2012 American Control Conference
Fairmont Queen Elizabeth, Montréal, Canada
June 27-June 29, 2012

Distributed multi-camera synchronization for smart-intruder detection

Markus Spindler Fabio Pasqualetti Francesco Bullo

Abstract— **This work proposes algorithms to control the trajectory of a team of cameras for video surveillance. We consider a chain of cameras installed in an environment. These cameras are used to detect *smart* intruders, who are aware of the cameras' configuration at each time instant, and who schedule their motion to avoid detection, if possible. For this problem setup, we first obtain a lower bound on the worst-case and on the average detection time of a smart intruder. We then propose a team trajectory for the cameras, called the *equal-waiting trajectory*, with minimum worst-case detection time and constant-factor optimal average detection time. Additionally, we design and analyse a distributed algorithm to steer the cameras in finite time towards the equal-waiting trajectory. Finally, the performance and robustness of our distributed algorithm is further analyzed through a simulation study.**

I. Introduction

Coordinated teams of autonomous agents have recently been used for many tasks requiring repetitive execution, including the monitoring of oil spills [1], the detection of forest fires [2], the track of border changes [3], and the patrol (surveillance) of an environment [4]. The surveillance of an area of interest requires the agents to continuously and repeatedly sweep the environment, and the challenging problem consists of scheduling the agents' trajectories so as to optimize a certain performance criteria.

A. Problem description

In this work we consider a network of identical Pan-Tilt-Zoom 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 problem of perimeter surveillance is a special case of this framework. We assume the cameras to be subject to physical constraints, e.g., 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 the f.o.v. of a camera. Regarding intruders, we assume them 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, if possible. We study the problem of scheduling the cameras trajectory as to minimize the *worst case detection time* and the *average detection time* of a smart intruder.

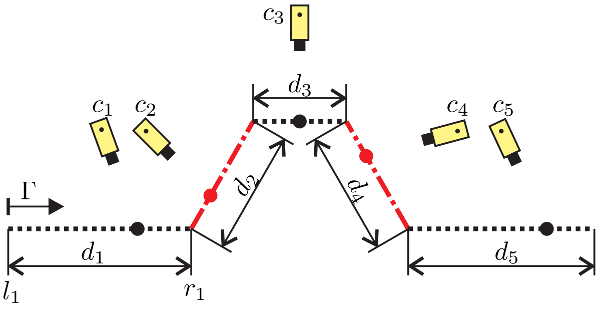

Fig. 1. This figure shows five cameras installed along a one dimensional path. The field of view of each camera is a point on the path. The cameras synchronize their motion to detect smart moving intruders along the path.

B. Related work

The problem of patrolling an environment by means of a team of autonomous robots has received attention from scientists interested in mobile robotics. Typically, (i) the environment is represented by a graph on which the agents' motion is constrained, and (ii) the patrolling performance is given by the *worst-case* detection time of a *static* event. In [5], [6] an empirical evaluation of certain patrolling heuristics is performed. In [4] and [2], an efficient and distributed solution to the (worst-case) perimeter patrolling problem for robots with zero communication range is proposed. In [7] 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 focus on the average detection time, as opposed to the worst case detection time.

In the context of camera networks, the perimeter patrolling problem has recently been studied in [8], [9]. In these works, distributed algorithms are proposed for the cameras to partition a one-dimensional environment, and to synchronize along a trajectory with minimum worst-case detection time of static events. We improve the results along this direction by showing that the strategies proposed in [8], [9] generally fail at detecting smart intruders, and by focusing on the average detection time of smart intruders while maintaining minimum worst case detection time.

C. Contribution

The contribution of this work is threefold. First, we mathematically formalize the concept of camera trajectory

This work was supported in part by NSF Award CPS 1035917 and ARO Award W911NF-11-1-0092.

Markus Spindler, Fabio Pasqualetti, and Francesco Bullo are with the Center for Control, Dynamical Systems and Computation, University of California, Santa Barbara, `spindlermarkus@gmail.com`, `{fabiopas,bullo}@engineering.ucsb.edu`.

978-1-4577-1094-0/12/$26.00 ©2012 AACC

and smart intruder, then we prove a lower bound for the worst-case and average detection time of smart intruders. Second, we propose the *equal-waiting* camera trajectory. This trajectory achieves minimum worst case detection time, and an average detection time within a constant factor from the minimum. Third and finally, we develop a distributed synchronization algorithm to steer the cameras towards an equal-waiting trajectory. In our distributed setup, only neighboring cameras need to communicate at some specific times. Our synchronization algorithm converges in finite time, which we characterize, and it requires only minimal information to be implemented. Moreover, we perform a simulation study to show that our synchronization algorithm is robust against camera failures and motion uncertainties.

