Nikolaos Kariotoglou
Automatic Control Laboratory,
Department of Electrical and
Information Engineering,
ETH Zurich,
Physikstrasse 3,
Zurich 8092, Switzerland
e-mail: karioto@control.ee.ethz.ch

Davide M. Raimondo
Identification and Control of
Dynamic Systems Laboratory,
Dipartimento di Ingegneria Industriale
e dell'Informazione,
Università degli Studi di Pavia,
Via Ferrata 1,
Pavia 27100, Italy
e-mail: davide.raimondo@unipv.it

Sean J. Summers
Automatic Control Laboratory,
Department of Electrical and
Information Engineering,
ETH Zurich,
Physikstrasse 3,
Zurich 8092, Switzerland
e-mail: ssummers@control.ee.ethz.ch

John Lygeros
Automatic Control Laboratory,
Department of Electrical and
Information Engineering,
ETH Zurich,
Physikstrasse 3,
Zurich 8092, Switzerland
e-mail: jlygeros@control.ee.ethz.ch

Multi-Agent Autonomous Surveillance: A Framework Based on Stochastic Reachability and Hierarchical Task Allocation

We develop and implement a framework to address autonomous surveillance problems with a collection of pan-tilt (PT) cameras. Using tools from stochastic reachability with random sets, we formulate the problems of target acquisition, target tracking, and acquisition while tracking as reach-avoid dynamic programs for Markov decision processes (MDPs). It is well known that solution methods for MDP problems based on dynamic programming (DP), implemented by state space gridding, suffer from the curse of dimensionality. This becomes a major limitation when one considers a network of PT cameras. To deal with larger problems we propose a hierarchical task allocation mechanism that allows cameras to calculate reach-avoid objectives independently while achieving tasks collectively. We evaluate the proposed algorithms experimentally on a setup involving industrial PT cameras and mobile robots as targets. [DOI: 10.1115/1.4028589]

1 Introduction

As a consequence of technological advancements in the field of intelligent embedded systems, surveillance systems are increasing in number and complexity. Densely populated areas such as transportation hubs are nowadays equipped with camera systems used to detect potential threats. The growing number of surveillance devices in such areas increases the amount of gathered information and makes supervision tasks almost impossible for human operators. Motivated by this fact, several communities are working on the automation of surveillance tasks.

Camera surveillance systems typically consist of a space to be surveilled and a collection of pan-tilt-zoom (PTZ) and static cameras that are tracking objects while obtaining high resolution images. A general overview of existing methods in the context of intelligent distributed systems and the particular problem of target tracking can be found in the reviews of Refs. [1–4]. Most approaches address the surveillance problem using a combination of static and PTZ cameras where feedback from static cameras is used to position the moving ones (see, for example, Refs. [5–10]). To reduce the complexity on large camera networks, several of these works focus on distributed implementations and in some cases relax the requirement of static cameras (see, for example, [10–12]).

Several autonomous surveillance methods suggested in literature incorporate the dynamic behavior of targets in the design of PTZ trajectories. A key difference between methods is the model chosen to describe objects, ranging from general stochastic models to simple linear ones. While approaches based on simple models can be quite effective, the approximation of the stochastic movement of the targets with mean and variance only can result in poor performance (see, for example, Ref. [13] for a tracking problem with crossroads). On the other hand, more complex models lead to computationally demanding algorithms revealing a trade-off between complexity and accuracy. In Refs. [14] and [15], the authors use different filtering techniques to predict the future behavior of the system and optimize camera movements using model predictive control. Several state-of-the-art algorithms (see, for example, Refs. [16,17]) are based on interacting multiple model (IMM) filtering which provides a very powerful framework combining different model complexities. A general limitation of IMM filtering is the assumed knowledge of the transition probability from one model to another. In the context of objects that are actively trying to avoid detection, this assumption may be strong. An approach using more general stochastic models for the target movement is presented in Ref. [18], where the objective is to maximize the number of observed targets while guaranteeing a predefined resolution. In Ref. [19], the authors propose a model based on temporal logic for synthesizing control strategies for camera networks in the form of objective specifications.

Camera surveillance problems can also be viewed in the context of pursuit-evasion games [20–23] where the evader actively

Contributed by the Dynamic Systems Division of ASME for publication in the Journal of Dynamic Systems, Measurement, and Control. Manuscript received January 29, 2014; final manuscript received September 15, 2014; published online October 21, 2014. Assoc. Editor: Dejan Milutinovic.

tries to avoid detection by the pursuer. Even though surveillance methods based on game theory are very interesting, they can result in nonrobust behavior whenever the evader strategy or the environment differ from the assumed one (see Ref. [24]). Related work includes optimal perimeter patrolling, addressed in Refs. [25–27] and design and control of surveillance unmanned aerial vehicle swarms, similar to the work of Ref. [28], where area monitoring is treated as a static covering problem and cameras are attached to flying quadrotors.

In this work, we pose surveillance problems as reach-avoid problems similar to [29–31] and propose tractable methods to approximate their solution. Our goal is to address the problem with a mathematical framework that can be extended to the multicamera case. We model camera dynamics as MDPs and develop a unified framework for dealing with problems of target tracking, target acquisition and acquisition while tracking. While there exist many works on autonomous tracking, less papers focus on the problem of acquiring targets currently not in the field of view (FOV) of the cameras (see, for example, Ref. [19]). Compared to other methods designed for object tracking or acquisition, our approach has the following features: We address dynamic tracking and the less studied problem of acquisition using the same mathematical formulation, we deal with uncertainty in a natural way even in situations where the prediction of object trajectories follow complicated distributions and we utilize the values of the optimal cost function (they correspond to success probabilities) to design high level tasks. Using reachability analysis in the spirit of Refs. [29] and [30], we cast different surveillance problems as optimal control problems for MDPs. In particular, we deal with the selection of PT angles for PTZ surveillance cameras such that a finite horizon reach-avoid problem is solved. To express tracking and acquiring objectives as reach-avoid problems we utilize a notion of stochastic sets and covering functions as described in Ref. [31]. In contrast to Ref. [31], where the authors prove the validity of the mathematical framework, we focus on practicality and adopt the framework to the autonomous surveillance problem. To achieve this we redefine all abstract sets and processes and provide illustrative examples clarifying the approach. Moreover, we present an extension of the framework to multicamera setups where several PT cameras are responsible for monitoring the same area. The proposed algorithm is based on the fact that each surveillance camera can calculate its optimal trajectory independently of others. We provide a systematic way to allocate surveillance tasks among cameras such that global objectives are achieved by the combined system. Although we develop the task allocation system for the particular problem of camera surveillance, it would be straightforward to extend the idea to other reachability formulations with multiple agents. Our experimental setup is similar to Refs. [5–9] since we are using a global eye camera, but our framework can be used with any type of global position sensor. We dedicate a complete section to elaborate on the implementation details of the proposed approach and carry out a series of experiments to analyze the overall performance. In this way we want to illustrate the applicability of the method to similar setups and provide the interested reader with a detailed guide on how to address problems of multicamera surveillance using a versatile framework based on stochastic reachability.

Preliminary work on autonomous surveillance with stochastic reachability and task allocation has been presented in Refs. [32] and [33]. Extending the work in Ref. [32], we provide intuition and details of all the different processes involved in the formulation of a stochastic reachability problem with random sets. Moreover, in the experimental validation of the proposed methods, we go beyond the initially documented results and evaluate the performance of the system under continuous operation using two cameras. In terms of task allocation, we analyze the method proposed in Ref. [33] in depth and carry out a complexity analysis to highlight the reduction achieved by transforming a single problem of exponential complexity to an exponential number of small problems. As part of the extended experimental validation we analyze the effect of some key system variables in the overall performance of the task allocation system.

The rest of this paper is organized as follows: In Sec. 2, we revisit the definition of stochastic reachability for MDPs with fixed sets and describe the type of reach-avoid problems that we are interested in together with their extension in the random set case. We then proceed to Sec. 3, where we define the relevant sets and maps in the context of autonomous surveillance and provide an algorithm to calculate the reach-avoid objectives for a single camera. In Sec. 4, we present the extension of the framework to the multicamera case via an approximation technique and discuss the computational benefits of the proposed method. We conclude with an algorithm outlining the proposed task allocation system and a discussion comparing our method's characteristics to the existing literature. In Sec. 5, we discuss the implementation issues related to the presented algorithms and provide an experimental study for the single- and multicamera cases illustrated on a two-camera two-evader system.

2 Stochastic Reachability

2.1 Stochastic Reachability With Fixed Sets. For the benefit of users not familiar with reachability theory, we provide a brief overview of stochastic reachability for MDPs. The material in this section is based on Refs. [29–31]. We consider finite horizon reachability problems for discrete time systems and inherit the mathematical framework introduced in Ref. [29] used to formulate the safety reachability problem as a stochastic optimal control problem. The safety problem aims to characterize the probability that the trajectory of the system, starting at a certain initial condition, stays in the safe region of the state space over a predefined time horizon $\mathcal{T} = \{0, \ldots, N-1\}$. Consider a state space $\mathcal{X}$, a control space $\mathcal{U}$, a safe set $K' \subseteq \mathcal{X}$, and a feedback policy $\mu : \mathcal{X} \to \mathcal{U}$ for an MDP $\{x_k\}_{k \in \mathcal{T}} \subseteq \mathcal{X}$ evolving according to a controlled stochastic kernel $Q : \mathcal{B}(\mathcal{X}) \times \mathcal{X} \times \mathcal{U} \to [0,1]$. Note that throughout the paper $\mathcal{B}(\cdot)$ denotes the Borel σ-algebra of the argument set. For a given initial state x_0 and state feedback policy μ, the authors in Ref. [29] show that the probability $p^{\mu}_{x_0}(K')$ of staying in K' can be expressed as the expected value of a product of indicator functions $p^{\mu}_{x_0}(K') = \mathbb{E}^{\mu}_{x_0}[\prod_{k=0}^{N-1} \mathbb{1}_{K'}(x_k)]$. Furthermore, they show that $p^{\mu}_{x_0}(K')$ can be maximized over the set of admissible feedback policies by solving a DP recursion. Extending the results in Ref. [29], the authors in Ref. [30] show that the probability of reaching a set $K \subseteq \mathcal{X}$ while staying in another set $K' \subseteq \mathcal{X}$ under a policy μ can be expressed as the expected value of a sum-multiplicative function comprising indicator functions

$$r^{\mu}_{x_0}(K, K') := \mathbb{E}^{\mu}_{x_0}\left[\sum_{k=0}^{N-1}\left(\prod_{i=0}^{k-1} \mathbb{1}_{K'\setminus K}(x_i)\right)\mathbb{1}_K(x_k)\right] \tag{1}$$

Note that if the reach-avoid objective is achieved at some time $\bar{k} < N-1$, the value of $\sum_{k=0}^{N-1}\left(\prod_{i=0}^{k-1} \mathbb{1}_{K'\setminus K}(x_i)\right)\mathbb{1}_K(x_k)$ is set to 1 for all times between $\bar{k}$ and $N-1$. In this way, the cost function expresses the minimum time reach-avoid probability. To provide some insight to the cost function in Eq. (1), which subsumes the cost function of the original safety problem, we consider the case of three equally probable trajectories initialized at x_0, denoted by the "square," "circle," and "diamond" trajectories in Fig. 1. Figure 2 depicts the value of the indicator functions $\mathbb{1}_K(x_k), \mathbb{1}_{K'}(x_k)$ in all cases. Note that $\mathbb{1}_{K'\setminus K}(x_k) = \mathbb{1}_{K'}(x_k) - \mathbb{1}_K(x_k)$. For the square trajectory in Fig. 2(*a*) we have that $\sum_{k=0}^{5}(\prod_{i=0}^{k-1} \mathbb{1}_{K'\setminus K}(x_i))\mathbb{1}_K(x_k) = 0$ since $\mathbb{1}_K(x_k) = 0$ for all k. For the circle trajectory in Fig. 2(*b*) we have that $\sum_{k=0}^{5}(\prod_{i=0}^{k-1} \mathbb{1}_{K'\setminus K}(x_i))\mathbb{1}_K(x_k) = 1$ since the objective is achieved at $k=4$. Finally, for the diamond trajectory in Fig. 2(*c*), we have that $\sum_{k=0}^{5}(\prod_{i=0}^{k-1} \mathbb{1}_{K'\setminus K}(x_i))\mathbb{1}_K(x_k) = 0$ since $\mathbb{1}_{K'\setminus K}(x_1) = 0$. Assuming the trajectories in Fig. 1 are the only possible ones and are all equally probable, we have that

