

# A Distributed Approach to Interactive Surveillance Using a UAV Mounted Camera Network

Nicolò Bargellesi  
dept. of Automation Engineering  
Università degli Studi di Padova  
1206181

Luca Facin  
dept. of Automation Engineering  
Università degli Studi di Padova  
1197485

Lorenzo Marchini  
dept. of Automation Engineering  
Università degli Studi di Padova  
1207385

**Abstract**—The aim of this paper is to discuss a possible solution to the Interactive Surveillance problem using multiple UAV mounted cameras, from a top down perspective, in known environments. The coordinated patrolling problem is approached in a distributed fashion using a Bayesian-based Greedy algorithm with State Exchange. The results of this strategy are compared in restricted environments with the optimal paths obtained through linear programming. Furthermore, the overall performance is evaluated on large scale environments, where the optimal problem becomes infeasible. Kalman's theory is employed to implement a smart target tracking algorithm with camera zoom control optimized for target containment and information loss minimization. Several simulations are performed tracking a target with different trajectory models and varying the sampling frequency of the filter and the accuracy of the detection. Finally, the robustness of such an algorithm to measurement errors and camera failures is put to the test.

## I. INTRODUCTION

Multi-Camera surveillance systems have become a modern trend in many fields both for research and for the industry. These systems are being increasingly used to enhance safety and security in public places such as parks, airports, and banks, as well as in restricted areas like government and military facilities. Furthermore, the possibility to mount cameras on Unmanned Aerial Vehicles (UAVs) extends the capabilities of these surveillance systems to a whole new level. Indeed, the employment of flying cameras allow for a substantial reconfigurability of the network and enable for new perspectives on the environment and wider coverage. Therefore, the use of such camera extends the use scenarios even to harsh environments for topographic monitoring [1], search and rescue operations [2], and disaster management activities [3]. Moreover, UAV based surveillance systems are becoming a trend even in the consumer industry for home security [4]. However, unlike many of the networks that are based on CCTV architectures, where human operators are in charge of the camera control and data interpretation, for these innovative systems the coordination of multiple flying objects and the analysis of the acquired data require some form of automatization. This task can be achieved either by centralized processing or by adopting a distributed approach [5] for better scalability.

This work addresses UAV mounted multi-camera systems for interactive surveillance purposes in known environments.

The surveillance task is herein divided into the following three subtasks:

- *Patrolling* - the activity of moving around or through the environment at regular intervals for security purposes;
- *Event Detection* - the activity of recognizing an object, a person, or a situation from the camera data stream, usually via Computer Vision (CV) techniques;
- *Tracking* - the activity of following the event through the environment for information gathering.

To approach the coordinated patrolling problem, a Greedy algorithm with State Exchange is implemented for camera coordination, starting from the Bayesian-based mathematical formalism presented in [6]. Such a distributed strategy is tested in restricted environments achieving comparable results with the ones from the optimal paths obtained through linear programming techniques. Furthermore, a simulation in a real-like use case scenario is performed on a large scale environment, where optimal control becomes infeasible.

The event detection problem is heavily dependent on the application context. In particular, the CV techniques employed are usually designed ad hoc for the detection of specific shapes and color schemes. Therefore, the in-depth analysis of the problem is beyond the scope of this work and the events are considered to be automatically detected.

To achieve the smart target tracking, an algorithm that combines Kalman filter theory with an optimized camera zoom control is designed. In particular, the optimal zoom is achieved managing the trade-off between target containment and information loss minimization, relying on the covariance matrix of Kalman filter prediction errors. Several simulations changing the target trajectory models and varying the sampling frequency of the filter are performed. The obtained results are in line with the ones expected. In particular, running the filter at frequencies higher than 5Hz yields almost perfect tracking in many scenarios. Moreover, the robustness of such algorithm to measurement errors and camera failures is evaluated.

In the next section, a brief overview of the state-of-the-art is conducted and the contributions of the paper are described. Afterward, in Section III, the interactive surveillance problem is formalized highlighting all the assumptions made and the desired working specifications. The following two sections deeply describe and present the corresponding simulation results of our approach to the coordinated patrolling of the

environment (Section IV) and the smart tracking of moving objects (Section V). Finally, the paper ends with conclusions and open issues for future research.

## II. RELATED WORK

Existing works related to multi-camera surveillance systems mainly focus on object detection and tracking using Computer Vision techniques [7], activity analysis [8], person re-identification [9], security and privacy protection [10], self-reconfiguration [11] [12], and coordination and control [13]. However, the types of cameras usually included in these works are Pan-Tilt-Zoom (PTZ), omni-directional, and smart cameras, with fixed location in the environment. In contrast, in this paper the multi-camera coordination and control is extended to moving cameras, referring to several contributions on different multi-agent systems such as UAV swarms and Multi-Robot Systems (MRS).

As concerns the patrolling problem, the research focused on two different types of patrolling [14]: adversarial patrol [15] and regular patrol [16]. In adversarial patrol, the team of agents assumes the existence of an intruder and the aim is to coordinate the research to capture the opponent as quickly as possible. On the other hand, regular patrolling refers to keeping the agents visiting the desired points of interest continuously so that the time lag between two visits can be minimized. Henceforth, the focus of this work is mainly on the latter, and in particular on the regular area patrol, i.e. the activity of going through an area, differently from going around an area (regular perimeter patrol).

In several theoretical contributions, it has been shown that the optimal patrolling can be obtained if all agents follow the same Traveling Salesman Path or Hamilton Cycle, equally distributed in time and space (see e.g. [17]). However, these cycles are not trivial to compute in sparse topologies (the case of most real-world environments), NP-hard to compute, or not even existing. Furthermore, in this study scenario, where the agents can sense nearby nodes without visiting them, the solutions presented in the literature become suboptimal. Therefore, in this paper, a linear programming technique addressing the optimal coordinated patrolling of the environment using such specific agents is presented.

To face the tracking problem we rely on the Kalman filter theory. These filters are well known for their efficiency in solving vision problems such as object modeling, target tracking, surveillance, and object recognition [18]. In particular, a wide variety of Kalman based algorithms for target tracking have been proposed in the literature, e.g. Adaptive Kalman Filters (AKF), Extended Kalman Filters (EKF), and Unscented Kalman Filters (UKF) [19]. The typical purpose of these filters is to estimate the position, the velocity, and sometimes the acceleration of the target, relying on noisy and uncertain visual measurements of its locations. Hence, through these estimates, the agent is able to follow the target, minimizing the risk of losing it once it has been detected. Moreover, in the specific application of surveillance systems, it becomes of primary importance also

to obtain the maximum resolution of the target, in order to facilitate classification tasks (e.g. people identification) [20]. However, meeting the desired requirements for resolution and tracking robustness is not a trivial task. This is due to the fact that increasing tracking accuracy usually leads to decreasing classification efficiency and vice versa. Therefore, in this paper, an innovative approach to target tracking is proposed combining the traditional Kalman filter with an optimized camera zoom control. In this way, the trade-off between the target containment and information gathered can be balanced.

To summarize, the contributions of this work to the state-of-the-art are as follows:

- formalization of a flexible and adaptable approach to the interactive surveillance problem in the context of UAV mounted camera networks;
- definition of the linear programming problem for optimal patrolling paths-finding, in the context of nearby nodes aware agents;
- implementation of a distributed and scalable Bayesian-based Greedy algorithm addressing the coordinated patrolling problem;
- definition of a smart tracking system with optimized zoom control through the use of Kalman filters;
- effectiveness evaluation of the presented solutions through MATLAB simulations.

## III. PROBLEM FORMALIZATION

In this paper, the *Interactive Surveillance* is intended as the mechanism by which multiple UAV mounted cameras collect information on the desired environment and share such information with the other agents through communication in order to perform the desired surveillance task collaboratively. In particular, as outlined in Section I, the cameras are asked to patrol a known environment looking for possible events that may occur. If an event is detected, then the cameras are required to track it continuously at high resolution, for maximal information gathering, to predict its possible trajectory.

In the context of this work, it is crucial to formalize all the different aspects of the: surveilled environment, camera model, and communication network.

### A. Surveilled Environment

The focus of this case study is on the surveillance of fixed and known planar environments. Moreover, since in most of the applications of a surveillance system the scene consists of polygonal shapes or can be approximated by a collection of them, we approximate the whole region by a polygon. Furthermore, we allow the environment to have cavities that represent potential No-Fly-Zones (NFZ).

As stated in [21], “*visual coverage is an important quantifiable property of camera networks, describing from a pragmatic standpoint what the system can see*”. Therefore, to facilitate the definition of the visible area from each camera, the environment is represented as an occupancy grid. This decision is also supported by the fact that the evaluation

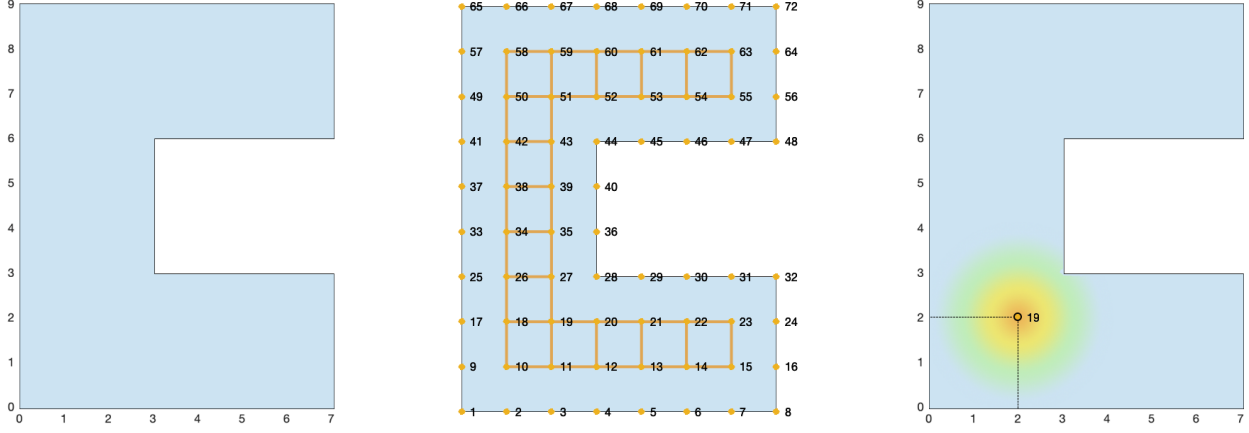


Fig. 1. **Left:** Polygonal Environment. **Middle:** Cellular representation (note that environment borders must be covered but are considered to be NFZ). **Right:** Heat update after detection in vertex  $v_{19} = (2, 2)$ .

of the optimal camera trajectories is usually infeasible in the continuous domain. Several different cell shapes can be adopted (e.g. square, hexagonal, etc.); however, such choice appears to be heavily dependent on the implementation [22]. For the implementation of the coordinated patrolling, a squared cell representation with a selectable resolution is adopted and formalized as an undirected navigation graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  with  $v_i = (x_{v_i}, y_{v_i}) \in \mathcal{V}$  vertices and  $e_{i,j} \in \mathcal{E}$  edges. Each vertex represents a specific location that must be covered regularly and each edge corresponds to an allowed camera movement, having a weight  $|e_{i,j}|$  equal to the environment resolution.

The cellular representation also simplifies the introduction of the heat map  $\mathcal{H}$  associated with the scene, representing prior knowledge on detected event position or additional information about the weak spots of the environment. In particular, each vertex  $v_i$  of the graph  $\mathcal{G}$  is initialized with heat  $\mathcal{H}(v_i) = 1$ . Once a specific condition occurs in  $\mathbf{x} = (x, y)$  (i.e. event detection, tracking lost, etc.) the heat map is updated using a 2D Gaussian kernel as follows:

$$\mathcal{H}(v_i) = \mathcal{H}(v_i) + \alpha e^{-\gamma \|\mathbf{x} - v_i\|^2}. \quad (1)$$

Moreover,  $\mathcal{H}(v_i)$  is reset to its initial state whenever any camera passes on the  $v_i$ .

An illustration of the cellularization process and of a possible heat update is given in Fig. 1

### B. Camera Model

Before proceeding with the detailed description of the camera model, we introduce five important camera parameters for the current work:

- *Absolute Position X*: camera position  $X = (x_c, y_c, z_c)$  in space;
- *Grid Position V*: vertex of the environment in which the UAV is located;
- *Angle of View (AoV)*: the angular extent of a given scene that is imaged by a camera;

- *Field of View (FoV)*: part of the world visible through the camera in a particular position and orientation in space. Here the FoV is expressed as the projection along the view cone associated with the AoV;
- *Spatial Resolution*: the ratio between the total number of pixels excited by the projection of a real-world object and the object size. A higher spatial resolution produces more detailed and sharpened images.

