Journal of Intelligent and Robotic Systems manuscript No.
(will be inserted by the editor)

Cooperative decision-making under uncertainties for multi-target surveillance with multiples UAVs

J. Capitan · L. Merino · A. Ollero

Received: date / Accepted: date

Abstract Surveillance is an interesting application for Unmanned Aerial Vehicles (UAVs). If a team of UAVs is considered, the objective is usually to act cooperatively to gather as much information as possible from a set of moving targets in the surveillance area. This is a decision-making problem with severe uncertainties involved: relying on imperfect sensors and models, UAVs need to select targets to monitor and determine the best actions to track them. Partially Observable Markov Decision Processes (POMDPs) are quite adequate for optimal decision-making under uncertainties, but they lack scalability in multi-UAV scenarios, becoming tractable only for toy problems. In this paper, we take a step forward to apply POMDP methods in real situations, where the team needs to adapt to the circumstances during the mission and foster cooperation among the team-members. We propose to split the original problem into simpler behaviors that can be modeled by scalable POMDPs. Then, those behaviors are auctioned during the mission among the UAVs, which follow different policies depending on the behavior assigned. We evaluate the performance of our approach with extensive simulations and propose an implementation with real quadcopters in a testbed scenario.

*This work has been supported by the FP7 EC-SAFEMOBIL Project (grant number 288082) and the PAIS-MultiRobot (TIC-7390) Regional Project.

J. Capitan
University of Seville, Seville, Spain
E-mail: jcapitan@us.es

L. Merino
Pablo de Olavide University, Seville, Spain
E-mail: lmercab@upo.es

A. Ollero
University of Seville, Seville, Spain
E-mail: aollero@us.es

1 Introduction

Tracking a set of targets of interest is a relevant application for surveillance. For example, this functionality is relevant in search and rescue missions, environmental monitoring, traffic control, etc. Given their dynamic and sensing capabilities, the use of Unmanned Aerial Vehicles (UAVs) for these kind of missions is spreading widely [6,12,3]. It is usually the case that UAVs provide wider fields of view and can access more hazardous places than other vehicles.

Also, in complex scenarios, the use of cooperative teams of UAVs for tracking can be of capital importance. Consider that, with a single UAV operating, the targets may be too dynamic or numerous, and the surveillance area too large. In other cases, the co-existence of several UAVs collaborating is a way to achieve goals faster and improve the efficiency of the mission, which may be critical for some applications.

In a surveillance application where several UAVs have to track the existing targets, serious uncertainties are involved. An example is depicted in Fig. 1: the positions of the targets are unknown and can only be observed with imperfect sensors; occlusions can occur due to elements in the scenario or due to other UAVs; the dynamic models for the UAVs and the trackers are not perfect; and so forth. Considering these uncertainties when solving the tracking problem is key in order to ensure optimal solutions.

The problem of target tracking has been extensively considered from the point of view of sensing, which means to maintain an estimation of the targets' positions and their associated uncertainties [2]. For this purpose, many different stochastic filters integrating observations from sensors have been proposed. There also exist multi-robot filters that fuse information from

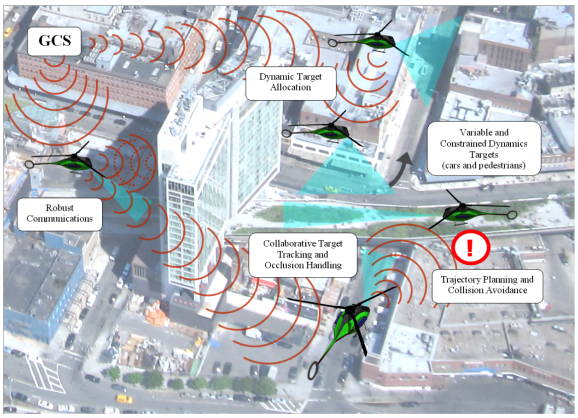

Fig. 1: Typical scenario for cooperative surveillance. A team of UAVs has to track a set of (non-cooperative) targets, such as cars or people. Reasoning about uncertainties due to occlusions, imperfect sensors, targets' motion, and so on, is mandatory. A problem to solve is how to assign UAVs to the different targets considering the issues above and some criteria to optimize (GCS stands for Ground Control Station).

all team-mates in situations where several robots cooperate in the tracking mission [18,7].