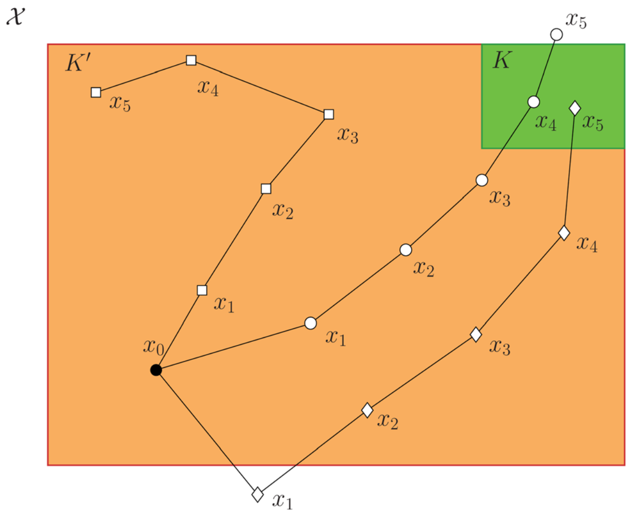


Fig. 1 An example of three system trajectories starting at x_0. The reach-avoid probability is the expected value of a sum-multiplicative cost of indicator functions over the different trajectories.

$$\mathbb{E}^{\mu}_{x_0}\left[\sum_{k=0}^{N-1}\left(\prod_{i=0}^{k-1}\mathbb{1}_{K'\setminus K}(x_i)\right)\mathbb{1}_K(x_k)\right] = \frac{1}{3}\cdot 0+\frac{1}{3}\cdot 1+\frac{1}{3}\cdot 0=\frac{1}{3} \quad (2)$$

In principle, given the policy μ, the transition kernel Q and the initial state x_0, the reach-avoid probability $r^{\mu}_{x_0}(K,K')$ can be calculated via the expectation integral. Calculating this integral is (in general) computationally prohibitive and the authors in Ref. [30] use the same machinery as in Ref. [29] to prove that $r^{\mu}_{x_0}(K,K')$ can be evaluated and maximized over the set of admissible feedback policies via a dynamic programming recursion of the form

$$\begin{aligned}
V^*_k(x) &= \sup_{u\in\mathcal{U}}\left\{\mathbb{1}_K(x)+\mathbb{1}_{K'\setminus K}(x)\int_{\mathcal{X}}V^*_{k+1}(z)Q(dz|x,u)\right\}\\
V^*_{N-1}(x) &= \mathbb{1}_K(x).
\end{aligned} \quad (3)$$

Under generic assumptions, this allows one to compute an optimal feedback policy at each stage k using the value functions $V^*_k:\mathcal{X}\rightarrow[0,1]$

$$\mu^*_k(x) = \arg\sup_{u\in\mathcal{U}}\left\{\mathbb{1}_K(x)+\mathbb{1}_{K'\setminus K}(x)\int_{\mathcal{X}}V^*_{k+1}(z)Q(dz|x,u)\right\} \quad (4)$$

2.2 Stochastic Reachability With Random Sets. The mathematical framework used to formulate surveillance tasks is based on a slight extension of the reach-avoid problem in Ref. [30] to the case where both target and safe sets are varying according to a stochastic process (initially used in Ref. [32] and formally analyzed in Ref. [31]). Consider an additional space $\mathcal{Y}$ and a stochastic process $\{y_k\}_{k\in\mathcal{T}}\subseteq\mathcal{Y}$ evolving according to a collection of stochastic kernels $G_k:\mathcal{B}(\mathcal{Y})\times\mathcal{Y}\rightarrow[0,1]$. Moreover, consider two Borel measurable set-valued functions $\gamma:\mathcal{Y}\rightarrow\mathcal{B}(\mathcal{X}), \gamma':\mathcal{Y}\rightarrow\mathcal{B}(\mathcal{X})$ such that $\gamma(y)\subseteq\gamma'(y), \forall y\in\mathcal{Y}$. Define $K_k:=\gamma(y_k)$ and $K'_k:=\gamma'(y_k)$ as stochastic target and safe sets, respectively. To facilitate computations on stochastic sets we define the following covering functions for the safe and target sets, respectively,

$$\begin{aligned}
p_{K_k}(x) &:= \int_{\mathcal{Y}}\mathbb{1}_{\gamma(y_k)}(x)G_k(dy_k|y_k)=\mathbb{E}\left[\mathbb{1}_{\gamma(y_k)}(x)\right]\\
p_{K'_k}(x) &:= \int_{\mathcal{Y}}\mathbb{1}_{\gamma'(y_k)}(x)G_k(dy_k|y_k)=\mathbb{E}\left[\mathbb{1}_{\gamma'(y_k)}(x)\right]
\end{aligned} \quad (5)$$

where expectations are taken with respect to G_k. The covering function of a stochastic set is analogous to the indicator function for a deterministic set. That is, given the distribution y_k and a state $x\in\mathcal{X}, p_{K_k}(x)$, and $p_{K'_k}(x)$ are expressing the probability that x belongs to K_k or K'_k with respect to the distribution of the sets. Since, by definition, $\gamma(y)\subseteq\gamma'(y), \forall y\in\mathcal{Y}$, it is easy to see that $p_{K'_k\setminus K_k}(x)=p_{K'_k}(x)-p_{K_k}(x)$ for every stage $k\in\mathcal{T}$. Using the defined covering functions, reach-avoid problems with random sets are concerned with evaluating and maximizing the probability that an MDP $\{x_k\}_{k\in\mathcal{T}}$ will reach the (stochastic) target set K_k at some time in the horizon while remaining inside K'_k at all prior times. The maximization is carried out over the set of admissible feedback policies for the MDP. The safe and target sets are evolving according to the stochastic process $\{y_k\}_{k\in\mathcal{T}}$ and as a result, the reach-avoid probability varies according to $\{y_k\}_{k\in\mathcal{T}}$. The authors in Ref. [31] show that given μ, x_0, y_0, the reach-avoid probability can be expressed by the following sum-multiplicative function, very similar to the one in Eq. (1)

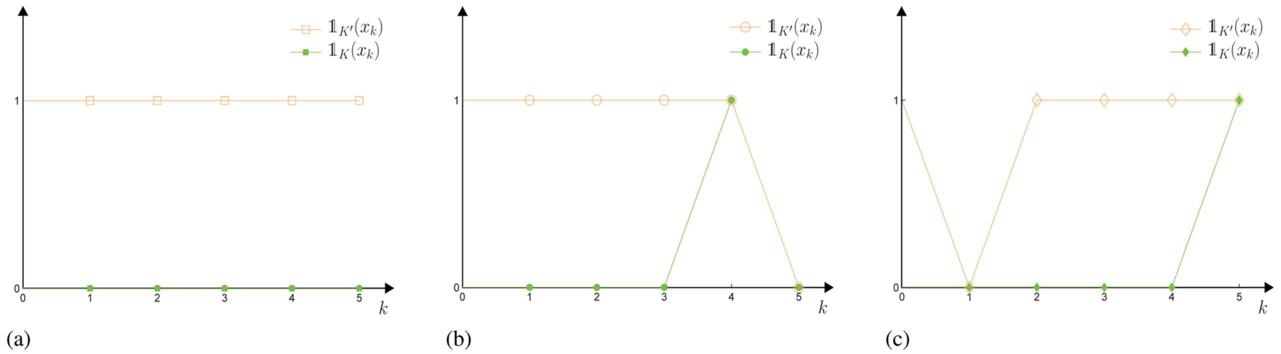
$$r^{\mu}_{x_0,y_0}(\gamma,\gamma') = \mathbb{E}^{\mu}_{x_0}\left[\sum_{k=0}^{N-1}\left(\prod_{i=0}^{k-1}p_{K'_i\setminus K_i}(x_i)\right)p_{K_k}(x_k)\right] \quad (6)$$

Moreover, it was shown that one can maximize this probability over the set of admissible feedback policies by solving the following recursion for the value function $V^*_0(x)$ where $V^*_k:\mathcal{X}\rightarrow[0,1]$

$$\begin{aligned}
V^*_k(x) &= \sup_{u\in\mathcal{U}}\left\{p_{K_k}(x)+p_{K'_k\setminus K_k}(x)\int_{\mathcal{X}}V^*_{k+1}(z)Q(dz|x,u)\right\}\\
V^*_{N-1}(x) &= p_{K_{N-1}}(x)
\end{aligned} \quad (7)$$

Under generic assumptions, the optimal feedback policy over $\mathcal{X}$ is retrieved by

$$\mu^*_k(x) = \arg\sup_{u\in\mathcal{U}}\left\{p_{K_k}(x)+p_{K'_k\setminus K_k}(x)\int_{\mathcal{X}}V^*_{k+1}(z)Q(dz|x,u)\right\} \quad (8)$$



(a) (b) (c)

Fig. 2 Reach-avoid indicator function values for different state trajectories. (*a*) Indicator functions for the square trajectory in Fig. 1. (*b*) Indicator functions for the circle trajectory in Fig. 1, and (*c*) indicator functions for the diamond trajectory in Fig. 1.

2.3 Reach-Avoid With Random Sets for Autonomous Surveillance.

We focus on three types of reach-avoid objectives, defined specifically for autonomous surveillance: target tracking, target acquisition, and target acquisition while tracking introduced in Ref. [32]. In target tracking, we are interested in keeping evaders within the FOV of the surveilling camera for every step of a predefined time horizon. In target acquisition, the goal is to bring evaders within the FOV as soon as possible within the time horizon. Acquisition while tracking is a combination of the above objectives and relevant only when multiple evaders are present. Since the movement of evaders determines the safe and target sets for the cameras, it is natural to adopt the reachability framework with random sets to express reach-avoid objectives.

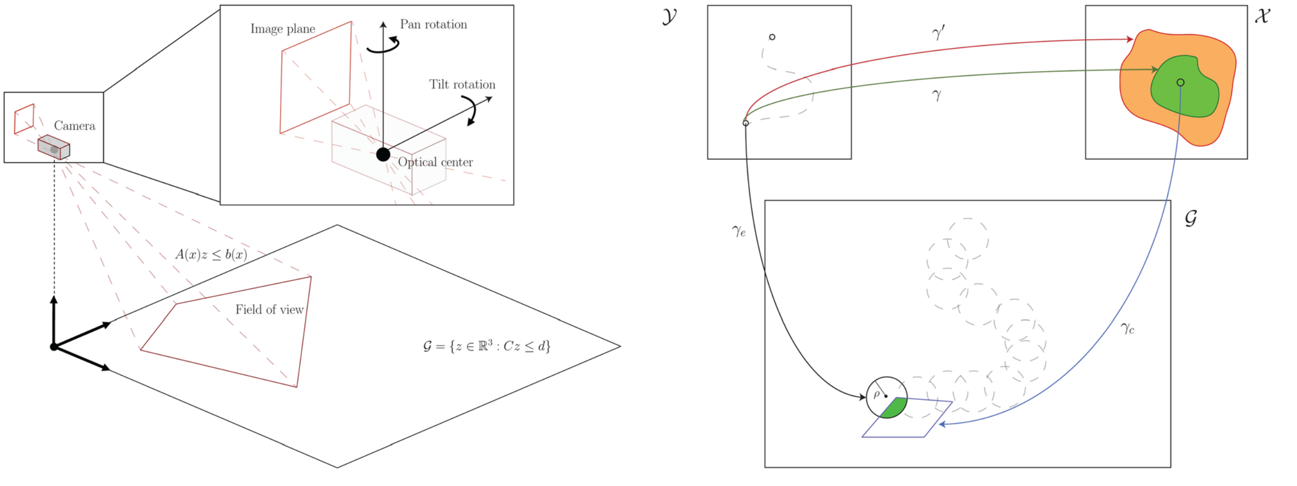
We consider stochasticity coming from two different sources, namely the evader movement and the inaccuracy in the operation of the camera. We consider the camera movement to be nearly deterministic (assuming that the low-level controllers are successfully tracking motor angles) and assign low probabilities in transitioning to a wrong PT position. As a result, the camera state is a random variable whose distribution is almost entirely concentrated around the desired PT angle. For the evaders, we assume the existence of a mathematical model able to characterize every step of their distribution. In this way, the evader state becomes a random variable with known distribution over the surveillance horizon. Naturally, the estimate of the distribution becomes worse as the horizon length increases. Both camera and evader dynamics are modeled as stochastic processes in the spaces $\mathcal{X} \subseteq \mathbb{R}^n$ and $\mathcal{Y} \subseteq \mathbb{R}^m$, respectively. The surveillance tasks of interest are formulated as finite horizon problems for a sequence of N discrete time steps $\mathcal{T} = \{0, \ldots, N-1\}$. The camera stochastic process $\{x_k\}_{k \in \mathcal{T}} \subseteq \mathcal{X}$ is assumed to be Markovian. We denote by $\mathcal{U}$ a compact set containing the possible control actions of the camera process and define a controlled transition probability function $Q : \mathcal{B}(\mathcal{X}) \times \mathcal{X} \times \mathcal{U} \to [0,1]$ which assigns a probability measure $Q(\cdot|x,u)$ on $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$ for every pair $(x,u) \in \mathcal{X} \times \mathcal{U}$. Given a camera state x and control input u, the transition probability function Q uniquely characterizes the distribution of the corresponding next state. The evader is modeled by a stochastic process $\{y_k\}_{k \in \mathcal{T}} \subseteq \mathcal{Y}$ the distribution of which is approximated, along the lines of Ref. [31], by a sequence of probability distributions $G_k : \mathcal{B}(\mathcal{Y}) \to [0,1]$. This type of approximation neglects time dependence in the distribution of the random variables y_k but has enormous computational advantages (in the calculation of covering functions) over using time-coupled probability distributions.