As outlined in Section II, the analysis in this paper focuses on multiple UAV mounted cameras. For manageability, we model them as Pan-Zoom (PZ) cameras with a top-down view. In particular, we consider panning as a UAV coplanar movement and zoom as a change in height. Neglecting tilt capabilities we ensure the camera FoV to be always regular

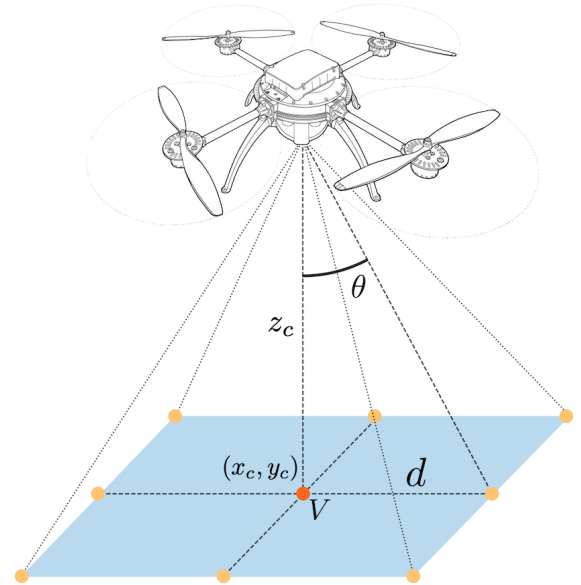


Fig. 2. Vision model of an UAV mounted camera. Yellow dots correspond to the 8-neighbors and the red one to the grid position  $V$ .

(i.e. square or round). Moreover, we assume a half-Angle of View  $\theta = \text{AoV}/2 = 40^\circ$ . This is a valid assumption since many of the commercial UAVs equipped with cameras share similar AoVs [23] [24].

Cameras are assumed to be aware of the *a priori* map of the environment, i.e. the navigation graph  $\mathcal{G}$  described in Section III-A. Since a square cell representation is adopted for the environment, the FoV of the cameras is considered to be a square of side  $2d$ , where  $d = z_c \tan \theta$ . Moreover, we assume that the UAVs can only move from a minimum height  $z_{min}$  to a maximum height  $z_{Max}$  defined as

$$z_{Max} = \frac{\text{environment resolution}}{\tan \theta}, \quad (2)$$

that allows an 8-neighbors coverage of the cellularized environment, i.e. information can be gathered from nine nodes. In what follows,  $A_v(v_i)$  denotes the set of nine vertices viewed by a camera at  $z_c = z_{Max}$  on the vertex  $v_i$ .

A schematic representation of the camera vision model is presented in Fig. 2.

Since event detection is strongly dependent on the application context, an event is assumed to be detected automatically and instantaneously once it lays inside the FoV of a camera. However, possible errors in the event detection are taken into consideration by modeling the detection as a boolean random variable  $E_d(\cdot)$  with  $\mathbb{P}(E_d(\cdot)=0) = e_d$ , where  $E_d(\cdot) = 1$  corresponds to the correct detection and  $E_d(\cdot) = 0$  corresponds to a lost detection.

Considering that the Computer Vision algorithms return an estimate of the event location in the image domain, the measurement error is modeled as white Gaussian noise with standard deviation  $\sigma_{r\%}$  expressed as a percentage of the FoV. This assumption is motivated by the fact that such error is dependent on the spatial resolution of the camera and therefore it is related to the camera height. In particular, it is reasonable to consider a lower error when the camera is close to the object, i.e. the spatial resolution is high, and a higher error when the camera is far from the ground.

### C. Communication Design

All the operations involved in the surveillance tasks can only be carried out if there is proper coordination among the camera nodes. Therefore, the accurate design of the communication system is required. The communication design is herein tackled by analyzing the following four aspects:

- *Network Topology*: the element arrangement (links, nodes, etc.) of the communication network, usually formalized as a graph;
- *Broadcast Technology*: the transmission media (wired or wireless) adopted for data sharing;
- *Information Exchanged*: the actual transmitted data;
- *Communication Protocol*: the set of rules that regulates the information exchange in the network.

The choice of the Network Topology is strictly related to the specific application. Many topologies are present in literature (see [25]) and they differ in: time dependence (static or

dynamic), dimensionality, and representation (line, tree, ring, mesh, bus, etc.). In the context of this work we consider  $N$  cameras  $\mathcal{C} = \{c_1, \dots, c_N\}$  distributed in the environment and interconnected by a wireless *all-to-all* (fully-connected) communication graph, meaning that local information is known by all agents across the network.

The information exchanged through the camera network is dependent on the task performed by each camera, i.e. patrolling or tracking. In particular, patrolling cameras should be aware of their number  $n_p$ , grid positions  $V_t^i$ , movement intentions  $V_{t+1}^i$ , and if necessary of new event detections or tracking losses location. On the other hand, the tracking camera is asked to broadcast its state and position, if an event is detected or the tracking is lost. Therefore, two different transmission packages  $\mathcal{P}$  are defined:  $\mathcal{P}_0 = \{0, V_t, V_{t+1}\}$  for patrolling data and  $\mathcal{P}_1 = \{1, V_{Ed}\}$  for tracking data, where the first bit represents the camera task.

The distributed approach adopted to face the surveillance problem allows for the formalization of the camera communication protocol as an *Asynchronous Protocol*. This means that the agents transmit data intermittently, avoiding the problems associated with the use of an external clock signal or clocks synchronization. In particular, the cameras are supposed to broadcast as soon as the computation phase is completed, regardless of the time. For this purpose, in the simulations the transmission order is randomized. Moreover, interferences and communication errors are taken into consideration by modelling the transmission as a boolean random variable  $T_x(\mathcal{P})$  with  $\mathbb{P}(T_x(\mathcal{P})=0) = e_{tx}$ , where  $T_x(\mathcal{P}) = 1$  corresponds to the correct data transmission and  $T_x(\mathcal{P}) = 0$  corresponds to a package lost.

## IV. COORDINATED PATROLLING

In this section, we study how to efficiently patrolling a given environment with an arbitrary number of robots. In particular, in section IV-A a Binary Linear Programming (BLP) strategy relying on pre-computed offline routes is presented, while in section IV-B a Greedy distributed approach based on the Bayes theory is reported. In order to compare the performance of these two different patrolling algorithms, it is important to establish some formal evaluation metrics. From literature, the most common criteria proposed to evaluate the effectiveness of multi-agent patrolling strategies are based on the idleness of the vertices, the distance covered by agents, or the frequency of passages (see [26]). In this work, both the idleness of the vertices and the mean coverage period  $\overline{T_c}$  are considered. This choice is motivated by the fact that the former can provide an estimate of the average detection time and the latter an estimate of the energetic efficiency, since the flying time of UAVs (i.e. battery life) is limited and we want to guarantee the highest possible amount of coverages before the battery drains out.

### A. Binary Linear Programming Strategy

Leveraging on the fact that the environment is known, an extended set of possible paths  $\mathcal{W}$  can be pre-computed offline.

---

**Algorithm 1** BnB Paths Extraction
 

---

```

 $\mathcal{C}_G^b \leftarrow \text{getBaseCycles}(T_G)$ 
 $\mathcal{W} \leftarrow \mathcal{C}_G^b$ 
# BRANCH
for all  $w_i, w_j \in \mathcal{W}$  do
   $w_{i,j} = w_i \text{ xor } w_j$ 
  # BOUND
  if  $\text{isConnected}(w_{i,j})$  then
     $\mathcal{W} \leftarrow \{\mathcal{W}, w_{i,j}\}$ 
    # BRANCH
     $k \leftarrow 1$ 
    while  $k \leq \frac{|w_{i,j}|}{2}$  do
      for all  $W_{i,j,k} \leftarrow \text{set of subpaths of } w_{i,j} \text{ of length } k$  do
         $\mathcal{W} \leftarrow \{\mathcal{W}, W_{i,j,k}, \overline{W_{i,j,k}}\}$ 
      end for
       $k \leftarrow k + 1$ 
    end while
  end if
end for
return  $\mathcal{W}$ 

```

---

For this purpose, the following Branch and Bound (BnB) algorithm is employed. Starting from a spanning tree  $T_G$  of the environment graph  $\mathcal{G}$ , the set of fundamental cycle basis  $\mathcal{C}_G^b$  of  $\mathcal{G}$  can be found selecting the cycles formed by the combination of a path in the tree with a single edge outside the tree. The set of all cycles  $\mathcal{C}_G$  is then obtained evaluating the exclusive disjunction (XOR) for all the combinations of cycles in  $\mathcal{C}_G^b$  and taking the ones resulting in a single connected component (i.e. single closed path). Finally, in order to obtain the complete set of paths  $\mathcal{W}$ , all cycles in  $\mathcal{C}_G^b$  are split in open paths of different length. Note that such open paths are considered to be covered twice, in order to return to the starting point. The pseudocode for the BnB paths extraction is reported in Algorithm 1 and a graphical representation of the steps can be appreciated in Fig. 3.

As outlined in section III-A, the heat map  $\mathcal{H}$  associated with the scene provides some prior knowledge of the environment. Therefore, it is useful to introduce the heat of each path  $\mathcal{H}(w_i)$ ,  $w_i \in \mathcal{W}$ , before providing a detailed description of the optimization problem. Thus, we define it as:

$$\mathcal{H}(w_i) = \sum_{v_j \in w_i} \max \left\{ \mathcal{H}(v_k) : v_k \in A_v(v_j) \right\}. \quad (3)$$

Note that  $\mathcal{H}(w_i)$  intrinsically takes into account the path length, therefore we will refer to it also as *weighted length* of  $w_i$ .

Once the precompiled set of paths  $\mathcal{W}$  is available, the coordinated patrolling can be interpreted as the problem of finding the set of  $n_p$  paths  $\mathbf{w}_{n_p}^* = \{w_1^*, \dots, w_{n_p}^*\} \subset \mathcal{W}$  which yields the complete coverage of the environment (i.e. the vision of each vertex  $v_i \in \mathcal{V}$ ). The common objective is to minimize the coverage time, taking into account the prior knowledge given by the heat map  $\mathcal{H}$ . By the fact that the

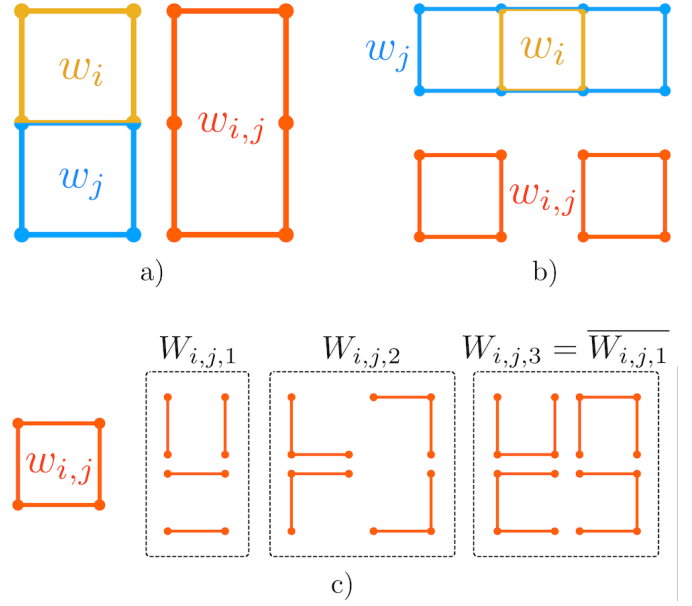


Fig. 3. Graphical representation of the Paths Extraction. **a)** Exclusive disjunction of the cycles  $w_i$  and  $w_j$  resulting in the cycle  $w_{i,j}$ . **b)** Exclusive disjunction of the cycles  $w_i$  and  $w_j$  resulting in  $w_{i,j}$  composed of two distinct connected components and therefore discarded. **c)** Extraction of the sets of subpaths  $W_{i,j,k}$  from  $w_{i,j}$  (note how the evaluation of  $W_{i,j,3}$  can be neglected directly considering  $W_{i,j,3} = \overline{W_{i,j,1}}$ )

complete coverage time is proportional to the weighted length of the longest path, the coordinated patrolling problem can be formalized as follows:

$$\begin{aligned} \arg \min_{\mathbf{w}_{n_p}} \quad & \max_{w_i \in \mathbf{w}_{n_p}} \{ \mathcal{H}(w_i) \} \\ \text{s.t.} \quad & \forall v_k \in \mathcal{V}, \quad \exists w_j \in \mathbf{w}_{n_p} : v_k \in A_v(w_j), \end{aligned} \quad (4)$$

where  $A_v(w_j)$  corresponds to the set of vertices viewed by the camera following the path  $w_j$ .

Moreover, it can be observed that (4) is equivalent to the problem of finding among the sets of  $n_p$  paths that achieve the complete coverage of the environment with a minimal sum of weighted length the set that presents similar heat  $\mathcal{H}(w_i)$  for all the paths  $w_i \in \mathbf{w}$ , i.e.:

$$\begin{aligned} \arg \min_{\mathbf{w}_{n_p}} \quad & \sum_{w_i \in \mathbf{w}_{n_p}} \mathcal{H}(w_i) + \sum_{i \neq j} |\mathcal{H}(w_i) - \mathcal{H}(w_j)| \\ \text{s.t.} \quad & \forall v_k \in \mathcal{V}, \quad \exists w_j \in \mathbf{w}_{n_p} : v_k \in A_v(w_j), \end{aligned} \quad (5)$$

Note that in (5) it becomes clear how paths going through nodes of high heat result to be shorter than the ones surveilling low heat nodes, as it can be observed in Fig. 4.