However, the problem we face in this paper is also a decision-making problem. We are interested not only in estimating the position of the targets with moving sensors (UAVs), but also in controlling those sensors to optimize some objectives. There are multiple targets and multiple UAVs, so we need to decide dynamically which UAVs track each target, and then, how they move in order to do so. The criteria to optimize may vary depending on the application, but they usually consider the percentage of time that each target is within field of view, the degree of uncertainty on the targets' positions, the fuel consumption, communication constraints, and so on.

From the point of view of decision making, the problem of target tracking is often formulated as an stochastic optimal control problem where an utility function is optimized [1]. Many approaches [11,5,24,18,9] have also been proposed to solve multi-target tracking with a team of several vehicles.

As said before, target estimations are uncertain and maximizing the information gathered from the targets is usually quite relevant. This information can be quantified by means of different metrics, such as entropy or mutual information. Many works assume Gaussian uncertainties and Kalman (or Information) Filters as the underlying estimation frameworks, defining utility functions based on these information metrics in order to determine the actuations [10,23,16]. However, tracking applications can result in multi-modal distributions when estimating the targets' positions. Hence, other works also consider alternative representations, such as discrete Bayes filters [5,24] or Particle Filters [18].

Many of the previous works in the literature propose information-gathering approaches based on heuristics or rigid optimization problems. Those approaches lack usually adaptability to different scenarios and optimization criteria. *Partially Observable Markov Decision Processes* (POMDPs) provide a sound mathematical framework for planning under uncertainties [14]. Resulting policies reason about uncertainties and can combine easily multiple objectives, such as maximizing information and fuel consumption. Moreover, these policies do not constrain to specific scenarios, since they can be recomputed by adapting the models involved or the required objectives.

POMDPs suit very well planning problems under uncertainty from a theoretical perspective, and policy solvers for models in the domain of discrete states are rife. However, they present a severe problem of scalability for multi-robot decision making. Optimal policies in a POMDP reason about future decisions (actions) and sensor measurements (observations) in a receding time horizon. Besides, the computational complexity for obtaining these policies depends on the size of the observation and action spaces. In our surveillance problem, increasing the number of robots or targets entails an exponential growth in these spaces, leading to intractable problems for a time horizon greater than 1, unless we consider very small scenarios.

Some works try to alleviate the computational complexity of the original problem by decoupling it into a set of simpler sub-models. Thus, suboptimal policies are obtained by exploiting weak interdependences between variables and restricting the states where multiple robots should interact [17,15]. Factored models [21] can also be used to reduce the complexity of POMDPs. These models represent variables with several factors and define the utility functions over those factors. Due to conditional independences between the factors, the original model can be reduced. Moreover, the use of factored models allows us to considered mixed observability in the problem [19], assuming some factors as observable and neglecting their associated uncertainty.

Our main idea is to come up with behaviors that can be modeled as simpler POMDPs. We call them factored behaviors because they can be extracted from the factorization of the original model. Each factored behavior can be emulated by means of a policy that is computed from a factored POMDP simpler than the original. Therefore, we take the original POMDP and derive some simpler factored models to compute a set of policies. Then, we use those policies to emulate different behaviors that can be combined to achieve co-

operation with a multi-robot team. Thus, the mission is somehow split into different behaviors represented by policies that are derived from scalable POMDP models.

In this paper we extend our previous work [8] for multi-target surveillance, where we proposed a decentralized auction of behaviors based on POMDPs for target tracking with multiple robots. Those behaviors where assigned dynamically to the robots during the execution of the mission in order to foster cooperation. Here, we exploit the factorization of the models in order to apply the same approach to multi-target tracking without increasing the problem complexity. For this purpose, different behaviors are derived by applying the same policy to different factored sub-models. This allows us to re-use policies when emulating the behaviors, reducing dramatically the computational complexity.

In addition, our approach is totally decentralized and scalable, which is an advantage for multi-UAV applications with hard communication constraints. The UAVs access only local observations and communicate with others in their neighborhood. Even though information is shared when the UAVs are within communication range, they can still take actions when only local information is available.

The paper is organized as follows: Section 2 introduces POMDPs for single and multiple robots as well as factored models; Section 3 describes our online auction for factored behaviors; Section 4 details the factored models proposed for multi-target surveillance; Section 5 presents experimental results and Section 6 gives the conclusions and future work.

2 Background

This section describes POMDP models for single and multiple robots (UAVs in our case). The notion of factored POMDPs is also explained.

