Continuous Graph Partitioning for Camera Network Surveillance

D. Borra * **F. Pasqualetti** ** **F. Bullo** **

* *Department of Mathematical Sciences, Polytechnic of Turin, Italy*
domenica.borra@polito.it
** *Center for Control, Dynamical Systems and Computation,*
University of California, Santa Barbara, USA
{fabiopas,bullo}@engineering.ucsb.edu

Abstract: This work focuses on the problem of designing surveillance trajectories for a network of autonomous cameras. As performance criterion we consider the worst-case detection time of static intruders. First, we represent the environment by means of a robotic roadmap. We show that optimal trajectories can be designed via a continuous graph partitioning problem. This minimization problem is convex and not differentiable. Second, we derive an auxiliary convex and differentiable minimization problem whose minimizer provides a solution to the original problem. Third and finally, we develop three distributed algorithms, for the cameras to partition the roadmap, and, consequently, synchronize along a trajectory with minimum worst-case detection time. Different communication protocols are used for the three algorithms.

1. INTRODUCTION

Remote surveillance of human activities for military and civil applications is receiving considerable attention form the research community. Public places such as banks, art galleries, homes, prisons, department stores, and parking lots, are now equipped with camera networks to detect important activities. From a scientific perspective, one of the main challenge consists of developing efficient algorithms for the cameras to autonomously and distributively complete tracking, surveillance, and recognition tasks.

In this work we focus on the problem of detecting intruders by means of a network of autonomous cameras. In particular, we consider camera networks installed at important locations in an environment. We assume cameras to be able to move their *field of view* (f.o.v.) to sweep the whole environment. We develop algorithms for the cameras to self-organize in order to detect *intruders* in the environment. We consider static intruders, which appear at arbitrary locations and at arbitrary times. As performance criteria we consider the *worst-case detection time*, that is the longest time needed for the cameras to detect intruders. Our setup is illustrated in Fig. 1.

Related work. In mobile robotics, the patrolling problem consists of repeatedly surveying a region with a team of autonomous agents in order to detect intruders or important events, e.g., see Alberton et al. [2012], Baseggio et al. [2010], Pasqualetti et al. [2011]. Although related, the patrolling problem and the problem considered in this paper significantly differ. First, cameras are fixed at predetermined locations, while robots are usually allowed to travel the whole environment. Second, the cameras f.o.v. must lie within the cameras visibility constraints, while robots do not have visibility contraints, since they can usually travel anywhere in the environment. Because of these reasons, algorithms developed for teams of robots are, in general, not applicable in our setup.

In the context of camera networks, the perimeter patrolling problem has recently been studied in Baseggio et al. [2010], Carli et al. [2011], Spindler et al. [2012]. 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 intruders. We improve the results along this direction by developing cameras trajectories for general environment topologies.

Paper contributions. The main contributions of this work are as follows. We consider the problem of detecting (static) intruders by means of a team of autonomous cameras. We adopt the worst-case detection time as performance function. We model the environment to be monitored as a roadmap, in which vertices represent important locations in the environment, and edges denote the possibility for a camera installed at an end-vertex to survey the physical edge. We show that, for acyclic roadmaps, cameras trajectories with minimum worst-case detection time can be designed by solving a continuous roadmap partitioning problem. We also conjecture that this is the case for cyclic roadmaps. Second, we show that the above continuous roadmap partitioning problem is convex but non-differentiable. Hence, we derive an auxiliary (strictly) convex and differentiable minimization problem. We show that for general roadmaps the unique minimizer of the smooth problem is also a minimizer for the original problem. Third, we develop three distributed algorithms for the cameras to partition the roadmap, and hence to self-organize along optimal trajectories. These three algorithms rely upon different communication assumptions.

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

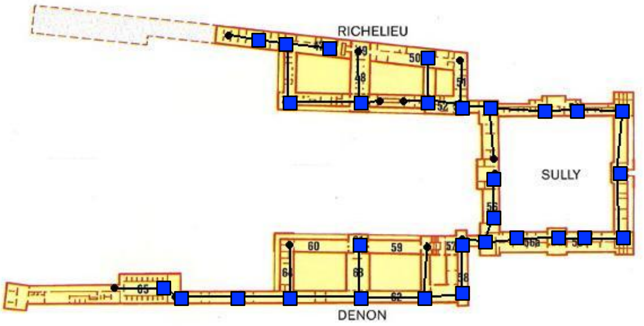

Fig. 1. As example environment we choose the Louvre museum, a map of which is here reported. We select 38 locations in the map (black circles and blue squares), and we install cameras at 29 of these locations (blue squares). The environment is represented by a roadmap, whose vertices correspond to the 38 locations, and whose edges are reported in the picture as solid black lines.

In particular: our first algorithm assumes a synchronous mode of operation of the cameras; our second algorithm assumes an asymmetric broadcast communication model and extends the class of block-coordinate descent algorithms to the constrained case; and our third algorithm only requires gossip communication. We show convergence of all three algorithms, and we show their convergence properties via a numerical experiment.

Paper organization. A description of the problem and some remarks are stated in Section 2. Section 3 deals with the design of an optimal camera trajectory and an equivalent minimization problem is presented. In Section 4 three distributed partition algorithms are presented with convergence results, and numerical results are shown in Section 5 to validate the effectiveness of the proposed algorithms. Finally Section 6 contains the conclusions of this work and some perspectives related to future research.

