Simultaneous Boundary Partitioning and Cameras Synchronization for Optimal Video Surveillance ⋆

F. Zanella * **F. Pasqualetti** ** **R. Carli** * **F. Bullo** **

* *Department of Information Engineering, Università di Padova, Via Gradenigo 6/a, 35131 Padova, Italy*
{fzanella,carlirug}@dei.unipd.it
** *Center for Control, Dynamical Systems and Computation, University of California, Santa Barbara, USA*
{fabiopas,bullo}@engineering.ucsb.edu

Abstract: This paper proposes a real-time distributed algorithm for a team of smart cameras to self-organize and perform video surveillance of an open boundary. In particular, our algorithm simultaneously partitions the boundary among the cameras, and synchronizes the motion of the cameras to optimize the surveillance performance. We focus on the detection of *smart* intruders, who are aware of the cameras configuration at each time instant, and who schedule their motion to avoid detection for as long as possible. We consider both the *worst-case* and the *average* detection times of smart intruders. Our algorithm achieves minimum worst-case detection time, and, under some reasonable assumptions, constant-factor optimal average detection time.

1. INTRODUCTION

Networks of coordinated autonomous agents have recently been used for many tasks requiring repetitive execution, including the monitoring of oil spills (see Clark and Fierro (2007)), the detection of forest fires (see Kingston et al. (2008)), the tracking of border changes (see Susca et al. (2008)), and the patrolling (surveillance) of an environment (see Elmaliach et al. (2008)). The surveillance of an area of interest requires the agents (cameras in this case) to continuously and repeatedly sweep the environment, and the challenging problem consists in scheduling the cameras trajectories so as to optimize a performance criteria.

Problem description. In this work we consider a network of identical Pan-Tilt-Zoom (PTZ) cameras for video surveillance, subjected to physical constraints: each camera is located in a fixed position with limited field of view (f.o.v.) of the scene and limited motion capability. The cameras in fact use only the panning motion to span a one dimensional perimeter. It is also assumed that cameras are equipped with a low-level routine to detect intruders that intersect their f.o.v.. 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. Regarding our optimality criteria, we consider both the *worst-case detection time* as well as the *average detection time* of a smart intruder.

We focus on the development of algorithms for (i) optimally assigning a subpart of the environment to each camera (*partitioning problem*), and (ii) optimally scheduling the motion of the cameras for the detection of moving intruders (*cameras synchronization problem*). By solving these two problems, our algorithms allow the cameras to self-organize along a boundary to provide optimal surveillance against moving intruders.

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 *static* events. In Machado et al. (2003); Chevaleyre (2004) an empirical evaluation of certain patrolling heuristics is performed. In Elmaliach et al. (2008) and Kingston et al. (2008), an efficient and distributed solution to the (worst-case) perimeter patrolling problem for robots with zero communication range is proposed. In Pasqualetti et al. (2011) 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 moving intruders, as opposed to static ones, and we focus not only on the worst-case but also on the average detection times.

In the context of camera networks, the perimeter patrolling problem has recently been studied in Baseggio et al. (2010) and Carli et al. (2011). 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

⋆ This work was supported in part by "Fondazione Ing. Aldo Gini" scholarship and NSF grants IIS-0904501 and CNS-1035917, and by ARO grant W911NF-11-1-0092.

by showing that the strategies proposed in Baseggio et al. (2010) and Carli et al. (2011) generally may fail in terms of worst-case at detecting smart moving intruders, and by focusing on the average detection time of smart intruders. A more detailed description of the differences between our approach and those in Baseggio et al. (2010), Carli et al. (2011) is given in Sec. 2.

Paper contributions. Our solution builds upon the results presented in Spindler et al. (2012); Alberton et al. (2012). We propose a distributed algorithm to simultaneously partition a one dimensional open path among a set of fixed cameras, and synchronize the cameras motion to achieve optimal performance. Our partitioning algorithm is asymptotically optimal, in the sense that it asymptotically computes a partition of the path that minimizes the length of the longest cluster. Our algorithm only requires distributed computation. Regarding our communication model, we adopt a gossip communication model among neighboring cameras, and we require communication only when the f.o.v. of the two neighboring cameras overlaps. [1] Our synchronization algorithm inherits the optimality properties discussed in Spindler et al. (2012); Alberton et al. (2012). In particular, the cameras trajectory induced by our algorithm asymptotically achieves minimum worst-case detection time and bounded average detection time of smart intruders. In fact, we show that our trajectory achieves also minimum average detection time in some particular scenarios such as, for instance, in the absence of cameras range constraints. Finally, we validate our findings through simulations.