The remainder of the paper is organized as follows. In Section II we introduce our notation and we formulate the considered problem. In Section III we present and characterize our equal-waiting trajectory and our synchronization algorithm. Section IV and Section V contain, respectively, an illustrative example and a proof of our theoretical results. Finally, our conclusion and final remarks are in Section VI.

II. Problem formulation

Consider a set of $n \in \mathbb{N}$ identical cameras installed along a one dimensional open path Γ (cf. Fig 1). Assume that (i) the f.o.v. of each camera is a point on Γ, and that (ii) the motion of each f.o.v. is uniquely determined by the pan movement of the corresponding camera. Furthermore, we assume that, for a sufficiently high image resolution, the f.o.v. of each camera cannot move faster than $\dot{x}_{\max}$. Without affecting generality, we let $|\dot{x}_{\max}| = 1$. Consequently, we assume each camera to be equipped with a low-level controller that maintains the speed of its f.o.v. at $\dot{x}_{\max}$.[1] Let Γ_0 and Γ_f be the two extremes of Γ. For simplicity, we label the cameras in increasing order from c_1 to c_n according to their distance from Γ_0 on Γ. Let $x_i(t)$ and $\dot{x}_i(t)$ be the position and the velocity of the i-th f.o.v., where $l_i \leq x_i(t) \leq r_i$ and $l_i, r_i \in \Gamma$. We refer to $s_i = [l_i, r_i]$ as to the i-th cluster, and we let d_i be the distance on Γ between l_i and r_i. We additionally assume that $l_i = r_{i-1}$, with $i = 2, \dots, n$, so that $\{s_1, \dots, s_n\}$ is in fact a partition of Γ. A *(camera) trajectory* is an array $X(t) = \{x_1(t), \dots, x_n(t)\}$ of n continuous and *periodic* functions describing the motion of the camera f.o.v. on Γ, with uniform period $T \in \mathbb{R}_{\geq 0}$. Hence, it holds $X(t+T) = X(t)$, that is $x_i(t+T) = x_i(t)$ for $i \in \{1, \dots, n\}$.

In this work, we focus on the problem of detecting moving objects, referred here as to *intruders*, by means of a set of cameras. We consider the case of intruders with speeds greater than or equal to the camera speed. We represent the intruder as a point on Γ, and we let $t_0 \in \mathbb{R}_{\geq 0}$ be the time at which the intruder appears on Γ. Moreover, we let the continuous map $p : \mathbb{R}_{\geq t_0} \mapsto \Gamma$ describe the position of the intruder at a certain time $t \geq t_0$. We say that an intruder is detected at time $t_d \in \mathbb{R}_{\geq t_0}$ if $p(t_d) \in X(t_d)$. We focus on *smart* intruders that appear uniformly distributed on Γ. We assume that intruders have full knowledge of the cameras trajectory, and choose their trajectory $p(t)$ to avoid detection 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 of a smart intruder $p^*_{t_0,p_0}(t)$ is such that

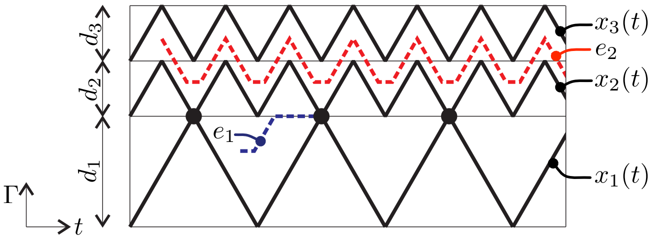

Fig. 2. The figure shows a cameras trajectory in which cameras 1 and 2 are synchronized, while cameras 2 and 3 are not synchronized. Notice that, because of the synchronization, intruder e_1 is eventually detected when the f.o.v. of cameras 1 and 2 occupy the same position. Instead, intruder e_2 may avoid detection by properly choosing its trajectory.

$$p^*_{t_0,p_0}(t) = \arg\max\{t^*_d(p) - t_0 \mid p \in \Phi(t_0, p_0)\},$$

where $\Phi(t_0, p_0)$ is the set of continuous maps $p : \mathbb{R}_{\geq t_0} \mapsto \Gamma$ with $p(0) = p_0$, and

$$t^*_d(p) = \min\{t \mid t \geq t_0, p(t) \in X(t)\}.$$

Notice that the trajectory $p^*_{t_0,p_0}(t)$ is, in general, not unique. In what follows, we design camera trajectories that minimizes the *worst case detection time* (WDT) and the *average detection time* (ADT) of a smart intruder. In particular,

$$\mathrm{WDT}(X(t)) = \max_{p_0, t_0} t^*_d(p) - t_0, \tag{1}$$

and

$$\mathrm{ADT}(X(t)) = \frac{1}{TL}\int_0^T \int_\Gamma (t^*_d(p^*_{\tau,\gamma}) - \tau) d\gamma \, d\tau, \tag{2}$$

where $T = 2\max\{d_1, \dots, d_n\}$, and $L = \sum_{i=1}^{n} d_i$.