2.1 Single-robot POMDP

Formally, a discrete POMDP is defined by the tuple $\langle S, A, Z, T, O, R, D, \gamma \rangle$ [14]:

- The *state space* is the finite set of possible states $s \in S$, for instance the poses of targets and the UAV.
- The *action space* is defined as the finite set of possible actions that the UAV can take, $a \in A$.
- The *observation space* consists of the finite set of possible observations $z \in Z$ from the sensors on board the UAV.
- After performing an action a, the state transition is modeled by the conditional probability function $T(s', a, s) = p(s'|a, s)$, which indicates the probability of reaching state s' if action a is performed at state s.
- The observations are modeled by the conditional probability function $O(z, a, s') = p(z|a, s')$, which gives the probability of getting observation z given that the state is s' and action a is performed.
- The reward obtained performing action a at state s is $R(s, a)$.

The state is non-observable; at every time step the UAV has only access to observations z that give incomplete information about the state. Thus, a belief function b is maintained by using the Bayes rule. The new belief b' obtained after applying action a at belief b and getting observation z is given by:

$$b'(s') = \tau(b, a, z) = \eta O(z, a, s') \sum_{s \in S} T(s', a, s) b(s), \quad (1)$$

where the normalization constant is defined as the probability of obtaining a certain observation z after executing action a for a belief b:

$$\eta = p(z|b, a) = \sum_{s' \in S} O(z, a, s') \sum_{s \in S} T(s', a, s) b(s). \quad (2)$$

This POMDP model assumes that, at every step, an action is taken, an observation is made and a reward $R(s, a)$ is obtained. The objective is to determine the policy $a = \pi(b)$ that maximizes the expected cumulative reward earned during D time steps. This metric is called *value* $V^\pi(b)$ and depends on the current belief:

$$V^\pi(b) = R(b, \pi(b)) + \gamma \sum_{z \in Z} p(z|b, a) V^\pi(\tau(b, \pi(b), z)), \quad (3)$$

where $R(b, a) = \sum_s R(s, a) b(s)$ is the expected immediate reward. Rewards are weighted by a discount factor $\gamma \in [0, 1)$ to ensure that the sum is finite when $D \to \infty$. The value of the optimal policy π^* is usually denoted by $V^*(b)$.

The same formulation could be cast using costs instead of rewards. Note that, once the system is correctly modeled through the transition and observation functions, the reward (or cost) function is critical, since it is the way the desired behavior is incorporated into the system.

2.2 Multi-robot POMDP

If a set of n robots or UAVs is considered, each UAV i can execute an action a^i from a finite set A^i and can

measure an observation z^i from a finite set Z^i. The transition function $T(s', a^J, s)$ is now defined over the set of joint actions $a_J \in A^1 \times \cdots \times A^n$ (the actions that the team as a whole can perform); and the observation function $O(z^J, a^J, s')$ relates the state and the joint action to the joint observation $z^J \in Z^1 \times \cdots \times Z^n$. The team reward is defined over the joint set of states and actions $R : S \times A^1 \times \cdots \times A^n \rightarrow R$. The goal in the multi-robot POMDP is also to compute an optimal joint policy $\pi^* = \{\pi^1, \cdots, \pi^n\}$ that maximizes the expected discounted reward.

On the one hand, a policy for a multi-robot POMDP (MPOMDP) can be computed in a centralized fashion [22], assuming that each robot has access to the complete observation vector z. On the other hand, the policy can be computed with a Decentralized POMDP (Dec-POMDP) model [4], assuming that each robot has only access to its local observation z^i.

The main issue when computing policies for a multi-robot model with the above approaches is that the computational complexity of the problem increases exponentially with the number of robots, since this complexity depends on the observation and action spaces. Moreover, the computational complexity of solving a Dec-POMDP is significantly higher than that of an MPOMDP (NEXP-complete [4] vs. PSPACE-complete [20]). This precludes the direct application of these models except for very simple scenarios with small teams.

2.3 Factored POMDP

Factored models [21] are commonly used to simplify POMDPs. These models decompose some variables into factors, and the probability and reward functions are expressed over those factors. In general, in a factored POMDP the state consists of a set of d variables or factors: $s = (s_1, s_2, \ldots, s_d)$. A similar decomposition can be applied to the action and observation variables.

The idea lying behind a factored model is to exploit some degree of conditional independence of the factored variables in order to obtain a more compact representation of the model. Thus, the transition, observation and reward functions may depend only on a subset of the factors, simplifying the original model. Moreover, these functions could be defined as products of simpler functions depending on subsets of factors.

