Distributed Synthesis of Surveillance Strategies for Mobile Sensors

Suda Bharadwaj[1] and Rayna Dimitrova[2] and Ufuk Topcu[1]

Abstract— **We study the problem of synthesizing strategies for a mobile sensor network to conduct surveillance in partnership with static alarm triggers. We formulate the problem as a multi-agent reactive synthesis problem with surveillance objectives specified as a temporal logic formula. In order to avoid the state space blow-up arising from a centralized strategy computation, we propose a method to *decentralize* the surveillance strategy synthesis by decomposing the multi-agent game into subgames that can be solved independently. We also decompose the global surveillance specification into local specifications for each sensor and show that sensors satisfying their local surveillance specifications guarantees the sensor network as a whole will satisfy the global surveillance objective. Thus, our method is able to guarantee global surveillance properties in a mobile sensor network while synthesizing completely decentralized strategies with no need for communication between the sensors. We also present a case study where we demonstrate an application of decentralized surveillance strategy synthesis.**

I. INTRODUCTION

The importance of surveillance in our daily life has been constantly growing in the past couple of decades, and with that, also the need for more efficient and sophisticated mechanisms for surveillance. One of the major challenges comes from the need to perform surveillance in large and complex environments, where it is not always feasible or cost effective to have complete surveillance coverage of the entire area at all times. Furthermore, sensors might not always be able to classify threats, and often require human intervention to assess the threat level. It can thus be necessary to deploy multiple mobile sensors, that work together with conventional static sensors to maintain a sufficient level of knowledge on the location of a potential threat. This is particularly crucial in applications where it is necessary to monitor a potential threat which can move over a large area until it can be accounted for.

In a formal setting, designing a surveillance strategy for a (mobile) sensor network dealing with a potentially adversarial target can be modelled as a two-player game in which one player represents the sensor network and the other player represents the adversary. There are several variants of such games, including pursuit-evasion games [1] and graph-searching games [2]. In such games, the problem is formulated as enforcing eventual detection, which is, in essence, a search problem – once the target is detected, the game ends. These types of games are too restrictive for applications where the goal is not to capture, but instead to maintain information about the location of the adversary for an unbounded time horizon.

[1]Suda Bharadwaj and Ufuk Topcu are with the University of Texas at Austin

[2]Rayna Dimitrova is with the University of Leicester, UK.

Another class of games used in physical security are *Stackelberg* games, also known as leader-follower games. In such games the defender acts first, for example by placing their defence system, and the attacker follows with his action, possibly after obtaining information about the placed defence system. In recent years Stackelberg games have seen use in, among others, LAX airport, [3] and the US Coast Guard [4]. These games aim to compute randomized policies for the defender to protect target locations from an attacker. Extensions of this model [5] have been proposed to generate infinite-horizon patrolling strategies either for mobile resources alone or in concert with static alarm triggers [6], [7]. However, these models cannot be used to reason about the uncertain set of possible locations of dynamic threats.

Our objective in this work is not to just compute a patrolling strategy for the mobile sensors, but also to quantify the sensor network's knowledge of the possible locations of active threats and use this information to synthesize strategies for the mobile sensors that provide knowledge guarantees on the threat location over an infinite-time horizon.

As a motivating case study in this paper, we consider the use of autonomous drones working in cooperation with static sensors in wildlife conservation. UAVs are increasingly being adopted for monitoring of illegal hunting and poaching [8], though they are mostly remotely controlled [9]. In Kenya, for example, remotely controlled drones were deployed in 2014 in an attempt to reduce poaching by providing constant surveillance [10], allowing authorities to arrest rhino poachers when they are sensed by the drones. Autonomous UAVs have not been used in this setting yet, and proposed plans involve drones following pre-programmed paths [11]. In this paper, we propose a method for automatically constructing *autonomous reactive surveillance* strategies for multiple mobile sensors (like UAVs) working in concert with static sensors in the field.

We study the problem of synthesizing strategies for enforcing *temporal surveillance objectives*, such as the requirement to never let the sensor network's uncertainty about the target's location exceed a given threshold, or recapturing the target every time it escapes. To this end, we consider surveillance objectives specified in linear temporal logic (LTL), equipped with basic surveillance predicates. Our computational model is that of a two-player game played on a finite graph, whose nodes represent the joint possible locations of all the mobile sensors and the target, and whose edges model the possible (deterministic) moves between locations. The mobile sensors play the game with partial information, as they can only observe the target when it is in the area of sight of one of the sensors. The target, on the other hand, always has full information about the location of the entire sensor network. In that way, we consider a model with one-sided partial information, making the computed strategy for the agent robust against a potentially more powerful adversary.

We formulate surveillance strategy synthesis as the problem of computing a joint winning strategy for the multiple mobile sensors in a partial-information game with a surveillance objective. Partial-information games with LTL objectives have been well studied [12], [13] and it is well known that the synthesis problem is EXPTIME-hard [14], [15]. In a companion submission to CDC 2018 we describe a framework for formalizing *single-agent* surveillance synthesis as a two-player game with partial information, and propose an abstraction-based method for solving such games. The interested reader is referred to the extended version [16] for details about the abstraction-based synthesis method. The price of resorting to abstraction is the potential overapproximation of the set of possible target locations (that is, loss of precision in the sensors' knowledge) which may make satisfying surveillance requirements difficult. There is thus a trade-off between strictness of surveillance requirements, i.e, how closely a target needs to be tracked, and the size of the abstract game necessary for a surveillance strategy to exist.

Sensor networks with a large number of dynamic sensors as well as static sensors can achieve better coverage, and thus, in general, can make do with much coarser abstractions to satisfy a given surveillance objective. However, even when using abstraction, the size of the game is exponential in the number of sensors. To address the blow-up of the state space incurred by large number of sensors, we propose a *decentralized* synthesis method that aims to compute a surveillance strategy for each mobile sensor separately. Our main contribution is as follows:

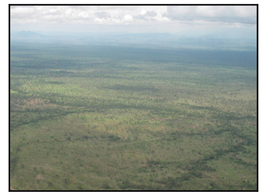
We decompose the original surveillance game into a set of subgames, one for each sensor. Accordingly, the global surveillance objective is broken up into a local objective for each subgame. Our reduction guarantees, that if the local strategy in each subgame satisfies the local surveillance objective, then the composition of the strategies fulfills the global surveillance objective. This allows us to solve each subgame under its local surveillance objective independently, using off-the-shelf reactive synthesis tools.

There has been work in decentralized synthesis for GR(1) specifications, however, the synthesis process often involves a centralized computation as in [17] or synchronization [18], [19]. Our approach, on the other hand is fully decentralized and the sensors require no communication as simply satisfying their local properties guarantees the global objective.

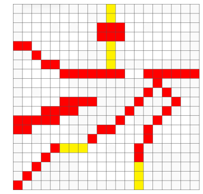
II. MOTIVATING CASE STUDY

We first describe the multi-agent surveillance synthesis problem informally, in the context of a motivating case study.