In order to tackle the coordinated patrolling as a Binary Linear Programming problem, (5) must be expressed as:

$$\begin{aligned} \arg \min_{\mathbf{x}} \quad & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} \quad & \mathbf{A} \mathbf{x} \leq \mathbf{b} \\ & \text{and } \mathbf{x} \in \{0, 1\}^{|\mathcal{W}|}. \end{aligned} \quad (6)$$

For this purpose the following proposition is stated.

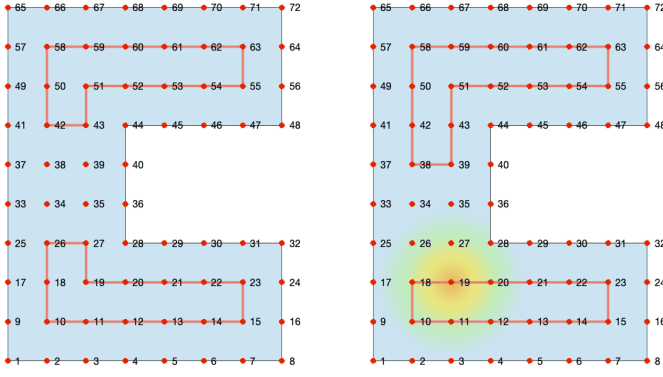


Fig. 4. Optimal paths associated to the patrol of the environment with two cameras. **Left:** All nodes with the same heat  $\mathcal{H}(v_i)$ . **Right:** Heat map after detection in vertex  $v_{19}$ .

**Proposition 1.** *The optimization problem described by*

$$\begin{aligned} \arg \min_{\mathbf{w}_{n_p}} \quad & \sum_{w_i \in \mathbf{w}_{n_p}} e^{\mathcal{H}(w_i)} \\ \text{s.t.} \quad & \forall v_k \in \mathcal{V}, \quad \exists w_j \in \mathbf{w}_{n_p} : v_k \in A_v(w_j) \end{aligned} \quad (7)$$

is equivalent to (5).

*Proof.* Provided in Appendix A.  $\square$

In conclusion, the Binary Linear Programming problem associated to (7) can be stated as follows:

$$\begin{aligned} \arg \min_{\mathbf{x}} \quad & [e^{\mathcal{H}(w_1)} \dots e^{\mathcal{H}(w_{|\mathcal{W}|})}] \mathbf{x} \\ \text{s.t.} \quad & \begin{bmatrix} 1 & \dots & 1 \\ -\mathbf{A}_v(w_1) & \dots & -\mathbf{A}_v(w_{|\mathcal{W}|}) \end{bmatrix} \mathbf{x} \leq \begin{bmatrix} n_p \\ -\mathbf{1} \end{bmatrix} \\ \text{and} \quad & \mathbf{x} \in \{0, 1\}^{|\mathcal{W}|}, \end{aligned} \quad (8)$$

where  $\mathbf{A}_v(w_i) \in \{0, 1\}^{|\mathcal{V}|}$  is the vector associated to the vertices viewed by the camera following the path  $w_i$ . That is, if a vertex  $v_k \in A_v(w_j)$  then the  $k$ -th element of  $\mathbf{A}_v(w_i)$  is equal to 1, otherwise the  $k$ -th element of  $\mathbf{A}_v(w_i)$  is equal to 0.

Due to the combinatory behavior of Algorithm 1, increasing the size of environments, the set of paths  $\mathcal{W}$  offline extraction quickly becomes an NP-Hard problem. Moreover, since the dimensionality of the BLP problem is dependent on the set of paths cardinality  $|\mathcal{W}|$ , i.e.  $\mathbf{x} \in \{0, 1\}^{|\mathcal{W}|}$ , the computation within reasonable time of the optimal set of paths  $\mathbf{w}_{n_p}^*$  could be infeasible.

### B. Greedy State Exchange Bayesian Strategy

In order to overcome the problems that emerged approaching the coordinated patrol with a BLP strategy, the dependence of patrolling algorithm on pre-computed offline routes should be removed. For this purpose, the coordinated patrolling problem can be reinterpreted as follows.

Intuitively, a good patrolling strategy should minimize the time elapsed between two consecutive passages on the same

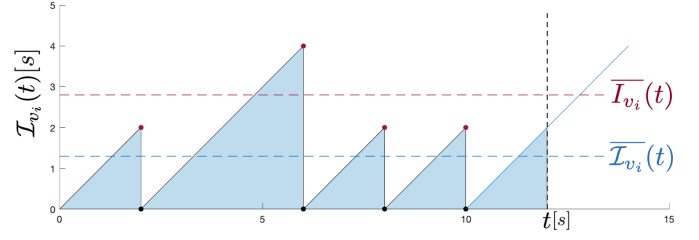


Fig. 5. Graphical representation of the average values for the idleness. Black dots correspond to the time instants when the vertex  $v_i$  is viewed by any camera, i.e.  $C_i = 4$ . Red dots correspond to the maximum idleness peaks.

place, for all places. Referring to this assumption, the instantaneous idleness of a vertex  $v_i \in \mathcal{V}$  at the time step  $t$  is defined as the elapsed time since any agent in the team viewed  $v_i$  last, i.e.:

$$\mathcal{I}_{v_i}(t) = t - t_l, \quad (9)$$

where  $t_l$  corresponds to the last time instant when the vertex  $v_i$  was viewed by any camera.

Consequently, two different averages can be defined for the instantaneous idleness: the *mean idleness*

$$\overline{\mathcal{I}}_{v_i}(t) = \frac{1}{t} \sum_{k=0}^t \mathcal{I}_{v_i}(k), \quad (10)$$

and the *average peak idleness*

$$\overline{\mathcal{I}}_{v_i}(t) = \frac{\overline{\mathcal{I}}_{v_i}(t_l^-) C_i(t) + \mathcal{I}_{v_i}(t)}{C_i(t) + 1}, \quad (11)$$

where  $C_i(t)$  represents the number of times  $v_i$  has been viewed at the time step  $t$ . The difference between these two averages can be appreciated in Fig. 5. In particular, it can be observed how (10) represents the average time needed to view  $v_i$  (i.e. the average time needed to detect an event occurred in  $v_i$ ), while (11) corresponds to the average time elapsed between two consecutive passes on  $v_i$ .

Finally, in order to obtain a generalized measure, the averages are extended to the whole graph as follows:

$$\overline{\mathcal{I}}_{\mathcal{G}}(t) = \frac{1}{|\mathcal{V}|} \sum_{i=1}^{|\mathcal{V}|} \overline{\mathcal{I}}_{v_i}(t), \quad (12)$$

$$\overline{\mathcal{I}}_{\mathcal{G}}(t) = \frac{1}{|\mathcal{V}|} \sum_{i=1}^{|\mathcal{V}|} \overline{\mathcal{I}}_{v_i}(t). \quad (13)$$

Given these definitions, the coordinated patrolling can be reinterpreted as the problem of finding the set of paths  $\mathbf{w}^* = \{w_1^*, \dots, w_{n_p}^*\}$  which yields a complete coverage of the environment (i.e. the vision of each vertex  $v_i \in \mathcal{V}$ ), with the common objective of minimizing  $\overline{\mathcal{I}}_{\mathcal{G}}$ :

$$\begin{aligned} \arg \min_{\mathbf{w}} \quad & \overline{\mathcal{I}}_{\mathcal{G}} \\ \text{s.t.} \quad & \forall v_k \in \mathcal{V}, \quad \exists w_j \in \mathbf{w} : v_k \in A_v(w_j), \end{aligned} \quad (14)$$

where  $A_v(w_j)$  corresponds to the set of vertices viewed by the camera following the path  $w_j$ .

Greedy Bayesian Strategies (GBS) have been successfully used in similar optimization problems, where finding a global optimum in reasonable time is impracticable (see i.e. [6]). The idea is that, after reaching a vertex  $v_0$  of the navigation graph, each camera decides the direction it should travel next, among all adjacent vertices  $v_A \in A(v_0)$ , making a locally optimal choice, with the intent of minimizing the average peak idleness of the graph  $\bar{I}_G$ .

In order to make the locally optimal choice we define two fundamental random variables. The act of moving (or not) from the current vertex  $v_0$  to a neighbour vertex  $v_A$  is defined as the random variable  $move(v_A) = \{true, false\}$ , while the gain  $G_A$  associated to the movement from  $v_0$  to  $v_A$ , assuming constant speed  $c$ , as:

$$G_A(t) = \min \left\{ \frac{c}{|e_{0,A}|} \sum_{v_k \in A_v(v_A)} \mathcal{I}_{v_k}(t), M \right\}, \quad (15)$$

where the value  $M > 0$  can be obtained considering an upper bound on  $\mathcal{I}_{v_i}(t)$ . Moreover, the likelihood associated to  $G_A$  is modelled as the exponential function:

$$\mathbb{P}(G_A | move(v_A)) = L \cdot e^{\frac{\ln(1/L)}{M} G_A}, \quad (16)$$

with  $0 < L < 1$  and  $G_A \leq M$ , so that higher values of gain become rapidly more influential on the movement decision.

$L$  and  $M$  are the parameters that allow the likelihood to be finely tuned. More specifically,  $L$  controls the probability value for lower gains and  $M$  is the gain saturation beyond which the probability values are maximum. Intuitively, a good value for  $L$  should be close to 0, while  $M$  can be empirically tuned considering the expected maximum gain  $G_A$ . In this work, the values considered are  $L = 0.1$  and  $M = 9|\mathcal{V}|/n_p$ .

In collective operations with a common objective, coordination between agents plays a fundamental role in making the mission successful. In situations where the number of cameras per area is high, agents will tend to compete to arrive in the same region, resulting in overlapping FoVs. In order to prevent this situation, a State Exchange paradigm is implemented taking advantage of the communication network.

By collecting intentions of other cameras, all agents are capable of computing how much each vertex should be considered in their decision process. In particular, the greater the number of teammates intentioned to view a vertex, the less likely for the camera to move in that direction. This behavior is herein described with an approximation of a geometric sequence of ratio  $1/2$ , resulting in the following likelihood:

$$\mathbb{P}(S_A | move(v_A)) = \frac{2^{n_p - (S_A + 1)}}{2^{n_p} - 1}, \quad (17)$$

where

$$S_A = \sum_{v_k \in A_v(v_A)} |\{c_i \in \mathcal{C} : v_k \in A_v(V_{t+1}^i)\}| \quad (18)$$

is the penalty term associated to the number of cameras intended to view one of the vertices that can be covered moving to  $v_A$ .

Finally, the probability of moving to a specific vertex  $v_A$  given its gain  $G_A$  and penalty factor  $S_A$ , is calculated applying the Bayes rule:

$$\mathbb{P}(move(v_A) | G_A, S_A) \propto \mathbb{P}(move(v_A)) \mathbb{P}(G_A | move(v_A)) \cdot \mathbb{P}(S_A | move(v_A)), \quad (19)$$

where  $\mathbb{P}(move(\cdot))$  represents prior knowledge or assumptions on the environment. In this particular application, the prior encoded in the heat map  $\mathcal{H}$  can be formalized as follows:

$$\mathbb{P}(move(v_A)) = \frac{\sum_{v_k \in A_v(v_A)} \mathcal{H}(v_k)}{\sum_{\tilde{v}_A \in A(v_0)} \sum_{\tilde{v}_k \in A_v(\tilde{v}_A)} \mathcal{H}(\tilde{v}_k)}. \quad (20)$$

Note that the proportionality in (19) is given by the fact that the denominator term of the Bayes rule should be  $\mathbb{P}(G_A) \mathbb{P}(S_A)$  and it is regarded as a normalization factor.

The high-level pseudo-code of the State Exchanged Bayesian Strategy (SEBS) running locally on each camera is presented in Algorithm 2. This Greedy approach to the coordinated patrolling is heavily modular: each decision is independent and the agents have the ability to choose the action which has the greatest expectation of utility, weighted by the effects of all possible actions. Thus, each camera's patrol route is built progressively, at each decision step, adapting to the system's needs and changes in the team size  $n_p$ .