Paper organization. Section 2 contains the mathematical description of the problem under investigation and some preliminary results for our performance functions. In Section 3 we present our main contributions, that is a distributed algorithm to simultaneously partition an open boundary among cameras and to synchronize the motion of the cameras. Section 4 and Section 5 contain, respectively, our illustrative examples and our conclusions.

2. PROBLEM FORMULATION AND PRELIMINARY RESULTS

In this section we describe the surveillance problem we are considering and the mathematical tools for its analysis.

Problem setup Consider a set of N identical cameras $\{1, \dots, N\}$ installed along a one dimensional open path (boundary) $\mathcal{B} = [0\ L]$, $L > 0$ (see Fig. 1). Assume that (i) the f.o.v. of each camera is a point on $\mathcal{B}$, (ii) the motion of each f.o.v. is uniquely determined by the pan movement of the corresponding camera, and (iii) the speed $v_i : \mathbb{R}_{\geq 0} \to \mathbb{R}$ of the f.o.v. of the i-th camera satisfies $|v_i| \leq v_i^{\max}$, with $v_i^{\max} \in \mathbb{R}_{>0}$.

For the ease of notation we label the cameras in increasing order according to their position on $\mathcal{B}$. We define the *patrolling window* D_i as the subpart of $\mathcal{B}$ that camera i can monitor due to physical panning constraints, i.e., $D_i = [\underline{d}_i\ \overline{d}_i] \subseteq \mathcal{B}$ for some $\underline{d}_i < \overline{d}_i$. In order to guarantee

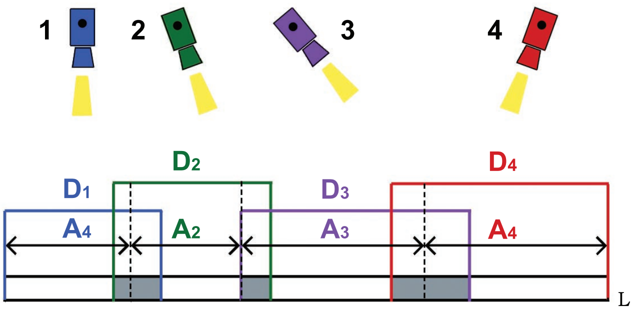

Fig. 1. Four cameras are installed along an open boundary $\mathcal{B} = [0\ L]$. For each camera i, the figure shows the patrolling window D_i, and the active boundary A_i. Notice that patrolling windows may overlap, and that the active boundaries form a partition of $\mathcal{B}$.

that the boundary $\mathcal{B}$ can be fully patrolled by the cameras, i.e., $\cup_{i=1}^N D_i = \mathcal{B}$, we let $\underline{d}_1 = 0$ and $\overline{d}_N = L$, and assume that the following *interlacing physical coverage constraints* hold:

$$\underline{d}_i \leq \underline{d}_{i+1} \leq \overline{d}_i \leq \overline{d}_{i+1}, \quad i = 1, \dots, N-1. \qquad (1)$$

Let $A_i(t) = [\ell_i(t)\ r_i(t)]$ denote the *active boundary* of the i-th camera at time t, i.e., the subpart of D_i assigned to the i-th camera at time t. In other words the following *patrolling window constraints* are satisfied:

$$A_i(t) \subseteq D_i, \quad \forall i \in \{1, \dots, N\}, \forall t \in \mathbb{R}_{\geq 0}. \qquad (2)$$

Additionally, let $|A_i(t)|$ be the distance on $\mathcal{B}$ between $\ell_i(t)$ and $r_i(t)$. Finally, define the *i-th sweeping time* as

$$\tau_i(t) := \frac{|A_i(t)|}{v_i^{\max}},$$

and

$$\tau^{\max}(t) := \max_{i \in \{1,\dots,N\}} \tau_i(t).$$

Cameras trajectories Let $\mathcal{F}(\mathbb{R}_{\geq 0}, \mathcal{B})$ be the Banach space of all continuous and bounded functions from $\mathbb{R}_{\geq 0}$ into $\mathcal{B}$ endowed with the supremum norm (see Kreyszig (1989)). The position of the i-th camera is a function $x_i \in \mathcal{F}(\mathbb{R}_{\geq 0}, \mathcal{B})$, $x_i : \mathbb{R}_{\geq 0} \to D_i$ that describes the position of the f.o.v. of the i-th camera as a function of the time variable t. In particular the i-th camera position is a *S-asymptotically T_i-periodic* function (see Nicola and Pierri (2009)), i.e.,

$$\exists\, T_i > 0 \ : \ \lim_{t\to\infty} x_i(t + T_i) - x_i(t) = 0.$$

In this case we say that T_i is the asymptotic period of x_i.