We consider wildlife conservation in Africa, in particular, at the Selous Game Reserve (SGR) located in Tanzania, where the African Black Rhinoceros population is under serious threat due to poaching. We are motivated by a recommended anti-poaching initiative in the SGR by the World Heritage Centre, to study the use of a sensor network for tracking the position of a potential poacher with user-specified precision. Since the SGR is a very large area, the network consists of both mobile and static sensors. We apply the distributed synthesis method proposed in this paper to synthesize surveillance strategies for the mobile sensors that satisfy the desired tracking requirement.



(a) SGR interior landscape [20]



(b) Gridworld representation of landscape in 1a

Fig. 1: The landscape in 1a is coarsely represented as the gridworld in 1b. The red regions represent impassable terrain. The yellow areas covered are by static sensors.

Figure 1 shows a section of the SGR that we represent as a gridworld which will form the state space of the game. Each static sensor monitors a given area of the grid (shown in yellow) and detects any presence of the target (i.e., threat) in these states, but cannot determine the target's exact location. The tracking requirement is to ensure that over and over again, the set of potential locations of the target is reduced to 5 cells. In other words, every time the target escapes from the vision of all the sensors, the network guarantees that eventually the uncertainty about its position will be reduced to 5 grid cells.

III. GAMES WITH SURVEILLANCE OBJECTIVES

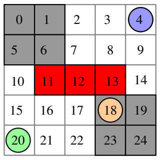
We begin by providing a formal model for describing multi-agent surveillance strategy synthesis problems, in the form of a two-player game between the mobile sensors in a network and a target, in which the sensors have partial information about the target's location.

A. Multi-Agent Surveillance Game Structures

We define a *multi-agent surveillance game structure* to be a tuple $G = (S, s^{\mathsf{init}}, T, vis_1, \ldots, vis_n)$ where:

- $S = L_s \times L_t$ is the set of states, where $L_s = L_1 \times L_2 \times \cdots \times L_n$ is the set of joint locations of the n mobile sensors, L_i is the set of possible locations of sensor i, and L_t is the set of possible locations of the target;
- $s^{\mathsf{init}} = (l_1^{\mathsf{init}}, \ldots, l_n^{\mathsf{init}}, l_t^{\mathsf{init}})$ is the initial state;
- $T \subseteq S \times S$ is the transition relation describing the possible joint moves of the sensors and the target; and
- $vis_1, \ldots, vis_n$ are the *visibility functions* for the n sensors, where $vis_i : L_i \times L_t \to \mathbb{B}$ maps a state (l_i, l_t) to *true* iff *position* l_t *is in the area of sight of* l_i.

Additionally, we define the *joint visibility function* $Vis : S \to \mathbb{B}$ that maps a state (l, l_t) to *true* if the set $\mathcal{I} = \{i \mid vis_i(l_i, l_t) = true\}$ is non-empty. Informally, $Vis(l_s, l_t)$ is *true* if the target is in view of *at least one* of the sensors.



(a) Surveillance arena

$$((20,4),\{18\}) \rightarrow \begin{cases} ((15,3),\{19,23\}) \\ ((15,3),\{19,23\}) \\ ((21,9),\{19,23\}) \\ ((21,9),\{17\}) \\ ((15,9),\{19\}) \\ ((21,9),\{18,23\}) \end{cases}$$

(b) Some possible transitions from the initial state in the belief-set game from Example 2. Note that, since the set of static sensors is empty, it is omitted from the states. For the sake of readability, some transitions are excluded.

Fig. 2: A simple surveillance game on a grid arena. Obstacles are shown in red. There are two sensors (at locations 20 and 4) coloured in blue and green respectively and the target (at location 18) is orange. The grey states are not visible to either sensor, i.e, $Vis((20,4,l_t)) = \mathit{false}$ for all grey l_t.

The transition relation T encodes the one-step move of the target and the n sensors: First, the target makes a move, and then, the sensors move jointly in a synchronized manner.

We denote with $T{\downarrow}i$ the projection of the transition relation T on the sets of locations of the target and the sensor with index i. Formally, we define $T{\downarrow}i = \{(l_i,l_t) \in L_i \times L_t \mid \exists l_1,\ldots,l_{i-1},l_{i+1},\ldots,l_n : (l_1,\ldots,l_n,l_t) \in T\}$.

For a state $(l_s,l_t) \in S$ we define $succ_t(l_s,l_t)$ to be the set of possible successor locations of the target:

$succ_t(l_s,l_t) = \{l'_t \in L_t \mid \exists l'_s.\ ((l_s,l_t),(l'_s,l'_t)) \in T\}$.

We extend $succ_t$ to sets of locations of the target by stipulating that the set $succ_t(l_s,L)$ consists of all possible successor locations of the target for states in $\{l_s\} \times L_t$. Formally, let $succ_t(l_s,L_t) = \bigcup_{l_t \in L_t} succ_t(l_s,l_t)$.

For a state (l_s,l_t) and a successor location of the target l'_t, we denote with $succ(l_s,l_t,l'_t)$ the set of successor locations of the sensors, given that the target moves to l'_t:

$succ(l_s,l_t,l'_t) = \{l'_s \in L_s \mid ((l_s,l_t),(l'_s,l'_t)) \in T\}$.

We assume that, for every state $s \in S$, there exists a state $s' \in S$ such that $(s,s') \in T$, that is, from every state there is at least one move possible (including self transitions). We also assume, that when the target moves to an invisible location, its position does not influence the possible one-step moves of the sensors. Formally, we require that if $Vis(l_s,l'_t) = Vis(l_s,\widehat{l}'_t) = \mathit{false}$, then $succ(l_s,l_t,l'_t) = succ(l_s,\widehat{l}_t,\widehat{l}'_t)$ for all $l_t,l'_t,\widehat{l}_t,\widehat{l}'_t \in L_t$. This assumption is natural in the setting where each of the sensors can move in one step only to locations that are in its sight.

Example 1: Figure 2 shows an example of a multi-agent surveillance game on a grid. The sets of possible locations L_i and L_t for the each of the sensors and for the target consist of the squares of the grid. The transition relation T encodes the possible one-step moves of all the sensors and the target on the grid, and incorporates all desired constraints. For example, moving to a location occupied by another sensor or the target, or to an obstacle, is not allowed. In this example, the function vis_i encodes straight-line visibility with a range of 2: a location l_t is visible to sensor i from location l_i if there is no obstacle on the straight line between them and the distance between the target and sensor i is not larger than 2. Initially the target is not in the area of sight of the sensors, but the initial position of the target is known. However, once the target moves to one of the locations reachable in one step, in this case, locations $\{17,19,23\}$, this might no longer be the case. More precisely, if the target moves to location 17, then the green sensor observes its location, but if it moves to one of the other two locations, then neither sensor can observe it, and its exact location will not be known. ■

B. Static Sensors

We now describe how we incorporate static sensors in the multi-agent surveillance game framework. Given a multi-agent surveillance game structure G as defined previously, a *static sensor M operating over a set of locations Λ* is a function $M_\Lambda : L_t \rightarrow \mathbb{B}$ that maps the target location to true iff it belongs to the set over which the sensor operates. Formally, given a set $\Lambda \subseteq L_t$ of target locations, we define $M_\Lambda(l_t) = \mathit{true}$ iff $l_t \in \Lambda$.