---

#### Algorithm 2 Greedy SEBS

---

```

 $v_n \leftarrow v_0$ 
 $\mathcal{I}_v \leftarrow \mathbf{0}$ 
while true do
  for all  $v_A \in A(v_n)$  do
     $G_A \leftarrow \frac{c}{|e_{0,A}|} \cdot \sum_{v_k \in A_v(v_A)} \mathcal{I}_{v_k}(t)$ 
     $\mathbb{P}(G_A | move(v_A)) \leftarrow L \cdot e^{\frac{\ln(1/L)}{M} G_A}$ 
     $S_A \leftarrow \sum_{v_k \in A_v(v_A)} |\{c_i \in \mathcal{C} : v_k \in A_v(V_{t+1}^i)\}|$ 
     $\mathbb{P}(S_A | move(v_A)) \leftarrow \frac{2^{n_p - (S_A + 1)}}{2^{n_p} - 1}$ 
     $\mathbb{P}(move(v_A)) \leftarrow \frac{\sum_{v_k \in A_v(v_A)} \mathcal{H}(v_k)}{\sum_{\tilde{v}_A \in A(v_0)} \sum_{\tilde{v}_k \in A_v(\tilde{v}_A)} \mathcal{H}(\tilde{v}_k)}$ 
     $\mathbb{P}(move(v_A) | G_A, S_A) \leftarrow \frac{\mathbb{P}(move(v_A)) \mathbb{P}(G_A | move(v_A)) \mathbb{P}(S_A | move(v_A))}{\mathbb{P}(G_A) \mathbb{P}(S_A)}$ 
  end for
   $v_{n+1} \leftarrow \arg \max \{\mathbb{P}(move(v_A) | G_A, S_A)\}$ 
   $\mathcal{P}_0 \leftarrow \{0, v_n, v_{n+1}\}$ 
  transmit( $\mathcal{P}_0$ )
  while moving to  $v_{n+1}$  do
    receive_transmissions()
    update( $\mathcal{I}_v$ )
    update( $\mathcal{H}$ )
  end while
   $v_n \leftarrow v_{n+1}$ 
end while

```

---



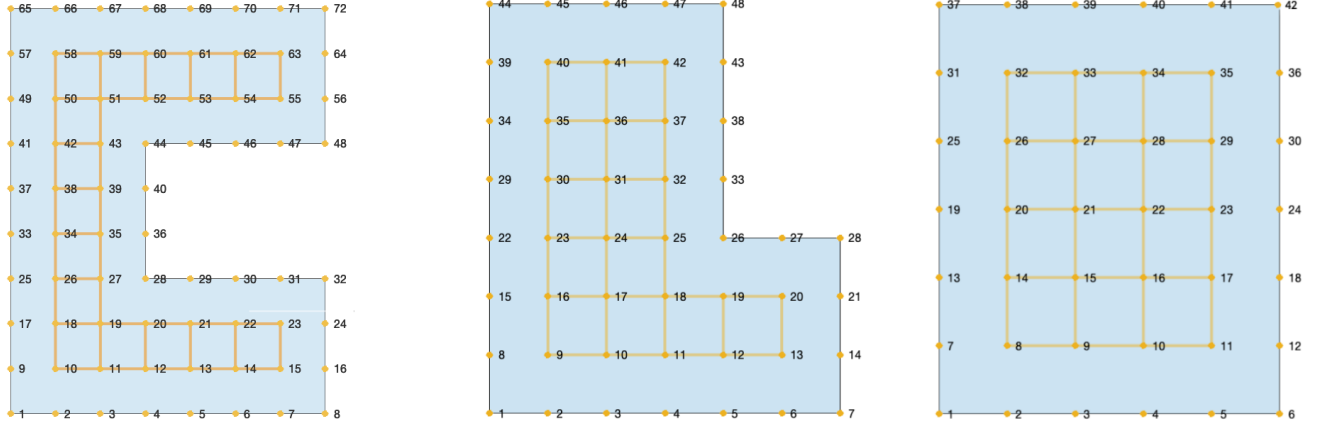


Fig. 6. **Left:** C-shaped environment. **Middle:** L-shaped environment. **Right:**  $6 \times 5$  rectangular environment (herein referred to as O-shaped).

### C. Simulations and Results

To evaluate the performance of the two approaches taken into account for the coordinated patrolling problem, the following numerical simulations are designed. In the first experiment, the results obtained by the two patrolling algorithms on restricted environments, where the BLP strategy is feasible, are compared relying on the mean idleness  $\overline{\mathcal{I}_G}$  and on the average coverage period  $\overline{T_c}$ . In the second experiment, the performance of the SEBS is evaluated on a large scale environment.

Before going into the analysis of the results, it is important to highlight that in the initialization phase of the two algorithms the instantaneous idleness  $\mathcal{I}_v$  of all vertices is set to 0, as if each vertex had just been visited. As a consequence, there is a transitory phase in which the values of  $\overline{\mathcal{I}_G}$  and  $\overline{T_c}$  tend to be different from the real case scenario. For this reason, the final values of the mean idleness and average coverage period are evaluated after the convergence in the stable phase. Moreover, while in the BLP strategy the paths are evaluated in their totality and repeated indefinitely, in the SEBS the path of each camera is determined step by step and can be non regular among the iterations over the environment. Therefore, for the SEBS the average value over 100 patrolling cycles is considered. Furthermore, since the steady-state paths of the SEBS appear to be heavily dependent on the initial conditions (i.e. the starting position of each camera), in what follows the results associated to such strategy are evaluated considering the mean values over 1000 simulations starting from random initial conditions.

1) *Experiment 1:* Fig. 6 presents three cellularized environments considered for the comparison between the BLP strategy and the SEBS. As anticipated, the decision of considering small scale maps is due to the long computational time associated to the extraction of the set of paths  $\mathcal{W}$  in the BLP strategy. For the sake of completeness, referring to the notation used in Fig. 6, the running time of Algorithm 1 and the cardinality of the resulting set of paths  $|\mathcal{W}|$  are reported in Table I.

TABLE I  
CARDINALITY  $|\mathcal{W}|$  AND RUNNING TIME OF ALGORITHM 1 (IN MINUTES).

	C-shaped	L-shaped	O-shaped
$ \mathcal{W} $	9664	29176	49963
time	36	112	187

For an unbiased comparison of the two techniques, the heat associated to each vertex  $\mathcal{H}_{v_i}(t)$  is considered to be unitary and constant. The results in terms of average coverage period  $\overline{T_c}$  and mean idleness  $\overline{\mathcal{I}_G}$ , obtained considering unitary resolution and constant speed  $c = 1\text{m/s}$ , are reported respectively in Table III and Table II. Note that the analysis loses meaning once the coverage period  $\overline{T_c}$  gets lower than one second. This is due to the fact that  $\overline{T_c} = 1$  corresponds to the situation in which all cameras are moving back and forth between the same two vertices. Therefore, the addition of a further agent would result in at least one of the team members not moving,

TABLE II  
OVERVIEW OF THE RESULTING MEAN IDLENESS  $\overline{\mathcal{I}_G}$  [s] WITH BLP AND SEBS (CONSIDERING UNITARY RESOLUTION AND CONSTANT SPEED  $c = 1\text{m/s}$ ).

$n_p$	C-shaped		L-shaped		O-shaped	
	BLP	SEBS	BLP	SEBS	BLP	SEBS
1	11.498	11.517	6.224	6.228	4.753	4.755
2	4.295	4.486	2.169	2.551	1.687	1.870
3	2.395	2.562	0.875	0.976	0.657	0.775
4	1.100	1.272	0.417	0.515	0.143	0.206
5	0.583	0.688	0.188	0.248		
6	0.361	0.422				
7	0.250	0.264				



TABLE III

OVERVIEW OF THE RESULTING AVERAGE COVERAGE PERIOD  $\overline{T_c}$  [s] WITH BLP AND SEBS (CONSIDERING UNITARY RESOLUTION AND CONSTANT SPEED  $c = 1\text{M/S}$ ).

$n_p$	C-shaped		L-shaped		O-shaped	
	BLP	SEBS	BLP	SEBS	BLP	SEBS
1	31	31.11	17	17.02	13	13.00
2	13	13.65	9	9.88	7	7.02
3	11	10.29	5	5.25	5	4.68
4	5	5.56	3	3.00	1	1.00
5	3	3.22	1	1.81		
6	3	3.15				
7	1	1.67				

i.e. such team member can be replaced by a fixed camera.

As expected, analyzing Table II it can be observed that the results obtained by the SEBS algorithm with  $n_p = 1$  appear to be extremely close in terms of  $\overline{T_G}$  to the ones obtained by the BLP strategy (maximum difference of 0.03%). When increasing the size of the patrolling team the percentage difference between the two strategies increases. This is because a higher number of agents introduces a higher complexity in the coordination tasks. However, neglecting the peculiar situation in which the BLP strategy presents  $\overline{T_c} = 1$  that leads to high percentage differences (maximum peak: 44%), the gap between the results of the SEBS and the ones of the BLP strategy is on average 11%, with a maximum of 23.5%. Therefore, since this type of surveillance system is meant to be adopted on large scale scenarios, in which achieving near unitary coverage period  $\overline{T_c}$  is unreasonable (otherwise a fixed camera network would be a better option), the obtained results are extremely encouraging.

As concerns the energetic efficiency, referring to Table III, the BLP strategy appears to be on average 0.21 seconds faster than the SEBS, and therefore more efficient. However, some

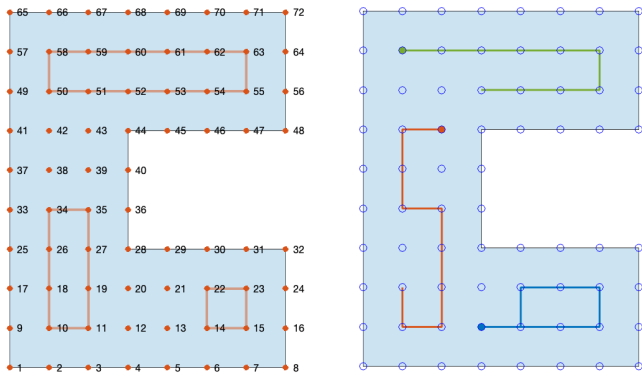


Fig. 7. Example of the results achieved by the two algorithms on the C-shaped environment with  $n_p = 3$ . **Left:** BLP paths with  $\overline{T_c} = 11\text{s}$  and  $\overline{T_G} = 2.395\text{s}$ . **Right:** SEBS paths with  $\overline{T_c} = 9\text{s}$  and  $\overline{T_G} = 2.668\text{s}$ .

TABLE IV

OVERVIEW OF THE AVERAGE PERCENTAGE OF ENVIRONMENT SPANNED BY THE SEBS IN THE TIME NEEDED BY THE BLP ALGORITHM TO COMPLETE A FULL COVERAGE.

$n_p$	C-shaped	L-shaped	O-shaped
1	99.5%	100.0%	100.0%
2	97.0%	96.5%	98.2%
3	99.1%	99.1%	99.4%
4	98.9%	99.6%	100.0%
5	99.2%	97.5%	
6	99.6%		
7	97.0%		

observations are needed for the results obtained with  $n_p = 3$  in the C-shaped and O-shaped environments. In those cases the average coverage period of the SEBS appears to be lower than the one obtained by the BLP strategy. This behavior may be due to the effect of the initial conditions on the steady-state paths, causing agents to get stuck in local optima. In these situations the SEBS returns short paths that cover the environment quickly but results in a higher average idleness. An example of such situation is reported in Fig 7.

A further analysis can be carried out relying on the area covered by the two algorithms in a fixed amount of time. In particular, the results in terms of percentage of environment spanned by the SEBS in the time needed by the BLP to complete a full coverage is reported in Table IV. Here it can be appreciated how the differences in  $\overline{T_c}$  are mapped into an average 98.8% coverage of the environment, with a minimum of 96.5%. These results also provide an idea on the regularity of the paths, i.e. the differences among multiple runs. In particular, the fact that the percentage associated to the two peculiar cases, obtained with  $n_p = 3$  in the C-shaped and O-shaped environments, is not 100% means that the short paths reported in Fig 7 are then compensated by much longer ones.

2) *Experiment 2:* In order to evaluate the performance of the SEBS in a real-like applicative scenario, the environment in Fig. 8 is adopted. Such environment is designed to present many of the possible situations in which a patrolling agent could get into to complete its task, e.g. bottlenecks, dead ends, obstacles to overcome, and open spaces. On such environment, in addition to the usual evaluation metrics  $\overline{T_c}$  and  $\overline{T_G}$ , the average time  $\overline{T_d}$  elapsed between an update of the heat map (i.e. an event detection or tracking loss) and the coverage of the hot zone is evaluated. For this purpose, a heat update is triggered in a random position of the environment at a random time instant.

The obtained results are summarized in Table V. In particular, it can be appreciated how the added heat attracts the agents in the desired position resulting in values of  $\overline{T_d}$  on average 52.7% lower than the average time needed to detect an event

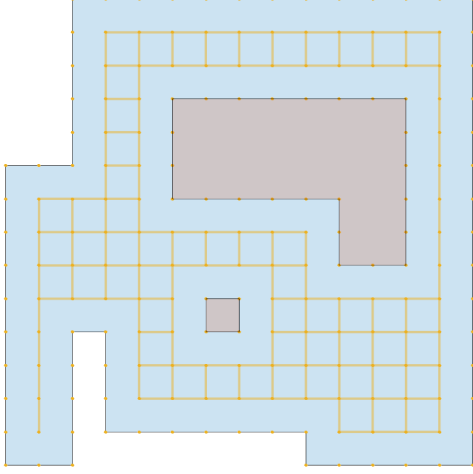


Fig. 8. Real-like large scale environment with bottlenecks, dead ends, and No Fly Zones (highlighted in red).

TABLE V  
OVERVIEW OF THE RESULTS ON THE REAL-LIKE LARGE SCALE ENVIRONMENT.

$n_p$	$\overline{T}_c[s]$	$\overline{T}_G[s]$	$\overline{T}_d[s]$
1	135.85	38.113	28.908
2	64.10	18.718	13.687
3	44.57	11.905	8.356
4	32.68	8.289	5.999
5	25.65	6.115	4.073
6	21.07	4.682	3.087
7	17.78	3.664	2.470
8	15.17	2.908	1.960
9	13.07	2.337	1.263
10	11.33	1.888	0.970

without prior knowledge, encoded by the average idleness  $\overline{T}_G$ .