A *(cameras) trajectory* as an array $X = \{x_1, \dots, x_N\}$ of N functions in $\mathcal{F}(\mathbb{R}_{\geq 0}, \mathcal{B})$ describing the motion of the cameras f.o.v. on $\mathcal{B}$. In particular, we focus on S-asymptotically *jointly* T-periodic cameras trajectories X, so that

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

Finally, we say that a cameras trajectory is *synchronized* if, for each pair of neighboring cameras i and j ($|i-j| = 1$), there exists $\tau \in [0\ T]$ such that $\lim_{t\to\infty} x_i(t+\tau) - x_j(t+\tau) = 0$, where T denotes the asymptotic periodicity of X.

Model of intruder In this work we consider the problem of detecting an *intruder* on the boundary $\mathcal{B}$. We represent

[1] Although conservative, this assumption makes our algorithm implementable for almost any real wireless radio or similar proximity communication method.

the intruder as an arbitrarily fast point of $\mathcal{B}$, and we let $t_0 \in \mathbb{R}_{\geq 0}$ be the time at which the intruder appears on $\mathcal{B}$. Let the continuous map $p : \mathbb{R}_{\geq t_0} \mapsto \mathcal{B}$ describe the position of the intruder at a certain time $t \geq t_0$. We say that the 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, which have full knowledge of the cameras trajectory and choose their trajectory p to avoid detection as long as possible. More formally, given an initial time $t_0 \in \mathbb{R}_{\geq 0}$, an initial point $p_0 = p(0) \in \mathcal{B}$ and a cameras trajectory X, the trajectory of a smart intruder $p^*_{t_0,p_0}$ is such that

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

where $\Phi(t_0, p_0) \in \mathcal{F}(\mathbb{R}_{\geq 0}, \mathcal{B})$ is the set of continuous maps p and

$$t^*_d(p) = \min\{t \ : \ p(t) \in X(t), \, p \in \Phi(t_0, p_0)\}.$$

Notice that the trajectory $p^*_{t_0,p_0}$ is, in general, non unique.

Performance functions The detection performance of a T-periodic cameras trajectory X is measured according to two criteria, namely the *worst-case* detection time (WDT), and the *average* detection time (ADT). The two criteria are formally defined as

$$\mathrm{WDT}(X) := \max_{p_0,t_0} \, t^*_d(p) - t_0, \tag{3}$$

and

$$\mathrm{ADT}(X) := \frac{1}{TL}\int_0^T \int_{\mathcal{B}} (t^*_d(p^*_{t,\gamma}) - t) d\gamma dt. \tag{4}$$

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 $\mathcal{B}$ and the periodicity T.

A *static* variation (WDTs) of the WDT criterion in equation (3) is defined in Alberton et al. (2012). In particular,

$$\mathrm{WDTs}(X) := \max_{p_0,t_0}\{t - t_0 \ : \ p_0 \in X(t), \, t \geq t_0\}, \tag{5}$$

and it corresponds to the longest time for the cameras to detect a static event along $\mathcal{B}$. Due the fact that $\mathcal{B}$ is an (open) boundary, a (periodic) cameras trajectory with minimum WDTs can be computed by *optimally* partitioning $\mathcal{B}$ among the cameras, and by letting each camera sweep its assigned segment A_i at maximum speed (see Alberton et al. (2012) and (Czyzowicz et al., 2011, Conjecture 1)). [2] It follows that

$$\mathrm{WDTs}^* := \min_X \mathrm{WDTs}(X) = 2 \min_{A_1,\dots,A_N} \tau^{\max} =: 2\tau^*, \tag{6}$$

where $\cup_{i=1}^N A_i = \mathcal{B}$, and $A_i \cap A_j$ has zero measure on $\mathcal{B}$ for every distinct cameras i and j. Clearly,

$$\mathrm{WDT}^* := \min_X \mathrm{WDT}(X) \geq \mathrm{WDTs}^*. \tag{7}$$

In particular, it can be shown that $\mathrm{WDT}(X) = \mathrm{WDTs}^*$ for any WDTs^*-periodic and synchronized cameras trajectory (see Spindler et al. (2012)). Hence, any WDTs^*-periodic and synchronized cameras trajectory achieves minimum WDT of smart intruders. On the other hand, notice that for any non-synchronized cameras trajectory X, it holds $\mathrm{WDT}(X) = \infty$. Indeed, a smart intruder may avoid detection by moving between two neighboring cameras (see Spindler et al. (2012)).

We now derive a lower bound for ADT criterion.