A surveillance game can have multiple (or none) static sensors. Let $\mathcal{M} = \{M_{\Lambda_1},\ldots,M_{\Lambda_m}\}$ be the set of m static sensors for a multi-agent surveillance game structure. We define $J(l_t)$ to be the set of all indices of static sensors such that l_t belongs to the corresponding set of locations, i.e, $J(l_t) = \{j \mid M_{\Lambda_j}(l_t) = \mathit{true}\}$. We refer to $J(l_t)$ as the set of *triggered static sensors* at location l_t. We also define $J(L) = \bigcup_{l_t \in L} J(l_t)$ for a set of locations $L \subseteq L_t$.

We assume that sensors do not suffer from false positives or negatives (studying these is an avenue for future work). Thus, by definition we have that the target must lie in the intersection of the state spaces of the triggered sensors, i.e., $l_t \in \bigcap_{j \in J(l_t)} \Lambda_j$. Also, when no static sensor is triggered, then we know that the target cannot be in the union of the state spaces in which they operate, i.e, $l_t \notin \bigcup_i^m \Lambda_i$.

C. Belief-Set Game Structures

In surveillance strategy synthesis, we need to state properties of, and reason about, the information which the sensors have, i.e, the *belief* about the location of the target. To this end, we can employ a powerset construction which is commonly used to transform a partial-information game into a perfect-information one, by explicitly tracking the joint knowledge of the sensors as a set of possible locations of the target. In that way we define a two-player game in which one player represents the whole sensor network, and the other player represents the target.

Given a set A, we denote with $\mathcal{P}(A) = \{A' \mid A' \subseteq A\}$ the powerset (set of all subsets) of A.

Given a multi-agent surveillance game structure $G = (S, s^{\mathsf{init}}, T, vis_1,\ldots,vis_n)$ with m static sensors $\mathcal{M} = \{M_{\Lambda_1},\ldots,M_{\Lambda_m}\}$, we define the corresponding *belief-set game structure* $G_{\mathsf{belief}} = (S_{\mathsf{belief}}, s^{\mathsf{init}}_{\mathsf{belief}}, T_{\mathsf{belief}})$ where:

- $S_{\mathsf{belief}} = L_s \times \mathcal{P}(L_t) \times \mathcal{P}(\{1,\ldots,m\})$ is the set of states, where L_s is the set of joint locations of the sensors, and $\mathcal{P}(L_t)$ the set of *belief sets* describing information about the location of the target, and $\mathcal{P}(\{1,\ldots,m\})$ is the set of possible sets of triggered sensors;
- $s^{\mathsf{init}}_{\mathsf{belief}} = (l^{\mathsf{init}}_1,\ldots,l^{\mathsf{init}}_n,\{l^{\mathsf{init}}_t\},J(l^{\mathsf{init}}_t))$ as initial state;

- $T_{\mathsf{belief}} \subseteq S_{\mathsf{belief}} \times S_{\mathsf{belief}}$ is the transition relation where $((l_s, B_t, J), (l'_s, B'_t, J')) \in T_{\mathsf{belief}}$ iff $l'_s \in succ(l_s, l_t, l'_t)$ for some $l_t \in B_t$ and $l'_t \in B'_t$, $J' \subseteq J(B'_t)$, and one of these following conditions is satisfied:
  (1) $B'_t = \{l'_t\}$, $l'_t \in succ_t(l_s, B_t)$, $Vis(l_s, l'_t) = true$;
  (2) $B'_t = \{l'_t \in succ_t(l_s, B_t) \mid Vis(l_s, l'_t) = false\} \cap \bigcap_{j \in J'} \Lambda_j$, and $J' \neq \emptyset$;
  (3) $B'_t = \{l'_t \in succ_t(l_s, B_t) \mid Vis(l_s, l'_t) = false\} \setminus \bigcup_{j=1}^{m} \Lambda_j$, and $J' = \emptyset$.

Condition (1) captures the successor locations of the target that can be observed from one of the mobile sensors' current locations. Condition (2) captures the cases when the target moves to a location that cannot be observed by the mobile sensors, but triggers a non-empty set J' of static sensors. Finally, condition (3) corresponds to the successor locations of the target not visible from the current location of any of the mobile sensors, and not triggering any static sensors.

Example 2: Consider the surveillance game structure from Example 1. The initial belief set is $\{18\}$, as the target's initial position is known. Figure 2b shows some of the possible successor states of the initial state $((20, 4), \{18\})$ in the belief-set game G_{belief}. ■

Based on T_{belief}, we can define the functions $succ_t : S_{\mathsf{belief}} \to \mathcal{P}(\mathcal{P}(L_t) \times \mathcal{P}(\{1, \ldots, m\}))$ and $succ : S_{\mathsf{belief}} \times \mathcal{P}(L_t) \times \mathcal{P}(\{1, \ldots, m\}) \to \mathcal{P}(L_s)$ similarly to the corresponding functions defined for G.

A *run* in G_{belief} is an infinite sequence $s_0, s_1, \ldots$ of states in S_{belief}, where $s_0 = s^{\mathsf{init}}_{\mathsf{belief}}$, $(s_i, s_{i+1}) \in T_{\mathsf{belief}}$ for all i.

A *strategy for the target in* G_{belief} is a function $f_t : S^{+}_{\mathsf{belief}} \to \mathcal{P}(L_t) \times \mathcal{P}(\{1, \ldots, m\})$ such that $f_t(\pi \cdot s) = (B_t, J)$ implies $(B_t, J) \in succ_t(s)$ for every $\pi \in S^{*}_{\mathsf{belief}}$ and $s \in S_{\mathsf{belief}}$. That is, a strategy for the target suggests a move resulting in some belief set reachable from a location in the current belief, and a set of triggered sensors.

A *joint strategy for the sensors in* G_{belief} is a function $f_s : S^{+}_{\mathsf{belief}} \times \mathcal{P}(L_t) \to S_{\mathsf{belief}}$ such that, if, $f_s(\pi \cdot s, B_t, J) = (l'_s, B'_t, J'')$ then, $B'_t = B_t$, $J' = J$, and $l'_s \in succ_s(s, B_t)$ for every $\pi \in S^{*}_{\mathsf{belief}}$, $s \in S_{\mathsf{belief}}$ and $B_t \in \mathcal{P}(L_t)$. Intuitively, a strategy for the sensors suggests a joint move based on the observed history of the play, the current belief about the target's position, and the set of currently triggered sensors.