Moreover, referring to the graphs in Fig. 9, it can be observed how an upgrade of the system  $\Delta n_p$  (i.e. passing from  $n_p$  to  $n_p + 1$  agents) affects the patrolling performance. This analysis becomes really useful for the selection of the optimal number of agents, since increasing  $n_p$  does not lead to a linear increase in performance.

## V. SMART TRACKING

In this section, a Smart Target Tracking system composed of a Kalman filter and an optimized Camera Zoom Control (CZC) is proposed. More precisely, starting from the one step ahead predictions of the Kalman filter, we present an algorithm relying on the covariance matrix of the prediction error to return a UAV height (i.e. the zoom level) that takes into account the trade-off between target containment and information loss minimization. In this way, the tracking camera follows the target with the highest possible resolution, guaranteeing the

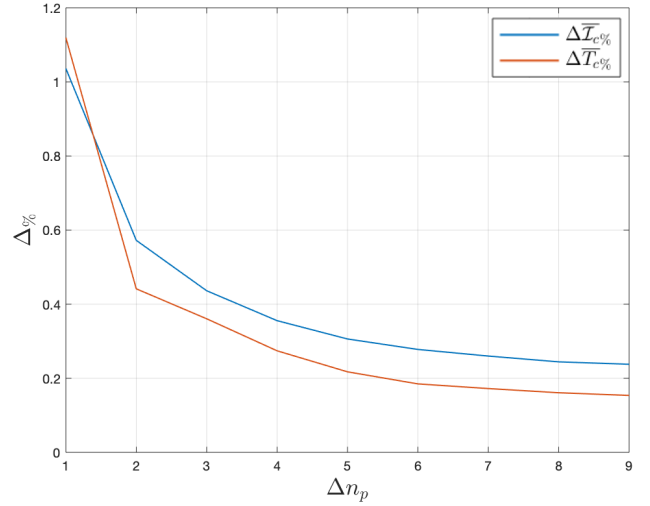


Fig. 9. Percentage increase of the evaluation metrics  $\overline{T}_c$  and  $\overline{T}_G$  associated with multiple upgrades of the system  $\Delta n_p$ .

desired confidence. In particular, in section V-A the structure of the Kalman filter, with both the process and measurement models, is presented. In sections V-B and V-C the optimal problem for zoom control is formalized. Finally, numerical simulations changing the target trajectory models and varying the sampling frequency of the filter are provided in section V-D.

### A. Tracking Kalman Filter

As previously mentioned, the problem of tracking a moving target is addressed resorting to the Kalman filter theory. For easy reference, standard Kalman filter notations are given. Consider the following linear discrete-time system with sampling time  $T$ :

$$\begin{cases} \mathbf{x}(t+T) = F \mathbf{x}(t) + \mathbf{v}(t) \\ \mathbf{y}(t) = H \mathbf{x}(t) + \mathbf{w}(t) \end{cases} \quad (21)$$

where  $\mathbf{x}(t)$  denotes the state and  $\mathbf{y}(t)$  the measurements. The process noise  $\mathbf{v}(t)$  and the measurement noise  $\mathbf{w}(t)$  are zero-mean Gaussian noises. These two processes are considered uncorrelated with covariance  $Q$  and  $R$  respectively.

The well-known equations of the discrete-time Kalman filter for the one-step-ahead prediction of the state  $\hat{\mathbf{x}}(t+T|t)$  and the Kalman gain  $K(t)$  are

$$\hat{\mathbf{x}}(t+T|t) = F \hat{\mathbf{x}}(t|t-T) + K(t)[\mathbf{y}(t) - H \hat{\mathbf{x}}(t|t-T)], \quad (22)$$

$$K(t) = F P(t|t-T) H^\top (H P(t|t-T) H^\top)^{-1}, \quad (23)$$

where the prediction error covariance matrix  $P(t+T|t)$  is obtained from the solution of the Differential Riccati Equation (DRE) as follows:

$$\begin{aligned} P(t+T|t) = & F P(t|t-T) F^\top + Q \\ & - F P(t|t-T) H^\top (H P(t|t-T) H^\top)^{-1} \\ & \cdot H P(t|t-T) H^\top. \end{aligned} \quad (24)$$

For a time-invariant system under the usual observability and controllability conditions, the solution of the DRE converges to a unique steady-state covariance. Such covariance value, that corresponds to the solution of the Algebraic Riccati Equation (ARE), results in the reliability of prediction being upper bounded.

This type of tracking filter works relying on a target motion model. The common approach is to model the unknown motion through a second-order two-dimensional linear system driven by white noise process (see e.g. [27]). For this purpose, the state vector  $\mathbf{x}$  is defined as

$$\mathbf{x}(t) = [x(t) \ y(t) \ \dot{x}(t) \ \dot{y}(t)]^\top, \quad (25)$$

where  $(x, y)$  denote the object position and  $\dot{x}$  and  $\dot{y}$  its velocity along the  $x$ -axis and  $y$ -axis respectively. Therefore, assuming a dynamic model with constant velocity during the sampling interval  $T$ , the transition matrix from  $t = kT$  to  $(k+1)T$  is expressed as

$$F = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (26)$$

In the context of tracking systems, the most important design parameters of the Kalman filter are the elements of the covariance matrix of the process noise  $Q$ . Such matrix must be designed to achieve a tracking error that is as small as possible. In [28] it has been observed that, among the most commonly used noise processes in conventional tracking systems, a random acceleration (RA) process often yields better performance. Therefore, we assume the covariance matrix of the process noise  $Q$  to be as follows:

$$Q = \begin{bmatrix} T^4/4 & 0 & T^3/2 & 0 \\ 0 & T^4/4 & 0 & T^3/2 \\ T^3/2 & 0 & T^2 & 0 \\ 0 & T^3/2 & 0 & T^2 \end{bmatrix} \sigma_q^2, \quad (27)$$

where the variance  $\sigma_q^2$  is treated as a design parameter and empirically selected relying on some assumptions on target motion. In this work we consider  $\sigma_q^2 = 10$ .

For what concerns the measurement system, the filter relies on the measurement  $(\hat{x}, \hat{y})$  of the position of the target provided by the event detection framework. A Position Only Measurement System (POMS) is thence assumed, where the measurement matrix  $H$  and the measurement noise covariance matrix  $R$  are expressed as follows:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad (28)$$

$$R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \sigma_{r\%}^2, \quad (29)$$

where  $\sigma_{r\%} = d \cdot r\%$ , for a reasonable value  $r\% \in (0, 1)$ .

In this context, the measurement matrix  $H$  can be considered time-variant. In particular,  $H$  is set to zero whenever no measure occurs in  $T$  (i.e. there is a camera failure or the target is out of camera FoV). As a consequence, the Kalman algorithm propagates the estimate of the previous step increasing the covariance of the estimation error. Such propagation continues for a maximum of  $N_k$  consecutive updates of the filter without event detection. After that, the target is considered to be lost and the camera returns in patrolling.  $N_k$  encodes the resilience of the camera, i.e. the inclination to keep tracking or return patrolling, and it is considered a design parameter to be properly chosen.

### B. Optimal Zoom

As stated in the introductory section, the purpose of this algorithm is to obtain the optimal zoom value  $z^*$  considering the trade-off between information loss minimization and target containment.

Recalling that  $z_{min} \leq z_c \leq z_{Max}$ , the information loss  $I(z)$  can be considered null at minimum height and increasing exponentially going up in altitude. Such a behavior can be encoded via the exponential function reported hereunder:

$$I(z) \triangleq e^{z-z_{min}} - 1, \quad (30)$$

for  $z \in [z_{min}, z_{Max}]$ , in what follows the minimum height is considered to be  $z_{min} = z_{Max}/4$ . The shape of  $I(z)$  considering unitary resolution is represented in Fig. 10.

It is important to underline that a lot of different functions could have been used to model the information loss. In particular, function (30) could be adjusted according to any further information available on the CV system or camera behavior. To minimize the information lost, the height  $z_c$  of the camera

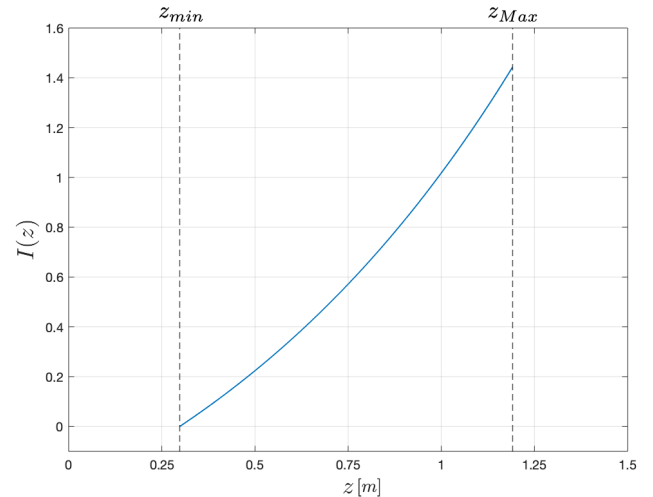


Fig. 10. Information loss function in the feasible region  $z \in [z_{min}, z_{Max}]$

should be as close as possible to  $z_{min}$ . However, decreasing in altitude results in higher tracking loss probability.

In order to evaluate the target containment, the Confidence Interval (CI) for the prediction of the target position need to be gathered from the Kalman prediction errors covariance matrix (that is our information about the estimate reliability). Hence, in this work the optimal problem uses as input  $P(t+T|t)$  to find a proper value of UAV height  $z^*$  that guarantee to follow the target at the highest possible resolution minimizing the probability to loose it.

As first step, from equation (25) it can be observed that the state of the model takes into account both the position and speed. Hence,  $P(t+T|t)$  is a  $4 \times 4$  matrix in which the diagonal contains the variance of the prediction errors associated with the position (first two elements) and speed (last two elements), i.e.  $P_{i,i} = \{\sigma_x^2, \sigma_y^2, \sigma_{\dot{x}}^2, \sigma_{\dot{y}}^2\}$ . Moreover, due to the structure of the process covariance matrix  $Q$  and equation (24),  $\sigma_x^2 = \sigma_y^2$  and  $\sigma_{\dot{x}}^2 = \sigma_{\dot{y}}^2$ .

In the situation when the camera is close to the target (i.e. the FoV is narrow), an error in the speed prediction could cause the tracking loss. To also account for the effect of errors in the speed prediction on the tracking task, all the standard deviations  $\sigma_x$ ,  $\sigma_y$ ,  $\sigma_{\dot{x}}$ , and  $\sigma_{\dot{y}}$  are enclosed in the following standard deviation:

$$\sigma_p = \max \left\{ \frac{\sigma_x + \sigma_y}{2}, \frac{\sigma_{\dot{x}} + \sigma_{\dot{y}}}{2} T \right\}, \quad (31)$$

where the standard deviation of the velocity prediction errors  $\sigma_{\dot{x}}$  and  $\sigma_{\dot{y}}$  is converted in meters multiplying it by the sampling time  $T$ . Note that in (31) the average values are considered to limit the effect of computational errors that cause  $\sigma_x^2 \simeq \sigma_y^2$  and  $\sigma_{\dot{x}}^2 \simeq \sigma_{\dot{y}}^2$  and not exactly equal.

In this way, the tracking error  $e_p$  can be expressed as a single Gaussian r.v. with zero mean and variance  $\sigma_p^2$ . Therefore, the confidence interval for the predicted position  $\hat{x}$  can be easily evaluated as follows:

$$CI_k = [\hat{x} - k\sigma_p, \hat{x} + k\sigma_p], \quad (32)$$

where the coverage factor  $k$  encodes the required confidence level. This is a conservative definition for the CI of  $\hat{x}$  since the extended uncertainty is evaluated with the highest standard deviation. Relying on (32) and on the vision model of the cameras, described in Section III-B, the following linear constraint on the height of the UAV can be defined:

$$k\sigma_p \leq z_c \tan \theta = d. \quad (33)$$

This means that camera height must be at least such that the desired confidence interval is contained within the FoV.

Usually, the coverage factor  $k$  is considered to be fixed and between 2 and 3, which corresponds to a confidence level in (32) between 95.44% and 99.74%. However, in this work, a dynamic approach to the selection of a suitable value for the coverage factor  $k \in [k_{min}, k_{Max}]$  is proposed. In particular, we consider  $k_{min} = 2$  and  $k_{Max} = 3$ . This decision is motivated by the fact that it is reasonable to consider that in some situations a camera should increase the risk of losing

the tracking to gather more information. For this purpose, the relation between the tracking confidence and the coverage factor  $k$  is obtained as follows.

