ell_i}^{r_i} (T_{\text{det}}(I_{t,\gamma}) - t) \, d\gamma dt = \int_0^{\tau_{\text{max}} - \tau_i} \int_{\ell_i}^{x_i(t)} (\tau_{\text{max}} - t) \, d\gamma + \int_{x_i(t)}^{r_i} (2 \tau_{\text{max}} - t) \, d\gamma dt.
\]
Since \( x_i = r_i + (t - (\tau_{\text{max}} - \tau_i)v_i^{\text{max}}) \) (see Trajectory 1), it follows from the above expression that
\[
\int_{\tau_i}^{\tau_{\text{max}}} \int_{\ell_i}^{r_i} (T_{\text{det}}(I_{t, \gamma}) - t) \, d\gamma \, dt = \int_{\tau_i}^{\tau_{\text{max}}} (2\tau_{\text{max}} - t) d_i - \tau_{\text{max}}(\tau_{\text{max}} - t)v_i^{\text{max}} \, dt
\]
\[
= \frac{1}{2} \tau_{\text{max}}^2 \tau_i^2 v_i^{\text{max}} + \frac{1}{2} d_i \tau_i^2.
\]
(A-3)

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

We now conclude with a proof of Theorem III.1.

**Proof of Theorem III.1:** From Lemma VI.1 and VI.2 we have
\[
\frac{\text{ADT}(X^{\text{eq}})}{\text{ADT}^*} = \frac{\frac{1}{2} \sum_{i=1}^{n} d_i \tau_i}{\frac{1}{2} \sum_{i=1}^{n} \tau_i^{\text{max}} v_i^{\text{max}} \tau_i^2} = \frac{\frac{1}{2} \sum_{i=1}^{n} d_i \tau_i}{\frac{1}{2} \sum_{i=1}^{n} d_i \tau_i} \leq \frac{1}{2} + \frac{L \tau_{\text{max}}}{\tau_{\text{max}} \sum_{i=1}^{n} d_i} = \frac{\tau_{\text{max}} + \tau_{\text{min}}}{2\tau_{\text{min}}},
\]
where we have used \( L = \sum_{i=1}^{n} d_i \) and \( \tau_{\text{min}} \leq \tau_i \) for all \( i \in \{1, \ldots, n\} \). To show the second bound notice that
\[
\frac{\text{ADT}(X^{\text{eq}})}{\text{ADT}^*} = \frac{\frac{1}{2} \sum_{i=1}^{n} d_i \tau_i}{\frac{1}{2} \sum_{i=1}^{n} \tau_i^{\text{max}} v_i^{\text{max}} \tau_i^2} \leq \frac{1}{2} + \frac{\sum_{i=1}^{n} d_i \tau_i}{\sum_{i=1}^{n} d_i} = \frac{1}{2} + \frac{\sum_{i=1}^{n} d_i \tau_i}{\sum_{i=1}^{n} d_i} \leq \frac{1}{2} + \frac{L d_{\text{max}}}{\sum_{i=1}^{n} d_i} \leq \frac{1}{2} + \frac{n d_{\text{max}}}{2d_{\text{min}}} \leq \frac{(n + 1)d_{\text{max}}}{2d_{\text{min}}}.
\]

We now show the last part of the Theorem. Assume that all cameras move at unitary speed and, without affecting generality, that \( d_1 = d_{\text{max}} \). Notice that
\[
\frac{\text{ADT}(X^{\text{eq}})}{\text{ADT}^*} = \frac{\frac{1}{2} \sum_{i=1}^{n} d_i \tau_i}{\frac{1}{2} \sum_{i=1}^{n} \tau_i^{\text{max}} v_i^{\text{max}} \tau_i^2} = \frac{\frac{1}{2} \sum_{i=1}^{n} d_i d_i}{\frac{1}{2} \sum_{i=1}^{n} d_i d_i} = \frac{1}{2} \left( 1 + \frac{\sum_{i=1}^{n} y_i}{1 + \sum_{i=1}^{n} y_i^2} \right),
\]
where \( y_i = d_i/d_1 \). Consider the minimization problem
\[
\min_{\{K, x_2, \ldots, x_n\}} K,
\]
subject to \( 1 + \sum_{i=2}^{n} y_i \leq K (1 + \sum_{i=2}^{n} y_i^2) \),
(A-4)

and the associated Lagrangian function [22]
\[
\mathcal{L} = K + \lambda \left( 1 - K + \sum_{i=2}^{n} y_i - K y_i^2 \right).
\]
Following standard optimization theory, necessity optimality conditions for the minimization
problem (A-4) are
\[
\frac{\partial L}{\partial K} = 0 \implies 1 - \lambda \left(1 + \sum_{i=2}^{n} y_i^2\right) = 0,
\]
\[
\frac{\partial L}{\partial y_i} = 0 \implies \lambda \left(1 - 2K y_i\right) = 0, \text{ for } i \in \{2, \ldots, n\},
\]
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