Problem 1 ***(Trajectory design)*** *For a set of n cameras on an open path, design a camera trajectory $X^*(t)$ such that*

$$X^*(t) = \arg\min_{X(t)} \ \mathrm{ADT}(X(t)).$$

With the above definition of $X^*(t)$, we let

$$\mathrm{ADT}^* = \mathrm{ADT}(X^*(t)).$$

Finally, we say that a camera's trajectory is *synchronized*, if, for each pair of neighboring cameras c_i and c_{i+1}, there exists $t \in [0, T]$ such that $x_i(t) = x_{i+1}(t)$.

Remark 1 ***(Worst-case detection time)*** *It can be shown that the minimum value of* $\mathrm{WDT}(X(t))$ *equals* $2d_{\max}$*, where* $d_{\max} = \max\{d_1, \dots, d_n\}$ *[7], [8]. Observe that any* $2d_{\max}$*-periodic synchronized cameras trajectory attains minimum worst-case detection time of smart intruders (cf. Fig. 2). In the next section we propose a particular* $2d_{\max}$*-periodic*

[1] For instance, the controller may set the panning velocity of the i-th camera to $\dot{\alpha}_i = \dot{x}_{\max}/(a_i \sec^2(\alpha))$, where α denotes the panning angle, and a_i is the distance of the i-th camera from Γ.

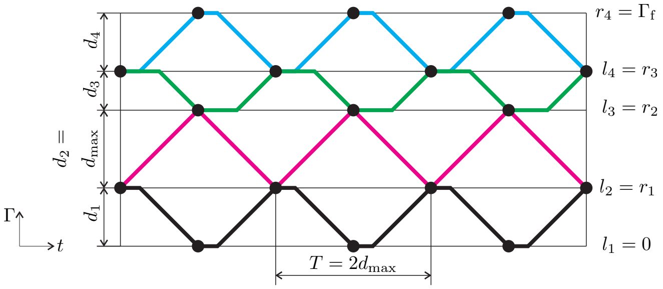

Fig. 3. This figure shows the equal-waiting trajectory for 4 cameras. Notice that (i) the cameras are synchronized, (ii) the trajectory is $2d_{\max}$-periodic, and (iii) the waiting time of each camera is the same at both its boundaries.

synchronized camera trajectory as solution to Problem 1. Hence, our proposed trajectory is also optimal with respect to the worst-case detection time of smart intruders. Finally note that, for the trajectories described in [8], the worst-case detection time of a smart intruder is unbounded (cf. Fig. 2).

A second focus of this paper is on developing a distributed synchronization algorithm for the cameras to converge to the desired trajectory. In our distributed setting we allow communication only between neighboring cameras.

Problem 2 ***(Distributed Algorithm design)*** *For a set of n cameras on an open path, design a distributed algorithm to steer the cameras towards a trajectory with minimum average detection time of smart intruders.*

III. Main results

In this section we describe an approximate solution to Problem 1, and we design a distributed algorithm for the cameras to converge to such a trajectory. We remark that, for some cases, an exact solution to Problem 1 could be computed through standard optimization techniques [10]. Such computation, however, is not scalable with the number of cameras, and it is not amenable to distributed implementation. Our approximate solution, instead, is extremely simple and efficient to compute, and its performance is shown to be within a certain bound of the optimum. Moreover, our approximate solution is valid for every number of cameras and environment configuration. The camera trajectory we propose can informally be described as follows.

> *(Informal description)* Each camera continuously sweeps its cluster at maximum speed, and it stops for a certain waiting time when its f.o.v. reaches a boundary. The waiting time of each camera is the same at both boundaries. Additionally, all cameras have the same period, and the trajectories are synchronized such that two neighboring cameras arrive at their shared boundary at the same time.

Since we let each camera wait the same interval at its two boundaries, 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 Remark 1, the equal-waiting cameras trajectory is optimal with respect to the worst-case detection time criterion. Indeed, by construction, the equal-waiting camera trajectory is synchronized and $2d_{\max}$-periodic. Next we show that the equal-waiting camera trajectory is constant factor optimal with respect to the average detection time criterion. A proof of this result is postponed to Section V.