2. PROBLEM SETUP AND PRELIMINARY DEFINITIONS

Consider a set of $n \in \mathbb{N}$ identical cameras $\{c_1, \dots, c_n\}$ for the task of monitoring a two dimensional environment. We represent the environment with a *roadmap* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ (see LaValle [2006]), where the vertex set $\mathcal{V}$ corresponds to a set of locations in the environment, and the edge set reflects the visibility properties of the cameras. In particular, we assume the cameras to be installed at the vertices $\mathcal{V}_c \subseteq \mathcal{V}$, with $|\mathcal{V}_c| = n$, and we let camera c_i be installed at vertex $v_i \in \mathcal{V}_c$. Let the undirected edge $\{v_i, v_j\} \in \mathcal{E}$ if there exists a point in the segment $[v_i,\ v_j]$ visible by the camera installed at v_i or v_j. Let $\mathcal{E}_c = (\mathcal{V}_c \times \mathcal{V}_c) \cap \mathcal{E}$, and let the distance between vertices v_i and v_j be the weight of the edge $\{v_i, v_j\}$, defined by $L_{ij} = \mathrm{dist}(v_i, v_j)$. Thus the camera network is the undirected weighted graph $\mathcal{G}_c = (\mathcal{V}_c, \mathcal{E}_c, \boldsymbol{L})$, where $\boldsymbol{L}$ denotes the vector of L_{ij}. If $\{v_i, v_j\} \in \mathcal{E}_c$, then c_i and c_j are *neighboring cameras*. Let $\mathcal{N}_i^{\text{in}}$ denote the set of vertices in $\mathcal{V}_c$ adjacent to v_i. Finally, let $\mathcal{N}_i^{\text{out}}$ denote the set of vertices in $\mathcal{V} \setminus \mathcal{V}_c$ adjacent to v_i.

Notice that the construction of the roadmap $\mathcal{G}$ depends upon the cameras visibility constraints. We assume that:

(A1) at each point in time, the i-th f.o.v. is a point along the segment $[v_i,\ v_j]$ for some $v_j \in \mathcal{N}_i^{\text{in}}$, and

(A2) the speed of the i-th f.o.v. belongs to the set $\{0, 1\}$.[1]

Let $x_i(t)$ denote the position at time t of the i-th f.o.v.. We assume that each camera has a limited visibility range along each adjacent edge. In particular, at each time t,

(A3) if $x_i(t) \in [v_i\ \ v_j]$, then $\ell_{ij} L_{ij} \le \mathrm{dist}(x_i(t), v_i) \le u_{ij} L_{ij}$, for some constants $\ell_{ij}, u_{ij} \in [0, 1]$.

Let $\boldsymbol{\ell} \in [0,1]^{\mathcal{E}_c}$ and $\boldsymbol{u} \in [0,1]^{\mathcal{E}_c}$ be the vectors of ℓ_{ij} and u_{ij}, respectively. Finally we assume that the whole roadmap $\mathcal{G}$ is jointly visible by the cameras. Our setup is in Fig. 1.

A *cameras trajectory* is a set of n continuous functions $X = \{x_1, \dots, x_n\}$, where x_i describes the position of the i-th f.o.v. along the roadmap $\mathcal{G}$. We focus on periodic cameras trajectories, for which there exists $T \in \mathbb{R}_{>0}$ such that $X(t+T) = X(t)$ for $t \in \mathbb{R}_{>0}$. Let $T_i \in \mathbb{R}_{>0}$ be the period of camera c_i. Then, $T = \mathrm{l.c.m}(T_i)$ is the least common multiple of the periods T_i. Define the image of the i-th camera as the set of points covered by the i-th f.o.v. in any period of length T_i, i.e., $\mathrm{Im}(x_i) = \cup_{t \in [0, T_i]} x_i(t)$. Finally, let the image of the cameras trajectory be $\mathrm{Im}(X) = \cup_{i=1}^{n} \mathrm{Im}(x_i)$, and the *cameras image set* be $\mathcal{I}^X = \{\mathrm{Im}(x_1), \dots, \mathrm{Im}(x_n)\}$.

We consider the problem of designing cameras trajectories to detect *static intruders* along the roadmap. In particular, we allow an intruder to appear at any time $t_{\text{arr}} \in \mathbb{R}_{>0}$, and any point p along the roadmap $\mathcal{G}$. We assume that the intruder is detected at time t_{det} if $p \in X(t_{\text{det}})$. We evaluate a trajectory X according to its *worst-case detection time*, that is the longest time for the detection of an intruder. Specifically, we define the worst-case detection time of a cameras trajectory as

$$\mathrm{WDT}(X) := \max_{t_0, p_0}\ t^*(t_0, p_0, X),$$

where

$$t^*(t_0, p_0, X) = \min\{t - t_0\ :\ t \ge t_0,\ p_0 \in X(t)\}.$$

In this work we design cameras trajectories with minimum worst-case detection time for the case of acyclic roadmaps.

3. CAMERAS TRAJECTORIES WITH MINIMUM WORST-CASE DETECTION TIME

In this section we design a camera trajectory with minimum worst-case detection time of static intruders. In particular, we assume the roadmap $\mathcal{G}$ and the cameras locations $\mathcal{V}_c$ to be given, and we design the cameras trajectory $X = \{x_1, \dots, x_n\}$ that achieves minimum worst-case detection time of static intruders. This section is organized as follows. First, we show that the trajectory design problem can be cast as a convex, non-differentiable, minimization problem (Section 3.1). Then we derive an auxiliary convex and differentiable minimization problem for computing optimal trajectories (Section 3.2). In Section 4 we show that the latter formulation is amenable to distributed implementation.

[1] If $\dot{x}_i$ denotes the linear speed of the i-th f.o.v., the related angular speed $\dot{\theta}_i = \dot{x}_i / (a_i \sec^2(\theta_i))$ depends on the distance a_i of the camera from the f.o.v. path, and the Pan-angle θ_i.