The outcome of given strategies f_s and f_t for the sensors and the target in G_{belief}, denoted $outcome(G_{\mathsf{belief}}, f_s, f_t)$, is a run $s_0, s_1, \ldots$ of G_{belief} such that for every $i \geq 0$, we have $s_{i+1} = f_s(s_0, \ldots, s_i, B^i_t)$, where $B^i_t = f_t(s_0, \ldots, s_i)$.

D. Temporal Surveillance Objectives

We consider a set of *surveillance predicates* $\mathcal{SP} = \{p_b \mid b \in \mathbb{N}_{>0}\}$, where for $b \in \mathbb{N}_{>0}$ we say that a state (l_s, B_t) in the belief game structure satisfies p_b (denoted $(l_s, B_t) \models p_b$) iff $|\{l_t \in B_t \mid Vis(l_s, l_t) = false\}| \leq b$. Intuitively, p_b is satisfied by the states in the belief game structure where the size of the belief set does not exceed the threshold $b \in \mathbb{N}_{>0}$.

We study surveillance objectives expressed in a fragment of linear temporal logic (LTL) over surveillance predicates. We consider *safety surveillance objectives* expressed using the temporal operator $\Box$ and *liveness surveillance objectives* expressed using the temporal operators $\Box$ and $\Diamond$.

A *safety surveillance objective* $\Box p_b$ requires that the size of the belief-set never exceeds the given threshold b. More formally, an infinite sequence of states $s_0, s_1, \ldots$ in G_{belief} satisfies the safety property $\Box p_b$ if and only if for every $i \geq 0$ it holds that $s_i \models p_b$. A *liveness surveillance objective* $\Box\Diamond p_b$, on the other hand, requires that the size of the belief is smaller or equal to the bound b infinitely often. That is, $s_0, s_1, \ldots$ in G_{belief} satisfies $\Box\Diamond p_b$ if for every $i \geq 0$ there exists $j > i$ such that $s_j \models p_b$.

In this paper we consider safety and liveness surveillance objectives, as well as conjunctions of such objectives. We remark the following equivalences of surveillance objectives:

- $\Box p_a \wedge \Box p_b \equiv \Box p_{\min(a,b)}$;
- $\Box\Diamond p_a \wedge \Box\Diamond p_b \equiv \Box\Diamond p_{\min(a,b)}$;
- if $a \leq b$, then $\Box p_a \wedge \Box\Diamond p_b \equiv \Box p_a$.

Using these equivalences, we can restrict our attention to surveillance objectives of one the following forms: $\Box p_b$, $\Box\Diamond p_b$ or $\Box p_a \wedge \Box\Diamond p_b$, where $a > b$.

E. Multi-Agent Surveillance Synthesis Problem

A *multi-agent surveillance game* is a triple $(G, \mathcal{M}, \varphi)$, where G is a surveillance game structure, $\mathcal{M}$ is a set of static sensors, and φ is a surveillance objective. A *winning strategy for the sensors for* $(G, \mathcal{M}, \varphi)$ is a joint strategy f_s for the sensors in the corresponding belief-set game structure G_{belief} such that for every strategy f_t for the target in G_{belief} it holds that $outcome(G_{\mathsf{belief}}, f, f_t) \models \varphi$. Analogously, a *winning strategy for the target for* $(G, \mathcal{M}, \varphi)$ is a strategy f_t such that, for every strategy f_s for the mobile sensors in G_{belief}, it holds that $outcome(G_{\mathsf{belief}}, f_s, f_t) \not\models \varphi$.

Problem statement: Given a multi-agent surveillance game $(G, \mathcal{M}, \varphi)$, compute a joint strategy for the mobile sensors that is winning for the game $(G, \mathcal{M}, \varphi)$.

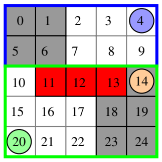
In the remainder of the paper we show how to solve the multi-agent surveillance synthesis problem in a compositional manner. The key idea is to decompose the problem into a set of single-sensor surveillance games over smaller sets of locations, and solve each of these games separately.

IV. DISTRIBUTED SURVEILLANCE GAMES

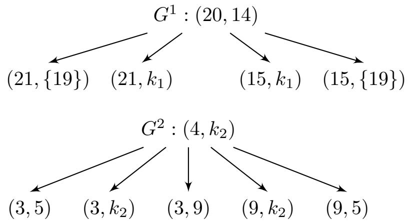
In the sequel we assume that $L_1 = L_2 = \cdots = L_t \triangleq L$ in the surveillance game structure, i.e, all n sensors and the target operate in the same state space. For the remainder of the paper, let $G = (S, s^{\mathsf{init}}, T, vis_1, \ldots, vis_n)$ be a multi-agent surveillance game structure defined over L, and let $\mathcal{M} = \{M_{\Lambda_1}, \ldots, M_{\Lambda_m}\}$ be a set of static sensors. We define a *state-space partition* of size n of the set L of locations in a game structure G to be a tuple $\widetilde{L} = (\widetilde{L}_1, \ldots, \widetilde{L}_n)$ of subsets of L_i such that $\bigcup_i^n \widetilde{L}_i = L$, and $\widetilde{L}_i \cap \widetilde{L}_j = \emptyset$ for $i \neq j$.

A. Surveillance Subgames

We now describe how, given a state-space partition $\widetilde{L} = (\widetilde{L}_1, \ldots, \widetilde{L}_n)$, to construct a tuple of single-agent surveillance game structures $\widetilde{G} = (G^1, \ldots, G^n)$ that contains one *surveillance subgame* G^i for each mobile sensor i. Each



(a) Multi-agent surveillance game partitioned into two subgames.



(b) Transitions from initial state in subgames.

Fig. 3: Partitioning of the state space of a surveillance game into two subgames with locations $\widetilde{L}_1$ (green) and $\widetilde{L}_2$ (blue).

subgame, G^i is defined over the subset of locations $\widetilde{L}_i$. Since the target and sensors operate on the same state space we will have $\widetilde{L}^i_s = \widetilde{L}^i_t = \widetilde{L}_i$. Additionally, to each $\widetilde{L}^i_t$ we add an *auxiliary location* k_i that encapsulates all possible locations of the target that are outside of this subgame's region, i.e., all locations in $L \setminus \widetilde{L}_i$. We then model transitions leaving or entering $\widetilde{L}^i_t$ as transitions to or from location k_i respectively. We require that the initial location l^{init}_i of sensor i is in $\widetilde{L}_i$.