The use of factored models also eases the integration of mixed observability into POMDPs [19]. Mixed observability refers to considering part of the state as observable, taking out that part from the belief computation. In general, this alleviates the complexity of the model, since the belief space is reduced when searching for optimal policies. Therefore, in a factored model, some of the factors could be considered observable if the degree of uncertainty associated with them is not so relevant. For instance, if a team of UAVs can operate with a highly precise localization (e.g., differential GPS receivers), the uncertainty associated with their poses may be negligible compared to that of the target poses, assuming the former as observable.

3 Online auction of factored behaviors

In our previous work [8], we proposed to approximate multi-robot POMDPs by auctioning behaviors. Those behaviors were modeled by simpler POMDPs and encoded by polices precomputed offline. The objective was to perform cooperative tracking of a target without computing a policy for the joint multi-robot model, which does not scale with the number of robots. In this paper, we take a step forward and introduce the concept of factored behaviors. This will allow us to represent different behaviors with the same policy, and hence, obtaining a solution for the multi-target case without increasing the computational complexity.

3.1 Decentralized estimation of joint belief

The first manner to foster cooperation in the team is to share information in order to maintain a common belief of the state. Therefore, all UAVs have access to a joint estimation that incorporates observations from all team-members. This is accomplished by running a decentralized filter for data fusion [7].

Basically, the joint belief is estimated by each UAV using local information and exchanging that information with other neighbors. UAVs employ local sensor data to update their local belief, and then *share* those beliefs with other neighbors at certain time instants. The beliefs are received by other team-members and fused with their local estimations in order to obtain a joint belief that incorporates information from the whole team. For instance, in our multi-target surveillance application, this allows UAVs to gather information relative to all detected targets, even if they are being tracked by other UAVs.

3.2 Factored behaviors

We cast behaviors as POMDP models, so each behavior has a reward function associated and is defined over a state space. If the models are factored, we can introduce the notion of factored behaviors, where different behaviors can be obtained by applying the same POMDP model to different subset of state factors.

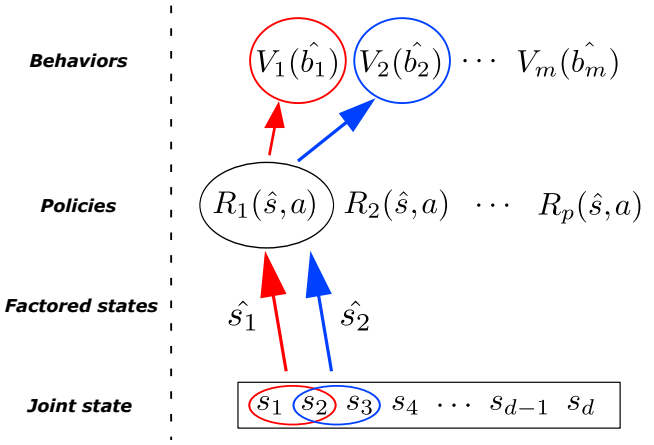

Fig. 2: A set of m factored behaviors are obtained from the original joint model. Each behavior j is defined over a subset of factors $\hat{s}_j$ and has a value function associated $V_j(\hat{b}_j)$. A set of $p \leq m$ reward functions will suffice to model all the behaviors. Evaluating the same value function based on a certain reward function (e.g., R_1) with different factors (e.g., $\hat{s}_1$ and $\hat{s}_2$) could lead to policies corresponding to different behaviors.

The concept of factored behaviors is depicted in Fig. 2. Assuming that we have a joint factored state, each behavior j is defined over a subset of factors $\hat{s}_j$. The behavior is represented by a policy, which comes from maximizing a certain value function defined over that factored belief $V_j(\hat{b}_j)$. In general, we could define a reward function for each behavior, which would lead to its corresponding value function. That would mean to compute a policy per behavior. However, if the behaviors keep certain symmetry properties, we can match several behaviors to the same reward function. This idea is shown in Fig. 2, where a single reward function (R_1) is used to generate different behaviors. Thus, we can formulate a factored POMDP model with reward function R_1, and compute a policy and a value function associated with it. Then, if we evaluate that value function with a factored state $\hat{s}_j$, we obtain the value corresponding to behavior j, as well as its associated optimal action. The same process can be repeated to reproduce other beharviors.

This approach suits our multi-target surveillance problem. Provided that the targets move independently, and the targets and UAVs present homogeneous models; a policy for tracking a target will always be the same, regardless of the UAV and target considered. Thus, assuming that the different behaviors correspond to tracking different targets, if we evaluate a sole value function with the factors associated with target j, we will obtain optimal actions corresponding to that behavior j (tracking target j).