3.1 Trajectory design and graph partitioning

We define a relative order among the cameras as follows. For the neighboring cameras c_i and c_j, with $i < j$, let the time t be such that the i-th f.o.v. and the j-th f.o.v. lie on the edge $\{v_i, v_j\}$. Then we say that $c_i \leq c_j$ if $\text{dist}(x_i(t), v_i) \leq \text{dist}(x_j(t), v_i)$. If $x_i(t), x_j(t)$ lie on different edges at time t, our convention is $x_i(t) \leq x_j(t)$. A camera trajectory is *order-invariant* if the relative order of the cameras along the edges is preserved over time.

Theorem 3.1. ***(Order-invariant cameras trajectory)*** For any cameras trajectory X along the roadmap $\mathcal{G}$, there exists an order-invariant cameras trajectory $\bar{X}$ with $\text{WDT}(X) = \text{WDT}(\bar{X})$.

Proof. For any couple of adjacent cameras $\{v_i, v_j\} \in \mathcal{E}_c$ if there exists $t \geq 0$ such that $x_i(t) = x_j(t)$ define $t_{ij}^0 = \min\{t \geq 0 \ : \ x_i(t) = x_j(t)\}$ and $t_{ij}^n = \min\{t > t_{ij}^{n-1} \ : \ x_i(t) = x_j(t)\}, n \in \mathbb{N}$. An order-invariant trajectory can be derived from X permuting the cameras labels as follows

$$\bar{x}_i(t) = x_j(t), \bar{x}_j(t) = x_i(t), \text{ if } t_{ij}^{2k} \leq t \leq t_{ij}^{2k+1}$$

with $k = 0, 1, \ldots$ and define $\bar{x}_i(t) = x_i(t), \bar{x}_j(t) = x_j(t)$ otherwise. Each point along the roadmap receives the same visits at the same time instants, thus $\text{WDT}(\bar{X}) = \text{WDT}(X)$. □

Notice that in general the images of neighboring cameras may overlap. A camera trajectory is called *non-overlapping* if for any $\{v_i, v_j\} \in \mathcal{E}_c$, it holds $\text{Int}(\text{Im}(x_i)) \cap \text{Int}(\text{Im}(x_j)) = \emptyset$, where $\text{Int}(\cdot)$ denotes the interior of a set. Let $|\text{Im}(x_i)|$ denote the length on $\mathcal{G}$ of the image $\text{Im}(x_i)$.

Theorem 3.2. ***(Non-overlapping cameras trajectory)*** For any cameras trajectory X on an acyclic roadmap, there exists an order-invariant non-overlapping cameras trajectory $\bar{X}$ with $\text{WDT}(X) \geq \text{WDT}(\bar{X})$.

Proof. We can assume X to be order invariant, if not find an equivalent order invariant trajectory as in Theorem 3.1. Define a camera trajectory $\bar{X} = \{\bar{x}_1, \ldots, \bar{x}_n\}$ in such a way that $\bar{x}_i(t) = x_i(t)$ for any $i \in \mathcal{V}_c$ with $deg(i) = 1$, hence $\text{Im}(\bar{x}_i) = \text{Im}(x_i)$. Since the worst-case detection time for the point v_i is $t^*(t_0, v_i, X) = T_i$, then $\max_{p \in \text{Im}(x_i)} t^*(t_0, p, X) = \max_{p \in \text{Im}(\bar{x}_i)} t^*(t_0, p, \bar{X}) = T_i$. Up to relabeling the nodes, suppose $\bar{x}_1, \ldots, \bar{x}_r$ have already been assigned, where r denotes the number of nodes with unitary degree. Define the subsets of the roadmap $\text{Im}(\bar{x}_i) = cl(\text{Im}(x_i) \backslash \left(\cup_{j=1}^{i-1} \text{Im}(\bar{x}_j)\right)), i = r + 1, \ldots, n$ where $cl(\cdot)$ denotes the closure of a set, and we choose the assignment order in such a way that node i can be selected if it has at least one assigned neighbor. Consider $p \in \partial(\text{Im}(\bar{x}_i)) \backslash \left(\cup_{j=1}^{i-1} \text{Im}(\bar{x}_j)\right), i = r + 1, \ldots, n$ where $\partial(\cdot)$ denotes the boundary points of a set. Referring to trajectory X, for sufficiently small $\epsilon > 0$, note that $p + \epsilon$ along edge $\{v_i, v_j\}$ can only be visited by node i. Thus $\max_{p \in \text{Im}(x_i)} t^*(t_0, p, X) \geq 2 |\text{Im}(\bar{x}_i)| - 2\epsilon$. Moreover in limit $\epsilon \to 0$, it holds $\max_{p \in \text{Im}(x_i)} t^*(t_0, p, X) \geq 2 |\text{Im}(\bar{x}_i)|$ since the trajectories are continuous. Given the partition of the roadmap $\{\text{Im}(\bar{x}_i)\}_{i=1}^n$, define the overlapping trajectory $\bar{X}$ assuming that the cameras sweep periodically at maximum speed their image. Therefore $\max_{p \in \text{Im}(\bar{x}_i)} t^*(t_0, p, \bar{X}) = 2 |\text{Im}(\bar{x}_i)|$, in other words $\text{WDT}(\bar{X}) = 2 \max_i |\text{Im}(\bar{x}_i)| \leq \text{WDT}(X)$. □

Remark 1. ***(Order invariance and non-overlapping properties for cyclic graphs)*** While Theorem 3.1 applies to general roadmaps, Theorem 3.2 only applies to acyclic roadmaps. It is our conjecture that analogous results can be stated for cyclic graphs. We leave this important aspect as the subject of future research.