Theorem 2.1. ***(Lower bound for ADT)*** Consider N cameras installed along a boundary $\mathcal{B}$ of length L. Let A_i be the active boundary of camera i. Assume $\cup_{i=1}^N A_i = \mathcal{B}$, and $A_i \cap A_j$ has zero measure on $\mathcal{B}$ for every distinct pair $i, j \in \{1, \dots, N\}$. Then,

$$\mathrm{ADT}^* = \min_X \mathrm{ADT}(X) \geq \frac{1}{L}\sum_{i=1}^N |A_i|\tau_i = \frac{1}{L}\sum_{i=1}^N \frac{|A_i|^2}{v_i^{\max}}, \tag{8}$$

where X is a $\tau^{\max}$-periodic function.

Theorem 2.1 generalizes the result in Spindler et al. (2012) to the case of cameras with different maximal speeds. In the interest of space, we omit a proof of Theorem 2.1, which closely the arguments used in Spindler et al. (2012)

Communication model In this work we consider a distributed scenario in which cameras i and 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 develop algorithms implementable with many low-cost communication devices.

Problem 1. ***(Distributed surveillance)*** Consider a set of N cameras with given patrolling windows on a boundary $\mathcal{B}$. Design a distributed algorithm for the cameras to converge to a synchronized trajectory X with $\mathrm{WDT}(X) = \mathrm{WDT}^*$ and $\mathrm{ADT}(X) < \infty$.

It should be observed that, although we make explicit use of the techniques developed by Spindler et al. (2012) and by Alberton et al. (2012), this paper extends the cited works in the following ways: with respect to Spindler et al. (2012), (i) the boundary partition and hence its dimension are not given a priori, and, instead, they are computed by the cameras via a distributed algorithm, (ii) we consider cameras with different maximal speeds, and (iii) with respect to Alberton et al. (2012), we consider the WDT and the ADT of moving and smart intruders, as opposed to only the WDTs of static intruders. As it will be clear in the sequel, these extensions are not trivial. Finally, we remark that in order to simultaneously partition the boundary $\mathcal{B}$ among the robots and synchronize the motion of the robots, the length $\tau^{\max}$ of the longest partition needs to dynamically tracked by every camera via distributed computation. This tracking problem is not posed in Spindler et al. (2012) because they start from the assumption that the A_i partitions are known a priori. Instead, we address this problem by considering S-asymptotically cameras trajectories, and the machinery introduced in the next section.

3. SYMMETRIC-GOSSIP PARTITIONING ALGORITHM WITH EQUAL-WAITING TRAJECTORIES GENERATION

In this section we introduce the *symmetric-gossip partitioning with equal-waiting trajectories generation algorithm* (denoted as SGPEWT hereafter). This algorithm allows the cameras to simultaneously partition the boundary and synchronize their motion via distributed computation. Loosely speaking, the i-th camera sweeps back and

[2] The computation of non-periodic cameras trajectories with optimal performance is, to the best of our knowledge, still an open problem, as well as a proof of the fact that the family of trajectories based on partitions of $\mathcal{B}$ contains optimal camera trajectories.

forth at the maximum speed its patrolling windows $A_i(t)$, and updates $A_i(t)$ upon communication with neighboring cameras. The update of $A_i(t)$ is performed so that, as t grows, 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. In an equal-waiting trajectory, for each camera, 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 $\tau^{\max}$ needs to be propagated among 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 $\tau^{\max}$.

We assume that:

(A1) each camera knows its initial active boundary $A_i(0) = [\ell_i(0), r_i(0)]$ and the patrolling speeds of its neighbors,

(A3) initial active boundaries satisfy the patrolling window constraint, i.e., $A_i(0) \subseteq D_i$, the *boundary conditions*, i.e. $l_1(0) = \underline{d}_1$ and $r_N(0) = \overline{d}_N$, and that the set $\{A_1(0), \ldots, A_N(0)\}$ is a partition of $\mathcal{B}$, and

(A4) the position $x_i(0)$ is randomly chosen within $A_i(0)$.

The SGPEWT is formally described as follows:

Processor states: For each $i \in \{1, \ldots, N\}$ and for all $t \in \mathbb{R}_{\geq 0}$, camera i maintains in memory the following quantities:

- the extremes $\ell_i(t)$ and $r_i(t)$ defining its active boundary $A_i(t)$;
- the estimated maximum sweeping time $\hat{\tau}_i^{\max}(t)$;
- the value of the variable $q_i(t)$.

Initialization: For $i \in \{1, \ldots, N\}$, set the estimated maximum sweeping time $\hat{\tau}_i^{\max}(0)$ and the variable $q_i(0)$ equal to $\tau_i(0)$ and i, respectively. Moreover at time $t = 0$ camera i starts moving at speed $v_i^{\max}$ toward the furthest extreme.