3 Autonomous Surveillance

To formulate surveillance tasks in the stochastic reachability framework we first describe the models used for the camera, the evader and their interaction space (ground plane/surveillance space) and then provide an explicit expression to the functions γ and γ' for the problems of tracking, acquiring and acquiring while tracking. We approximate the evaders by flat circles of radius ρ moving on an elevated plane $\mathcal{G}$. For a fixed zoom level, the FOV of a camera on $\mathcal{G}$ is completely determined by its PT angles. Extensions to evaders moving in a three-dimensional space and PTZ cameras are conceptually straightforward but computationally much more demanding.

We model the camera dynamics as a MDP of PT angles $\{x_k\}_{k \in \mathcal{T}}, x_k \in \mathcal{X} \subseteq \mathbb{R}^2$, defined by a kernel Q such that $x_{k+1} \sim Q(dx_{k+1}|x_k, u_k)$. The camera process $\{x_k\}_{k \in \mathcal{T}}$ can be mapped to a set-valued process on $\mathcal{B}(\mathcal{G})$ by the FOV map $\gamma_c : \mathcal{X} \to \mathcal{B}(\mathcal{G})$ where $x \mapsto \gamma_c(x) = \{z \in \mathbb{R}^3 : A(x)z \le b(x) \wedge Cz \le d\}$. The elements $A(x)$, $b(x)$ define the camera FOV as a polytope in $\mathbb{R}^3$ and are different for each PT configuration while the ground plane $\mathcal{G}$ is defined by C and d and remains constant (see Fig. 3(*a*) and Ref. [15] for more details). The intersection of the camera FOV with $\mathcal{G}$ completely determines the area of the surveillance space visible to the camera for each PT angle configuration. Note that since $\mathcal{G}$ is parallel to the ground, it is a subset of $\mathbb{R}^2$ with a constant shift, implying that one dimension of $\mathcal{G}$ is constant. Every evader is modeled as a process $\{y_k\}_{k \in \mathcal{T}}, y_k \in \mathcal{Y} \subseteq \mathbb{R}^3$ defined through a collection of probability distributions $G_k : \mathcal{B}(\mathcal{Y}) \to [0,1]$ such that $y_k \sim G_k(dy_k)$. The process $\{y_k\}_{k \in \mathcal{T}}$ can be mapped to a set-valued process on $\mathcal{B}(\mathcal{G})$ by the measurable map $\gamma_e : \mathcal{Y} \to \mathcal{B}(\mathcal{G})$ where $y \mapsto \gamma_e(y) = \{g \in \mathcal{G} : \| g - y \|_2 \le \rho\}$, representing all the elements of the ground plane that are within the evader circle.

The different process spaces and maps are summarized in Fig. 3. The reachability objectives that we consider are posed on the space of PT angles $\mathcal{X}$ where target and safe sets are related to whether the corresponding FOV intersects the evader process or not on the ground plane $\mathcal{G}$. In Fig. 3(*b*), we illustrate an example realization of the stochastic process y on $\mathcal{Y}$, mapped to $\mathcal{G}$ via γ_e. The safe and target sets in $\mathcal{B}(\mathcal{X})$ are calculated at each time step by considering whether $\gamma_e(y)$ and $\gamma_c(x)$ intersect or not. The objective in the example of Fig. 3(*b*) is to capture the entire evader while moving in states where the evader is (at least) partially seen. We consider $K = \gamma(y) = \{x \in \mathcal{X} : \gamma_e(y) \subseteq \gamma_c(x)\}$, $K' = \gamma'(y) = \{x \in \mathcal{X} : \gamma_c(x) \cap \gamma_e(y) \neq \emptyset\}$ where γ and γ' depend on the maps γ_e, γ_c. Note that this objective would be satisfied at the time k at which $\gamma_e(y) \subseteq \gamma_c(x)$. In what follows we encode the



(a)

(b)

Fig. 3 Maps between spaces. (*a*) The FOV projection on the ground plane. (*b*) Visualization of the transformation maps between the different spaces. The dotted line on space $\mathcal{Y}$ corresponds to a realization of the evader process while the dotted circles on $\mathcal{G}$ are the mapping through γ_e.

particular surveillance objectives that we are interested in and precisely define safe and target set covering functions on an example with two robotic evaders and a single PT camera. Note that to simplify the notation we have dropped the dependence of time k in the definition of γ and γ' (both x, y are different at each time k) and explicitly state if any map changes at some $k \in \mathcal{T}$.

3.1 Tracking. For the objective of target tracking we assume that the target is intersecting with the FOV at time $k=0$. The purpose is then to maximize the probability that the evader intersects the FOV over the whole horizon $\mathcal{T}$. To encode this objective, the safe set is defined as $\gamma'(y) = \{x \in \mathcal{X} : \gamma_{\mathrm{c}}(x) \cap \gamma_{\mathrm{e}}(y) \neq \emptyset\}$ for all time k to make sure that the camera moves in states where the evader is (at least partially) seen. The target set is defined as $\gamma(y) = \{x \in \mathcal{X} : \gamma_{\mathrm{c}}(x) \cap \gamma_{\mathrm{e}}(y) \neq \emptyset\}, \forall y \in \mathcal{Y}$ at time $k=N-1$ and is left empty $(\gamma(y) = \emptyset, \forall y \in \mathcal{Y})$ at all previous time to make sure that tracking is complete at the end of the horizon $\mathcal{T}$. The functions γ and γ' are used to determine the covering functions (5) at each time step and solve the dynamic recursion in Eq. (7). In order to track more evaders with a single camera, we can augment the map γ' and include the intersection of each of the evader sets γ_{e} with the camera FOV γ_{c}.

3.2 Acquiring. For the objective of target acquisition, we assume that the target is not intersecting with the FOV at time $k=0$. The purpose is then to maximize the probability that the FOV intersects the evader at some time in the horizon $\mathcal{T}$. We define the target and safe sets to be $\gamma(y) = \{x \in \mathcal{X} : \gamma_{\mathrm{c}}(x) \cap \gamma_{\mathrm{e}}(y) \neq \emptyset\}$ and $\gamma'(y) = \mathcal{X}, \forall y \in \mathcal{Y}$ at all times in the horizon. The covering functions are again defined as in Eq. (5) where in this special case $p_{K'_k}(x) = 1, \forall x \in \mathcal{X}$ and $\forall k \in \mathcal{T}$. In the case of acquiring multiple evaders at the same time, we can define a separate map γ_{e} for each evader and define the map γ using the intersection of each of the evader sets γ_{e} with the camera FOV γ_{c}.

3.3 Acquire While Tracking. For the objective of target acquisition while tracking we consider the presence of two evaders modeled by processes y_k^1 and y_k^2 where y_k^1 is intersecting with the FOV at time $k=0$ while y_k^2 is not. The purpose is to maximize the probability of acquiring process y_k^2 at some point in the horizon $\mathcal{T}$ while minimizing the probability of losing track of process y_k^1 before that time. The target and safe sets are defined as $\gamma(y) = \{x \in \mathcal{X} : \gamma_{\mathrm{c}}(x) \cap \gamma_{\mathrm{e}}(y^1) \neq \emptyset, \gamma_{\mathrm{c}}(x) \cap \gamma_{\mathrm{e}}(y^2) \neq \emptyset\}$ and $\gamma'(y) = \{x \in \mathcal{X} : \gamma_{\mathrm{c}}(x) \cap \gamma_{\mathrm{e}}(y^1) \neq \emptyset\}$ for $y = (y^1, y^2) \in \mathcal{Y} \times \mathcal{Y}$. The covering functions are defined as in Eq. (5) where both $p_{K_k}(x), p_{K'_k}(x)$ take values in $[0, 1], \forall x \in \mathcal{X}$ and $\forall k \in \mathcal{T}$.

3.4 Reach-Avoid Algorithm Description. In Algorithm 1, we present the sequence of steps needed to calculate reach-avoid probabilities for different surveillance objectives. We assume the existence of a global sensor that provides estimates of the evaders position. A model of the evader dynamics is used to construct the transition kernel of the associated stochastic process and approximate the discrete distributions of each evader i, $\mathbf{y}_{\mathbf{k}}^i$ for $k \in \mathcal{T}$. We then use the discrete distributions to calculate tracking, acquiring, and acquiring while tracking probabilities. The camera PT state is measured with on-board encoders. Using this information, we solve Eqs. (7) and (8) and apply the obtained control policies. Note that step 5 of Algorithm 1 involves an important assumption about the joint evader dynamics. In order to create an estimate for the spatial distribution of all M evaders over time, we have assumed both time-independence in the evolution of each evader as well as spatial independence between evaders. These assumptions are necessary to avoid the exponential growth in the dimension of the joint evader distribution both as a function of time and space. As illustrated in our experimental study, the approximate distributions $\mathbf{y}_{\mathbf{k}}^i$ used are accurate enough for this work. Note that step 8 in Algorithm 1 requires solving a dynamic recursion which for our implementation is of combinatorial complexity in the dimension of $\mathcal{X} \times \mathcal{U}$. As a result, the proposed method is limited by the curse of dimensionality. In the presence of more cameras, the dimension of $\mathcal{X} \times \mathcal{U}$ increases creating fundamental computational problems since we need a tractable way to calculate reach-avoid probabilities on the combined space of cameras. In what follows, we propose a way to approximate the reachability calculation by posing reach-avoid objectives for each camera independently without taking into account the state of all other cameras. In this way we can transform the problem of solving Eq. (7) for the camera collective into finding the best (potentially suboptimal) allocation of reach-avoid objectives to the different cameras. In what follows we present the mechanism used to evaluate and order different objective allocations. The result of the proposed method is to transform the combinatorial complexity of a monolithic dynamic recursion defined on the collection of cameras into a combinatorial number of lower dimensional dynamic recursions defined for each camera. In Sec. 5, we discuss the implementation issues related to Algorithm 1 and illustrate by an experimental study that the task allocation mechanism can be used to deal with system sizes that are intractable using the monolithic formulation.

Algorithm 1. Reach-avoid for track, acquire and acquire while track.

```
1: Select a reach-avoid objective from $\{T_i, A_i, T_jA_i\}$ for evaders
   $i \neq j$. $\{T_i, A_i, T_jA_i\}$ denote tracking evader $i$, acquiring
   evader $i$ and acquiring $i$ while tracking $j$.
2: Select the appropriate form of the functions $\gamma, \gamma'$ corresponding
   to the selected objective.
3: Read the current state $x$ of the camera pan-tilt angle.
4: Read the current estimates of centers $\mathbf{y}_{\mathbf{0}}^i$ for all $i \in M$
   evaders.
5: Create an estimate for each distribution $\mathbf{y}_{\mathbf{k}}^i, \forall k \in \mathcal{T} \backslash \{0\}$.
6: Calculate the functions $p_{K_k}(x), p_{K'_k}(x), \forall k \in \mathcal{T}$ using $\gamma$, $\gamma'$
   corresponding to the selected objective.
7: Initialize $V_{N-1}^*(x) = p_{K_{N-1}}(x)$.
8: Solve the recursion in Eq. (7) storing the policy obtained
   from Eq. (8).
9: Evaluate $\mu_k^*(x), \forall k \in \mathcal{T}$ and implement the obtained $N-1$
   step control policy sequence starting with $\mu_0^*(x)$.
```

4 Multicamera System

We consider a surveillance multicamera system to be a collection of d cameras controlled by an autonomous task allocation mechanism. We define a collection of spaces $\mathcal{X} = \prod_{i=1}^d \mathcal{X}_i, \mathcal{U} = \prod_{i=1}^d \mathcal{U}_i, K = \prod_{i=1}^d K_i, K' = \prod_{i=1}^d K'_i$ where $\mathcal{X}_i, \mathcal{U}_i \subseteq \mathbb{R}^2, K_i, K'_i \in \mathcal{B}(\mathbb{R}^2)$. Each camera has the same state, control, target, and safe set interpretation as the ones introduced in Sec. 3 (the cameras FOV interact in the common space $\mathcal{G}$—see Fig. 4). The main objective of the allocation mechanism is to assign tasks (surveillance objectives) to each camera such that the collective reaches a state at which each evader is in the FOV of at least one camera.