Following Theorem 3.1 and 3.2 we focus on periodic order-invariant and non-overlapping cameras trajectories. Consider the set $\mathcal{I} = \{\mathcal{I}_1, \ldots, \mathcal{I}_n\}$, where $\mathcal{I}_i \subseteq \mathcal{G}$ is associated with camera c_i. The set $\mathcal{I}$ is a *continuous partition* of the roadmap $\mathcal{G}$ if $\mathcal{G} = \cup_{i=1}^n \mathcal{I}_i$ and $\text{Int}(\mathcal{I}_i) \cap \text{Int}(\mathcal{I}_j) = \emptyset$ for all $i \neq j$. Given a partition $\mathcal{I}$ of $\mathcal{G}$, let the dimension of $\mathcal{I}$ equal $\max_i |\mathcal{I}_i|$. For each $\{v_i, v_j\} \in \mathcal{E}_c, i < j$, define the parameters $\alpha_{ij} \in [\ell_{ij}, u_{ij}]$, where ℓ_{ij} and u_{ij} are the previously defined cameras contraints. Let $\boldsymbol{\alpha} \in [0, 1]^{\mathcal{E}_c}$ be the vector of the α_{ij} with the constraint $\boldsymbol{\ell} \leq \boldsymbol{\alpha} \leq \boldsymbol{u}$[2]. Notice that the vector $\boldsymbol{\alpha}$ defines a continuous partition of $\mathcal{G}$, precisely

$$\mathcal{I}_i = \mathcal{I}_i^{\text{in}+} \cup \mathcal{I}_i^{\text{in}-} \cup \mathcal{I}_i^{\text{out}},$$

where

$$\begin{aligned}
\mathcal{I}_i^{\text{in}+} &= \cup_{j \in \mathcal{N}_i^{\text{in}}, i<j} [v_i, \ v_i + \alpha_{ij}(v_j - v_i)], \\
\mathcal{I}_i^{\text{in}-} &= \cup_{j \in \mathcal{N}_i^{\text{in}}, i>j} [v_i, \ v_i + (1 - \alpha_{ij})(v_j - v_i)], \\
\mathcal{I}_i^{\text{out}} &= \cup_{j \in \mathcal{N}_i^{\text{out}}} [v_i, \ v_j].
\end{aligned}$$

Remark 2. ***(Finite worst-case detection time)*** For any periodic cameras trajectory X, it holds $\text{WDT}(X) < \infty$ *only if* the roadmap $\mathcal{G}$ is included in the image set $\mathcal{I}^X$.

Given a continuous partition $\mathcal{I}^{\text{df}}$, in Algorithm 2 we design a trajectory X^{df}, called *DF-Trajectory* (Depth First-Trajectory). In Theorem 3.3 we show that the DF-Trajectory achieves minimum worst case detection time among the cameras trajectories with image set $\mathcal{I}^{\text{df}}$ (see Diestel [2000] for the concept of depth-first search).

Algorithm 1 DF-Trajectory

```
(Step A) Compute the partition $\mathcal{I}$ from $\boldsymbol{\alpha} \in [0,1]^{\mathcal{E}_c}$

(Step B) Design DF-Trajectory
for $i \in \mathcal{V}_c$ do
    $t_0 = 0, x_i(0) = v_i$;
    for $j \in \mathcal{N}_i^{\text{in}}$ do
        while $t_0 \leq t \leq t_0 + 2\alpha_{ij}$ do
            $x_i(t) = (t - t_0)v_j + (\alpha_{ij} - (t - t_0))v_i$;
                $t \leq t_0 + \alpha_{ij}$;
            $x_i(t) = -x_i(t - \alpha_{ij}), t > t_0 + \alpha_{ij}$;
            $t_0 = t_0 + 2\alpha_{ij}$;
        end while
    end for
end for
```

From Theorem 3.2, we next show that an optimal worst-case detection time is given by an optimal partition.

Theorem 3.3. ***(Worst-case detection and partitions)*** Consider an acyclic roadmap $\mathcal{G}$, and let X be a cameras trajectory with $\mathcal{I}^X$ a continuous partition of $\mathcal{G}$. Then,

$$\text{WDT}(X) \geq 2 \max_{i \in \{1, \ldots, n\}} |\mathcal{I}_i^X|.$$

[2] $\boldsymbol{a} \leq \boldsymbol{b}$, where $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^n$ means $a_i \leq b_i$ for all $i = 1, \ldots, n$.

Additionally, the DF-Trajectory X^{df} with images set $\mathcal{I}^{\mathrm{df}} = \mathcal{I}^X$ in Algorithm 2 satisfies

$$\mathrm{WDT}(X^{\mathrm{df}}) = 2 \max_{i\in\{1,\dots,n\}} |\mathcal{I}_i{}^{\mathrm{df}}|.$$

Proof. The first statement can be achieved reasoning along the lines of the proof of Theorem 3.2. Moreover it is straightforward to show that $WDT(X^{\mathrm{df}})$ reaches the lower bound. □

We conclude this section by noticing that, as a consequence of Theorem 3.3, a cameras trajectory with minimum worst-case detection time can be designed by addressing the problem of finding a continuous partition of $\mathcal{G}$ with smallest dimension.

3.2 Continuous min-max roadmap partitioning

In this section we introduce the graph partitioning problem we address and the necessary notation. Define the *load* of the i-th camera, with $i \in \mathcal{V}_c$, by

$$W_i = \sum_{j\in\mathcal{N}_i^{\mathrm{in}}, i<j} \alpha_{ij} L_{ij} + \sum_{j\in\mathcal{N}_i^{\mathrm{in}}, i>j} (1-\alpha_{ij}) L_{ij} + \sum_{j\in\mathcal{N}_i^{\mathrm{out}}} L_{ij}, \tag{1}$$

where the parameters α_{ij} are defined as in Section 3.1, and L_{ij} denotes the length of the edge $\{v_i, v_j\}$. Let W be the vector of the loads W_i.