Transmission steps: Consider cameras i and $i+1$, and notice that they communicate as soon as their f.o.v.s meet at the common extreme, that is, as soon as $x_i(t) = r_i(t)$ and $x_{i+1}(t) = \ell_{i+1}(t)$ for a certain instant $t > 0$. This communication link is bidirectional, namely, camera i sends to camera $i+1$ the values $\ell_i(t)$, $r_i(t)$, $\hat{\tau}_i^{\max}(t)$, $q_i(t)$ while camera $i+1$ sends to camera i the values $\ell_{i+1}(t)$, $r_{i+1}(t)$, $\hat{\tau}_{i+1}^{\max}(t)$, and $q_{i+1}(t)$. Based on this information, cameras i and $i+1$ perform instantaneously the operations described in *Extremes' update* and *Maximum sweeping time estimation.*

Extremes' update: Cameras i and $i+1$ update the extremes r_i and ℓ_{i+1} according to the *symmetric gossip partitioning* algorithm introduced in Alberton et al. (2012). Specifically, the agents perform the following two actions in order. First, following the *neighbors' equal traveling time criterion*, they compute the point

$$m^* := \frac{\ell_i(t) v_{i+1}^{\max} + r_{i+1}(t) v_i^{\max}}{v_i^{\max} + v_{i+1}^{\max}}$$

Second they check if the intervals $[\ell_i(t), m^*]$ and $[m^*, r_{i+1}(t)]$ satisfy the physical constraints, and they update their extremes by setting

$$r_i(t^+) = \ell_{i+1}(t^+) = \begin{cases} m^* & \text{if} \quad m^* \in [\underline{d}_{i+1}, \overline{d}_i], \\ \overline{d}_i & \text{if} \quad m^* > \overline{d}_i, \\ \underline{d}_{i+1} & \text{if} \quad m^* < \underline{d}_{i+1}, \end{cases}$$

where the symbol t^+ denotes any time instant after t, i.e. $t^+ > t$. Instead, ℓ_i, r_{i+1} don't change, i.e. $\ell_i(t^+) = \ell_i(t)$, $r_{i+1}(t^+) = r_{i+1}(t)$. Finally, both cameras compute and store the values $\tau_i(t^+) = (r_i(t^+) - \ell_i(t^+))/v_i^{\max}$ and $\tau_{i+1}(t^+) = (r_{i+1}(t^+) - \ell_{i+1}(t^+))/v_{i+1}^{\max}$.

Maximum sweeping time estimation: Cameras i and $i+1$ update q_i, q_{i+1}, $\hat{\tau}_i^{\max}$ and $\hat{\tau}_{i+1}^{\max}$, based on the current values of these quantities. In particular we distinguish among four cases:

(M1) if $q_i(t) = i-1$ and $q_{i+1}(t) = i+2$, then

$$\hat{\tau}_i^{\max}(t^+) = \hat{\tau}_{i+1}^{\max}(t^+) = \max\left\{\hat{\tau}_i^{\max}(t), \hat{\tau}_{i+1}^{\max}(t)\right\}.$$

Moreover, if the above maximum is attained at $\hat{\tau}_i^{\max}(t)$ then

$$q_i(t^+) = i-1 \quad \text{and} \quad q_{i+1}(t^+) = i,$$

otherwise

$$q_i(t^+) = i+1 \quad \text{and} \quad q_{i+1}(t^+) = i+2.$$

(M2) if $q_i(t) = i-1$ and $q_{i+1}(t) \in \{i, i+1\}$, then

$$\hat{\tau}_i^{\max}(t^+) = \hat{\tau}_{i+1}^{\max}(t^+) = \max\left\{\hat{\tau}_i^{\max}(t), \tau_i(t^+), \tau_{i+1}(t^+)\right\},$$

Moreover, if the above maximum is attained at $\hat{\tau}_i^{\max}(t)$, then

$$q_i(t^+) = i-1 \quad \text{and} \quad q_{i+1}(t^+) = i,$$

otherwise

$$q_i(t^+) = q_{i+1}(t^+) = \arg\max_{i,i+1}\left\{\tau_i(t^+), \tau_{i+1}(t^+)\right\}.$$

(M3) the case $q_{i+1}(t) = i+2$ and $q_i(t) \in \{i, i+1\}$ work analogously to the previous case (M2);

(M4) if $q_i(t), q_{i+1}(t) \in \{i, i+1\}$, then

$$\hat{\tau}_i^{\max}(t^+) = \hat{\tau}_{i+1}^{\max}(t^+) = \max\left\{\tau_i(t^+), \tau_{i+1}(t^+)\right\},$$

and

$$q_i(t^+) = q_{i+1}(t^+) = \arg\max_{i,i+1}\left\{\tau_i(t^+), \tau_{i+1}(t^+)\right\}.$$

Equal-waiting synchronization: After performing the update steps, cameras i waits in its current position for an additional time $\omega_i = \hat{\tau}_i^{\max}(t^+) - \tau_i(t^+)$, which is the time that each camera is required to wait for X to converge to an equal-waiting trajectory (see Spindler et al. (2012)). [3] Then, camera i moves its f.o.v. toward the opposite extreme ℓ_i, and it stops at ℓ_i until it communicates with camera $i-1$. Analogously, camera $i+1$ moves its f.o.v. towards r_{i+1}, and it stops at r_{i+1} until it communicates with camera $i+2$.