Trajectory 1: *Equal-waiting trajectory (camera i)*

```
Input : $d_{\max}, r_i, l_i, d_i$;
Set   : $\tau_i(k) := (k+1)d_{\max} - d_i,\ k = 0, 2, \dots$;
if $i$ is odd then
    $x_i(t) := r_i$ for $\tau_i(k-1) + d_i \leq t \leq \tau_i(k)$;
    $x_i(t) := -t + \tau_i(k) + r_i$ for $\tau_i(k) \leq t \leq \tau_i(k) + d_i$;
    $x_i(t) := l_i$ for $\tau_i(k) + d_i \leq t \leq \tau_i(k+1)$;
    $x_i(t) := t - \tau_i(k+1) + l_i$ for $\tau_i(k+1) \leq t \leq \tau_i(k+1) + d_i$;
else if $i$ is even then
    $x_i(t) := l_i$ for $\tau_i(k-1) + d_i \leq t \leq \tau_i(k)$;
    $x_i(t) := t - \tau_i(k+1) + l_i$ for $\tau_i(k) \leq t \leq \tau_i(k) + d_i$;
    $x_i(t) := r_i$ for $\tau_i(k) + d_i \leq t \leq \tau_i(k+1)$;
    $x_i(t) := -t + \tau_i(k) + r_i$ for $\tau_i(k+1) \leq t \leq \tau_i(k+1) + d_i$;
```

Theorem III.1 ***(Performance of equal-waiting trajectories)*** *For a set of n cameras with cluster lengths $d_1, \dots, d_n$ and periodic trajectories, let $X(t)$ be the equal-waiting trajectory defined in Trajectory 1.*

1) *The optimal average detection time for a smart intruder satisfies the lower bound:*
$$\mathrm{ADT}^* \geq \frac{\sum_{i=1}^n d_i^2}{L}, \tag{3}$$
where $L = \sum_{i=1}^{n} d_i$.
2) *The equal-waiting trajectory $X(t)$ has performance*
$$\mathrm{ADT}(X(t)) = \frac{1}{2}d_{\max} + \frac{1}{2}\frac{\sum_{i=1}^n d_i^2}{L}, \tag{4}$$
where $d_{\max} = \max\{d_1, \dots, d_n\}$.
3) *The equal-waiting trajectory $X(t)$ has performance within a constant factor of the optimum, that is,*
$$\frac{\mathrm{ADT}(X(t))}{\mathrm{ADT}^*} \leq \min\left\{\frac{1}{2}\left(1 + \frac{d_{\max}}{d_{\min}}\right), \frac{3+\sqrt{n}}{4}\right\}, \tag{5}$$
where $d_{\min} = \min\{d_1, \dots, d_n\}$.

The following facts follow from Theorem III.1. First, the performance of the equal-waiting trajectory is within a constant factor of the optimal if either $d_{\max}/d_{\min}$ or n are constant. Second, if all clusters have the same length, i.e. $d_{\max} = d_{\min}$, then Trajectory 1 is an optimal solution to Problem 1. Third, the lower bound is independent of the cluster arrangement. We now design a distributed feedback algorithm that steers the cameras towards an equal-waiting trajectory. The algorithm is informally described as follows.

Algorithm 2: *Distributed camera synchronization along an equal-waiting trajectory (camera i)*

```
Input : $d_{\max}, l_i, r_i, d_i$;
Set   : $t_{i,w} := d_{\max} - d_i$, $x_0(t) := l_1$ and $x_{n+1}(t) := r_n$ for all $t$;
1 Move towards $l_i$ with $|\dot{x}_i(t)| = 1$;
2 while True do
3     if $x_i(t) = x_{i-1}(t)$ or $x_i(t) = x_{i+1}(t)$ then
4         wait until time $t + t_{i,w}$;
5         move towards the opposite boundary with $|\dot{x}_i(t)| = 1$;
6     else wait;
```

(Informal description) Each camera moves to its left boundary, and waits until the f.o.v. of its left neighboring camera occupies the same position. Then, both cameras stop as specified in Trajectory 1, and finally move towards the opposite boundary.