For an acyclic roadmap $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with corresponding camera network $\mathcal{G}_c = (\mathcal{V}_c, \mathcal{E}_c, \boldsymbol{L})$, let $A \in \mathbb{R}^{\mathcal{V}_c \times \mathcal{E}_c}$ be the *weighted incidence matrix* of the subgraph of $\mathcal{G}$ with vertices $\mathcal{V}_c$ and edges $\mathcal{E}_c$. Specifically,

$$A_{i,e} = \begin{cases} L_{ij} = -A_{j,e} \text{ if } e = \{v_i, v_j\} \in \mathcal{E}_c,\ i<j, \\ 0, \text{ otherwise.} \end{cases} \tag{2}$$

Define the vector $\boldsymbol{b} \in \mathbb{R}^{\mathcal{V}_c}$ as

$$\boldsymbol{b}_i = \sum_{j\in\mathcal{N}_i^{\mathrm{in}}, i>j} L_{ij} + \sum_{j\in\mathcal{N}_i^{\mathrm{out}}} L_{ij}. \tag{3}$$

Notice that the loads vector W can be written as

$$W = A\boldsymbol{\alpha} + \boldsymbol{b}.$$

Recall from Theorem 3.3 that a minimum worst-case detection time cameras trajectory can be designed from a continuous partition of $\mathcal{G}$ with minimum dimension (see also Algorithm 2). Thus we address the following convex minimization problem (*continuous min-max partitioning*)

$$\min_{\boldsymbol{\ell}\le\boldsymbol{\alpha}\le\boldsymbol{u}} \|A\boldsymbol{\alpha} + \boldsymbol{b}\|_\infty. \tag{4}$$

Notice that, although the the minimization problem (4) is not differentiable, it can be solved efficiently by means of an LP solver (see Boyd and Vandenberghe [2004]). On the other hand, since the infinity norm in (4) is not differentiable, distributed solvers are difficult to find. We next derive an equivalent differentiable minimization problem, which is instead amenable to distributed implementation.

Theorem 3.4. ***(Auxiliary minimization problem)*** Consider the minimization problem

$$\min_{\boldsymbol{\ell}\le\boldsymbol{\alpha}\le\boldsymbol{u}} \|A\boldsymbol{\alpha} + \boldsymbol{b}\|_2^2, \tag{5}$$

where A and $\boldsymbol{b}$ are as in equation (2) and (3), respectively, and $\boldsymbol{\ell}, \boldsymbol{u}$ denote the cameras constraints. Let $\boldsymbol{\alpha}^*$ be the unique solution to the minimization problem (5). Then,

$$\|A\boldsymbol{\alpha}^* + \boldsymbol{b}\|_\infty = \min_{\boldsymbol{\ell}\le\boldsymbol{\alpha}\le\boldsymbol{u}} \|A\boldsymbol{\alpha} + \boldsymbol{b}\|_\infty.$$

Proof. In the following we denote $\alpha_{ji} = 1 - \alpha_{ij}$ whenever $\{v_i, v_j\} \in \mathcal{E}, i > j$. Firstly we prove that (5) has a unique minimizer. Consider $f : [0,1]^{\mathcal{E}_c} \to \mathbb{R}_{\ge 0}$, defined by

$$f(\boldsymbol{\alpha}) = \sum_{j=1}^n W_j^2 = \sum_{j=1}^n \left(\sum_{i\in\mathcal{N}_j^{\mathrm{in}}} \alpha_{ij} L_{ij} \right)^2.$$

This function is defined on a convex compact and nonempty feasibility set and it is strictly convex, therefore its minimum is unique. Let $\boldsymbol{\alpha}^*$ denote this minimizer and $W(\boldsymbol{\alpha}^*) = \{W_1^*, \dots, W_n^*\}$ the corresponding loads set.
Secondly, we characterize the minimizer as follows. Consider any $j \in \mathcal{V}_c$ such that $W_j^* = \|\boldsymbol{W}^*\|_\infty$ and partition the neighbor set as follows $\mathcal{N}_j^{\mathrm{in}} = \mathcal{N}_j^1 \cup \mathcal{N}_j^2$, where $W_i^* < W_j^*, i \in \mathcal{N}_j^1$ and $W_i^* = W_j^* = \|\boldsymbol{W}^*\|_\infty, i \in \mathcal{N}_j^2$. For any $i \in \mathcal{N}_j^1$ we claim that $\alpha_{ji}^* = \ell_{ji}$, namely the parameter is saturated on the lower bound starting from j. By absurd, if there exists $i \in \mathcal{N}_j^1$ such that $\alpha_{ji}^* > \ell_{ji}$, then consider the loads set $\hat{W}$ defined as W^* for all the parameters α except for $\hat{\alpha}_{ji} = 1 - \hat{\alpha}_{ij} = \alpha_{ji}^* - \epsilon$ such that $\hat{\alpha}_{ij} < \hat{\alpha}_{ji}$, or equivalently

$$\alpha_{ij}^* + \epsilon < \alpha_{ji}^* - \epsilon. \tag{6}$$

Define the constants $C_1 = \sum_{k\in\mathcal{N}_i^{\mathrm{in}}, k\ne j} \alpha_{ik}^* L_{ik}$ and $C_2 = \sum_{k\in\mathcal{N}_j^{\mathrm{in}}, k\ne i} \alpha_{jk}^* L_{ik}$, the following contradiction is achieved

$$\begin{aligned} f(W^*) - f(\hat{W}) &= \big(C_1 + \alpha_{ij}^* L_{ij}\big)^2 - (C_1 + \hat{\alpha}_{ij} L_{ij})^2 \\ &+ \big(C_2 + \alpha_{ji}^* L_{ji}\big)^2 - (C_2 + \hat{\alpha}_{ji} L_{ji})^2 \\ &= 2\epsilon L_{ij}(C_2 - C_1 + L_{ij}(\alpha_{ji}^* - \alpha_{ij}^* - \epsilon)) \\ &> 2\epsilon L_{ij}^2(2(\alpha_{ji}^* - \alpha_{ij}^*) - \epsilon) > 6\epsilon^2 L_{ij}^2 > 0 \end{aligned}$$