We make use of an auxiliary finite state machine $\mathcal{P}$ defined for the collective in order to choose the best among different task allocations. The states of $\mathcal{P}$ correspond to whether an evader or a combination of evaders is inside a FOV or not. For example, in the case of two evaders the states of $\mathcal{P}$ are $\{S_{00}, S_{01}, S_{10}, S_{11}\}$ corresponding to having no evader, only evader 1, only evader 2, and both evaders, respectively, in the FOV of at least one camera. The main idea is to encode the transition probabilities from each state of $\mathcal{P}$ using reach-avoid probabilities of target tracking, acquiring and acquiring while tracking calculated by the camera network. In Fig. 5, the transitions are labeled in a pessimistic way assuming that the only objective achieved is the labeled one. In the case of two evaders, if the camera network is at a state $x = [x_1, \ldots, x_d] \in \mathcal{X}$ where each evader is inside a FOV, $\mathcal{P}$ is in S_{11}. We encode a transition from S_{11} to S_{01} with the reach-avoid probability of only tracking the first evader (denoted by $\Pr(T_1)$ in Fig. 5) assuming that no other objective is achieved. The probability of a self transition from S_{11} to S_{11} is encoded by the

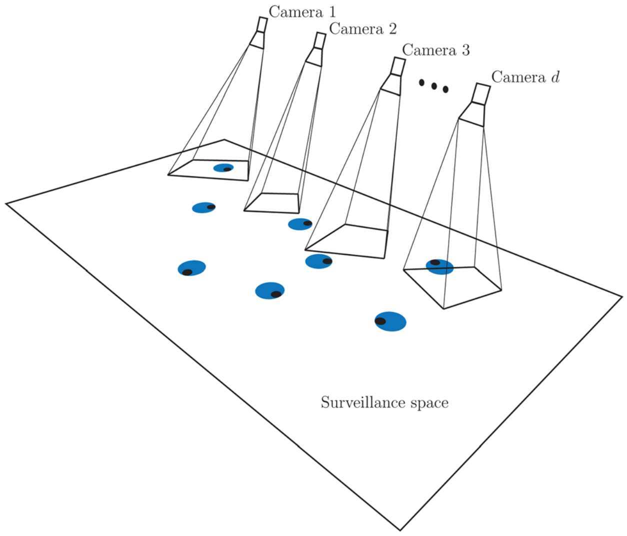


Fig. 4 Visualization of the multicamera network. The flat circles correspond to potential evaders

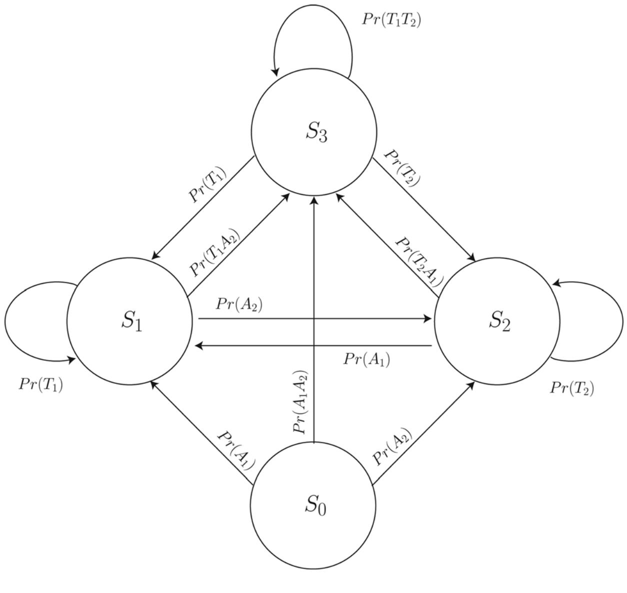


Fig. 5 Auxiliary state machine

reach-avoid probability of tracking both evaders (denoted by $\Pr(T_1T_2)$ in Fig. 5). All other transitions are encoded in a similar way.

4.1 Reachability on the Collective. We can pose reach-avoid problems for the augmented spaces in the exact same way we did for a single MDP. For a target set $K \in \mathcal{B}(\mathcal{X})$ and a safe set $K' \in \mathcal{B}(\mathcal{X})$ we are again looking for a state feedback policy that will manage to guide the MDP to K within N steps, staying in $K' \backslash K$ for all previous time. The resulting DP defined on the augmented spaces $\mathcal{X}, \mathcal{U}$ is not tractable due to the curse of dimensionality (in the case of two cameras we already have $\mathrm{DIM}(\mathcal{X} \times \mathcal{U}) = 8$). To circumvent this difficulty we approximate the collective reach-avoid probabilities by the product of individual reach-avoid probabilities. For example, one way to approximate the probability of the collective tracking all evaders is to use the product of the probabilities of each individual camera to track one of the evaders. Similarly, the probability of the collective tracking a single evader can be approximated by the maximum tracking probability among cameras assigned to track it. Using this idea, we replace the objective of solving a reach-avoid problem for a state $x = [x_1, \ldots, x_d] \in \mathcal{X}$ to solving all reach-avoid problems for each $x_i \in \mathcal{X}_i$. The final task is to select for each camera a reach-avoid objective that makes the collective benefit the most. To make this more precise, we introduce the following maps where 2^A denotes the power set of the set A.

$\mathcal{L}_i(m)$: For a given number of evaders m, $\mathcal{L}_i(m)$ denotes the set of all possible reach-avoid value functions that can be computed by camera i. For example, in the case $m=2$, $\mathcal{L}_i(2) = \{V_{T_1}, V_{A_1}, V_{T_2A_1}, V_{T_2}, V_{A_2}, V_{T_1A_2}\}$ where $V_{T_k}, V_{A_k}, V_{T_kA_l}$ correspond to the value functions of tracking evader k, acquiring evader k and acquiring l while tracking k.

$L_i : \mathcal{X} \rightarrow 2^{\mathcal{L}_i(m)}$. Using this map we select the subset of $\mathcal{L}_i(m)$ that includes all reach-avoid problems with probability over a threshold α for camera i. Naturally, this depends on the state of the camera x_i. For example, a camera at x_i will have $V_{T_1}(x_i) < \alpha$ if evader 1 is initially not in the FOV. The map may differ between cameras (hence the subscript i) depending on their position in the environment.

$\mathcal{F} : \prod_{i=1}^{d} L_i(x) \rightarrow \mathcal{P}$. A map from an allocation of reach-avoid objectives to the state of $\mathcal{P}$. This map will be used to assign to every reach-avoid value function combination $l = [l_1, \ldots, l_d]$, $l_i \in L_i(x)$, a state $s \in \mathcal{P}$. The state s corresponds to the resulting system state if each camera successfully completes the reach-avoid objective indicated by the allocation l. For example, if there are two cameras and two evaders, candidate allocations $l^1, l^2, l^3 \in \prod_{i=1}^{d} L_i(x)$ could be $l^1 = \{V_{T_1}(x_1), V_{T_2}(x_2)\}, l^2 = \{V_{A_2}(x_1), V_{T_1A_2}(x_2)\}$ and $l^3 = \{V_{A_2}(x_1), V_{A_2}(x_2)\}$ where $\mathcal{F}(l^1) = \mathcal{F}(l^2) = S_{11}$ and $\mathcal{F}(l^3) = S_{01}$. We consider l^1 feasible if x_1, x_2 are such that evader 1 is included in the FOV of camera 1 and evader 2 is included in the FOV of camera 2. Similarly, the allocation l^2 is feasible if evader 1 is included in the FOV of camera 2 and l^3 is feasible if none of the evaders is in either of the cameras FOV.

$\mathcal{H} : \mathcal{P} \rightarrow \mathbb{R}_+$. A hierarchy (order) assignment between states in $\mathcal{P}$. The idea is to give the freedom to the surveillance system operator to select which state (which evader or combination of evaders) is more important to be tracked or acquired. For example, if evaders 2 and 3 are more important than evaders 4 and 5, $\mathcal{H}(S_{00110}) \gg \mathcal{H}(S_{11000})$. Note that $\mathcal{H}$ can also be camera dependent, capturing the fact that some PT cameras are more strategically placed than others or have greater image processing capabilities.

The final task is to order the different possible allocations that can emerge from all the combinations $l \in \prod_{i=1}^{d} L_i(x)$ of feasible reach-avoid objectives. For that we calculate $s = \mathcal{F}(l)$ and use $\mathcal{H}(s)$ to scale the product of the selected reach-avoid probabilities corresponding to each l_i, denoted by $\mathbb{P}[l_i]$. The optimization problem that we end up solving has the following form:

$$\begin{aligned} &\max_{l} \quad \mathcal{H}(s) \prod_{i=1}^{d} \mathbb{P}[l_i] \\ &\mathrm{s.t} \quad \mathrm{l} \in \prod_{\mathrm{i}=1}^{\mathrm{d}} \mathrm{L_i(x)} \\ &\quad s = \mathcal{F}(l) \end{aligned} \tag{9}$$

The elements in $\prod_{i=1}^{d} L_i(x)$ are finite, the maps $\mathcal{F}, \mathcal{H}$ are set a-priori and each l_i is a label that corresponds to a reach-avoid probability (i.e., a value function associated with a surveillance objective for camera i—denoted by $\mathbb{P}[l_i]$) evaluated at the current x_i. In principle, the map L_i is not needed since objectives with very low probability will anyway reduce the product in Problem (9) and never be selected. However, pruning the very obvious ones (e.g., tracking objectives when the evader is initially outside all FOV or acquisition objectives when evader is initially inside one FOV) using L_i reduces computations significantly. Problem (9) is a combinatorial problem and the number of candidate solutions scales combinatorially with the number of evaders and number of cameras. Even so, the experimental validation in Sec. 5 indicates that the method is successful in approximating reach-avoid problems that would otherwise be intractable. This is achieved via an implicit objective independence assumption which allows using $\prod_{i=1}^{d} \mathbb{P}[l_i]$ to approximate the reach-avoid probabilities defined for the collective. The proposed allocation will be suboptimal compared to the one obtained if one could solve the monolithic DP defined on the camera network.

4.2 Task Allocation Algorithm Description. In Algorithm 2 we summarize the task allocation mechanism which is a three-step scheme. In the first step, the system operator decides on the importance of different states in the camera collective by fixing the values of $\mathcal{H}, \forall s \in \mathcal{P}$ (these values can be decided offline or change dynamically while the algorithm is running). We have suggested an evader-dependent definition of states in $\mathcal{P}$ and $\mathcal{H}$ but this can be extended to different types of states (e.g., related to characteristics of the surveillance space) and hierarchies. In the second step, every camera solves a collection of local reach-avoid problems (see Algorithm 1) that depend on the current estimate of every evader's position. In the final step of the process, the success probabilities calculated by each camera are communicated to the task allocation mechanism that solves Problem (9) and assigns to each camera the objective corresponding to the optimal allocation. Using the state feedback policies that each camera calculated, a control action is applied. Note that steps 6–11 of Algorithm 2 contain most of the computation burden and are completely independent between cameras. The only thing that the task allocation mechanism needs from each camera i in order to carry out steps 14–16 is a vector of scalars corresponding to the value functions of all reach-avoid objectives at the state x_i—denoted by $\mathbb{P}[L_i(x)]$. An alternative approach would be to communicate the PT angle vector x to the whole network and calculate all possible reach-avoid value functions (for all cameras) locally with each camera. Problem (9) could then be solved independently by each camera removing the need of a central task allocating mechanism. Another approach to decentralize the method would be to implement an all-to-all communication of value function values (instead of x) and solve problem (9) again on each camera independently. Such approaches would increase the computational burden of each camera but make the system completely decentralized. The block diagram in Fig. 6 illustrates the flow of information in the current implementation. Note that the estimates of the evader positions are filtered globally and communicated to each camera independently. In principle, this requirement can be removed by implementing a distributed estimation algorithm. Finally, note that in step 19 we only apply the first control action to the camera network (receding horizon implementation). This is not necessary and one could also use the whole block of control policies and apply a sequence of control actions ranging from 1 to $N-1$, as in Algorithm 1.