Formally, given a subset $\widetilde{L}_i \subseteq L$ we define the *subgame* of G corresponding to sensor i as the tuple $G^i = (\widetilde{S}_i, \widetilde{s}^{\mathsf{init}}_i, \widetilde{T}_i, \widetilde{vis}_i)$ where:

- $\widetilde{S}_i = \widetilde{L}_i \times (\widetilde{L}^i_t \cup k_i)$ is the set of states.
- $\widetilde{s}^{\mathsf{init}}_i = (l^{\mathsf{init}}_i, \widetilde{l}_t)$ is the initial state, where $\widetilde{l}_t = l^{\mathsf{init}}_t$, if $l^{\mathsf{init}}_t \in \widetilde{L}_i$, and $\widetilde{l}_t = k_i$ otherwise.
- The set $\widetilde{T}_i$ consists of two types of transitions: the transitions in $T{\downarrow}i$ that originate and end in the subgame's region are preserved as they are. Transitions of the target exiting or entering $\widetilde{L}^i_t$ are replaced by transitions to and from location k_i respectively, since k_i represents all target locations outside of $\widetilde{L}^i_t$. Formally, for every pair of states $(\widetilde{l}_i, \widetilde{l}_t) \in \widetilde{S}_i$ and $(\widetilde{l}'_i, \widetilde{l}'_t) \in \widetilde{S}_i$ we have that $((\widetilde{l}_i, \widetilde{l}_t), (\widetilde{l}'_i, \widetilde{l}'_t)) \in \widetilde{T}_i$ if and only if there exists a transition $((l_i, l_t), (l'_i, l'_t)) \in T{\downarrow}i$ for which the following conditions are satisfied:
  - if $\widetilde{l}_t \in \widetilde{L}^i_t$ and $\widetilde{l}'_t \in \widetilde{L}^i_t$, then $\widetilde{l}_t = l_t$ and $\widetilde{l}'_t = l'_t$, that is, we have a *transition internal for the region* $\widetilde{L}^i_t$;
  - if $\widetilde{l}_t \in \widetilde{L}^i_t$ and $\widetilde{l}'_t = k_i$, then $l_t \in \widetilde{L}^i_t$ and $l'_t \notin \widetilde{L}^i_t$, that is, we have a *transition exiting the region* $\widetilde{L}^i_t$;
  - if $\widetilde{l}_t = k_i$ and $\widetilde{l}'_t \in \widetilde{L}^i_t$, then $l_t \notin \widetilde{L}^i_t$ and $l'_t \in \widetilde{L}^i_t$, that is, we have a *transition entering the region* $\widetilde{L}^i_t$;
  - if $\widetilde{l}_t = k_i$ and $\widetilde{l}'_t = k_i$, then $l_t \notin \widetilde{L}^i_t$ and $l'_t \notin \widetilde{L}^i_t$, that is, we have a *transition completely outside* $\widetilde{L}^i_t$.
- The visibility function $\widetilde{vis}_i$ in the subgame G^i agrees with the visibility function vis_i of sensor i in the original game when the target's location is in the subgame's region. Target locations outside of the region $\widetilde{L}^i_t$ (summarized by location k_i) are invisible to the sensor in the subgame. Formally, $\widetilde{vis}_i(\widetilde{l}_i, l_t) = vis_i(\widetilde{l}_i, l_t)$ $l_t \in \widetilde{L}^i_t$ and *false* if $l_t = k_i$

Example 3: In Figure 3a, we have two subgames: G^1 for the green mobile sensor and G^2 for the blue one. The states in the subgames are $s_1 = (20, 14)$, and $s_2 = (4, k_2)$. Recall that k_i is an indicator state to represent that the target is not subgame i. The possible transitions shown in Figure 3b show that the target has the ability to leave G^1 and enter G^2. ■

Note that in this construction, sensor i is not able to leave the region of locations $\widetilde{L}_i$. Furthermore, all the information about the target's behaviour outside of the subgame's region is completely hidden from the mobile sensor controller, since all locations outside of $\widetilde{L}^i_t$ are represented by the single location k_i. In section IV-C, we discuss the local knowledge (belief) of sensor i in the game structure G^i.

B. Static Sensors in Subgames

We assumed that all information about the target's behaviour outside of subgame G^i is completely hidden to sensor i. Hence, sensor i is only privy to static sensors that operate in the state space of the subgame G^i, i.e, static sensors M_{Λ_m} where $\Lambda_m \cap \widetilde{L}_i \neq \emptyset$. We define $Q_i = \{i \mid \Lambda_i \cap \widetilde{L}_i \neq \emptyset\}$ to be the set of static sensors operating in subgame G^i. We denote $\widetilde{\Lambda}_q = \Lambda_q \cap \widetilde{L}_i$ and define $J_i(\widetilde{l}_t) \subseteq Q_i$ as the set of triggered static sensors in location $\widetilde{l}_t \in \widetilde{L}_i$ in region i.

C. Local Beliefs in Surveillance Subgames

A surveillance subgame is a game structure with a single mobile sensor and some number of static sensors, and thus, is a special case of multi-agent surveillance game structure. With this, the definition of belief-set game structures from Section III-C directly applies to surveillance subgames.

In the belief-set game structure for a surveillance subgame $G^i = (\widetilde{S}_i, \widetilde{s}^{init}_i, \widetilde{T}_i, \widetilde{vis}_i)$ with static sensors Q_i, the belief sets represent the local belief of sensor i. More specifically, a belief set in G^i_{belief} is an element of $\mathcal{P}(\widetilde{L}^i_t \cup \{k_i\})$, and can thus contain the auxiliary location. Intuitively, if k_i is present in the sensor's current belief, then the target could possibly be outside of the local set of locations $\widetilde{L}_i$, or if the belief is the singleton $\{k_i\}$, then sensor i knows for sure that the target is outside of its region. Additionally, if there is a triggered static sensor in the region, the sensor will know that the target must be in the state space of the static sensor and k_i cannot be in the belief. We define the *global interpretation* $[\![B_t]\!]$ of a belief set B_t in G^i_{belief}, which is a set of locations in G, as

$$[\![B_t]\!] = \begin{cases} B_t & \text{if } k_i \notin B_t \\ B_t \cup (L \setminus \widetilde{L}_i) & \text{if } k_i \in B_t. \end{cases}$$

Strategies of sensor i in the belief-set game G^i_{belief} depend only on the sequence of states in this game, and thus, only on local information. Following the definitions in Section III-C, the outcome of a pair of given strategies f_i and f_{t_i} for the sensor and the target in G^i_{belief} is a sequence of states in G^i_{belief}, each of which is a pair consisting of a location of sensor i and a belief-set for sensor i in G^i_{belief}.

D. Distributed Surveillance Synthesis Problem

Given a state-space partitioning $\widetilde{L}$ and the corresponding tuple of subgames $\widetilde{G} = (G^1, \ldots, G^n)$ we will define a distributed surveillance strategy synthesis problem, which, intuitively, asks to synthesize strategies for the sensors in the individual belief subgames, such that together they guarantee the global surveillance objective. In this section we formalize this intuitive problem description. We first need to define what it means for the individual sensor strategies to jointly satisfy together a global requirement.