We now characterize the convergence properties of the SGPEWT algorithm.

[3] Notice that ω_i is intrinsically greater or equal to zero. This is trivial to see for the cases (M2)-(M4) in which $\hat{\tau}_i^{\max}(t^+) \geq \tau_i(t^+)$. For (M1) it is sufficient to consider that, for the property of the partitioning, the sweeping times τ_i, $i \in \{1, \ldots, N\}$, are not increasing functions. Thus in (M1) it holds $\hat{\tau}_i^{\max}(t^+) \geq \hat{\tau}_i^{\max}(t) \geq \ldots \geq \hat{\tau}_i^{\max}(t^\star) \geq \tau_i(t^\star) \geq \ldots \geq \tau_i(t) \geq \tau_i(t^+)$ where $t^\star < t^+$ is the first instant in which the maximum sweeping time estimation occurs in (M2) or (M3) or (M4). Otherwise if the evolution of $\hat{\tau}_i^{\max}$ permanently takes place in (M1) we trivially have $\hat{\tau}_i^{\max}(t^+) \geq \hat{\tau}_i^{\max}(t) \geq \ldots \geq \hat{\tau}_i^{\max}(0) = \tau_i(0) \geq \ldots \geq \tau_i(t) \geq \tau_i(t^+)$.

Theorem 3.1. ***(Convergence of the SGPEWT)*** Consider a set of N cameras installed along a boundary $\mathcal{B}$ of length L. Let $\{v_1^{\max}, \dots, v_N^{\max}\}$ be the set of f.o.v. maximum speeds. Let the camera implement the SGPEWT algorithm. Then,

(i) the evolution of the partitions $t \mapsto \{A_i(t)\}_{i=1}^N$ generated by the SGPEWT algorithm satisfies the boundary conditions, the patrolling windows and interlacing physical constraints for all $t \in \mathbb{R}_{\geq 0}$, and the partition $\{A_i(t)\}$ satisfies

$$\tau^* = \lim_{t \mapsto \infty} \max_{i \in \{1,\dots,N\}} \frac{|A_i(t)|}{v_i^{\max}} = \min_{A_1,\dots,A_N} \max_{i \in \{1,\dots,N\}} \frac{|A_i|}{v_i^{\max}},$$

where $\{A_1, \dots, A_N\}$ is a feasible partition of $\mathcal{B}$,

(ii) the cameras trajectory X generated by the SGPEWT algorithm is S-asymptotically jointly $2\tau^*$-periodic, and converges to an equal-waiting trajectory.

Proof. Due to space constraints we just sketch the proof. First 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 ℓ_1 and r_N, respectively. In particular, it can be shown that any two neighboring cameras communicate within an interval of finite length. Statement (i) follows from (Alberton et al., 2012, Theorem IV.1).

Notice that, because of the persistency of communication among cameras, for any time t, the value $\tau^{\max}(t)$, which is decreasing in t, propagates in a finite time T_{prop} to every camera. 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 camera i is within 2ε of $2\tau^*$, for each $i \in \{1, \dots, N\}$. Statement (ii) follows by letting ε to zero. □

As a corollary of Thm. 3.1, the detection performance of the cameras trajectory generated by the SGPEWT algorithm are as follows.

Corollary 3.1. ***(Performance of the SGPEWT)*** Let X be the cameras trajectory generated by the SGPEWT algorithm and $\tau^\diamond := \lim_{t \mapsto \infty} \min_{i \in \{1,\dots,N\}} \tau_i(t)$. Then,

(i) $\displaystyle \lim_{t \to \infty} \frac{\text{WDT}(X)}{\text{WDT}^*} = 1,$

(ii) $\displaystyle \lim_{t \to \infty} \frac{\text{ADT}(X)}{\text{ADT}^*} \leq \frac{\tau^* + \tau^\diamond}{2\tau^\diamond}.$