Let consider the normalized version of tracking error  $e_p$ , i.e.  $Z \triangleq e_p/\sigma_p \sim \mathcal{N}(0,1)$ . Then, it is possible to calculate the confidence  $U(k)$  w.r.t. a given coverage factor  $k$  using the cumulative distribution function  $\Phi(\cdot)$  of the normal distribution as follows:

$$U(k) \triangleq \mathbb{P}(|Z| \leq k) = 2\Phi(k) - 1. \quad (34)$$

Since the cumulative distribution function  $\Phi(k)$  is very difficult to evaluate, we want an approximation of (34) that preserves as much as possible the relation between  $k$  and the confidence level. To this end, we consider the hereunder function:

$$\tilde{U}(k) \triangleq 1 - e^{-\frac{k}{b}}, \quad (35)$$

where  $b$  is a parameter to be accurately selected for reducing the difference between these two functions. Such a selection could be done either by a *trial & error* approach or solving the following LS optimization problem:

$$b^* = \arg \min_{b \in \mathbb{R}} \{U(k) - \tilde{U}(k)\}^2 \quad \text{s.t. } k \in [k_{min}, k_{Max}]. \quad (36)$$

From now on, when we will refer to (35) we consider  $b = b^* = 0.6161$ , i.e. the optimized value for such a parameter.

The functions  $U(k)$  and  $\tilde{U}(k)$  are reported in Fig. 11, where it can be appreciated that in the required interval  $k \in [2, 3]$  the behavior of these functions is extremely similar. Moreover, the good results obtained with (35), and reported in Section V-D, suggest that such function is a valid approximation in a restricted domain as chosen. Note that, (35) may not guarantee

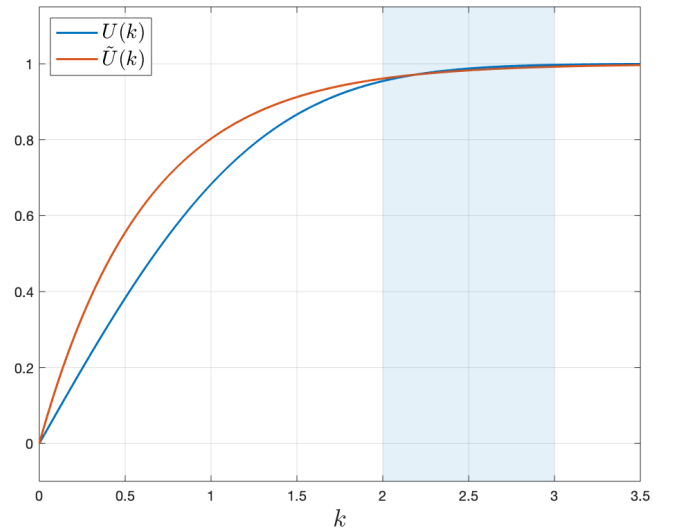


Fig. 11. Comparison between the true confidence function, depicted in blue, and the designed confidence, depicted in orange. The interval  $k \in [2, 3]$  is highlighted in blue.

the same results so far as the similarity between the two functions is required for a larger interval. In such a situation further analysis is necessary.

Once the functions  $I(z)$  and  $\tilde{U}(k)$  have been designed, it can be observed that these relations are in conflict with each other. More precisely, while the former is required to be minimal, the latter is asked to be as high as possible. Therefore, the following non-linear optimization problem with linear constraint can be stated:

$$\begin{aligned} (z^*, k^*) = \arg \min \quad & I(z) - \gamma \cdot \tilde{U}(k) \\ \text{s.t.} \quad & k\sigma_p \leq z \tan \theta = d \\ \text{and} \quad & z \in [z_{min}, z_{Max}] \\ & k \in [k_{min}, k_{Max}], \end{aligned} \quad (37)$$

where the trade-off is regulated by the hyper-parameter  $\gamma$ . The value of such a hyper-parameter determines the importance given to  $I(z)$  or  $\tilde{U}(k)$  respectively (i.e. information loss minimization or target containment) and must be chosen wisely.

---

**Algorithm 3** Smart Tracking with Optimized Zoom

---

**On event detected in  $(\tilde{x}, \tilde{y})$  with w.g.n. of s.d.  $\sigma_{r\%}$**

$\hat{x}(T|0) \leftarrow [\tilde{x} \ \tilde{y} \ 0 \ 0]$

$$P(T|0) \leftarrow \begin{bmatrix} \sigma_{r\%}^2 & 0 & 0 & 0 \\ 0 & \sigma_{r\%}^2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$n_k \leftarrow 0$

$t \leftarrow T$

**while**  $n_k \leq N_k$  **do**

$X(t) \leftarrow \text{getCameraPosition}()$

$y(t) \leftarrow \text{event\_detection}()$

**if**  $y(t).isEmpty()$  **then**

$H \leftarrow 0$

$n_k \leftarrow n_k + 1$

**else**

$$H \leftarrow \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$n_k \leftarrow 0$

**end if**

$[\hat{x}(t+T|t), P(t+T|t)] \leftarrow \text{Kalman\_filter}()$

$(x^*, y^*) \leftarrow [\hat{x}_1(t+T|t), \hat{x}_2(t+T|t)]$

**if**  $t > k_s \cdot T$  **then**

$$\sigma_p \leftarrow \max \left\{ \frac{\sigma_x + \sigma_y}{2}, \frac{\sigma_{\tilde{x}} + \sigma_{\tilde{y}}}{2} T \right\}$$

$z^* \leftarrow \text{optimal\_zoom}(\sigma_p)$

**else**

$z^* \leftarrow z_{Max}$

**end if**

$X^*(t+T) \leftarrow [x^* \ y^* \ z^*]$

$\text{camera.Move}(X^*(t+T))$

$t \leftarrow t + T$

**end while**

---

Due to the model selected for the design of the Kalman filter in Section V-A, the first complete prediction is obtained after  $T$  seconds. Indeed, as soon as the camera detects an event in  $(\tilde{x}, \tilde{y})$  the prediction provided by the filter is, clearly, that the target is not moving, i.e.  $\hat{x}(T|0) = [\tilde{x}, \tilde{y}, 0, 0]$ . This accumulated delay translates into a transitory phase in which the predictions of the Kalman filter are not extremely accurate. For this reason, the camera is fixed at  $z_{Max}$  for the first  $k_s \cdot T$  seconds. Finally, the high-level pseudocode for the Smart Tracking with Optimized Zoom is reported in Algorithm 3.

### C. Zoom Control

Usually the UAVs present high maximum horizontal speed (e.g. 47-72 km/h) and vertical speed between  $-3\text{m/s}$ , while descending, and  $5\text{m/s}$ , while ascending (see [23] [24]). For the bare target tracking, where only the horizontal speed is relevant, the accurate design of a control system is not of primary importance for the scopes of this work. Thus, the camera can be assumed to move from the current position  $(x, y)$  to the desired one  $(x^*, y^*)$  with constant velocity and in the desired time  $T$ . On the other hand, since the actual height  $z_c(t)$  of the camera is extremely relevant both in the definition of the measurement noise covariance matrix  $R$  of the Kalman filter and therefore in the evaluation of the optimal zoom  $z^*(t+T)$  and the vertical velocity is limited, an accurate design of a height controller is required.

Considering that the Kalman filter working frequency is  $f = 1/T$  Hz, (37) yields a new reference zoom level  $z^*$  every  $T$  seconds. Therefore, it is reasonable to consider a height controller that satisfies the following specifications:

- perfect steady-state tracking of step references;
- step response with settling time  $t_{s,5\%} \leq T$ ;
- overshoot as small as possible, e.g.  $M_p \leq 1\%$  (to avoid exiting  $[z_{min}, z_{Max}]$ ).

For simplicity, the following continuous-time constant speed model for the vertical motion of the UAV is considered:

$$\begin{cases} \dot{z}(t) = F_z z(t) + G_z u(t) \\ y(t) = H_z z(t) \end{cases} \quad (38)$$

where the state  $z(t) = [z_c(t) \ \dot{z}_c(t)]^\top$ ,  $u(t)$  is the input speed and

$$F_z = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad G_z = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad H_z = \begin{bmatrix} 1 & 0 \end{bmatrix}. \quad (39)$$

Therefore, the available measure is  $y(t) = z_c(t)$ .

To achieve the desired specifications, a PID controller is designed. Due to the limited vertical velocity of the UAV, a saturation block with upper limit  $5\text{m/s}$  and lower limit  $-3\text{m/s}$  is introduced in the control scheme. Thus, an anti-windup scheme is desirable to avoid the integral action persistency when the speed is saturating. An overview of the adopted control scheme can be appreciated in Fig. 12.

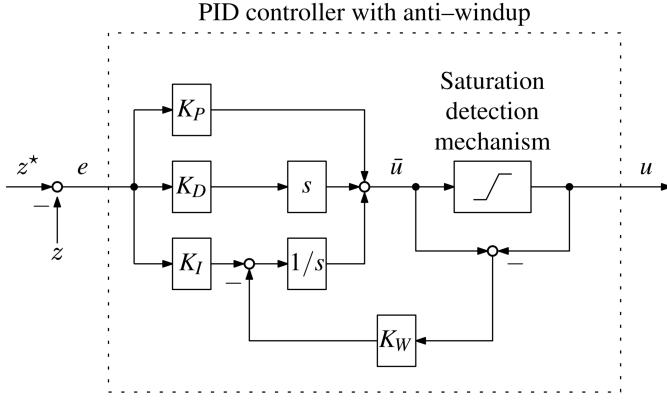


Fig. 12. PID controller with anti-windup scheme.

TABLE VI  
PID CONTROLLER GAINS.

$f$ [Hz]	$K_P$	$K_I$	$K_D$	$K_W$
10	431.77	1318.68	35.34	10
5	108.18	165.21	17.71	5
3.3	48.27	49.14	11.85	3.3
1	4.64	1.42	3.80	1

For the design of the PID controller, the damping factor  $\delta$ , the gain crossover frequency  $\omega_{gc}$ , and the phase margin  $\varphi_m$  can be derived from the time domain specifications as follows:

$$\delta = \frac{\log(1/M_p)}{\sqrt{\pi^2 + \log^2(1/M_p)}}, \quad \omega_{gc} \approx \frac{3}{\delta t_{s,5\%}}, \quad (40)$$

$$\varphi_m = \text{atan} \frac{2\delta}{\sqrt{1 + 4\delta^4} - 2\delta^2}.$$

The ratio between the integral and derivative times ( $\alpha = T_I/T_D$ ) is taken equal to 4 and the derivative action is implemented with a real derivative, consisting of a first order highpass filter of the type

$$L(s) = \frac{s}{T_L s + 1}, \quad (41)$$

with a cut-off frequency  $1/T_L \approx 10 \omega_{gc}$ .

The gains resulting from the Bode's method and the anti-windup gain  $K_W = 1/t_{s,5\%}$  considering different frequencies  $f = 1/T$  for the Kalman filter are reported in Table VI.

Note that, a different *ad-hoc* design of the control system should be considered so far as the actual model of the employed UAVs' dynamics is available.

#### D. Simulations and Results

Before evaluating the performance of the smart tracking system, the procedure followed to obtain the hyper-parameter  $\gamma$  is described in the first subsection. After that, several numerical simulations are performed to test the reliability of the predictions. Finally, the robustness of the optimized zoom control to measurement errors and camera failures is analyzed in the last subsection.

In what follows the resolution of the environment is considered to be unitary, therefore, from (2),  $z_{Max} \approx 1.19\text{m}$  and  $z_{min} = z_{Max}/4 \approx 0.3\text{m}$ . Moreover, the target trajectory is modeled as a random acceleration model with  $\sigma_q^2 = 15$ .

1) *Evaluation of Hyper-Parameter  $\gamma$* : A wise selection of the hyper-parameter  $\gamma$  is crucial for the correct functioning of the herein described optimal zoom control. To this aim, a rough estimate of the typical mid-range values of  $\sigma_p$  is obtained numerically performing the bare Kalman tracking on several simulated trajectories. Whereupon, the results of (37) are evaluated for different combinations of  $\sigma_p$  and  $\gamma$ . In particular, the curves obtained with  $\sigma_p = 0.13$ ,  $\sigma_p = 0.19$ , and  $\sigma_p = 0.25$  in the interval  $\gamma \in [1, 50]$  are depicted in Fig. 13. Referring to such curves, it can be noticed that when  $\gamma$  is high the weight of  $\tilde{U}(k)$  is higher than  $I(z)$ , hence minimizing the cost function returns  $k^* = k_{Max}$  and  $z^*$  such that  $CI_k|_{k=k_{Max}} \subseteq \text{FoV}$ . Vice versa, when  $\gamma$  is low (37) returns the minimum height value allowed considering  $k \in [k_{min}, k_{Max}]$ .

To achieve the desired behavior for mid-range values of  $\sigma_p$  an intermediate value of  $\gamma$  is chosen, i.e.  $\gamma = 12$ . Such a selection allows (37) to operate in the trade-off interval in most of the cases. This is a good strategy because if  $\gamma$  is too high or low only one between the functions  $I(z)$  and  $\tilde{U}(k)$  is considered.