Algorithm 2. Hierarchical task allocation.

```
1: Specify a hierarchy on global states by setting $\mathcal{H}(s), \forall s \in \mathcal{P}$.
2: Read the current estimates of centers $\mathbf{y}_\mathbf{0}^i$ for all $i \in M$ evaders.
3: Create an estimate for each distribution $\mathbf{y}_\mathbf{k}^i, \forall k \in \mathcal{T}\backslash\{0\}$.
4: Read pan-tilt angle vector $x$ where $x_i$ is the state of camera $i$.
5: for each camera $i$ do
6:     Use $x_i$ to calculate $L_i(x)$ and disregard objectives of no interest.
7:     for all $l_i \in L_i(x)$ do
8:         Select the appropriate form of the functions $\gamma, \gamma'$ corresponding to the objective $l_i$.
9:         Execute steps 6–8 of Algorithm 1 for each $l_i$ to obtain the probability $\mathbb{P}[l_i]$ that camera $i$ achieves objective $l_i$.
10:    end for
11:    Send to the task allocation mechanism a vector containing the probabilities $\mathbb{P}[l_i]$ of achieving each objective $l_i$.
12: end for
13: for all $l = [l_1,\ldots,l_d]$, $l_i \in L_i(x)$ do
14:    Calculate $g = \mathcal{F}(l)$ along with the corresponding $\mathcal{H}(s)$.
15:    Calculate the value of $\mathcal{H}(s)\prod_{i=1}^{d}\mathbb{P}[l_i]$.
16:    Label by $l^* = [l_1^*, \ldots, l_d^*]$ the combination of objectives that achieves the maximum in $\mathcal{H}(s)\prod_{i=1}^{d}\mathbb{P}[l_i]$.
17: end for
18: Obtain the optimal control actions $u_i^* = \mu_0^*(x_i)$, associated with each camera's optimal reach-avoid objective specified by the allocation $l^*$.
19: Apply the sequence of $u^* = [u_1^*, \ldots, u_d^*]$ to the camera network.
```

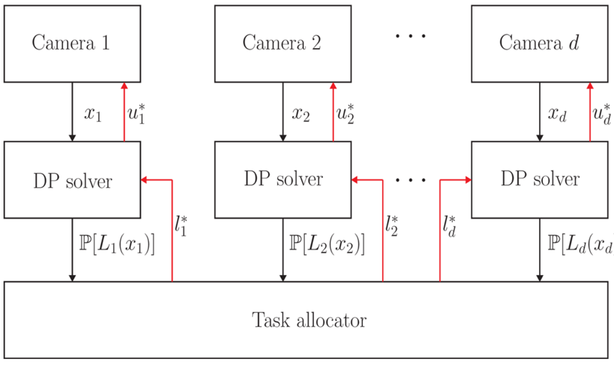


Fig. 6 Block diagram of the task allocator

4.3 Complexity. Denote by $O(C)$ the complexity of solving a four-dimensional N-step dynamic recursion defined for a single reach-avoid objective for one camera. The reach-avoid dynamic recursion defined on the collective of d cameras is of $\mathcal{O}(C^d)$ complexity since, as described in Sec. 5.1, we choose to grid the two-dimensional $\mathcal{X}_i$ and $\mathcal{U}_i$ spaces resulting in a four-dimensional grid for each one of the d cameras. Let m denote the total number of evaders which we assume to be an even number without loss of generality. Each camera can either track or acquire an evader which means it has to solve a maximum of m tracking and acquiring reach-avoid problems. The worst case total number of acquire while tracking objectives is $m^2/4$ and occurs when half of the evaders are currently in the FOV of the camera, assuming that each reach-avoid problem involves two evaders at a time (i.e., we assume that cameras do not track or acquire more than one evader). When reducing the monolithic dynamic recursion to local subproblems, the worst case total number of reach-avoid problems to be addressed is $d[(m^2/4)+m]$ which is $\mathcal{O}(dm^2)$. The choice of the best allocation (see Problem (9)) results in the comparison of at most $[(m^2/4)+m]^d$ possible solutions which is $\mathcal{O}(m^{2d})$.

Using the proposed task allocation mechanism we replace the calculation of the $4d$-dimensional DPs defined for each surveillance objective to a collection of four-dimensional ones, followed by a search over $\mathcal{O}(m^{2d})$ possible task allocations. Although both approaches scale combinatorially, the latter approximation method is feasible even when solving the monolithic DP is not. For the sake of comparison, consider that for $d=2$ cameras and $m=2$ evaders the DP recursion needs to be solved on an eight-dimensional grid for each reach-avoid objective. On the same problem size, the proposed task allocation method deals with at most 3 four-dimensional DPs per camera and chooses over nine possible allocations. For $d=5$ cameras and $m=6$ evaders each camera has to solve at most 15 four-dimensional DPs and the final allocation is chosen by ordering the resulting 759,375 allocations. Note that each camera can calculate the four-dimensional DPs locally, making this step of the algorithm completely decentralized. By comparison, the original DP recursions would be defined on a 20-dimensional space for each objective. In Fig. 7, we have plotted the complexity of addressing a single objective with the high-dimensional DP recursion against the complexity of using the task allocation method and solving all possible four-dimensional DPs (corresponding to all possible objectives) for different evader numbers. To generate the plot we assumed a

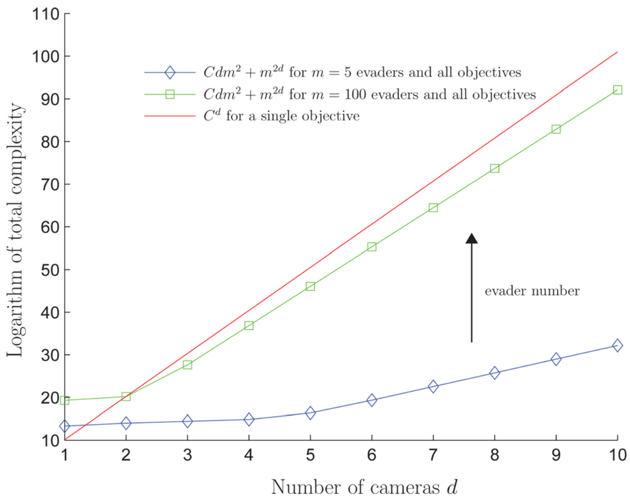


Fig. 7 Comparison of the total complexity. The *y*-axis is in logarithmic scale. The diamond line approaches the square one as *m* increases

complexity of $C = |\mathcal{X}| \cdot |\mathcal{U}|$ to solve a single four-dimensional DP and used the same cardinalities as the ones reported in the implementation section. The results are plotted in logarithmic scale. In terms of memory needed for the 20-dimensional DP, assuming ten grid points in each dimension and 1 byte of memory for each cell, the resulting grid would require 0.1 billion terabytes of storage. State of the art methods that include (uniform) gridding reach a maximum of 5–6 dimensions and as a result research is focusing on approximate dynamic programming (ADP) techniques, (see, for example, Refs. [27,34] and the books in Refs. [35,36] for a more general approach).

4.4 Discussion and Comparisons. We conclude this section by providing a high level comparison of the methods proposed here and existing methods in the literature. Our methods are only using information about the dynamical behavior of the targets to be tracked (these being people, vehicles, etc.) without incorporating important physical characteristics (faces, edges, reflecting materials, etc.). For this reason we consider our methods complementary to computer-vision-based detection algorithms and do not discuss these in detail here. In Table 1, we have listed some of the existing algorithms for target tracking using PTZ cameras. The categorization is based on the main focus of each work, the type of dynamics considered for cameras and targets, the planning horizon, the use of a global sensor for positioning and the size of tractable problems. In terms of size, we consider anything below five cameras and ten targets small scale and anything above ten cameras and ten targets, large scale.

In addition to successfully acquiring and tracking targets, we are also interested in accurately characterizing the related probabilities of success. In several works (see Refs. [6,10,11]), the authors implicitly involve probability metrics in characterizing the capabilities of a camera in tracking an object. These metrics are often static and have to be identified on the particular setup. This is different in our work where we explicitly calculate such probabilities online (as a function of the camera and target dynamics) in order to use them later in the task allocation mechanism.

One of the main limitations of our method is the need to communicate a vector of scalars (value function information) to a central controller that decides on the objective allocation between cameras. This is consistent with most architectures studied in the literature (and implemented in practice) where a central unit is responsible for camera coordination and multicast communication. Nevertheless, some camera networks do not include a central coordinator and rely instead on distributed algorithms (see, for example, Refs. [10–12]).

Concerning computational complexity, the methods we propose involve two major computation blocks. One is associated with solving the reachability DP defined in Eq. (7) and the other with selecting the optimal allocation of objectives between cameras by solving the combinatorial optimization Problem (9). In general, both problems are very difficult and the complexity of obtaining the optimal solution scales exponentially in the number of cameras and targets, respectively. This is a well-known fundamental limitation of DP and combinatorial optimization that cannot be circumvented by any known method. To ameliorate this problem, several approximation heuristics have been proposed in literature which run in polynomial time but produce suboptimal solutions. For ADP, we refer the reader to Ref. [35] and the recent work in Ref. [36] while for approximations to combinatorial optimization, the books in Refs. [37,38] are an excellent starting point. Note that the approximation methods proposed in these works can be readily used in our framework to solve Eq. (7) or Problem (9).

Several methods addressing the problem of target tracking with multiple cameras exhibit better scaling properties than the methods presented here. This is usually achieved by designing specialized cost functions that approximate the objective of tracking (see Refs. [11,12,18]) and using heuristic algorithms to enforce cooperation between agents (see, for example, Refs. [6,7,25]). The approach is different in our work where the objectives of target tracking and acquisition are naturally modeled as reach-avoid problems and the task allocation is solved to optimality under the assumption of decoupled objectives between cameras. Overall, other than the elegant mathematical framework, the main advantages of our approach are: the fact that we can deal with very general stochastic models both for the targets and cameras, the fact that our formulation is independent of the particular camera setup and the fact that the value function computations needed for the task allocation mechanism are naturally distributed between cameras (each camera is solving a collection of local problems).

5 Implementation and Experimental Results

5.1 Notes on the Implementation. The framework was tested on a surveillance setup consisting of two robotic evaders (Khepera III and e-puck differential robots) treated as flat circles of radius $\rho = 0.06$ m and two Ulysses PT units donated by Videotec S.p.A. The extrinsic and intrinsic camera parameters needed for projecting specific PT configurations to the FOV on the ground

Table 1 Comparison of related methods for surveillance using PTZ cameras

Algorithm	Focus	Dynamics (targets/camera)	Planning steps	Global sensor	Size
[10–12]	Distributed	Linear Gaussian/none	Single	No	Large
[5]	Camera subnetworks	Linear Gaussian/none	Single	Yes	Medium
[6]	Collaborative control	None/none	Multiple	Yes	Medium
[7]	Tracking assignments	None/none	Multiple	Yes	Medium
[14]	Bio inspired	Markov model/linear	Multiple	No	Small
[15]	EKF and PF combination	Stochastic/nonlinear	Multiple	No	Small
[18]	Observations versus resolution	Stochastic/MDP	Single	Yes	Medium
Reachability	Modeling framework	Stochastic/MDP	Multiple	Yes	Medium

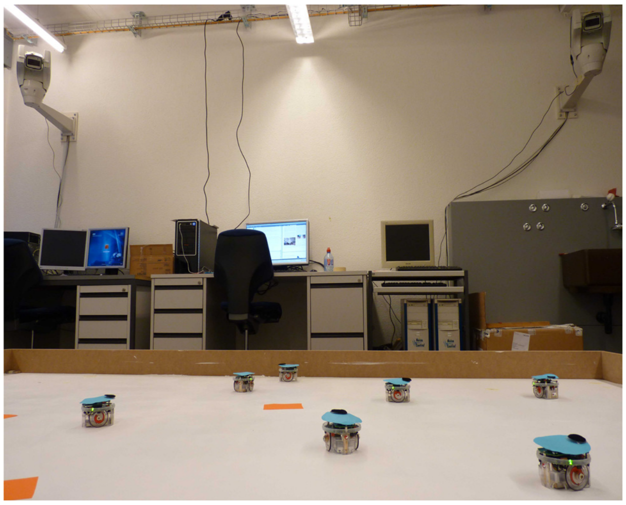


Fig. 8 View of the experimental setup

Table 2 Summary of used parameters

Pan grid	$[\pi/2, 0)$ rad
Tilt grid	$[\pi/2, \pi/2)$ rad
States N_x	2700
$\hat{Q}(u)$	$\mathbb{R}^{2700\times2700}$
$\mathcal{G}_x, \mathcal{Y}_x$ grid	$[1.24, 4.76)$ m
$\mathcal{G}_y, \mathcal{Y}_y$ grid	$[1.24, 4.76)$ m
States N_y, N_g	220,900
ν	0.33 m/s
Ω samples	1000
σ^2	$\pi/2$ rad^2/s^2
N	5
Δ_t	250 ms
B	$\mathbb{R}^{2700\times220900}$

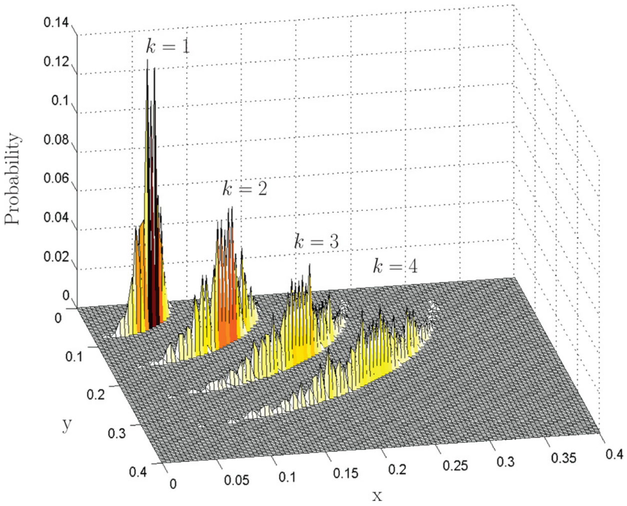


Fig. 9 Estimated distribution of the evader process

plane ($x \mapsto \gamma_c(x)$) were identified in Ref. [15]. The ground plane surveillance space consists of a 12 m^2 white table (see Fig. 8). We make use of a global eye camera placed above the surveillance space and filter the recorded images to provide estimates of the evader positions. Note that the PT cameras have a constant zoom selected to obtain high resolution images (over most of the surveillance space) as opposed to the global eye camera that is only used to provide rough estimates of the evader location.