Proof. We start from (i). We have already shown, recalling Equ. (6) and Spindler et al. (2012), that $\text{WDT}(\overline{X}) = \text{WDT}^* = \text{WDTs}^* = 2\tau^*$ for any $2\tau^*$-periodic and synchronized cameras trajectory $\overline{X}$. Thus (i) trivially follows by Thm. 3.1, since the cameras trajectory X is S-asymptotically jointly $2\tau^*$-periodic. Now consider (ii). Exploiting the technicalities to prove (Spindler et al., 2012, Thm. III.2) we have $\text{ADT}(\overline{X}) = \tau^*/2 + \sum_{i=1}^N v_i^{\max} \tau_i^2/(2L)$ where $\overline{X}$ is the $2\tau^*$-periodic equal-waiting trajectory to whom the cameras trajectory converge. Recalling (8), an upper bound of the ratio $\text{ADT}(\overline{X})/\text{ADT}^*$ is $(\tau^* \sum_{i=1}^N v_i^{\max} \tau_i / \sum_{i=1}^N v_i^{\max} \tau_i^2 + 1)/2$. Now the inequality

$$\frac{\tau^* \sum_{i=1}^N v_i^{\max} \tau_i}{2 \sum_{i=1}^N v_i^{\max} \tau_i^2} + \frac{1}{2} \leq \frac{\tau^* + \tau^\diamond}{2\tau^\diamond},$$

is trivially verified, in fact, after some manipulation, we obtain $\sum_{i=1}^N v_i^{\max} (\tau^\diamond - \tau_i) \tau_i \leq 0$, that holds since $\tau^\diamond \leq \tau_i$, $\forall i$. To conclude the proof note that $\lim_{t \to \infty} \text{ADT}(X)/\text{ADT}^* = \text{ADT}(\overline{X})/\text{ADT}^*$ follows from Thm. 3.1. □

4. ILLUSTRATIVE EXAMPLES

In this section we validate our algorithms thorough numerical simulations. We consider two scenarios with 5 cameras. All cameras start their trajectory from a random point $x_i(0) \sim \mathcal{U}\,[\ell_i(0), r_i(0)]$ [m]. The initial partitions $A_i(0)$ are generated as described in Section 3.

In the first scenario (Fig. 2) cameras have the same maximum speed $v_i^{\max} = 0.67$ [m/s], and they are not subject to patrolling windows constraints. In the case of same maximum speed, the patrolling windows D_i, the initial active boundaries $A_i(0)$, and the asymptotic partitions $A_i(\infty)$ are listed in Table 1. It can be seen from Fig. 2 that $\tau^{\max}$ is a decreasing function, and that the cameras converge to the optimally maximum sweeping time $\tau^* = 6.2023$ [s].

Table 1. Same cameras speed: values of D_i, $A_i(0)$ and $A_i(\infty)$, $i \in \{1, \dots, 5\}$.

	1	2	3	4	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]

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. In Table 2 we report the values of D_i, $A_i(0)$ and $A_i(\infty)$. As shown in Fig. 3, our algorithm is convergent also in this case ($\tau^* = 6.65$ [s]).

Table 2. Different cameras speeds: D_i, $A_i(0)$, $A_i(\infty)$ values for $i \in \{1, \dots, 5\}$.

	1	2	3	4	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]

Finally, notice from Fig. 2 and Fig. 3 that the cameras trajectories generated by our algorithm converge to synchronized trajectories.

5. CONCLUSIONS

This work studies the problem of simultaneously partitioning a boundary among a set of cameras and synchronizing the motion of the cameras. We measure the performance of a cameras trajectory in term of the worst-case detection time, and the average detection time of smart intruders. We show the convergence and characterize the performance of our algorithms.

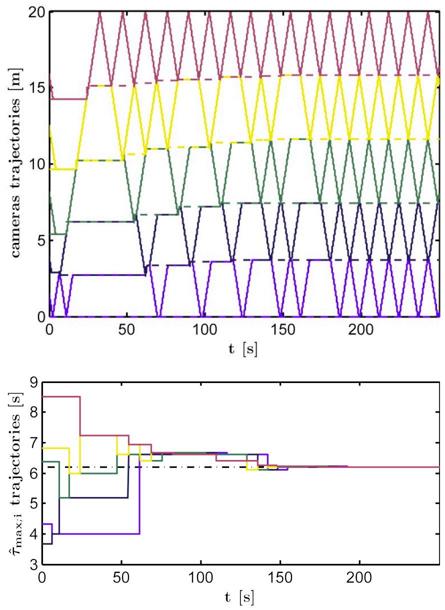

Fig. 2. Simulation of SGPEWT with $N = 5$ cameras running at the same maximum speed $v_i^{\max} = 0.67$ [m/s] with patrolling windows constraints. At the top are depicted the x_i trajectories of the cameras, starting form random positions $x_i(0)$. The dashed lines refer to the trajectories of the active boundaries. At the bottom are highlighted the dynamics of the maximum patrolling times $\hat{\tau}_i^{\max}$ that converge to the optimal $\tau^* = 6.2023$ [s], marked with a dash-dot line.