Our distributed algorithm is formally described in Algorithm 2. Two comments are in order. First, we set $x_0(t) := l_1$ (resp. $x_{n+1}(t) := r_n$) for all times t, because l_1 (resp. r_n) is the left (resp. right) extreme of the path Γ. Second, for the implementation of the proposed distributed algorithm, each camera is required to know only the endpoints of its cluster, the length of cluster, and the longest clusters length, and to be able of communicating with a neighboring camera. 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 cluster lengths $d_1, \dots, d_n$, let $X(t)$ be the camera trajectory generated by Algorithm 2. Let $\bar{t} = nd_{\max}$. Then, $X(t \geq \bar{t})$ is an equal-waiting trajectory.*

Proof: Notice that the f.o.v. of camera 1 coincides with the f.o.v. of camera 2 within time $\max\{2d_1, d_2\} \leq 2d_{\max}$. Then, the f.o.v. of camera c_i coincides with the f.o.v. of camera c_{i+1} within time $(i+1)d_{\max}$. Hence, within time $nd_{\max}$ the camera trajectory coincides with the equal-waiting trajectory in Trajectory 1. The statement follows. ■

Notice that In Algorithm 2 (line 1) the cameras could equivalently move to their right boundary.

IV. An illustrative example

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 clusters lengths $\{d_1, \dots, d_n\}$, where $d_1 = d_{\max} = 1$, and d_i, with $i = 2, \dots, n$, is uniformly distributed within the interval $(0, 1]$. For each configuration, we design the equal-waiting trajectory $X(t)$ and evaluate the cost $\text{ADT}(X(t))$. Additionally, for each configuration we compute the lower bound in equation (3). Fig. 4 a) shows the results of this study. Observed that, for large values of n and uniformly distributed cluster lengths, the expected average detection time is much smaller than the bound described in Theorem III.1.

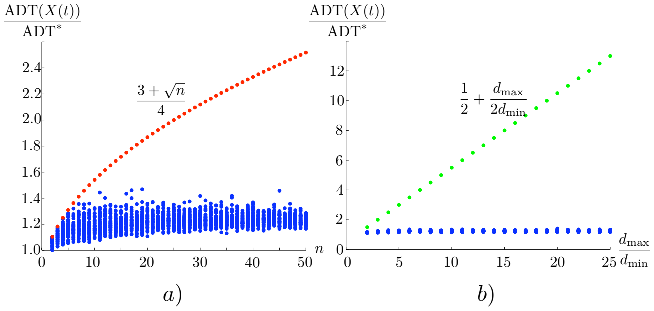

Fig. 4. Average detection time for the equal-waiting trajectory as a function of the number of cameras (a), and of the path partitioning (b). Notice that the bounds in Theorem III.1 for the equal-waiting trajectory are conservative.

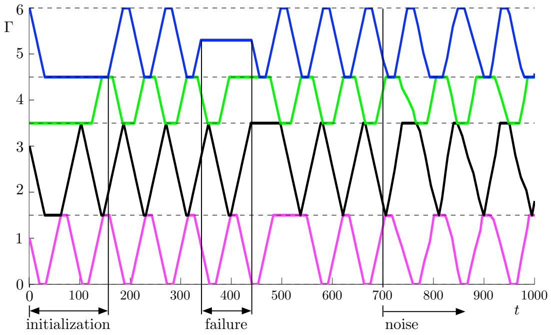

Fig. 5. This figure shows the trajectory obtained via Algorithm 2. Notice that (i) the cameras are synchronized after the initialization transient, (ii) the algorithm recovers after a temporary failure of camera 4 between time 340s and 440s, and (iii) the algorithm is robust when the cameras speed (unitary) is affected by noise starting from time $t = 700$s. For the simulation, the noise is 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 $d_{\max}/d_{\min}$ between 2 and 25. Specifically we let $d_1 = d_{\max} = 1$, and d_i, with $i = 2, \dots, 50$ be uniformly distributed within the interval $[d_{\min}/d_{\max}, 1]$. For each value of $d_{\max}/d_{\min}$ we generate 50 sets of cluster lengths $\{d_1, \dots, d_n\}$, then compute the equal-waiting trajectory $X(t)$, evaluate the cost $\text{ADT}(X(t))$, and compute the lower bound in equation (3). Fig. 4 b) shows that, for large values of $d_{\max}/d_{\min}$, the bound in Theorem III.1 is again conservative.

For our third simulation study, we consider a set of 4 cameras, and a fixed set of clusters with lengths $\{d_1, \dots, d_4\}$. Fig. 5 shows the trajectory obtained by means of Algorithm 2. Notice that (i) the cameras start at random initial positions, (ii) the algorithm is robust to temporary camera failures, and (iii) the average detection time degrades gracefully in the presence of motion uncertainties.