By exploiting the symmetry properties of the problem, we manage to emulate a set of behaviors by computing a lower number of policies, which alleviates the computational complexity. In particular, for our multi-target surveillance application, we convert the original problem into another problem that scales with the number of targets and UAVs, since only POMDPs with a single UAV and target will be used to compute policies.

3.3 Auction of factored behaviors

Once the behaviors are defined, they need to be executed in a cooperative fashion by the team in order to pursue the goals of the original multi-robot POMDP. We propose an online auction during the mission where the UAVs negotiate in a decentralized manner which is the best behavior they can select at each moment. Thus, behaviors are distributed optimally among the UAVs, which leads to a cooperative performance of the whole team. Moreover, this assignment is dynamic and the UAVs can switch between behaviors during their mission depending on the circumstances.

A UAV computes the bids for the auction by evaluating the value function of each behavior. These value functions provide the discounted cumulative reward expected from executing the different behaviors given the current belief. Therefore, UAVs will bid for behaviors with higher *values*, since that should produce more optimal team performances. As said before, a UAV can obtain bids for several behaviors by evaluating the same value function with several subset of factors. Recall that each UAV has an estimation of all joint state factors thanks to decentralized data fusion.

In the multi-target surveillance mission, the auction can be executed to allocate targets to UAVs in some optimal manner. Each UAV will select its target according to the probability it has to be rewarded by tracking it. Then, as long as the belief changes, the behavior or target distribution can vary too.

4 Factored models for multi-target surveillance

In this section we detail the factored models used for multi-target surveillance with a team of UAVs. The objective is to maintain an estimation of the targets' positions as certainly as possible during the mission. First, we present the joint factored POMDP considering all UAVs and targets. Then, a factored model for tracking a single target with a single UAV. We do not compute any policy for the complete joint model, but all behaviors are derived from the policy of the POMDP with a single target and UAV. We show in this section how to

emulate the different factored behaviors to track multiple targets with a single policy.

4.1 Factored POMDP for multiple UAVs

Our surveillance mission implies monitoring a set of m targets with a team of n UAVs. Given the probabilistic models and reward functions, the problem can be cast as a discrete multi-robot POMDP. The scenario is discretized into a cell grid and the state at each iteration is a vector with the following discrete factors: $s^{multi} = (t_1, \ldots, t_m, l_1, \ldots, l_n)$. The position in the grid of each target j is represented by a factor t_j, while the position of UAV i is represented by a factor l_i.

Each UAV is equipped with a camera sensor pointing downwards that provides (noisy) binary observations about the presence of targets. Therefore, the observation vector for each UAV i is $z^i = (o^i_1, \ldots, o^i_m)$, where o^i_j is a binary factor indicating whether UAV i has detected target j. At each iteration, each UAV can take an action $a^i \in \{stay, north, west, east, south\}$, in order to hover on the same cell or move to a neighboring cell. These actions issue waypoints to the corresponding UAV, which uses a low-level control algorithm to navigate to those waypoints.

UAVs maintain a state belief $b(s)$, which is updated by Eq. (1) with probabilistic transition and observation functions: $T(s', a^J, s)$ and $O(z^J, a^J, s')$. UAVs' actions are not deterministic, but noisy transition functions are considered. The movement of each target is modeled as random, existing the same probability to move to any of its neighboring cells at each time step. Moreover, a UAV can detect a target with a probability p_D if this is in one of its 9-connected cells. In addition, the complexity of the model is alleviated by considering mixed observability. In this case, UAVs' positions (l_i) are assumed observable. In general, UAVs can localize themselves with high precision thanks to on-board sensors (e.g., DGPS or vision), so it is reasonable to assume that this uncertainty is insignificant compared to the uncertainty on the targets' positions.

A joint reward function defined over the complete factored state can be designed to achieve the objectives of the mission. In particular, at each step, a *high* reward is given for each UAV located in the same cell as a target. If the target is already being monitored by another UAV, the reward is *lower*. Therefore, we foster cooperative surveillance, since UAVs will try to catch all the targets as many times as possible and they will distribute their sensing capabilities efficiently. Note that the complexity of this model increases exponentially with the number of UAVs or targets, since the state's, observation's and action's spaces do.

4.2 Factored POMDP for UAV-to-target policy

We do not compute a joint policy for the previous multi-robot model, since that is intractable in most cases and do not scale with the number of targets and UAVs in the scenario. Instead, we solve a policy for a POMDP model where a single UAV tracks a single target. From that policy, we can emulate different behaviors depending on the state factors used to apply the policy.