The surveillance requirements are defined in terms of the belief-states in G_{belief}, but strategies in the belief subgames are defined in terms of sequences of local belief states. Hence, we need to define a mapping of states of the form $((l_1, \ldots, l_n), B_t, J)$ to elements of $\mathcal{P}(\widetilde{L}_t^i \cup \{k_i\})$ for each i. Since, by definition, a strategy for sensor i in the corresponding belief subgame guarantees that it remains in $\widetilde{L}_i$, we only need to define the mapping for states $l_i \in \widetilde{L}_i$.

Formally, for a state $((l_1, \ldots, l_n), B_t, J)$ we define its projection on belief subgame i as $((l_1, \ldots, l_n), B_t, J){\downarrow}i = (l_i, B_t{\downarrow}i, J \cap Q_i)$, where

$$B_t{\downarrow}i = \begin{cases} B_t & \text{if } B_t \subseteq \widetilde{L}_i, \\ (B_t \cap \widetilde{L}_i) \cup \{k_i\} & \text{otherwise.} \end{cases}$$

The mapping extends to sequences of states in the usual way.

Intuitively, this mapping projects the joint knowledge of the sensors in G_{belief} onto the local belief of each sensor, where the sensors do not share their local beliefs with each other, that is, the sensors have no information about the target's position outside of their own region. The global, shared belief of the sensors is formed by the combination of their local beliefs. More precisely, this is the intersection of the global interpretation of the local beliefs. Indeed, it is easy to see that the property $B_t = \bigcap_{i=1}^n [\![B_t{\downarrow}i]\!]$ holds.

Now we are ready to define the joint strategy of the sensors in G_{belief} obtained by executing together a given set of sensor strategies in the individual subgames. Let $f_{s_1}, \ldots, f_{s_n}$ be strategies for the sensors in the belief subgames $(G^1_{\tilde{S}_{\mathsf{belief}}}, \ldots, G^n_{\tilde{S}_{\mathsf{belief}}})$. We define the *composition* $f_{s_1} \otimes \ldots \otimes f_{s_n}$ of $f_{s_1}, \ldots, f_{s_n}$, which is a joint strategy f_s for the sensors in G_{belief}, as follows: for every sequence $s_0, \ldots, s_k$ of states in G_{belief} and global belief $B_t \in \mathcal{P}(L_t)$, we let

$$f_s(s_0, \ldots, s_k, B_t, J) = (l_1, \ldots, l_n),$$

where $l_i = f_{s_i}((s_0, \ldots, s_k){\downarrow}i, B_t{\downarrow}i, J \cap Q_i)$ for each i.

Remark. If, for some i, the projection $(s_0, \ldots, s_k){\downarrow}i$ is undefined, then $f_s(s_0, \ldots, s_k, B_t)$ is undefined. However, by the definition of each f_{s_i} we are guaranteed that the projection is defined for every prefix consistent with f_{s_i}.

Intuitively, the joint strategy $f_{s_1} \otimes \ldots \otimes f_{s_n}$ makes decisions consistent with the choices of the individual strategies $f_{s_1}, \ldots, f_{s_n}$ in the respective belief subgames.

Our goal is to synthesize a joint strategy f_s that enforces a given surveillance property in the belief-set game G_{belief} by synthesizing individual strategies for all the sensors in the corresponding belief subgames. That is, we want to solve the following *distributed surveillance synthesis problem*.

Problem statement: Given a multi-agent surveillance game $(G, \mathcal{M}, \varphi)$ with n sensors, and a state-space partition $\widetilde{L}$, compute strategies $f_{s_1}, \ldots, f_{s_n}$ for the sensors in the belief subgames $G^1_{\mathsf{belief}}, \ldots, G^n_{\mathsf{belief}}$ respectively, such that the composed strategy $f_{s_1} \otimes \ldots \otimes f_{s_n}$ is a joint winning strategy for the sensors in the surveillance game $(G, \mathcal{M}, \varphi)$.

Thus, in the distributed surveillance synthesis problem we have to compute strategies $f_{s_1}, \ldots, f_{s_n}$ such that for every strategy f_t for the target in G_{belief} it holds that $outcome(G_{\mathsf{belief}}, f_{s_1} \otimes \ldots \otimes f_{s_n}, f_t) \models \varphi$. To this end, we have to provide *local surveillance objectives* for all the sensors, such that if all strategies are winning with respect to their local objectives, then their composition is winning with respect to the original surveillance objective. In this way we will reduce the multi-agent surveillance synthesis problem to n single-agent surveillance problems over smaller sets of locations. This reduction is the subject of the next section.

V. FROM GLOBAL TO LOCAL SPECIFICATIONS

In order to reduce the multi-agent surveillance synthesis problem for a given surveillance specification φ to solving a number of single-sensor surveillance subgames, we need to provide local surveillance objectives for the individual subgames. The local objectives should be such that by composing the strategies that are winning with respect to the local objectives we should obtain a strategy that is winning for the global surveillance objective. More precisely, we have to provide local surveillance specifications $\varphi_1, \ldots, \varphi_n$ such that if for each i it holds that f_{s_i} is a winning strategy for the sensor in (G^i, Q_i, φ_i), then the strategy $f_{s_1} \otimes \ldots \otimes f_{s_n}$ is a joint winning strategy for the sensors in $(G, \mathcal{M}, \varphi)$.

Recall that the surveillance objective φ is of the form $\Box p_b$, or $\Box\Diamond p_b$, or $\Box p_a \wedge \Box\Diamond p_b$, where $a > b$. We will provide translations for each of these types of specifications.

First, note that the belief sets in a belief subgame G^i_{belief} can contain the auxiliary location k_i, which represents all locations in $L \backslash \widetilde{L}_i$. Thus, when the local belief set contains k_i, the size of the global belief set depends on the local beliefs of the other agents as well. We have to account for this in the translation from global into local surveillance objectives.

Example 4: Consider the global safety surveillance specification $\Box p_5$ in a network with two mobile sensors. In this case we can reduce the multi-agent surveillance problem to two single-agent surveillance games, each of which has $\Box p_3$ as the local specification. To see why, consider the two possible cases of local belief set of sensor 1 whose size is less than or equal to 3. If k_1 is not part of the belief set of sensor 1, then the target is definitely in the region of sensor 1, meaning that the global belief is of size less than or equal to 3, and hence smaller that 5. If, on the other hand, k_1 is part of the local belief of sensor 1, then the target can be in at most 2 locations in $\widetilde{L}_1$. If at the same time we have that the local belief of sensor 2 is of size at most 3, this would guarantee that the size of the global belief does not exceed 5. Local specifications $\Box p_4$, on the other hand do not imply the global specification. Indeed, if at a given point in time both sensors have local beliefs of size 4, each of which contains the corresponding location k_i, the resulting global belief will be of size 6 and thus violate the global specification. ■

Generalizing the observations made in this example, for any number of sensors $n \geq 2$ and global safety surveillance objective $\Box p_b$, we define the local safety surveillance objective for each of the sensors, denoted $local(\Box p_b, n)$, as

$local(\Box p_b, n) \triangleq \Box p_c$, where $c = \lfloor \frac{b}{n} \rfloor + 1$. Since $n \geq 2$ and $b > 0$, we have $c \leq b$.