V. Proof of Theorem III.1

This section contains a sketch of the proof of Theorem III.1. All missing proofs can be found in [10]. In subsection V-A we characterize some useful properties of a minimum average detection time cameras trajectory. In subsection V-B

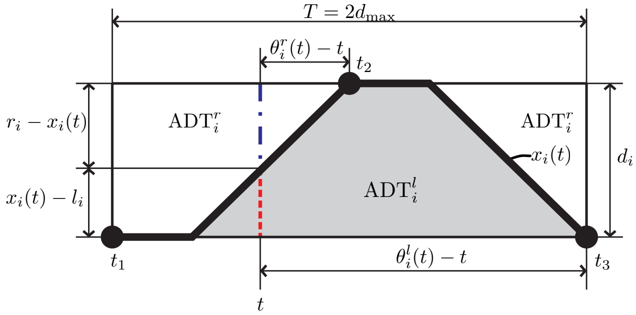

Fig. 6. This figure illustrates the computation of the average detection time of the periodic trajectory $x_i(t)$ within the period $[t_1, t_3]$. Three catch-points are considered for this example (t_1, t_2, t_3). Following Lemma V.1, the average detection time of $x_i(t)$ is computed as $\mathrm{ADT}_i^r + \mathrm{ADT}_i^l$. Smart intruders appearing at time t and in the interval $[l_i, x_i(t))$ are detected at time t_3. Likewise, smart intruders appearing at time t and in the interval $(x_i(t), r_i]$ are detected at time t_2. Finally, intruders appearing at time t and $x_i(t)$ are immediately detected.

we prove a lower bound for the minimum average detection time, we characterize the performance of the equal-waiting camera trajectory, and we conclude the proof. The following definition will be used in the proof. For a camera trajectory $X(t)$, the sets of *left catch-points* are defined as

$$\mathrm{CP}_i^l = \begin{cases} \{t \mid x_i(t) = x_{i-1}(t)\}, & \text{if } 2 \leq i \leq n, \\ \{t \mid x_i(t) = \Gamma_0\}, & \text{if } i = 1, \end{cases}$$

and the sets of *right catch-points* as

$$\mathrm{CP}_i^r = \begin{cases} \{t \mid x_i(t) = x_{i+1}(t)\}, & \text{if } 1 \leq i \leq n-1, \\ \{t \mid x_i(t) = \Gamma_{\mathrm{f}}\}, & \text{if } i = n. \end{cases}$$

A. Properties of an optimal trajectory

The performance function (2) can be rewritten in a more convenient form (see Fig. 6).

Lemma V.1 ***(Modified performance function)*** *Let $X(t) = \{x_1(t), \dots, x_n(t)\}$ be a T-periodic camera trajectory. Let CP_i^l and CP_i^r denote, respectively, the sets of left and right catch-points of $X(t)$. We have*

$$\mathrm{ADT}(X(t)) = \frac{1}{T \sum_{i=1}^{n} d_i} \sum_{i=1}^{n} \mathrm{ADT}_i^l(x_i(t)) + \mathrm{ADT}_i^r(x_i(t)), \tag{6}$$

where,

$$\mathrm{ADT}_i^l(x_i(t)) = \int_0^T (x_i(t) - l_i)(\theta_i^l(t) - t)dt,$$
$$\mathrm{ADT}_i^r(x_i(t)) = \int_0^T (r_i - x_i(t))(\theta_i^r(t) - t)dt,$$

and

$$\theta_i^l(t) = \min_{q \in \mathrm{CP}_i^l} q \geq t, \quad \theta_i^r(t) = \min_{q \in \mathrm{CP}_i^r} q \geq t.$$

Notice that a cameras trajectory is synchronized if and only if all the sets CP_i^l and CP_i^r are nonempty. It is now clear from Lemma V.1 that the camera trajectories $X(t)$ must be synchronized for $\mathrm{ADT}(X(t))$ to be finite.

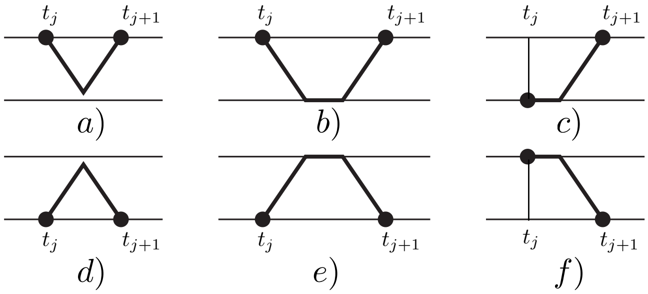

Fig. 7. This figure shows all possible trajectories between two catch-points leading to a minimum average detection time camera trajectory. Notice that the cameras either move at maximum speed, or they stop at a boundary.