using Eq. (6) and $C_2 - C_1 > L_{ij}(\alpha_{ji}^* - \alpha_{ij}^*)$, derived from $W_i^* < W_j^*$.
Finally, we prove that $\|W^*\|_\infty = \min_{\boldsymbol{\alpha}} \|A\boldsymbol{\alpha} + \boldsymbol{b}\|_\infty$. Suppose not, there exists $\hat{W} \ne W^*$ such that $\|\hat{W}\|_\infty = \min_{\boldsymbol{\alpha}} A\boldsymbol{\alpha} + \boldsymbol{b}$. Then suppose $W_j^* = \|W^*\|_\infty$ and $\mathcal{N}_j^2 = \emptyset$, hence for any $i \in \mathcal{N}_j^{\mathrm{in}}, \alpha_{ji} = \ell_{ji}$. If $\mathcal{N}_j^2 \ne \emptyset$, we can reason along the same lines, considering the subgraph $\mathcal{G}'$ given by all the connected nodes j whose component equals the infinity norm, and for the previous statement all their adjacent nodes in $\mathcal{G}\backslash\mathcal{G}'$ are saturated. Therefore $\hat{W}_j \le \|\hat{W}\|_\infty < \|W^*\|_\infty = W_j^* = \sum_{i\in\mathcal{N}_j^{\mathrm{in}}} \ell_{ji}$, but $\hat{W}_j \ge \sum_{i\in\mathcal{N}_j^{\mathrm{in}}} \ell_{ji}$ since $\boldsymbol{l} \le \boldsymbol{\alpha} \le \boldsymbol{u}$. This is a contradiction, thus the thesis holds, i.e., W^* achieves the minimum infinity norm. □

Remark 3. ***(Unconstrained solution)*** Notice that the vector W satisfies $\|W\|_1 = \sum_{\{v_i,v_j\}\in\mathcal{E}} L_{ij}$, for every parameters vector $\boldsymbol{\alpha}$. Then it can be verified that

$$\arg\min_{\boldsymbol{\alpha}} \|A\boldsymbol{\alpha} + \boldsymbol{b}\|_2^2 = \arg\min_{\boldsymbol{\alpha}} \|A\boldsymbol{\alpha} + \boldsymbol{b}\|_\infty = \boldsymbol{v}$$

where $\boldsymbol{v} = \dfrac{\sum_{\{v_i,v_j\}\in\mathcal{E}} L_{ij}}{n} \mathbf{1}$ and $\mathbf{1}$ is the vector of all ones.

4. DISTRIBUTED TRAJECTORY DESIGN

In this section we design three distributed, and thus scalable, algorithms for the continuous min-max partitioning problem. Once an optimal partition has been computed, it is immediate to design an optimal trajectory as in Algorithm 2.

The algorithms we present rely upon different cameras communication assumptions. We assume each camera to be equipped with a wireless sensor device. In all our algorithms, cameras perform the following operations: (i) receive parameters from (some) neighboring cameras, (ii) update the parameters corresponding to (some) adjacent edges, and (iii) transmit the new values to (some) neighboring cameras. These operations are detailed in the next sections. For convenience, let $S_i^t = \left\{\alpha_{ij} : j \in \mathcal{N}_i^{\text{in}}\right\}$ be the state of camera c_i at iteration $t \in \mathbb{N}$.

4.1 Synchronous Gradient Descent Partitioning algorithm

The distributed algorithm presented in this section assumes a synchronous mode of operation of the cameras, and it is inspired by the classical gradient projection method (see Bertsekas and Tsitsiklis [1997]). In particular, every camera performs operations at uniform time instants. This algorithm is detailed in Algorithm 2.

Algorithm 2 Synchronous Gradient Descent Partitioning

```
(Camera state) $S_i = \left\{\alpha_{ij} : j \in \mathcal{N}_i^{\text{in}}\right\}, i \in V_c$

(Initialize) $\alpha_{ij}^0 = \ell_{ij}$, for any $\{v_i, v_j\} \in \mathcal{E}_c$

(Transmit and compute at iteration $t \in \mathbb{N}$)
for $i \in \mathcal{V}_c$ do
    Camera $c_i$ receives $S_j^t$ from every $j \in \mathcal{N}_i^{\text{in}}$;
    Set $\alpha_{ij}^{t+1} \leftarrow \alpha_{ij}^t - \varepsilon L_{ij}(W_i^t - W_j^t)$;
    if $\alpha_{ij}^{t+1} < \ell_{ij}$ then $\alpha_{ij}^{t+1} = \ell_{ij}$;
    else if $\alpha_{ij}^{t+1} > u_{ij}$ then $\alpha_{ij}^{t+1} = u_{ij}$;
    end if
    Camera $c_i$ transmits $S_i^{t+1}$ to $j \in \mathcal{N}_i^{\text{in}}$;
end for
```

Theorem 4.1. ***(Synchronous Partitioning)*** The *Synchronous Partitioning* algorithm described in Algorithm 2 asymptotically converges to $\boldsymbol{\alpha}_{\text{SGD}}^* = \lim_{t\to\infty} \boldsymbol{\alpha}^t$. Moreover,

$$\min_{\boldsymbol{\ell}\leq\boldsymbol{\alpha}\leq\boldsymbol{u}} \|A\boldsymbol{\alpha} + \boldsymbol{b}\|_\infty^2 = \|A\boldsymbol{\alpha}_{\text{SGD}} + \boldsymbol{b}\|_\infty^2.$$