Note that this translation is conservative, since if according to the belief of sensor i the target could be outside its region, it should guarantee that the number of locations in its own region the target could be in is at most $\lfloor \frac{b}{n} \rfloor$, even if the target can possibly be in only one of the other regions. This conservativeness is necessary to guarantee soundness in the absence of communication between the sensors.

We now turn to liveness surveillance objectives. It is easy to see that each sensor guaranteeing a small enough local belief infinitely often is not enough to satisfy the global surveillance objective, since the local guarantees can happen in time-steps different for the different sensors.

Example 5: Consider the global surveillance specification $\Box\Diamond p_5$ for a network with two sensors. Suppose f_1 is a strategy for the sensor in G^1_{belief}, which ensures that every even step the size of the local belief is 10, and every odd step the local belief contains k_1 and its size is 3. Strategy f_2 in G^2_{belief}, is similar, but even and odd steps are interchanged: every even step the local belief contains k_2 and its size is 3, and every odd step the size of the local belief is 10. Thus, while f_1 and f_2 guarantee that their local belief is "small enough" infinitely often, they do this at different steps.

We circumvent the problem illustrated in this example by requiring that each sensor satisfies the liveness guarantee on its own. For this, we have to consider two cases. First, if from some point on sensor i always knows that the target is outside of its region, it has no obligation to satisfy the liveness surveillance guarantee. If, on the other hand, according to sensor i's belief the target could be in $\widetilde{L}_i$ infinitely often (note that this is true for at least one sensor), then it has to satisfy the corresponding liveness guarantee.

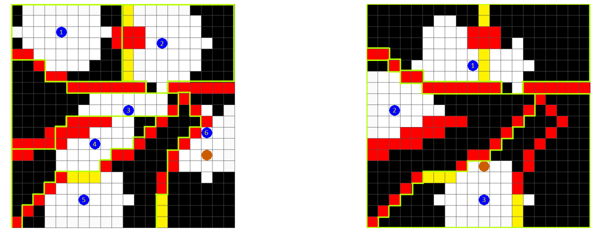
In order to capture this intuition, we need two additional types of surveillance predicates. First, we need to be able to express the negation of the property that the local belief of sensor i is the singleton $\{k_i\}$ (which means that sensor i knows that the target is outside $\widetilde{L}_i$). For this, we introduce the predicate $\mathit{belief} \neq \{k_i\}$. Second, in order to express that the local liveness guarantee, we need to be able to state that k_i is not in $\widetilde{L}_i$ (which means that sensor i knows that the target is in its region). The predicate we introduce for this property is $k_i \notin \mathit{belief}$. Both predicates can be interpreted over belief sets similarly to p_b and incorporated in LTL.

Formally, we define the local liveness specification for sensor i denoted $local_i(\Box\Diamond p_b)$ as

$$\begin{aligned} local_i(\Box\Diamond p_b) \quad \triangleq \quad & \big(\Box\Diamond(\mathit{belief} \neq \{k_i\})\big) \rightarrow \\ & \big(\Box\Diamond p_b \wedge (k_i \notin \mathit{belief})\big). \end{aligned}$$

Notice that here we use implication, which is not part of the fragment to which we restrict the global surveillance specifications. This is not a problem, since our surveillance strategy synthesis procedure for single-sensor games is applicable to such surveillance formulas as well.

This translation is again conservative, since it suffices that the liveness guarantee is satisfied by a single sensor. However, these can be different sensors for different behaviours of the target. Thus, we require that every sensor i satisfies $local_i(\Box\Diamond p_b)$. This requires that if the target crosses from one region to another infinitely often, then both sensors have to satisfy the liveness surveillance objective.



(a) The gridworld in 1b partitioned into 6 subgames.

(b) The gridworld in 1b partitioned into 3 subgames.

Fig. 4: Cases with 6 mobile sensors in Fig 4a and 3 mobile sensors in Fig 4b. The mobile sensors are blue circles and the target is represented in orange. The red cells represent impassable terrain (such as dense foliage) that cannot be seen through by the sensors. Black cells are locations not visible to any sensor.

Finally, for a global surveillance specification $\Box p_a \wedge \Box\Diamond p_b$, the local surveillance specification for sensor i is

$$local_i(\Box p_a \wedge \Box\Diamond p_b, n) \triangleq local(\Box p_a, n) \wedge local_i(\Box\Diamond p_b).$$

Slightly abusing the notation, we denote with $local_i(\varphi, n)$ the local surveillance specification for sensor i for any of the three types of global surveillance specifications.

The next theorem, which follows from the definition of the local specifications, states the soundness of the reduction.

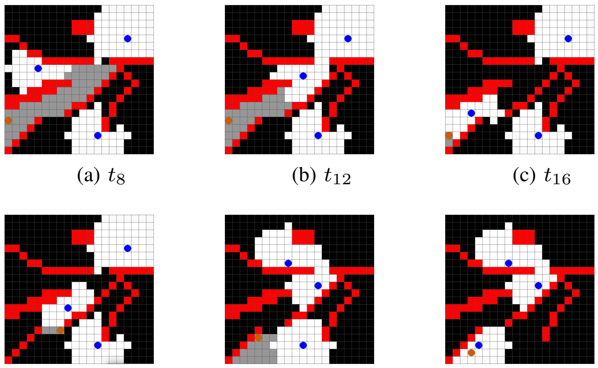
Theorem 1: Let $(G, \mathcal{M}, \varphi)$ be a multi-agent surveillance game with n sensors, and $\widetilde{L}$ be a state-space partition. Suppose that $f_1, \ldots, f_n$ are strategies for the sensors in the subgames $G^1_{\mathsf{belief}}, \ldots, G^n_{\mathsf{belief}}$ respectively, such that for each sensor i the strategy f_i is winning in the surveillance game $(G^i, Q_i, local_i(\varphi, n))$. Then, it holds that the composed strategy $f_{s_1} \otimes \ldots \otimes f_{s_n}$ is a joint winning strategy for the sensors in the surveillance game $(G, \mathcal{M}, \varphi)$.

VI. Numerical Experiments

We now return to the case study outlined in section II. We have implemented the simulation in `Python`, using the `slugs` reactive synthesis tool [21]. The experiments were performed on an Intel i5-5300U 2.30 GHz CPU with 8 GB of RAM.