Lemma V.2 ***(Necessity of synchronization)*** *Let $X(t) = \{x_1(t), \dots, x_n(t)\}$ be a periodic trajectory. Then*

$$\mathrm{ADT}(X(t)) < \infty$$

if and only if $\mathrm{CP}_i^l \neq \emptyset$ and $\mathrm{CP}_i^r \neq \emptyset$ for all $i = 1, \dots, n$.

Following Lemma V.1 it is possible to determine the shape of an optimal trajectory between two consecutive catch-points. Given the sets CP_i^l and CP_i^r, let $\mathrm{CP}_i = (t_0, \dots, t_k)$, where $t_i \in \mathrm{CP}_i^l \cup \mathrm{CP}_i^r$, $t_i < t_{i+1}$, and $t_k \leq t_0 + T$. Notice that CP_i is an ordered sequence of catch-points.

Lemma V.3 ***(Trajectory shape)*** *Let $\mathrm{CP} = \{\mathrm{CP}_1, \dots, \mathrm{CP}_n\}$ be a given sequence of catch-points. A camera trajectory $X(t)$ with CP as catch-points satisfies $\mathrm{ADT}(X(t)) = \mathrm{ADT}^*$ if and only if the trajectory of each camera i between any two consecutive points $t_j, t_{j+1} \in \mathrm{CP}_i$ is as in Fig. 7.*

As a consequence of Lemma V.2, if the catch-points are given, then an optimal cameras trajectory can be obtained from the trajectories in Fig. 7, where the speed of each camera is either zero or maximum. Hence, the problem of designing optimal camera trajectories reduces to the problem of finding a set of catch-points yielding optimal performance.

B. Performance bounds

We now derive a lower bound for the average detection time, and we characterize the performance of the equal-waiting trajectory. The lower bound is obtained as the sum of the lower bounds for the detection time of each cluster. The lower bound for a single cluster is obtained by assuming that, for each camera c_i, neighboring cameras c_{i-1} and c_{i+1} are such that $x_{i-1}(t) = l_i$ and $x_{i+1}(t) = r_i$ at all times.

Lemma V.4 ***(Sequence of catch-points)*** *For a camera trajectory $X(t)$, assume that $\mathrm{CP}_i^l = \{t \mid x_i(t) = l_i\}$ and $\mathrm{CP}_i^r = \{t \mid x_i(t) = r_i\}$ for each camera i. Let CP_i be the ordered sequence of catch-points within time $[0, T]$. Then, $\mathrm{ADT}(X(t)) = \mathrm{ADT}^*$ only if there exists no consecutive catch-points $t_j, t_{j+1}, t_{j+2} \in \mathrm{CP}_i$ such that $t_j, t_{j+1}, t_{j+2} \in \mathrm{CP}_i^l$ or $t_j, t_{j+1}, t_{j+2} \in \mathrm{CP}_i^r$, for each camera i.*

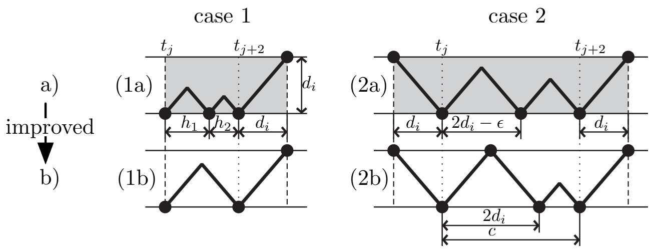

Fig. 8. Sequence of catch-points. Case 1 and case 2 illustrate the two possible cases of $t_j, t_{j+1}, t_{j+2} \in \text{CP}^l$. The cases in row a) show the trajectories with three consecutive catch-points on one boundary. Row b) shows how these trajectories can be improved in order to get a smaller ADT.

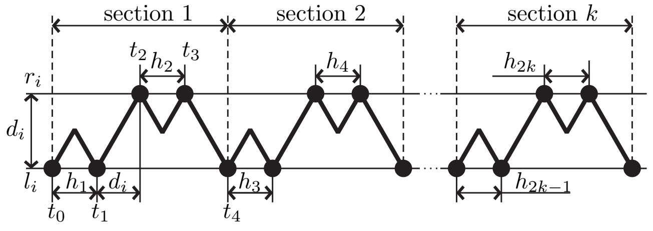

Fig. 9. A cameras trajectory is reported in this picture as a function of the parameters h_i and k. A minimum average detection time trajectory is obtained by optimizing the average detection time over values of h_i and k.