In this simpler model the state is $s^{local} = (t, c, l)$, where t and l are the positions of the target and UAV, respectively. The binary factor c specifies whether the target has been visited or not. Thus, when the UAV is in the same cell as the target, this variable is set to 1. Otherwise, if the target was already visited, there is a probability p_c at each time step that c switches back to 0 (not caught). This factor is used by the UAV to *forget* after some time that the target was detected, and go after it again. In the next section, we will see how this allows a single UAV to switch between different targets in a scenario with multiple targets.

Observations and actions are the same as in the multi-robot model, but restricted to a single UAV and a single target. The reward function is modified in order to take into account the new factor c. Basically, if the UAV is at the same position as the target and $c = 0$, a *high* reward is earned. If the UAV is at the same position as the target but $c = 1$, a *low* reward is earned. Otherwise, no reward is earned.

4.3 Factored behaviors for multi-target surveillance

In this surveillance mission, the original problem is split into m behaviors that can be executed in paralel. Each behavior is executed by a single UAV and represents tracking a specific target. Thus, if the behaviors are allocated dynamically to the UAVs with some optimization criteria, UAVs should be able to distribute the targets among them and monitor them cooperatively.

All behaviors are derived from the same policy. This policy corresponds to the factored model in Section 4.2 and is computed offline in our experiments (although it may be computed online too). Once we have that UAV-to-target policy, we can emulate different factored behaviors by selecting the state factors adequately, as it was explained in Section 3. For instance, the behavior corresponding to tracking target j with UAV i, will be defined over the factors (t_j, c_j, l_i). Plugging those into the UAV-to-target policy we can evaluate the benefit (value function) of executing behavior j with UAV i, and obtain the optimal action to perform that behavior. Thanks to factored models, we exploit the probabilis-

tic independence of the targets and tackle the problem without solving policies for multi-target models.

As explained in Section 3, decentralized data fusion is performed so that all the UAVs can maintain a multi-target belief with factors: $(t_1, \ldots, t_m, c_1, \ldots, c_m)$. This joint belief is useful for a UAV in order to evaluate the different behaviors considering information from other UAVs. Besides, during the execution of the mission, a decentralized auction is performed online among the UAVs in order to switch continuously between different behaviors (between targets in this case). Bids for each behavior correspond to evaluating the value function of that behavior with the adequate factors depending on the target and UAV involved.

This approach fosters cooperation between UAVs, since the targets can be re-assigned continuously to different vehicles. Note that a UAV gets a reward for monitoring a target, but that reward is even higher if that target has not been seen for a while (not caught). This, together with the joint belief available, precludes UAVs from tracking targets recently *caught* by other UAVs (their c factors will have higher probabilities), and forces UAVs to monitor all *free* targets in turns.

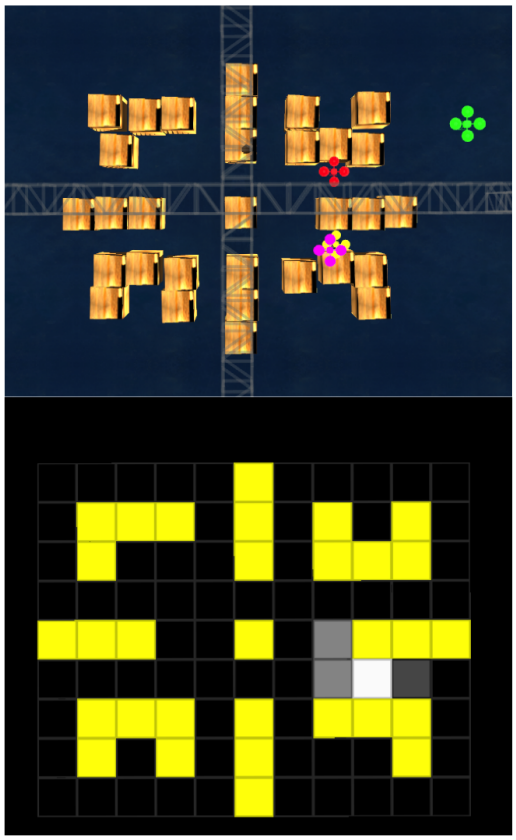

Fig. 3: Top: Simulated scenario. Two quadcopters act as targets (red and yellow), while two others act as the cooperative trackers (green and magenta). Bottom: the two trackers maintain a belief over the position of the two targets by using a decentralized data fusion filter. The figure shows the fused belief for one of the targets (the yellow one in this case). The scenario is discretized into a grid of cells (yellow tiles indicate non-flyable zones). The belief state is used to determine which tracker should be assigned to which target.