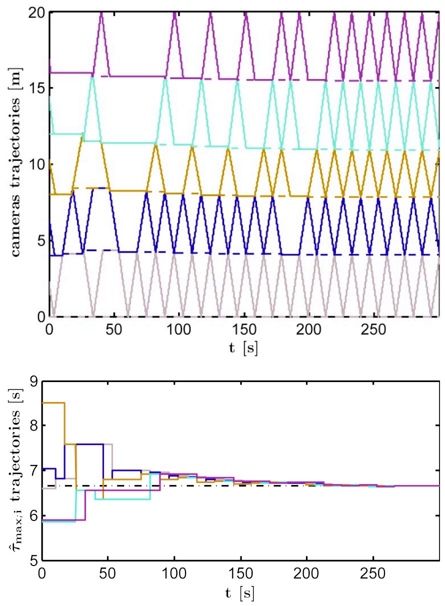

Fig. 3. Simulation of SGPEWT with $N = 5$ cameras running at different maximum speeds $v_i^{\max} \sim \mathcal{U}[0.45, 0.75]$ [m/s] with no patrolling windows constraints. At the top are depicted the x_i trajectories of the cameras, starting form random positions $x_i(0)$. The dashed lines refer to the trajectories of the active boundaries. At the bottom are highlighted the dynamics of $\hat{\tau}_i^{\max}$ that distributively converge to the optimal $\tau^* = 6.65$ [s], marked with a dash-dot line.

Several problems remain of interest. First, the design of finite time partitioning and synchronization algorithms. Second, the design of optimal non-periodic cameras trajectories. Third and finally, a proof of the fact that partitioned-based trajectories achieve optimal performance even when cameras have different maximal speeds.

REFERENCES

Alberton, R., Carli, R., Cenedese, A., and Schenato, L. (2012). Multi-agent perimeter patrolling subject to mobility constraints. In *American Control Conference*. Montréal, Canada. To appear.

Baseggio, M., Cenedese, A., Merlo, P., Pozzi, M., and Schenato, L. (2010). Distributed perimeter patrolling and tracking for camera networks. In *IEEE Conf. on Decision and Control*, 2093–2098. Atlanta, GA, USA.

Carli, R., Cenedese, A., and Schenato, L. (2011). Distributed partitioning strategies for perimeter patrolling. In *American Control Conference*, 4026–4031. San Francisco, CA, USA.

Chevaleyre, Y. (2004). Theoretical analysis of the multi-agent patrolling problem. In *IEEE/WIC/ACM Int. Conf. on Intelligent Agent Technology*, 302–308. Beijing, China.

Clark, J. and Fierro, R. (2007). Mobile robotic sensors for perimeter detection and tracking. *ISA Transactions*, 46(1), 3–13.

Czyzowicz, J., Gąsieniec, L., Kosowski, A., and Kranakis, E. (2011). Boundary patrolling by mobile agents with distinct maximal speeds. In *European Symposium on Algorithms*, 701–712. Saarbrücken, Germany.

Elmaliach, Y., Shiloni, A., and Kaminka, G.A. (2008). A realistic model of frequency-based multi-robot polyline patrolling. In *International Conference on Autonomous Agents*, 63–70. Estoril, Portugal.

Kingston, D.B., Beard, R.W., and Holt, R.S. (2008). Decentralized perimeter surveillance using a team of UAVs. *IEEE Transactions on Robotics*, 24(6), 1394–1404.

Kreyszig, E. (1989). *Introductory Functional Analysis with Applications*. Wiley.

Machado, A., Ramalho, G., Zucker, J.D., and Drogoul, A. (2003). Multi-agent patrolling: An empirical analysis of alternative architectures. In *Multi-Agent-Based Simulation II*, Lecture Notes in Computer Science, 155–170. Springer.

Nicola, S.H.J. and Pierri, M. (2009). A note on *s*-asymptotically periodic functions. *Nonlinear Analysis: Real World Applications*, 10(5), 2937–2938.

Pasqualetti, F., Franchi, A., and Bullo, F. (2011). On cooperative patrolling: Optimal trajectories, complexity analysis and approximation algorithms. *IEEE Transactions on Robotics*. To appear.

Spindler, M., Pasqualetti, F., and Bullo, F. (2012). Distributed multi-camera synchronization for smart-intruder detection. In *American Control Conference*. Montréal, Canada. To appear.

Susca, S., Martínez, S., and Bullo, F. (2008). Monitoring environmental boundaries with a robotic sensor network. *IEEE Transactions on Control Systems Technology*, 16(2), 288–296.