We analyze two scenarios. In Figure 4a, we have six available mobile sensors. We compare the surveillance strategy with the situation in Figure 4b where we have three. Our global surveillance task is $\Box\Diamond p_5$, i.e, we need to infinitely often bring the belief of the target location to 5 cells or lower.

Solving either case centralized is not computationally feasible as the state space grows exponentially with the number of sensors - we will have in the order of 400^6 and 400^3 states respectively. Thus, we partition the multi-agent surveillance game into subgames as shown in Figures 4a and 4b. We then solve each game individually with local specifications $local_i(\Box\Diamond p_5)$. We solve these *single-agent* surveillance game



(a) t_8 (b) t_{12} (c) t_{16}

(d) t_{18} (e) t_{20} (f) t_{22}

Fig. 5: Figures 5a - 5f are chronological snapshots at various points in time during the simulation of the surveillance game in Figure 4b. Grey states represent the global belief of the target's location.

using an abstraction-based method detailed in a companion submission to CDC 2018. We refer the reader to [16] for more details on this method. We report the synthesis times in Table I.

TABLE I: Synthesis times for each surveillance subgame

	Subgame	**Number of states**	**Synthesis time (s)**
6 sensors	Subgame 1	69	101
	Subgame 2	74	206
	Subgame 3	62	111
	Subgame 4	52	88
	Subgame 5	77	285
	Subgame 6	66	64
	Total	**400**	**855**
3 sensors	Subgame 1	142	473
	Subgame 2	113	306
	Subgame 3	145	372
	Total	**400**	**1151**

The multi-agent surveillance game in Figure 4a results in more subgames compared to the game in 4a. However, each game is much smaller and strategies can be synthesized faster in each subgame. Figure 5 shows snapshots in time of the simulation of the 3 sensor surveillance game in Figure 4b. The target is being controlled by a human and the sensors are following their synthesized local surveillance strategies. The global belief is depicted in Figure 5 as grey cells, meaning that the combined knowledge of all the sensors has restricted the location of the target into one of the grey cells.

We see, in Figures 5a - 5c, that the target is in the subgame corresponding to sensor 2. Hence, only sensor 2 is moving and trying to lower its belief to below 5 cells (which it does in Figure 5d). In Figures 5b - 5d, the target moves into subgame 3 at which point sensor 3 takes over. There is no communication between any of the agents, and each satisfy only their local surveillance specification. However, our construction guarantees that the global specification of $\Box\Diamond p_5$ will be satisfied.

VII. CONCLUSIONS

We presented a method for decentralized synthesis of surveillance strategies for a mobile sensor network working with static sensors. Problems that would otherwise be computationally intractable can be solved by decomposing the global game into local subgames for each sensor with individual surveillance specifications. We show that though each game is solved completely independently with no information sharing, we can still guarantee global surveillance properties.

For future work, we aim to incorporate false positives in static alarm triggers as well as noisy observations from the mobile sensors while still guaranteeing surveillance specifications.

REFERENCES

[1] T. H. Chung, G. A. Hollinger, and V. Isler, "Search and pursuit-evasion in mobile robotics," *Autonomous Robots*, vol. 31, p. 299, Jul 2011.

[2] S. Kreutzer, *Graph Searching Games*. Cambridge University Press, 2011, pp. 213–261.

[3] M. Jain, B. An, and M. Tambe, "An overview of recent application trends at the AAMAS conference: Security, sustainability and safety," *AI Magazine*, vol. 33, no. 3, p. 14, 2012.

[4] B. An, J. Pita, E. Shieh, M. Tambe, C. Kiekintveld, and J. Marecki, "Guards and protect: next generation applications of security games," vol. 10, pp. 31–34, 01 2011.

[5] N. Basilico, N. Gatti, and F. Amigoni, "Patrolling security games: Definition and algorithms for solving large instances with single patroller and single intruder," *Artif. Intell.*, vol. 184-185, June 2012.

[6] N. Basilico, G. De Nittis, and N. Gatti, "A security game combining patrolling and alarm-triggered responses under spatial and detection uncertainties." in *AAAI*, 2016, pp. 404–410.

[7] E. Munoz de Cote, R. Stranders, N. Basilico, N. Gatti, and N. R Jennings, "Introducing alarms in adversarial patrolling games," vol. 2, pp. 1275–1276, 01 2013.

[8] R. Schiffman, "Drones flying high as new tool for field biologists," 2014.

[9] M. Mulero-Pázmány, R. Stolper, L. Van Essen, J. J. Negro, and T. Sassen, "Remotely piloted aircraft systems as a rhinoceros anti-poaching tool in africa," *PloS one*, vol. 9, no. 1, p. e83873, 2014.

[10] G. Njeru, "Kenya to deploy drones in all national parks in bid to tackle poaching," *The Guardian*, April 2014.

[11] L. P. Koh and S. A. Wich, "Dawn of drone ecology: Low-cost autonomous aerial vehicles for conservation," *Tropical Conservation Science*, vol. 5, no. 2, pp. 121–132, 2012.

[12] L. Doyen and J. Raskin, *Games with Imperfect Information: Theory and Algorithms*. Cambridge University Press, 2011, pp. 185–212.

[13] K. Chatterjee, L. Doyen, and T. A. Henzinger, "A survey of partial-observation stochastic parity games," *Formal Methods in System Design*, vol. 43, no. 2, pp. 268–284, Oct 2013.

[14] J. H. Reif, "The complexity of two-player games of incomplete information," *J. Comput. Syst. Sci.*, vol. 29, no. 2, pp. 274–301, 1984.

[15] D. Berwanger and L. Doyen, "On the power of imperfect information," in *Proc. FSTTCS 2008*, ser. LIPIcs, vol. 2, 2008, pp. 73–82.

[16] S. Bharadwaj, R. Dimitrova, and U. Topcu, "Synthesis of surveillance strategies via belief abstraction," *CoRR*, vol. abs/1709.05363, 2017. [Online]. Available: http://arxiv.org/abs/1709.05363

[17] M. Kloetzer and C. Belta, "Hierarchical abstractions for robotic swarms," in *Proceedings 2006 IEEE ICRA 2006.*, May 2006, pp. 952–957.

[18] S. Moarref and H. Kress-Gazit, "Decentralized control of robotic swarms from high-level temporal logic specifications," in *2017 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*, Dec 2017, pp. 17–23.

[19] M. Kloetzer, X. C. Ding, and C. Belta, "Multi-robot deployment from ltl specifications with reduced communication," in *2011 50th IEEE Conference on Decision and Control and European Control Conference*, Dec 2011, pp. 4867–4872.

[20] W. H. Centre/IUCN, *Reactive Monitoring Mission Selous Game Reserve (United Republic of Tanzania)*. UNESCO, 2013.

[21] R. Ehlers and V. Raman, "Slugs: Extensible GR(1) synthesis," in *Proc. CAV 2016*, ser. LNCS, vol. 9780. Springer, 2016, pp. 333–339.