Proof. Note that the update step can be expressed in vector form as

$$\boldsymbol{\alpha}^{t+1} \leftarrow \boldsymbol{\alpha}^t - \varepsilon(A^T A\boldsymbol{\alpha}^t + A^T\boldsymbol{b}),$$

and that $A^T A\boldsymbol{\alpha} + A^T\boldsymbol{b}$ is the gradient of the quadratic function $\|A\boldsymbol{\alpha} + \boldsymbol{b}\|_\infty^2/2$. Therefore the Synchronous Partitioning algorithm coincides with the gradient projection method (Bertsekas and Tsitsiklis [1997]). To conclude the proof note that $\|A\boldsymbol{\alpha} + \boldsymbol{b}\|_2^2/2$ has a Lipschitz-continuous gradient with Lipschitz constant $K \in \mathbb{R}_{>0}$, thus for a sufficiently small step size the convergence of the algorithm to the optimal solution of Equation (5) is achieved, see e.g. [Bertsekas and Tsitsiklis, 1997, Proposition 3.4]. Finally apply Theorem 3.4 to achieve the optimality with respect to the infinity norm. □

Remark 4. (**Stepsize for Algorithm** 2) The stepsize ε in Algorithm 2 needs to satisfy $0 < \varepsilon < 2/K$, where K is Lipschitz constant of the function $A^T(A\boldsymbol{\alpha} + \boldsymbol{b})$. It is easy to show that $K \leq 2d_{\max}L_{\max}^2$, where $d_{\max}$ denotes the maximum degree of the nodes in $\mathcal{V}_c$ and $L_{\max}$ the maximum edge length. Thus it suffices to choose $\varepsilon < \frac{1}{d_{\max}L_{\max}^2}$. See Bertsekas and Tsitsiklis [1997] for further details. □

4.2 Asymmetric Broadcast Partitioning algorithm

The distributed algorithm presented in this section assumes an asymmetric broadcast communication protocol. In particular, at any time iteration only one camera updates its state, using only local information from its neighboring cameras. We assume that each camera initiate an update step according to a Poisson process which selects a camera after an exponentially distributed waiting times. From the Markovian property of Poisson processes (Ross [1983]), the sequence of selected cameras is a sequence of independent indexes uniformly distributed in $\mathcal{V}_c$. In order to guarantee the convergence of the algorithm, we require that every camera is selected at least once within a finite time horizon as described in Theorem 4.2. This algorithm is detailed in Algorithm 3.

Algorithm 3 Asymmetric Broadcast Partitioning

```
(Camera state) $S_i = \left\{\alpha_{ij} : j \in \mathcal{N}_i^{\text{in}}\right\}, i \in V_c$

(Initialize) $\alpha_{ij}^0 = \ell_{ij}$, for any $\{v_i, v_j\} \in \mathcal{E}_c$

(Transmit and computate at iteration $t \in \mathbb{N}$)
Camera $c_i$ is randomly selected;
Camera $c_i$ receives $S_j^t$ from every $j \in \mathcal{N}_i^{\text{in}}$;
for $j \in \mathcal{N}_i^{\text{in}}$ do
    $\alpha_{ij}^{t+1} \leftarrow \alpha_{ij}^t - \varepsilon L_{ij}(W_i^t - W_j^t)$;
    if $\alpha_{ij}^{t+1} < \ell_{ij}$ then $\alpha_{ij}^{t+1} = \ell_{ij}$;
    else if $\alpha_{ij}^{t+1} > u_{ij}$ then $\alpha_{ij}^{t+1} = u_{ij}$;
    end if
end for
Camera $c_i$ transmits $S_i^{t+1}$ to $j \in \mathcal{N}_i^{\text{in}}$;
```

Theorem 4.2. ***(Asymmetric Broadcast Partitioning)*** Consider the Asynchronous Broadcast Partitioning algorithm detailed in Algorithm 3. Assume the existence of a finite duration $B \in \mathbb{R}_{>0}$ such that, for any $t \in \mathbb{R}_{\geq 0}$, every camera in $\mathcal{V}_c$ is selected at least once in the time interval $[t, t+B]$ (*partial asynchronism assumption*). The Asynchronous Broadcast Partitioning algorithm asymptotically converges to $\boldsymbol{\alpha}_{\text{AB}}^* = \lim_{t\to\infty} \boldsymbol{\alpha}^t$. Moreover,

$$\min_{\boldsymbol{\ell}\leq\boldsymbol{\alpha}\leq\boldsymbol{u}} \|A\boldsymbol{\alpha} + \boldsymbol{b}\|_\infty^2 = \|A\boldsymbol{\alpha}_{\text{AB}}^* + \boldsymbol{b}\|_\infty^2.$$

Proof. Note that the constraints set $\{\boldsymbol{\alpha} \in [0,1]^{\mathcal{E}_c} : \boldsymbol{\ell} \leq \boldsymbol{\alpha} \leq \boldsymbol{u}\} = \prod_e [\ell_e, u_e]$ is a box, the gradient is Lipschitz continuous and the partial asynchronism assumption holds, thus [Bertsekas and Tsitsiklis, 1997, Proposition 5.3] applies. There exists $\varepsilon_0 \in \mathbb{R}_{>0}$ such that for any $\varepsilon < \varepsilon_0$, this partially asynchronous gradient projection algorithm converges to the optimal $\boldsymbol{\alpha}$. □

Remark 5. (**Stepsize for Algorithm 3**) The stepsize ε in Algorithm 3 needs to satisfy $0 < \varepsilon < \varepsilon_0$, with $\varepsilon_0 = \frac{1}{K(1+B+B|\mathcal{E}_c|)}$. The bound ε_0 depends on the Lipschitz constant K defined in Remark 4, the time horizon B and the number of edges $|\mathcal{E}_c|$. See Bertsekas and Tsitsiklis [1997] for further details. □

4.3 Symmetric gossip partitioning algorithm

The distributed algorithm presented in this section assumes a symmetric gossip-type communication protocol. In particular, at each time iteration only one component of a camera state is updated, and only two adjacent cameras are involved in the computation. As in the Asymmetric Broadcast Partitioning algorithm, neighboring cameras are selected according to a Poisson process. This algorithm is detailed in Algorithm 4.