Having defined the relevant covering functions in Sec. 3, the recursion in Eq. (7) was solved on uniform grids $\mathcal{X}_g = \{x_1, \ldots, x_{N_x}\}, \mathcal{Y}_g = \{y_1, \ldots, y_{N_y}\}, \mathcal{G}_g = \{g_1, \ldots, g_{N_g}\}$ and $\mathcal{U}_g = \{u_1, \ldots, u_{N_u}\}$ placed on $\mathcal{X}, \mathcal{Y}, \mathcal{G}$, and $\mathcal{U}$, respectively. In the following experiments we chose a 60×45 grid for $\mathcal{X}$ (leading to $N_x = 2700$ and a discretization step of 0.0262 deg in the pan and tilt dimensions) and a 470×470 grid on $\mathcal{Y}$ (leading to $N_y = 220{,}900$ and a discretization step of 0.0075 m in both evader space dimensions). The camera control actions were limited to a 3×3 grid $\mathcal{U}_g = \{$up, down, middle, left, right, upright, upleft, downright, downleft$\}$. All integrals were numerically calculated via summations over grid cells. Once the value functions associated with each objective were calculated $\forall x \in \mathcal{X}_g$ using the recursion in Eq. (7), the control actions were retrieved via the policy $\mu_0^*(x_0)$ obtained by solving Eq. (8) over $\mathcal{U}_g$ for the measured state of the camera x_0. After every calculation we rounded the states of the evader and camera processes to the nearest element of the corresponding grid.

The overall computational effort of this process is (in general) very high. The main challenge lies in the calculation of the covering functions $p_{K_k}, p_{K_k'}$ and the integral of V_{k+1}^* with respect to $Q(\cdot|x,u)$. Since all spaces $\mathcal{Y}_g, \mathcal{G}_g, \mathcal{X}_g, \mathcal{U}_g$ are finite, the calculation of vectors $p_{K_k}, p_{K_k'}$ is reduced to matrix–vector multiplications between the discrete distribution of the stochastic process $\mathbf{y_k}$ and a binary matrix B. Each element i of the vector represents the probability of the evader being in the FOV corresponding to i at time k. The matrix B was constructed offline as a map between the PT angles of each camera and the set of evader centers intersecting with the corresponding FOV. In particular, we constructed a matrix $B \in \mathbb{R}^{N_x \times N_y}$ where

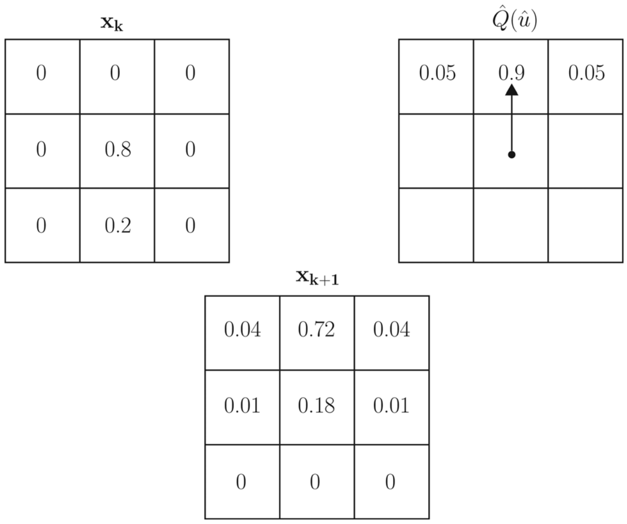
$$B_{i,j} = \begin{cases} 1 & \text{if} \quad \gamma_e(y_j) \cap \gamma_c(x_i) \neq \emptyset \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

Assuming $\mathbf{y_0}$ is measured, we constructed the distributions of $\mathbf{y_k}$ via sampling from $\Omega \sim \mathcal{N}(0, \sigma^2)$ and propagating a unicycle model for the evader

$$y(k+1) = \begin{bmatrix} y_1(k) + v\sin(y_3(k))\Delta_t \\ y_2(k) + v\cos(y_3(k))\Delta_t \\ y_3(k) + \Omega\Delta_t \end{bmatrix} \tag{11}$$

where $y_1(k)$ and $y_2(k)$ denote the coordinates of the evader center, $y_3(k)$ is the heading angle and Δ_t is the chosen sampling time. The linear velocity v is assumed to be known and constant[1]—see Table 2 for the exact values. The variable Ω corresponds to the rotational velocity of the evader and assuming it is random corresponds to random turning between time steps. Using Eq. (11), we approximate the distribution of $\mathbf{y_k}$ at every step of the horizon by sampling 1000 points from the distribution of Ω (see Fig. 9). As discussed in Sec. 2 the estimated distributions are used to calculate the corresponding covering functions. Increasing the variance σ^2 of Ω makes the assumed evader model more uncertain directly affecting the success probabilities of the reach-avoid objectives. In the conducted experiments, the evader movement was consistent with the chosen model. To evaluate the integral of V_{k+1}^* with respect to $Q(\cdot|x,u)$, we constructed a transition probability matrix $\hat{Q}(u) \in \mathbb{R}^{N_x \times N_x}$ for each possible control action $u \in \mathcal{U}_g$. For example, if the camera state, distributed according to $\mathbf{x_k}$, is commanded to go up ($\hat{u}$), the distribution $\mathbf{x_{k+1}}$ is given by $\mathbf{x_{k+1}} = \hat{Q}(\hat{u})\mathbf{x_k}$ where $\hat{Q}(\hat{u})$ is constructed such that going up in the PT space grid has probability 0.9 while diffusing to the position upright or upleft has probability 0.05. In this way, each transition probability matrix $\hat{Q}(\hat{u})$ is consistent with the idea that the camera angle is well tracked by the low-level controllers with a low probability of jumping to a wrong nearby PT state. Figure 10 illustrates an

[1] Even though we model the evader dynamics in $\mathbb{R}^3$ (where the third dimension is orientation) in the reach-avoid formulation we only need the location of the center to calculate $\gamma_e(y)$ (see Eq. (10)) and hence the covering functions.



0	0	0
0	0.8	0
0	0.2	0

0.05	0.9	0.05

0.04	0.72	0.04
0.01	0.18	0.01
0	0	0

Fig. 10 Camera kernel movement for $\hat{u} = up$

example move for $\hat{u} = up$. The rest of the considered PT movements are handled in a similar same way.

The system was tested at different prediction horizon lengths of $N \in \{1,\ldots,10\}$ with a sampling frequency ranging between 8 Hz and 1 Hz. A horizon length of $N=5$ resulted in a sampling frequency of 4 Hz which was adequate for tracking the evaders, moving at a maximum speed of 0.33 m/s. For faster evaders, the system would eventually be too slow to achieve tracking or acquisition. However, using the size of the respective grids, we can calculate the amount of time needed to carry out the computations of the DP recursion in advance and estimate the maximum evader speed that can be handled for a given horizon length. Note that solving Eq. (7) within this time frame was only possible due to the coarse quantization of control actions (only adjacent grid cells were considered as possible camera movements), making the maximization over $\mathcal{U}_g$ an easy operation. Moreover, in this particular application, all involved matrices are sparse making matrix multiplications relatively fast. For a single camera and 2 evaders, the slowest part of the algorithm is the set-point tracking of the low-level motor controllers. Note that as the system size increases (more cameras and evaders), the sampling time might have to be increased.

5.2 Overview of Experimental Study. To evaluate the performance of the overall system we conducted a series of experiments. We first evaluated the accuracy of the surveillance reach-avoid policies of tracking, acquiring and acquiring while tracking calculated using Algorithm 1 for a single camera and a pair of evaders. We then implemented Algorithm 1 in receding horizon, independently for each camera, in order to evaluate the system performance in continuous operation. Furthermore, we analyzed the effect of the horizon length N on the predicted reach-avoid probabilities, confirming the trade-off between the value function value of tracking and acquiring as N decreases or increases. To investigate the achieved coordination between cameras, we implemented the task allocation mechanism described in Algorithm 2 and evaluated the overall performance under two different scenarios: In the first we allowed the FOV of both cameras to move on the whole surveillance space while in the second we restricted the range of their respective FOV such that each camera is responsible for a different area. We measured the total time spent in each state of $\mathcal{P}$ and the results indicated that the allocation mechanism achieves cooperation between cameras. Finally, we investigated the total number of objective switches enforced by the task allocation mechanism and noticed an oscillatory behavior as the horizon length increases. This effect can be suppressed by low pass filtering the suggested objective allocation. During the course of all experiments, we assumed that the evaders do not wish to exit the surveillance space.

Table 3 Initial state values for the camera and the multiple evaders for the experimental tests

	$y^1(0)$	$y^2(0)$	x_0
A_1	[1.9, 2, 0.4636]	–	[−1.0455, 0.8069]
T_1	[1.9, 2, 0.4636]	–	[−0.9158, 0.7021]
T_1A_2	[1.9, 2, 0.4636]	[2.5, 2.6, −2.352]	[−0.9158, 0.6498]

Table 4 Validation of the surveillance framework via comparison of experimental results with the predicted performance computed by the dynamic program

	$V_0^*(x_0)$	Experiments
A_1	0.967	0.92
T_1	0.996	0.96
T_1A_2	0.787	0.76

5.3 Verification of Reach-Avoid Probabilities. In order to verify whether the developed reachability framework accurately quantifies the success probabilities of surveillance objectives, we implemented Algorithm 1 on our two-camera setup. The relevant parameters used are shown in Table 2. We considered a single camera and a pair of evaders for which we implemented separate controllers to make sure they move according to Eq. (11). We tried to quantify whether the reach-avoid surveillance framework is successfully characterizing the success probability of target tracking, target acquisition and target acquisition while tracking. Given an estimate of the distribution of evaders for every $k \in \mathcal{T}\backslash\{0\}$ and an initial PT configuration for the camera x_0, we calculated the $N-1$ optimal state feedback control policies for each reach-avoid objective and applied them in series using the measured camera state at every time. In particular, we calculated the reach-avoid probabilities for tracking evader 1, acquiring evader 1, and acquiring 2 while tracking 1 for different x_0 of the camera and $y^1(0)$, $y^2(0)$ for evader 1 and 2, respectively. We applied the control actions corresponding to the optimal policy maps 25 times and checked how many times the objectives $\hat{V}_{T_1}(x_0^{T_1}), \hat{V}_{A_1}(x_0^{A_1}), \hat{V}_{T_1A_2}(x_0^{T_1A_2})$ were fulfilled. The observed success probabilities and the initial positions of the camera and evaders are shown in Tables 3 and 4 where evader 1 is in the FOV of the camera while evader 2 is not. In order to carry out a full sensitivity analysis of the calculated reach-avoid probabilities to the system parameters, we would have to run a Monte Carlo type verification on the experimental platform for a range of parameters. Instead, we used a simulation platform designed to resemble the setup and checked the output of the reach-avoid algorithm for different initial conditions and parameters for the objective of target tracking. Figure 11 illustrates three scenarios labeled s_0, s_1, s_2. The relative parameters used and the associated probabilities calculated for the objective of target tracking are shown in Table 5.

The experimental results illustrate that the assumptions made are reasonable and the framework is successfully characterizing the reach-avoid probabilities. This was expected since the noise assumptions on the evader processes match their moving strategy while the chosen PT angles of the camera are adequately tracked by the low-level motor position controllers. As a result we can use the calculated probabilities before applying the associated control policies as an estimate of what the camera will be able to achieve during the horizon $\mathcal{T}$.