5 Experiments

Our approach for multi-target surveillance has been evaluated and compared to alternative methods. First, we present extensive simulations of our method under different circumstances in order to analyze its performance compared with simpler approaches. Second, we describe the implementation of the method with a real fleet of UAVs in a realistic testbed scenario.

5.1 Simulations

Simulations allow us to analyze our multi-target tracking approach under different circumstances (target dynamics) and compare it to simpler methods of target assignment. We take the simulated scenario from [13], since it presents interesting features for cooperative target tracking. That scenario is shown in Fig. 3, and it is used to run experiments where two targets are moving around. Simulations with two UAVs or a single UAV performing surveillance are presented.

In these experiments, although it is not a requirement of our technique, targets are not evading. Moreover, three different cases for targets' movement are considered: (i) *random*: targets move around randomly, with the same probability of going to any of their neighboring cells at each step; (ii) *route*: targets follow a fixed route unknown for the trackers, with the same probability of staying at the same cell or moving on along the route at each step; and (iii) *static*: targets stay at fixed positions during the whole experiment. Besides, three different methods for allocating behaviors (or targets) to the UAVs are tested for each experiment:

- *Fixed* allocation: UAVs are assigned to the same target during the whole experiment.
- Allocation based on *distance*: at each iteration, UAVs are assigned solving an auction based on distances to the targets. Therefore, UAV-target pairs are selected in terms of closeness. As the actual positions of the targets are not known by the UAVs, distances to targets are computed assuming targets at the positions with highest probability in the belief state.
- Allocation based on *value*: this is the approach presented in this paper. At each iteration, UAVs are assigned to targets solving the auction in Section 3,

Table 1: Average results and standard deviations for simulations with two UAVs tracking two targets. For each configuration of targets' movement and method to assign behaviors, 30 runs were performed.

Targets	***Random***			***Route***		
Behaviors	*Fixed*	*Distance*	*Value*	*Fixed*	*Distance*	*Value*
Normalized reward	0.023 ± 0.007	0.025 ± 0.007	0.024 ± 0.006	0.010 ± 0.005	0.011 ± 0.005	0.012 ± 0.004
Entropy target 0	2.64 ± 0.47	2.52 ± 0.41	2.51 ± 0.38	2.92 ± 0.29	2.99 ± 0.29	3.00 ± 0.27
Entropy target 1	2.37 ± 0.32	2.45 ± 0.41	2.42 ± 0.35	3.04 ± 0.29	3.03 ± 0.30	3.02 ± 0.29
Error target 0 (cells)	2.61 ± 1.18	2.27 ± 0.92	2.26 ± 0.92	3.51 ± 0.90	3.76 ± 1.01	3.74 ± 0.82
Error target 1 (cells)	2.03 ± 0.88	2.23 ± 1.01	2.22 ± 1.00	3.78 ± 0.80	3.75 ± 0.92	3.72 ± 0.94
Movement UAV 0 (%)	58.46 ± 10.23	56.40 ± 12.91	59.43 ± 8.36	68.26 ± 11.26	68.13 ± 11.23	70.53 ± 7.75
Movement UAV 1 (%)	56.10 ± 10.18	59.56 ± 8.79	59.80 ± 8.76	68.16 ± 9.02	66.70 ± 8.87	70.86 ± 7.42

Targets	***Static***		
Behaviors	*Fixed*	*Distance*	*Value*
Normalized reward	0.010 ± 0.007	0.010 ± 0.008	0.011 ± 0.008
Entropy target 0	2.20 ± 0.57	2.18 ± 0.61	2.14 ± 0.52
Entropy target 1	2.18 ± 0.62	2.16 ± 0.55	2.11 ± 0.51
Error target 0 (cells)	1.83 ± 1.76	1.90 ± 1.94	1.76 ± 1.67
Error target 1 (cells)	1.98 ± 1.36	2.01 ± 1.42	1.97 ± 1.40
Movement UAV 0 (%)	48.20 ± 25.60	42.40 ± 24.95	43.80 ± 23.79
Movement UAV 1 (%)	42.80 ± 19.15	49.13 ± 22.73	47.36 ± 20.69

Table 2: Average results and standard deviations for simulations with one UAV tracking two targets. For each configuration of targets' movement and method to assign behaviors, 30 runs were performed.