Algorithm 4 Symmetric Gossip Partitioning

```
(Camera state) $S_i = \{\alpha_{ij} : j \in \mathcal{N}_i^{\text{in}}\}, i \in V_c$
(Initialize) $\alpha_{ij}^0 = \ell_{ij}$, for any $\{v_i, v_j\} \in \mathcal{E}_c$
(Transmit and compute at iteration $t \in \mathbb{N}$)
Neighboring cameras $c_i$ and $c_j$ are randomly selected;
Camera $c_i$ (resp. $c_j$) receives $S_j^t$ (resp. $S_i^t$);
$W^* = (W_i^t + W_j^t)/2$;
$\alpha_{ij}^{t+1} = \big(W^* - \sum_{k \in \mathcal{N}_i^{\text{in}}, k \neq j} \alpha_{ik}^t L_{ik}\big)/L_{ij}$;
if $\alpha_{ij}^{t+1} < \ell_{ij}$ then $\alpha_{ij}^{t+1} = \ell_{ij}$;
else if $\alpha_{ij}^{t+1} > u_{ij}$ then $\alpha_{ij}^{t+1} = u_{ij}$;
end if
```

Theorem 4.3. ***(Symmetric Gossip Partitioning)*** Consider the Symmetric Gossip Partitioning algorithm detailed in Algorithm 4. Assume the existence of a finite duration $B \in \mathbb{R}_{>0}$ such that, for any $t \in \mathbb{R}_{\geq 0}$, every camera $\mathcal{V}_c$ is selected at least once in any time interval $[t, t+B]$ (partial asynchronism assumption). The Symmetric Gossip Partitioning algorithm converges to $\boldsymbol{\alpha}_{\text{SG}}^* = \lim_{t\to\infty} \boldsymbol{\alpha}^t$. Moreover,

$$\min_{\boldsymbol{\ell} \leq \boldsymbol{\alpha} \leq \boldsymbol{u}} \|A\boldsymbol{\alpha} + \boldsymbol{b}\|_\infty^2 = \|A\boldsymbol{\alpha}_{\text{SG}}^* + \boldsymbol{b}\|_\infty^2.$$

Proof. Define the energy function $U : \prod_e [\ell_e, u_e] \to \mathbb{R}$ $U(\boldsymbol{\alpha}) = \sum_{\{v_i,v_j\}\in\mathcal{E}} (W_i - W_j)^2$ as energy storage function. The convergence of $\boldsymbol{\alpha}^t$ can be retrieved by a reasoning along the lines of [Alberton et al., 2012, Theorem IV.1], and by applying Theorem 3.4. □

5. NUMERICAL EXAMPLES

In this section we validate our distributed algorithms through a numerical study. The roadmap considered for the simulations is in Fig. 1. Notice that $|\mathcal{E}| = 38$, and $n = |\mathcal{E}_c| = 29$. The stepsizes for Algorithm 3 and Algorithm 4 are chosen as in Remark 4 and Remark 5, respectively. The results of our simulation study are in Fig. 2. Notice that all algorithm converges to the desired value.

6. CONCLUSIONS

In this paper we design trajectories for a network of autonomous cameras to detect static intruders. As performance function we adopt the worst-case detection time, that is the longest time needed for an intruder to be detected. For acyclic roadmaps, we show that optimal trajectories can be designed by solving a min-max continuous graph partitioning problem. Then, we show that the min-max continuous graph partitioning problem is equivalent to a convex and differentiable minimization problem. Finally, we propose three different distributed algorithms for the min-max continuous graph partitioning problem.

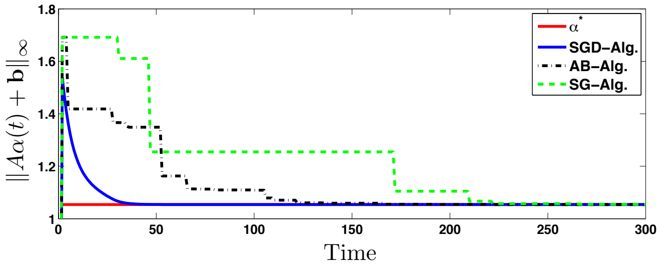

Fig. 2. This figure shows the convergence of the Synchronous Gradient Partitioning (blue solid line), the Asymmetric Broadcast Partitioning (black dash-dot line), and the Symmetric Gossip Partitioning algorithms (green dashed line) towards a solution of the continuous min-max partitioning problem. For the simulation we use the configuration in Fig. 1, with $\boldsymbol{\ell} = \mathbf{0}, \boldsymbol{u} = \mathbf{1}$ and $\alpha^0 = \mathbf{0}$.

Important aspects requiring further investigation include, (i) the design of optimal trajectories for cyclic roadmaps, (ii) more general intruder models, and (iii) the design of non-deterministic cameras trajectories.

REFERENCES

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

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*, pages 2093–2098, Atlanta, GA, USA, December 2010.

D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, 1997. ISBN 1886529019.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. ISBN 0521833787.

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

R. Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. Springer, 2 edition, 2000.

S. M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006. ISBN 0521862051. Available at `http://planning.cs.uiuc.edu`.

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

S. M. Ross. *Stochastic Processes*. Wiley Series in Probability and Mathematical Statistics, 1983.

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