5.4 Receding Horizon Implementation. We applied Algorithm 1 in receding horizon for each camera independently with two evaders moving randomly on the surveillance space (initialized as in the configuration in Fig. 12). We measured the state of each camera and evader, solved the recursion in Eq. (7) for each

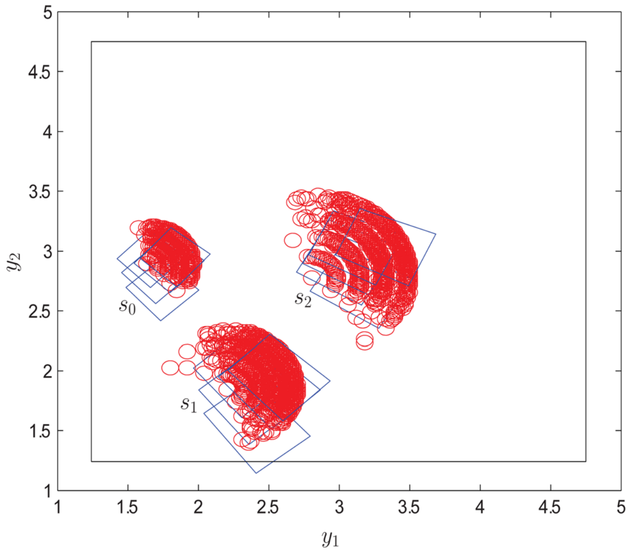


Fig. 11 Tracking trajectories under different initial conditions and parameters

Table 5 Summary of used parameters in Fig. 11

Scenario	ν (m/s)	σ^2 (rad^2/s^2)	V_0^* (tracking)	x_0	$y^1(0)$
s_0	0.33	$\pi/2$	0.9642	[−0.8110, 0.8070]	[1.5566, 2.8540, 0.4636]
s_1	0.5	π	0.8234	[−0.8896, 0.5974]	[2.0864, 1.7932, 0.4636]
s_2	0.7	$\pi/2$	0.7640	[−0.5228, 0.6236]	[2.6468, 2.6983, 0.4636]

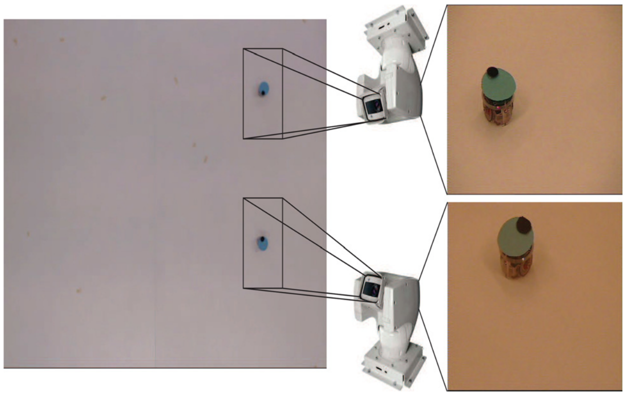


Fig. 12 Camera configuration

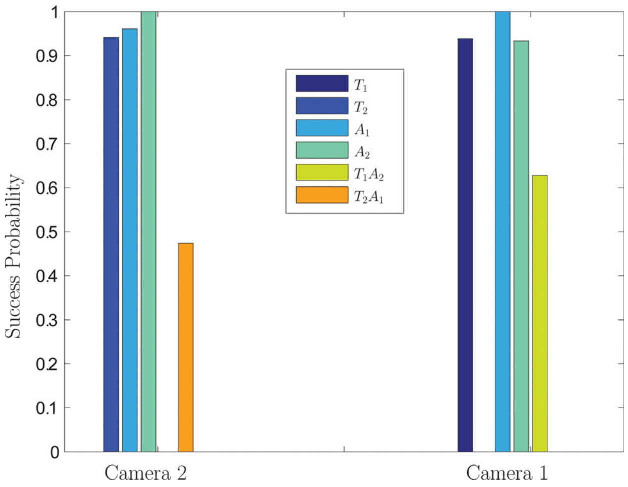


Fig. 13 An example of reach-avoid probabilities reported by the two cameras at a specific time instance

Table 6 Objective switching for camera 2 (top) and camera 1 (bottom)

Horizon	Switches
$N=2$	1
$N=5$	5
$N=8$	39

Horizon	Switches
$N=2$	1
$N=5$	7
$N=8$	39

camera/evader combination and applied the first of the $N-1$ step control policies. We then resolved the dynamic recursion for the newly measured position of the cameras x_0^1, x_0^2 and a newly estimated position for the evaders $y^1(0), y^2(0)$ and repeated the operation. The evader positions were estimated using a discrete time Extended Kalman filter tailored for the model in Eq. (11) with measurements coming from the global eye camera. Figure 13 illustrates the way by which each camera reports all reach-avoid probabilities for the specific $x_0^1, x_0^2, y^1(0), y^2(0)$ corresponding to the positions in Fig. 12 (notice that the evaders are facing each other thus making acquire-while-track objective probabilities nonzero). Our goal was to have each camera acquire an evader and track it thereafter. Whenever possible, each camera tried to achieve an acquire while tracking objective with respect to the evader that was not tracking. A full video demonstration of the receding horizon implementation on the noncooperative (no task allocation) two-camera, two-evader system is available on the web.[2] Two things are important to note: The view of the cameras retrieved directly from their respective frame grabber does not exactly correspond to their projection on the surveillance space shown in the video. This happens due to inaccuracies in the camera parameters that affect the nonlinear transformation between the camera PT angles and the FOV on the surveillance space. As a result, the frame grabber image is a few centimeters off the FOV used in the calculation of the reach-avoid probabilities. The effect varies as a function of the camera state and is greater in areas where the FOV is larger. The other thing to note is that whenever an acquisition can happen with probability 1 for multiple control policies (this can happen due to the bounded noise used in the implementation of the algorithm), the cameras choose to spend the least amount of energy and select $u=$ middle which corresponds to not moving.

5.5 Role of Horizon Length.

To investigate the effect of the horizon length N we recorded the reported reach-avoid probabilities (value function values) from each camera for different horizon length values with static evaders placed as in Fig. 12. The success probabilities reported in Fig. 14 are evaluations (at x_0) of the different V_0^* obtained by solving recursion (7) for each one of the surveillance objectives defined for camera 1. The values suggest that the effect of the horizon length on the value functions is very much related to whether the objective involves acquisition or tracking of an evader. The larger N is, the higher the value of the value function for an acquire objective since we are estimating more of the evader trajectory (square and diamond lines in Fig.

[2] www.youtube.com/user/ETHZurichIfA

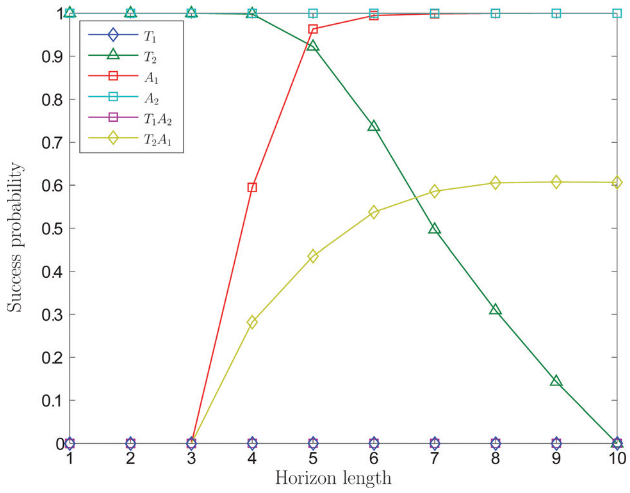


Fig. 14 Effect of horizon length

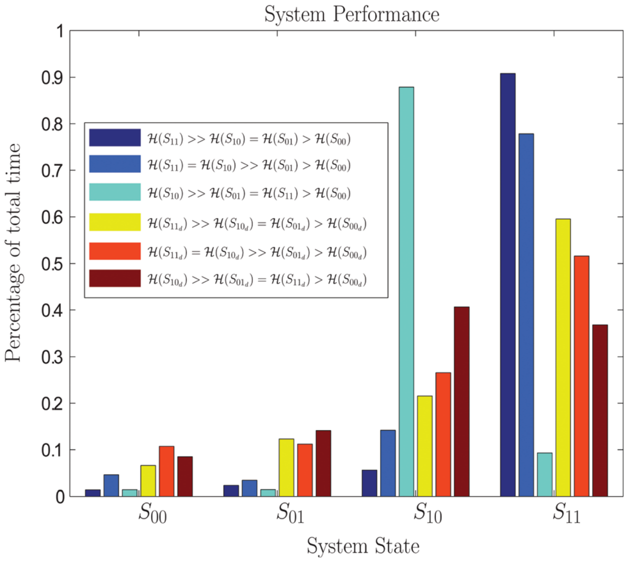


Fig. 15 Total percentage of time spent in each of the states $\{S_{00}, S_{01}, S_{10}, S_{11}\}$ of $\mathcal{P}$ under different choices of hierarchies $\mathcal{H}$. The subscript d denotes experiments where we decoupled the area of coverage of cameras. This is simple to implement by setting the columns of each camera's B matrix in Eq. (10) to 0.

14). The opposite is true concerning tracking objectives where the value function reduces since the variance of the estimated evader distribution (see Fig. 9) is increasing (triangle line in Fig. 14). We can therefore identify a trade-off between the reported value function values for acquire and acquire while track objectives as opposed to pure tracking. The reported values are used in the task allocation mechanism and the final horizon length choice depends on the physical setup, the surveillance scenario considered, the accuracy of the evader model, and whether it sufficiently captures the uncertain dynamics.

5.6 Task Allocation on the Camera System. Implementing the suggested task allocation scheme relies on the assumption that the overall sampling time chosen is enough to calculate all reach-avoid probabilities $\mathbb{P}[L_i(x)]$ for each camera i at x_i and solve Problem (9) to obtain l^*. Communicating $\mathbb{P}[L_i(x)]$ to the task allocation mechanism boils down to the transmission of a vector of scalars which is easy to handle with modern communication systems, even using wireless technology. Solving Problem (9) is a matter of scalar multiplications and ordering which is a very fast operation. The only drawback is the fact that all feasible combinations have to be tested in order to find the maximum. Up to a reasonable number of cameras and evaders (say 5 and 10), this computation is easy to deal with. Following this procedure, we manage to allocate surveillance tasks in a way that exploits the comparative advantages of the different cameras due to their different location in the surveillance setup and the estimated position of the evaders. We evaluated the performance of the task allocation mechanism, in two different surveillance scenarios, through the percentage of time that was spent in each of the states of $\mathcal{P}$. In the first scenario, we allowed the FOV of both cameras to move on the whole surveillance space and chose different reward values for each state in $\mathcal{P}$ by manipulating $\mathcal{H}$. Two robotic evaders were moving according to Eq. (11) on the whole surveillance space for approximately 30 min while the task allocation mechanism was (autonomously) assigning the surveillance task to each camera. The results shown in Fig. 15 indicate that choosing $\mathcal{H}$ a priori controls the total amount of time spent in a state. In particular, the operator can bias the system toward tracking a specific evader or combination of evaders. In the second scenario, we artificially split the surveillance space in half and allowed the FOV of each camera to evolve on one of the two decoupled spaces. In this experiment, we evaluated whether the surveillance of each evader was naturally handed-off to the camera responsible for the relevant area. We forced the evaders to cross from one space to the other simultaneously to observe whether the cameras exchange their targets. The results of this experiment are reported in Fig. 15 using a subscript d on the states. Videos demonstrating the performance of the system, in both scenarios, can be found on the web.[3] Note that the table has been split in half along the horizontal axis of the video. To implement this, we set the relevant entries of the matrix B in Eq. (10) to 0. For the rest of the videos $\mathcal{H}$ was chosen according to the legend in Fig. 15. Note that since we are interested in ordering, only the relative values of $\mathcal{H}$ (and not the absolute) are important. For the sake of completeness, the $\mathcal{H}$ values for the state of seeing no evaders (S_{00}, S_{00_d}) were always chosen to be $\mathcal{H}(S_{00}) = \mathcal{H}(S_{00_d}) = 0$, the high reward values were chosen to be 100 while the small values were chosen to be 10 (different states at each experiment—see the legend in Fig. 15).

5.7 Assignment Switching. An interesting issue to investigate with the objective allocation mechanism is how often objective allocations switch. It could be possible that the allocation mechanism switches tasks frequently from a camera to another resulting in oscillatory behavior in the overall operation. In our formulation, this can happen when the value of the cost function in Problem (9) oscillates between the values of different allocations, affecting the order between them. To investigate this effect we looked at the number of allocation switches in the two-camera, two-evader testbed. We noticed that allocation switching mainly happens when the evaders are spatially close and in areas where the cameras FOV size is similar. The total amount of switches over a long time depends on the prediction horizon length N. As discussed in Sec. 5.5, the horizon length determines which reach-avoid probabilities are feasible and which are not. Therefore, altering N can have a clear effect on the possible situations where objective switching may occur. Figure 16 depicts the allocation by the hierarchical controller to each camera for different values of N. The vertical axis denotes the state for each camera if the allocated objective is achieved. There are only two evaders which means S_{01} and S_{10} are the only options for each camera since the mechanism will never allocate objectives where the resulting state is S_{00} (the reward will be zero). The total number of switches in the allocation is reported in Table 6 as a function of N. It is clear from these results that a longer horizon results in more allocation switches. This can be partially explained by the evader model uncertainty. Longer horizons reduce tracking probabilities

[3] www.youtube.com/user/ETHZurichIfA

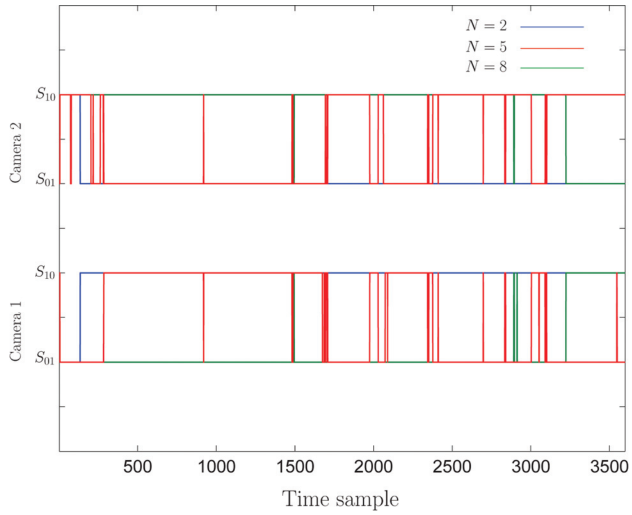


Fig. 16 Objective allocation for the two cameras for different horizon lengths

significantly (as illustrated in Fig. 14) and as a result the task allocation prefers acquisition objectives over tracking. For example, when camera 1 is in S_{01} and camera 2 in S_{10}, the probability of each one remaining in the same state (tracking during the whole horizon) might be very low and as a result the allocation mechanism would command them to acquire the evader which is not in their FOV resulting in switching between S_{01} and S_{10} and vice versa. To alleviate this effect one can filter the resulting task allocation disallowing frequent switching or reduce the horizon length N to reduce the variance in the evader distribution estimates.