The fine performance of the optimization with  $\gamma = 12$  can be appreciated in Fig. 14. In particular, it can be appreciated how for low values of  $\sigma_p$  the camera gets close to the target increasing the confidence level required in (33). This happens to minimize the probability of a tracking loss due to repentine movement of the target. As expected, in the mid-range, for  $\sigma_p \in [0.13, 0.25]$ , the algorithm operates in the trade-off interval. On the other hand, in the situation in which  $\sigma_p$  is high it is clear how the camera reduces the confidence level required in (33), increasing the risk of losing the tracking, to

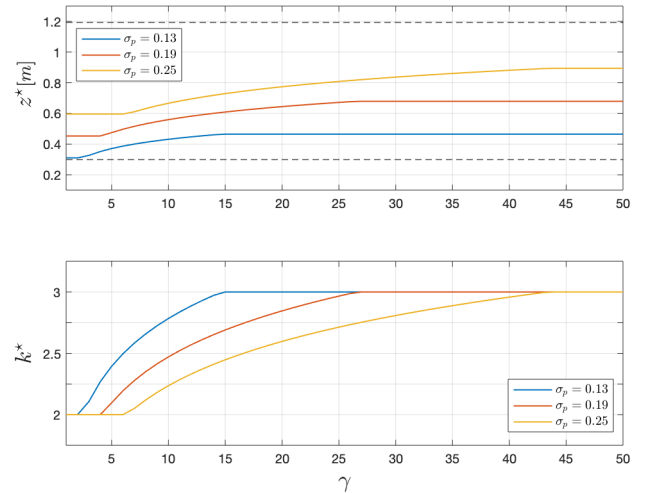


Fig. 13. Dependence of (37) on the hyper-parameter  $\gamma$ , considering different values for the standard deviation  $\sigma_p$  of the tracking error.



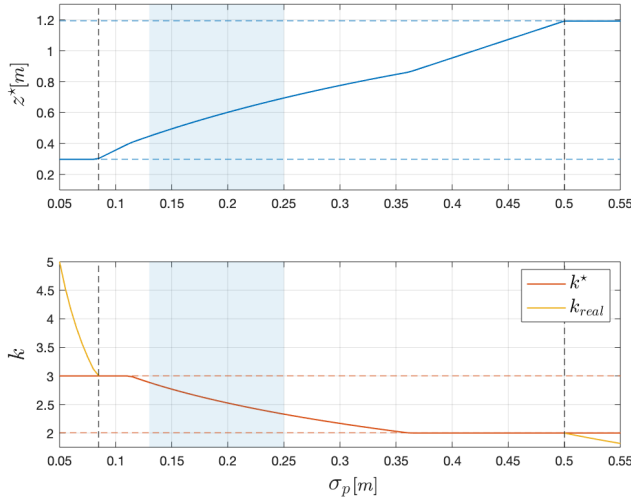


Fig. 14. Dependence of (37) on the standard deviation  $\sigma_p$  of the tracking error, considering the hyper-parameter  $\gamma = 12$ . The lines in yellow represent the coverage factor associated to the actual confidence level.

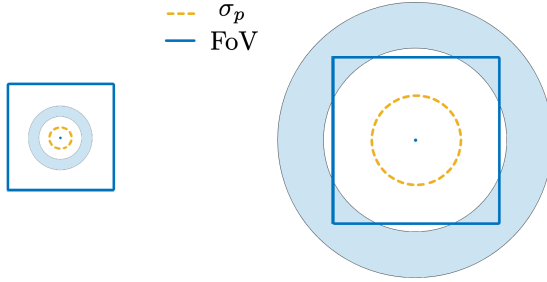


Fig. 15. Relation between the actual FoV of the camera and the tracking uncertainty  $\sigma_p$ . The highlighted area corresponds to the interval  $[2\sigma_p, 3\sigma_p]$ . **Left:**  $z_c = z_{min}$  and  $\sigma_p < 0.085$ . **Right:**  $z_c = z_{Max}$  and  $\sigma_p > 0.5$ .

get closer to the target.

It is worth noticing that for  $\sigma_p > 0.5$  and  $\sigma_p < 0.085$  further analysis is needed. Of particular interest is the situation where  $\sigma_p > 0.5$ , in which  $\nexists k \in [k_{min}, k_{Max}]$  such that (37) holds and even at  $z_{Max}$  the camera can not guarantee  $k^* = k_{min}$ . Whenever this situation occurs the camera is simply locked at the maximum height. Vice versa, in the situation in which  $\sigma_p = 0.085$  the optimal solution coincides with the boundaries of the feasible region of (37), i.e.  $z^* = z_{min}$  with  $k^* = k_{Max}$ . Therefore, for  $\sigma_p < 0.085$  the camera is locked at the minimum height, resulting in a real confidence level higher than 99.74%. To better visualize these peculiar situations two explicative examples are reported in Fig. 15.

2) *Target Tracking Analysis:* To evaluate the reliability of the Kalman filter predictions, i.e. tracking performance, the following numerical simulations are provided considering the euclidean distance between points of the real trajectory  $(x, y)$  and the estimated ones  $(\hat{x}, \hat{y})$  over the amplitude of the FoV  $d$ :

$$\frac{\|(x, y) - (\hat{x}, \hat{y})\|_2}{d}. \quad (42)$$

This value will be equal to 0 when the target is perfectly centered in the camera FoV and increases as the target moves towards the borders of the FoV. If (42) gets over 1 the target exited the FoV of the camera.

The results are evaluated on 1000 simulations of a RA trajectory, changing the filter sampling frequency  $f$  and the measurement error as percentage of FoV ( $r\%$ ). It is worth noticing that reducing the filter update frequency reduces the number of predictions. Hence, the probability of losing the target is expected to increase with the sampling time  $T$ . Similarly, increasing the measurement error (e.g. due difficult to detect events, inefficient CV detection algorithm, etc.) should reduce the prediction accuracy and therefore increase the probability of losing the tracking.

In Table VII the average and maximum values of the euclidean distance between points of the real trajectory  $(x, y)$  and the estimated ones  $(\hat{x}, \hat{y})$  over the amplitude of the FoV  $d$  are reported for multiple combinations of filter frequency  $f$  and percentage error  $r\%$ . Moreover, the values of the information lost  $I(z)$  associated to each combination can be appreciated in Table VIII.

As expected, setting the filter frequency at 10Hz and considering an efficient detection ( $r\% = 0.05$ ) yields the best results both in terms of tracking accuracy and information lost. However, running the filter at such a high frequency seems to yield too optimistic zoom values for  $r\% = 0.15$ . Indeed, at 10Hz and  $r\% = 0.15$  in order to keep  $I(z) = 0$  the effect of a particularly large single error in the measurement heavily

TABLE VII  
AVERAGE AND MAXIMUM VALUES OF (42) USING A RA TRAJECTORY MODEL.

$f$ [Hz]	Measurement Error in percentage of FoV ( $r\%$ )					
	5%		10%		15%	
	AVG	MAX	AVG	MAX	AVG	MAX
10	0.216	0.460	0.320	0.684	0.432	0.908
5	0.348	0.544	0.367	0.660	0.413	0.774
3.3	0.283	0.490	0.331	0.591	0.401	0.709
1	1.059	1.265	1.077	1.317	1.166	1.451

TABLE VIII  
AVERAGE INFORMATION LOST  $I(z)$  USING A RA TRAJECTORY MODEL.

$f$ [Hz]	Measurement Error in percentage of FoV ( $r\%$ )		
	5%	10%	15%
10	0	0	0
5	0.194	0.225	0.251
3.3	0.639	0.688	0.727
1	1.444	1.444	1.444



TABLE IX  
AVERAGE AND MAXIMUM VALUES OF (42) USING A RV TRAJECTORY MODEL.

$f$ [Hz]	Measurement Error in percentage of FoV ( $r\%$ )					
	5%		10%		15%	
	AVG	MAX	AVG	MAX	AVG	MAX
10	0.128	0.296	0.180	0.412	0.228	0.516
5	0.348	0.614	0.379	0.698	0.409	0.779
3.3	0.286	0.501	0.337	0.599	0.402	0.726
1	1.037	1.239	1.060	1.296	1.137	1.433

effects the tracking. Great results are also achieved for all the values of  $r\%$  with  $f = 5\text{Hz}$ . The tracking performance at  $f = 5\text{Hz}$  and  $f = 3.3\text{Hz}$  appears to be extremely similar. However, referring to the values in Table VIII, it can be appreciated how the average information loss running the filter at  $3.3\text{Hz}$  is considerably higher. Setting the frequency of the filter at  $1\text{Hz}$  leads to unfeasible tracking. Indeed, the average value of (42) exceeds 1 even at the maximum height, hence the tracking camera loses the target quickly.

As stated previously, the obtained results refer to the tracking of a random acceleration trajectory. However, similar results can be achieved also tracking different models. Indeed, the results associated to the tracking of a random velocity (RV) target trajectory can be appreciated in Table IX.

In conclusion, both with  $f = 10\text{Hz}$  and  $f = 5\text{Hz}$  the tracking is almost perfect and the camera never loses the target. An example of a simulated target trajectory and the corresponding tracking at  $f = 5\text{Hz}$  is depicted in Fig. 16.

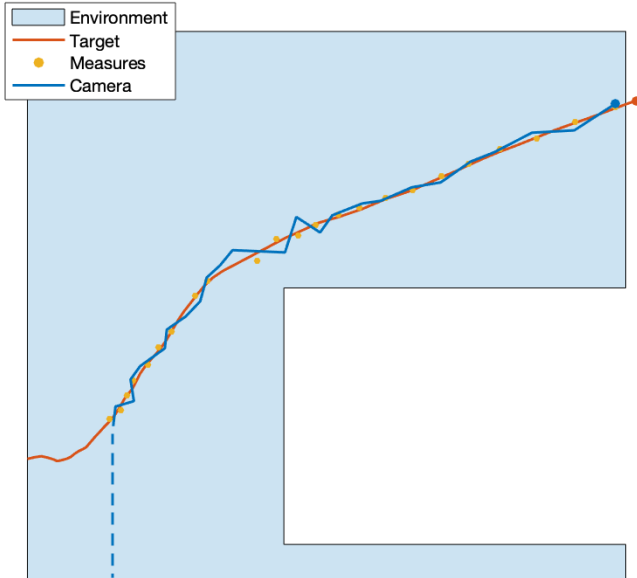


Fig. 16. Example of tracking at  $f = 5\text{Hz}$ . Camera tracking a target whose motion is modeled as a Random Acceleration (RA).

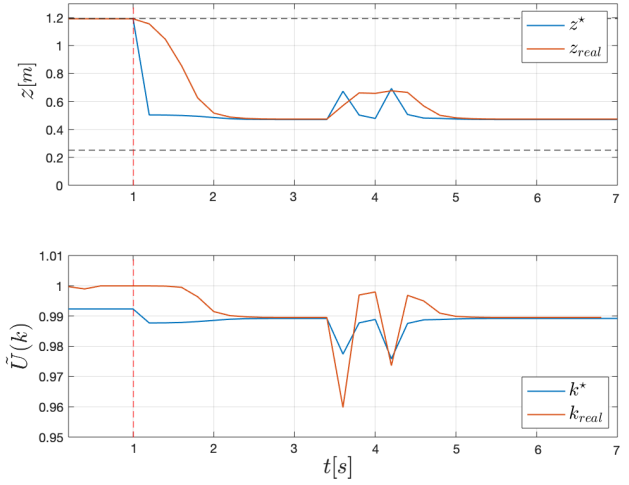


Fig. 17. Zoom control results associated to the tracking in Fig. 16. The red dashed line corresponds to the time  $k_s \cdot T$  in which the zoom control starts.

3) *Zoom Control Analysis:* As outlined in Section V-C, the actual height  $z(t)$  of the camera is extremely relevant both in the definition of the measurement noise covariance matrix  $R$  of the Kalman filter and therefore in the evaluation of the optimal zoom  $z^*(t + T)$ . Indeed, referring to the results of the optimized zoom control associated to the tracking in Fig. 16 and represented in Fig. 17, the effect of the limited vertical speed of the UAV can be appreciated. In particular, after the transitory phase of  $k_s \cdot T$  seconds at  $z = z_{Max}$ , while the reference height  $z^*$  drops instantly, the actual one  $z_{real}$  presents a smooth decrease. Clearly, this behaviour yields an increase in the confidence (i.e.  $\tilde{U}(k_{real}) > \tilde{U}(k^*)$ ) at the cost of higher information loss.

A further analysis on the robustness of the zoom control can be carried out referring to the results in Fig. 16 and Fig. 17. At  $t = 3.6\text{s}$  and  $t = 4.2\text{s}$  two measurement system failures are simulated. As expected, due to the propagation of the prediction in the Kalman filter, the tracking uncertainty  $\sigma_p$  increases and so does the optimal height  $z^*$ . Nevertheless, this time, due to the limited vertical speed of the UAV, the camera FoV contains a CI with confidence level  $\tilde{U}(k_{real}) < \tilde{U}(k^*)$ . The effect of the temporary failures in the measurement system can be appreciated also referring to the trajectories in Fig. 16. Indeed, the propagation of the prediction in the Kalman filter results in sharp changes of direction once the new measures are taken. However, despite the introduction of such errors in the measurement system, the camera can keep the target tracking thanks to the rise in height imposed by the zoom control.

## VI. CONCLUSIONS

In this work, a possible solution to the Interactive Surveillance problem using multiple UAV mounted cameras has been discussed. A BLP strategy for the patrolling problem has been formalized in the context of nearby nodes aware agents. A Bayesian-based Greedy algorithm with State Exchange has been designed and resulted in near optimal results in restricted

environment. Great results have also been achieved in real-like environments where the BLP strategy becomes unfeasible. The results obtained have demonstrated that the approach is highly scalable and self adjusting after a change in the teamsize.