Lemma V.4 can be proven by showing that the average detection time of a trajectory with three consecutive catch-points on the same boundary can be improved by modifying such trajectory to have only two consecutive catch-points (cf. Fig. 8). Following Lemma V.4 and Lemma V.3, we now derive a parameterized model for an optimal single camera trajectory. This model is illustrated in Fig. 9 and an explanation follows. Notice that there are no three consecutive catch-points at the same boundary, and that the camera moves at maximum speed between catch-points. Moreover, since we assume that the neighboring cameras behave as walls, the catch-points coincide with the instants at which camera c_i is at the boundaries. Finally, the parameters k and $h_1, \dots, h_{2k}$ fully describe the trajectory. Next, we compute a set of parameters yielding minimum average detection time.

Lemma V.5 ***(Lower bound)*** *For a set of n cameras with clusters lengths $d_1, \dots, d_n$, the optimal average detection time for a smart intruder satisfies the lower bound:*

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

where $L = \sum\limits_{i=1}^{n} d_i$.

We now characterize the performance of the equal-waiting trajectory.

Lemma V.6 ***(Equal-waiting trajectory performance)*** *For a set of n cameras with clusters lengths $d_1, \dots, d_n$, let $X(t)$ be the equal-waiting trajectory defined in Trajectory 1. Then*

$$\text{ADT}(X(t)) = \frac{\sum_{i=1}^{n} (d_{\max} + d_i) d_i}{2L},$$

where $d_{\max} = \max\{d_1, \dots, d_n\}$ and $L = \sum\limits_{i=1}^{n} d_i$.

A proof of Lemma V.6 follows from simple manipulation of equation (6) and it is omitted here.

We are now ready to prove the following result, which, combined with Lemma V.5 and Lemma V.6, concludes the proof of Theorem III.1.

Lemma V.7 ***(Equal-waiting trajectory approximation)*** *For a set of n cameras with clusters lengths $d_1, \dots, d_n$, let $X(t)$ be the equal-waiting trajectory defined in Trajectory 1. Then,*

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

where $d_{\min} = \min\{d_1, \dots, d_n\}$ and $d_{\max} = \max\{d_1, \dots, d_n\}$.

VI. Conclusion

In this work we address the problem of surveilling a one dimensional open path by means of a team of autonomous cameras against smart intruders. As performance function we define and adopt the average detection time criterion. We propose a camera trajectory, the equal-waiting trajectory, which is shown to be constant factor optimal with respect to the average detection time criterion. Additionally, we design a distributed algorithm to steer the cameras towards an optimal trajectory. The proposed algorithm is shown to be robust against cameras failures and motion uncertainties.

References

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

[2] D. B. Kingston, R. W. Beard, and R. S. Holt, “Decentralized perimeter surveillance using a team of UAVs,” *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1394–1404, 2008.

[3] S. Susca, S. Martínez, and F. Bullo, “Monitoring environmental boundaries with a robotic sensor network,” *IEEE Transactions on Control Systems Technology*, vol. 16, no. 2, pp. 288–296, 2008.

[4] Y. Elmaliach, A. Shiloni, and G. A. Kaminka, “A realistic model of frequency-based multi-robot polyline patrolling,” in *International Conference on Autonomous Agents*, Estoril, Portugal, May 2008, pp. 63–70.

[5] 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*, ser. Lecture Notes in Computer Science. Springer, 2003, pp. 155–170.

[6] Y. Chevaleyre, “Theoretical analysis of the multi-agent patrolling problem,” in *IEEE/WIC/ACM Int. Conf. on Intelligent Agent Technology*, Beijing, China, Sep. 2004, pp. 302–308.

[7] F. Pasqualetti, A. Franchi, and F. Bullo, “On cooperative patrolling: Optimal trajectories, complexity analysis and approximation algorithms,” *IEEE Transactions on Robotics*, Jan. 2011, to appear.

[8] M. Baseggio, A. Cenedese, P. Merlo, M. Pozzi, and L. Schenato, “Distributed perimeter patrolling and tracking for camera networks,” in *IEEE Conf. on Decision and Control*, Atlanta, GA, USA, Dec. 2010, pp. 2093–2098.

[9] R. Carli, A. Cenedese, and L. Schenato, “Distributed partitioning strategies for perimeter patrolling,” in *American Control Conference*, San Francisco, CA, USA, Jun. 2011, pp. 4026–4031.

[10] M. Spindler, “Distributed multi-camera synchronization for smart-intruder detection,” Master’s thesis, University of Stuttgart, Sep. 2011.