6 Conclusions

We illustrated how autonomous surveillance problems for PT cameras can be successfully cast into reach-avoid problems with stochastic sets. We explicitly defined the functions needed to calculate optimal control sequences for the objectives of tracking, acquiring, and acquiring while tracking for a single camera two-evader system, demonstrating the results on our experimental testbed. This approach is limited by the curse of dimensionality and is not directly extendable to systems consisting of multiple PT cameras. To deal with this issue, we introduced an approximation scheme where we extract control actions for all possible reach-avoid problems for each camera in the network and use simple hierarchical rewards (based on the expected surveillance outcome) to choose a control action for each camera. The approximation is based on an independence assumption between the reach-avoid objectives of each camera. Under similar assumptions, the proposed method can be used in different frameworks where reach-avoid objectives are defined for a collection of agents.

Applying the task allocation technique in a two-camera two-evader setup, we illustrated that the performance of the system can be controlled by the chosen hierarchical rewards which gives great power to the system operator in altering surveillance objectives while the system is running. Although the complexity of the proposed method is combinatorial in the number of evaders with a rate proportional to the number of cameras, it can be applied to networks where uniform gridding techniques are infeasible due to the size of the resulting grid. This was achieved by replacing a monolithic DP which is combinatorial in the size of the camera network with a combinatorial number of small DPs. The proposed approach introduces some conservatism since each objective is assigned to a single camera and we do not take into account situations where collections of cameras can cooperate in achieving an objective. In order to reduce this conservatism, we are currently looking into a sequential approach where each camera recalculates its optimal policy, taking into account the optimal policy of neighboring cameras. Another part of future work concentrates on approximate dynamic programing to make the reach-avoid recursion tractable in 3D surveillance spaces and incorporate camera zoom control. Finally, it would be interesting to develop a similar asynchronous task allocation mechanism that will be able to take full advantage of the fact that PT camera computation platforms are progressively improving, allowing the network to be completely decentralized without the synchronization imposed by the central task allocator.

Acknowledgment

The authors would like to thank Videotec S.p.A for donating the PT cameras and in particular Piero Donaggio, Arrigo Guizzo, Luca Galeazzo, and Giambattista Gennari for the collaboration on applying our framework in one of their surveillance setups. The research was partially supported by the European Commission under the European project MoVeS and the Swiss National Science Foundation under Grant No. 200021_137876.

References

[1] Valera, M., and Velastin, S., 2005, "Intelligent Distributed Surveillance Systems: A Review," Proc. IEEE Vision, Image Signal Process., **152**(2), pp. 192–204.

[2] Wang, X., 2012, "Intelligent Multi-Camera Video Surveillance: A Review," Pattern Recognit. Lett., **34**(1), pp. 3–19.

[3] Yilmaz, A., Javed, O., and Shah, M., 2006, "Object Tracking: A Survey," ACM Comput. Surv. (CSUR), **38**(4), pp. 1–45.

[4] Mazor, E., Averbuch, A., Bar-Shalom, Y., and Dayan, J., 1998, "Interacting Multiple Model Methods in Target Tracking: A Survey," IEEE Trans. Aerosp. Electron. Syst., **34**(1), pp. 103–123.

[5] Micheloni, C., Foresti, G., and Snidaro, L., 2005, "A Network of Co-Operative Cameras for Visual Surveillance," Proc. IEEE Vision, Image Signal Process., **152**(2), pp. 205–212.

[6] Krahnstoever, N., Yu, T., Lim, S.-N., Patwardhan, K., and Tu, P., 2008, "Collaborative Real-Time Control of Active Cameras in Large Scale Surveillance Systems," ECCV Workshop on Multi-Camera and Multi-Modal Sensor Fusion, Marseille, France, Oct. 18, Paper No. M2SFA2.

[7] Xu, Y., and Song, D., 2010, "Systems and Algorithms for Autonomous and Scalable Crowd Surveillance Using Robotic PTZ Cameras Assisted by a Wide-Angle Camera," Autonom. Rob., **29**(1), pp. 53–66.

[8] Park, J., Bhat, P. C., and Kak, A. C., 2006, "A Look-Up Table Based Approach for Solving the Camera Selection Problem in Large Camera Networks," Proceedings of the International Workshop on Distributed Smart Cameras (DCS'06), Boulder, CO, Oct. 31, pp. 72–76.

[9] Bellotto, N., Benfold, B., Harland, H., Nagel, H.-H., Pirlo, N., Reid, I., Sommerlade, E., and Zhao, C., 2012, "Cognitive Visual Tracking and Camera Control," Comput.Vision Image Understanding, **116**(3), pp. 457–471.

[10] Soto, C., Song, B., and Roy-Chowdhury, A. K., 2009, "Distributed Multi-Target Tracking in a Self-Configuring Camera Network," Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR), Miami, FL, June 20–25, pp. 1486–1493.

[11] Morye, A., Ding, C., Song, B., Roy-Chowdhury, A., and Farrell, J., 2011, "Optimized Imaging and Target Tracking Within a Distributed Camera Network," Proceedings of American Control Conference (ACC), San Francisco, CA, June 29–July 1, pp. 474–480.

[12] Morye, A., Ding, C., Roy-Chowdhury, A., and Farrell, J., 2013, "Constrained Optimization for Opportunistic Distributed Visual Sensing," Proceedings of American Control Conference (ACC), Washington, DC, June 17–19, pp. 6294–6301.

[13] Kirubarajan, T., Bar-Shalom, Y., Pattipati, K., and Kadar, I., 2000, "Ground Target Tracking With Variable Structure IMM Estimator," IEEE Trans. Aerosp. Electron. Syst., **36**(1), pp. 26–46.

[14] Avni, O., Borrelli, F., Katzir, G., Rivlin, E., and Rotstein, H., 2008, "Scanning and Tracking With Independent Cameras—A Biologically Motivated Approach Based on Model Predictive Control," Autonom. Rob., **24**(3), pp. 285–302.

[15] Raimondo, D. M., Gasparella, S., Sturzenegger, D., Lygeros, J., and Morari, M., 2010, "A Tracking Algorithm for PTZ Cameras," Proceedings of Estimation and Control of Networked Systems, Annecy, France, Sept. 13–14, pp. 61–66.

[16] Wang, H., Kirubarajan, T., and Bar-Shalom, Y., 1999, "Precision Large Scale Air Traffic Surveillance Using IMM/Assignment Estimators," IEEE Trans. Aerosp. Electron. Syst., **35**(1), pp. 255–266.

[17] Chen, B., and Tugnait, J. K., 2001, "Tracking of Multiple Maneuvering Targets in Clutter Using IMM/JPDA Filtering and Fixed-Lag Smoothing," Automatica, **37**(2), pp. 239–249.

[18] Natarajan, P., Hoang, T. N., Low, K. H., and Kankanhalli, M., 2012, "Decision-Theoretic Approach to Maximizing Observation of Multiple Targets in Multi-

Camera Surveillance," Proceedings of International Conference on Autonomous Agents and Multiagent Systems-Vol. 1, IFAAMAS, Valencia, Spain, June 4–8, pp. 155–162.

[19] Ozay, N., Topcu, U., Murray, R. M., and Wongpiromsarn, T., 2011, "Distributed Synthesis of Control Protocols for Smart Camera Networks," Proceedings of International Conference on Cyber-Physical Systems (ICCPS), IEEE/ACM, Chicago, IL, Apr. 12–14, pp. 45–54.

[20] Hespanha, J. P., Kim, H. J., and Sastry, S., 1999, "Multiple-Agent Probabilistic Pursuit-Evasion Games," Proceedings of the Conference on Decision and Control, Vol. 3, Phoenix, AZ, Dec. 7–10, pp. 2432–2437.

[21] Hespanha, J. P., Prandini, M., and Sastry, S., 2000, "Probabilistic Pursuit-Evasion Games: A One-Step NASH Approach," Proceedings of Conference on Decision and Control, Vol. 3, Sydney, NSW, Australia, Dec. 12–15, pp. 2272–2277.

[22] Spindler, M., Pasqualetti, F., and Bullo, F., 2012, "Distributed Multi-Camera Synchronization for Smart-Intruder Detection," American Control Conference (ACC), Montreal, QC, Canada, June 27–29, pp. 5120–5125.

[23] Zanella, F., Pasqualetti, F., Carli, R., and Bullo, F., 2012, "Simultaneous Boundary Partitioning and Cameras Synchronization for Optimal Video Surveillance," Proceedings of Workshop on Distributed Estimation and Control in Networked Systems, IFAC, Santa Barbara, CA, Sept. 14–15, pp. 1–6.

[24] Raimondo, D. M., Kariotoglou, N., Summers, S., and Lygeros, J., 2011, "Probabilistic Certification of Pan-Tilt-Zoom Camera Surveillance Systems," Proceedings of Conference on Decision and Control and European Control Conference (CDC-ECC), Orlando, FL, Dec. 12–15, pp. 2064–2069.

[25] Cenedese, A., Cerruti, F., Fabbro, M., Masiero, C., and Schenato, L., 2010, "Decentralized Task Assignment in Camera Networks," Proceedings of Conference on Decision and Control (CDC), Atlanta, GA, Dec. 15–17, pp. 126–131.

[26] Alberton, R., Carli, R., Cenedese, A., and Schenato, L., 2012, "Multi-Agent Perimeter Patrolling Subject to Mobility Constraints," Proceedings of American Control Conference (ACC), Montreal, QC, Canada, June 27–29, pp. 4498–4503.

[27] Park, M., Kalyanam, K., Darbha, S., Khargonekar, P., Pachter, M., and Chandler, P., 2012, "Sub-Optimal Stationary Policies for a Class of Stochastic Optimization Problems Arising in Robotic Surveillance Applications," ASME Paper No. DSCC2012-MOVIC2012-8610.

[28] Schwager, M., Julian, B. J., Angermann, M., and Rus, D., 2011, "Eyes in the Sky: Decentralized Control for the Deployment of Robotic Camera Networks," Proc. IEEE, **99**(9), pp. 1541–1561.

[29] Abate, A., Prandini, M., Lygeros, J., and Sastry, S., 2008, "Probabilistic Reachability and Safety for Controlled Discrete Time Stochastic Hybrid Systems," Automatica, **44**(11), pp. 2724–2734.

[30] Summers, S., and Lygeros, J., 2010, "Verification of Discrete Time Stochastic Hybrid Systems: A Stochastic Reach-Avoid Decision Problem," Automatica, **46**(12), pp. 1951–1961.

[31] Summers, S., Kamgarpour, M., Tomlin, C., and Lygeros, J., 2013, "Stochastic System Controller Synthesis for Reachability Specifications Encoded by Random Sets," Automatica, **49**(9), pp. 2906–2910.

[32] Kariotoglou, N., Raimondo, D. M., Summers, S., and Lygeros, J., 2011, "A Stochastic Reachability Framework for Autonomous Surveillance With Pan-Tilt-Zoom Cameras," Proceedings of Conference on Decision and Control and European Control Conference (CDC-ECC), Orlando, FL, Dec. 12–15, pp. 1411–1416.

[33] Kariotoglou, N., Summers, S., and Raimondo, D. M., 2013, "Hierarchical Task Allocation for Multi-Agent Systems Encoded by Stochastic Reachability Specifications," Proceedings of European Control Conference (ECC), EUCA, Zurich, Switzerland, July 17–19, pp. 2777–2782.

[34] Kariotoglou, N., Summers, S., Summers, T., Kamgarpour, M., and Lygeros, J., 2013, "Approximate Dynamic Programming for Stochastic Reachability," Proceedings of European Control Conference (ECC), Zurich, Switzerland, July 17–19, pp. 584–589.

[35] Powell, W. B., 2007, *Approximate Dynamic Programming: Solving the Curses of Dimensionality*, Vol. 703, Wiley, New York.

[36] Bertsekas, D. P., 2012, *Dynamic Programming and Optimal Control*, Vol. 2. Athena Scientific, Belmont, MA.

[37] Papadimitriou, C. H., and Steiglitz, K., 1998, *Combinatorial Optimization: Algorithms and Complexity*, Courier Dover Publications, New York.

[38] Ausiello, G., 1999, *Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties*, Springer, London, UK.