As concerns the tracking task, a smart tracking algorithm with optimized zoom control has been implemented and tested on different trajectory models and considering several combination of design parameters. The Kalman filter's update frequency resulted to be the most influent parameter on the tracking performance and, in particular, with  $f > 5\text{Hz}$  the tracking system achieves almost perfect results, never losing the target. Furthermore, the robustness of the presented framework to temporary failures in the measurement system has been proven.

In light of the promising results obtained, in future works it would be worth investigating also the following aspects. As concerns the patrolling problem, the SEBS can be further extended, due to its flexibility and simplicity, with the introduction of memory. Such introduction, taking in consideration the communication network, allows the patrolling cameras to unlock the capability of making autonomous decisions based on the past collective experience, i.e. at each vertex the decisions made previously by any agent will be considered for the next one.

As a first step towards the test on real UAVs of the proposed system, the accurate design of an *ad-hoc* height controller and the implementation of a suitable control algorithm for trajectory tracking are of primary importance.

With an eye on the cost efficiency of the surveillance system, the number of cameras  $N$  and the working frequency of the Kalman filters  $f$  can be optimized in order to achieve similar results to the ones obtained in this work but with lower energy consumption and overall cheaper systems. This specific topic is of particular interest because the fine balance of the two tasks could allow the targets lost due to lower frequencies of the filters to be quickly re-detected by a nearby patrolling agent.

## APPENDIX

### A. Proof of Proposition 1

*Proof.* By the positivity and monotonicity of the exponential

$$\begin{aligned} \arg \min_{\mathbf{w}_{n_p}} \sum_{w_i \in \mathbf{w}_{n_p}} e^{\mathcal{H}(w_i)} &\equiv \arg \min_{\mathbf{w}_{n_p}} e^{\mathcal{H}(w_i)}, \forall w_i \in \mathbf{w}_{n_p} \\ &\equiv \arg \min_{\mathbf{w}_{n_p}} \mathcal{H}(w_i), \forall w_i \in \mathbf{w}_{n_p} \end{aligned}$$

Therefore, the solution of (7) minimizes  $\sum \mathcal{H}(w_i), w_i \in \mathbf{w}_{n_p}$ .

By the fact that the exponential is strictly convex

$$\begin{aligned} e^{x+\Delta x} - e^x &> e^x - e^{x-\Delta x} \\ e^{x+\Delta x} + e^{x-\Delta x} &> 2e^x \end{aligned}$$

Moreover, if  $\sum_{i=1}^{n_p} \Delta x_i \geq 0$  the latter observation can be extended to a finite sum of  $n_p$  elements as follows:

$$\sum_{i=1}^{n_p} e^{x+\Delta x_i} > \sum_{i=1}^{n_p} e^x = n_p e^x$$

Therefore, it can be observed that the sum of exponentials is minimized when all exponents are equal, i.e.  $\Delta x_i = 0, \forall i = 1, \dots, n_p$ . In this context, the condition  $\sum_{i=1}^{n_p} \Delta x_i \geq 0$  is guaranteed by the fact that the complete coverage of the environment is required, thus

$$\arg \min_{\mathbf{w}_{n_p}} \sum_{w_i \in \mathbf{w}_{n_p}} e^{\mathcal{H}(w_i)} \Rightarrow |\mathcal{H}(w_i) - \mathcal{H}(w_j)| \simeq 0$$

$\forall w_i, w_j \in \mathbf{w}_{n_p}, i \neq j$ .

We can then conclude that the solution of (7) also minimizes  $\sum_{i \neq j} |\mathcal{H}(w_i) - \mathcal{H}(w_j)|$ .  $\square$

### B. Notation Overview

An overview of the notation and symbols used in this paper is presented in Table X.

TABLE X  
NOTATION OVERVIEW

Problem Formalization	
$\mathcal{G}$	Undirected navigation graph
$\mathcal{V}$	Set of vertices
$\mathcal{E}$	Set of edges
$v_i$	Vertex in position $(x_{v_i}, y_{v_i})$
$e_{i,j}$	Edge connecting $v_i$ and $v_j$
$\mathcal{H}$	Heat Map
$X$	Camera absolute position $(x_c, y_c, z_c)$
$V$	Camera grid position
$\theta$	Half Angle of View
$d$	Half edge of the Field of View
$A(v_i)$	Set of adjacent vertices of $v_i$
$A_v(v_i)$	Set of nine vertices viewed by a camera at $z_{Max}$
$e_d$	Event detection error
$\sigma_{r\%}$	Measurement error standard deviation as $d \cdot r\%$
$\mathcal{C}$	Set of cameras $\{c_1, \dots, c_N\}$
$V_t^i$	i-th camera grid position
$V_{t+1}^i$	i-th camera movement intention
$\mathcal{P}_0, \mathcal{P}_1$	Transmission packages
$e_{tx}$	Transmission error
$\mathbb{P}$	Probability

### Coordinated Patrolling

$n_p$	Number of patrolling cameras
$\mathcal{W}$	Set of precompiled paths
$\mathbf{w}_{n_p}$	Set of $n_p$ paths $\{w_1, \dots, w_{n_p}\}$
$W_{i,j,k}$	Set of subpaths of $w_{i,j}$ of length $k$
$A_v(w_i)$	Set of vertices viewed by a camera following $w_i$
$\vdots$	$\vdots$

$\vdots$	$\vdots$
$T_G$	Spanning tree of the environment graph
$C_G^b$	Set of fundamental cycle basis
$C_G$	Set of all simple cycles
$\mathcal{I}_{v_i}$	Instantaneous idleness
$\overline{\mathcal{I}_{v_i}}$	Mean idleness of $v_i$
$\overline{I_{v_i}}$	Average peak idleness of $v_i$
$C_i$	Number of times $v_i$ has been viewed
$\overline{\mathcal{I}_G}$	Graph mean idleness
$\overline{I_G}$	Graph average peak idleness
$\overline{T_c}$	Average coverage period
$\overline{T_d}$	Average re-detection time
$G_A$	Gain associated to the movement
$S_A$	Penalty factor associated to the movement
<b>Kalman Filter</b>	
$x(t)$	Filter state
$y(t)$	Noisy measures
$v(t)$	Process noise
$w(t)$	Measurement noise
$F$	Transition matrix
$H$	Measurement matrix
$Q$	Process noise covariance matrix
$R$	Measurement noise covariance matrix
$\sigma_q^2$	Process noise variance
$\sigma_{r\%}^2$	Measurement noise variance
$T$	Filter sampling time
$f$	Filter sampling frequency
$K(t)$	Kalman gain
$\hat{x}(t+T t)$	One-step-ahead prediction
$P(t+T t)$	Prediction covariance matrix
$N_k$	Tracking loss threshold
<b>Optimal Zoom</b>	
$I(z)$	Information loss function
$e_p$	Tracking error
$\sigma_p$	Tracking error standard deviation
$k$	Coverage factor
$CI_k$	Confidence Interval w.r.t. the coverage factor $k$
$\mathcal{Z}$	Normalized tracking error
$U(k)$	Confidence level w.r.t. the coverage factor $k$
$\tilde{U}(k)$	Aproximized confidence level w.r.t. the coverage factor $k$
$\gamma$	Hyper-parameter for trade-off managing
$k_s T$	Zoom control waiting time
$\vdots$	$\vdots$

$\vdots$	$\vdots$
<b>Zoom Control</b>	
$t_{s,5\%}$	Settling time
$M_p$	Percentage overshoot
$\delta$	Damping factor
$\omega_{gc}$	Gain crossover frequency
$\varphi_m$	Phase margin
$1/T_L$	Real derivative cutoff frequency
$k_w$	Anti-windup gain
$k_p$	Proportional gain
$k_i$	Integral gain
$k_d$	Derivative gain

## REFERENCES

- [1] JA Gonçalves and R Henriques. Uav photogrammetry for topographic monitoring of coastal areas. *ISPRS Journal of Photogrammetry and Remote Sensing*, 104:101–111, 2015.
- [2] Sonia Waharte and Niki Trigoni. Supporting search and rescue operations with uavs. In *2010 International Conference on Emerging Security Technologies*, pages 142–147. IEEE, 2010.
- [3] Markus Quaritsch, Karin Kruggl, Daniel Wischounig-Strucl, Subhabrata Bhattacharya, Mubarak Shah, and Bernhard Rinner. Networked uavs as aerial sensor network for disaster management applications. *e & i Elektrotechnik und Informationstechnik*, 127(3):56–63, 2010.
- [4] Sunflower-labs: Autonomous Home Security Drone. <https://www.sunflower-labs.com/>. Accessed: 2020/03/04.
- [5] Mac Schwager, Brian J Julian, Michael Angermann, and Daniela Rus. Eyes in the sky: Decentralized control for the deployment of robotic camera networks. *Proceedings of the IEEE*, 99(9):1541–1561, 2011.
- [6] David Portugal and Rui P Rocha. Distributed multi-robot patrol: A scalable and fault-tolerant framework. *Robotics and Autonomous Systems*, 61(12):1572–1587, 2013.
- [7] Maria Valera and Sergio A Velastin. Intelligent distributed surveillance systems: a review. *IEE Proceedings-Vision, Image and Signal Processing*, 152(2):192–204, 2005.
- [8] Brendan Tran Morris and Mohan Manubhai Trivedi. A survey of vision-based trajectory learning and analysis for surveillance. *IEEE transactions on circuits and systems for video technology*, 18(8):1114–1127, 2008.
- [9] Roberto Vezzani, Davide Baltieri, and Rita Cucchiara. People reidentification in surveillance and forensics: A survey. *ACM Computing Surveys (CSUR)*, 46(2):1–37, 2013.
- [10] Thomas Winkler and Bernhard Rinner. Security and privacy protection in visual sensor networks: A survey. *ACM Computing Surveys (CSUR)*, 47(1):1–42, 2014.
- [11] Juan C SanMiguel, Christian Micheloni, Karen Shoop, Gian Luca Foresti, and Andrea Cavallaro. Self-reconfigurable smart camera networks. *Computer*, 47(5):67–73, 2014.
- [12] Claudio Piciarelli, Lukas Esterle, Asif Khan, Bernhard Rinner, and Gian Luca Foresti. Dynamic reconfiguration in camera networks: A short survey. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(5):965–977, 2015.
- [13] Prabhu Natarajan, Pradeep K Atrey, and Mohan Kankanhalli. Multi-camera coordination and control in surveillance systems: A survey. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 11(4):1–30, 2015.
- [14] Li Huang, MengChu Zhou, Kuangrong Hao, and Edwin Hou. A survey of multi-robot regular and adversarial patrolling. *IEEE/CAA Journal of Automatica Sinica*, 6(4):894–903, 2019.
- [15] Nicola Basilico, Nicola Gatti, Thomas Rossi, Sofia Ceppi, and Francesco Amigoni. Extending algorithms for mobile robot patrolling in the presence of adversaries to more realistic settings. In *2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, volume 2, pages 557–564. IEEE, 2009.

- [16] David Portugal and Rui Rocha. A survey on multi-robot patrolling algorithms. In *Doctoral Conference on Computing, Electrical and Industrial Systems*, pages 139–146. Springer, 2011.
- [17] Fabio Pasqualetti, Antonio Franchi, and Francesco Bullo. On cooperative patrolling: Optimal trajectories, complexity analysis, and approximation algorithms. *IEEE Transactions on Robotics*, 28(3):592–606, 2012.
- [18] SY Chen. Kalman filter for robot vision: a survey. *IEEE Transactions on Industrial Electronics*, 59(11):4409–4420, 2011.
- [19] Zhongmin Li and Haochen Wu. A survey of maneuvering target tracking using kalman filter. In *2015 4th International Conference on Mechatronics, Materials, Chemistry and Computer Engineering*. Atlantis Press, 2015.
- [20] Eric Sommerlade and Ian Reid. Probabilistic surveillance with multiple active cameras. In *2010 IEEE International Conference on Robotics and Automation*, pages 440–445. IEEE, 2010.
- [21] Aaron Mavrinac and Xiang Chen. Modeling coverage in camera networks: A survey. *International journal of computer vision*, 101(1):205–226, 2013.
- [22] Uğur Murat Erdem and Stan Sclaroff. Automated camera layout to satisfy task-specific and floor plan-specific coverage requirements. *Computer Vision and Image Understanding*, 103(3):156–169, 2006.
- [23] SZ DJI Technology Co. <https://www.dji.com/>. Accessed: 2020/03/04.
- [24] Parrot SA. <https://www.parrot.com/>. Accessed: 2020/03/04.
- [25] Tse-yun Feng. A survey of interconnection networks. *Computer*, 14(12):12–27, 1981.
- [26] Luca Iocchi, Luca Marchetti, and Daniele Nardi. Multi-robot patrolling with coordinated behaviours in realistic environments. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2796–2801. IEEE, 2011.
- [27] Bertil Ekstrand. Some aspects on filter design for target tracking. *Journal of Control Science and Engineering*, 2012, 2012.
- [28] Kenshi Saho. Kalman filter for moving object tracking: Performance analysis and filter design. *Kalman Filters-Theory for Advanced Applications*, 2017.