Targets	***Random***			***Route***		
Behaviors	*Fixed*	*Distance*	*Value*	*Fixed*	*Distance*	*Value*
Normalized reward	0.023 ± 0.013	0.037 ± 0.014	0.030 ± 0.015	0.008 ± 0.007	0.013 ± 0.006	0.014 ± 0.009
Entropy target 0	3.46 ± 0.36	3.26 ± 0.55	3.29 ± 0.54	3.46 ± 0.27	3.54 ± 0.28	3.42 ± 0.26
Entropy target 1	2.91 ± 0.49	3.15 ± 0.52	3.17 ± 0.47	3.47 ± 0.24	3.53 ± 0.25	3.58 ± 0.29
Error target 0 (cells)	3.49 ± 1.41	3.53 ± 1.36	3.54 ± 1.47	4.44 ± 0.93	5.03 ± 1.0145	4.62 ± 1.15
Error target 1 (cells)	2.85 ± 1.43	3.14 ± 1.46	3.14 ± 1.37	4.55 ± 0.92	4.65 ± 1.36	4.86 ± 1.11
Movement UAV (%)	61.30 ± 11.14	54.30 ± 11.74	59.50 ± 9.35	74.73 ± 9.99	73.46 ± 8.12	72.86 ± 6.03

Targets	***Static***		
Behaviors	*Fixed*	*Distance*	*Value*
Normalized reward	0.007 ± 0.008	0.009 ± 0.009	0.012 ± 0.012
Entropy target 0	2.59 ± 0.52	3.19 ± 0.87	3.16 ± 0.68
Entropy target 1	3.62 ± 0.38	2.98 ± 0.86	3.01 ± 0.74
Error target 0 (cells)	2.15 ± 1.09	3.10 ± 2.30	2.75 ± 1.88
Error target 1 (cells)	3.46 ± 2.08	3.20 ± 2.24	2.77 ± 2.08
Movement UAV (%)	47.46 ± 22.72	40.20 ± 23.17	59.06 ± 11.41

which means that UAVs select targets whose value function are maximized.

At each iteration, UAVs decide their target to track and execute an action following the factored behavior (POMDP policy) corresponding to that target. For each configuration, 30 simulations were run with random initial positions for the targets and UAVs. Moreover, the single POMDP policy for the UAV-to-target model was computed offline with Symbolic Perseus [21]. In this POMDP model, $p_D = 0.9$ and $p_c = 0.04$; the *high* reward for catching a non-visited target was 500, and the *low* reward for catching a visited target was 1.

Table 1 shows the results of the simulations with two UAVs tracking two targets, whereas Table 2 shows the results of the simulations with one UAV tracking two targets. Several metrics are used to compare the different methods. The metric *normalized reward* shows the discounted cumulative reward for the team, which is computed using actual states during the experiments and normalized dividing by the maximum possible cumulative reward. *Entropies* for the belief estimation of the targets at each step ($\sum_{\forall cell} -p_{cell} \log(p_{cell})$) are averaged. The *errors* estimating the targets' positions are computed by averaging the distances between the actual cell of the targets (ground truth) and the cell with higher probability in the belief. The metric *movement* measures the percentage of time steps that the UAV is not staying at the same cell, i.e., it is moving.

There are several conclusions about the comparison for the multi-UAV simulations (Table 1). In terms of entropy and error in the estimations, all assignment methods behave better when the targets move randomly in-

stead of along a fixed route (the standard deviations are bigger in the case of *random* targets due to the higher randomness of the simulations, this effect should be reduced with more runs.). Despite the fact that the second option constrains more the targets' movements, UAVs travel more and achieve worse tracking results. The conclusion is that the model for the target movement (*random* for all our policies) is quite relevant. Thus, when the targets move along the path UAVs are not able to take advantage of that information, since they assume a different target model. Performances are also better when the targets are static, but that is expected, since the UAVs can monitor easily static targets once they have been detected.

Also, our assignment method (*value*) is slightly better than the others in terms of entropy and error (averaged for the two targets) when the targets are *random* or *static*. However, the results for the three methods do not differ significantly. Note that in this simple setup, the number of UAVs and targets are equal and the scenario is not excessively complex. Therefore, the matching between UAVs and targets is not tough regardless of the assignment method used, which prevents from seeing bigger differences between methods. Moreover, the method that assigns targets reasoning about distances can perform relatively well for this particular application, but it implies incorporating additional information from the problem at hand. The value-based approach is more general, as it can be applied to any application using POMDPs, and reasons about future steps (not only current distances). Hence, we expect to obtain more efficient performances with our method in complex planning scenarios.

Additionally, we can reason that our method should be more adequate than others for cases where UAVs need to visit targets in turns in order to monitor all of them (more targets than UAVs). The other assignment methods do not penalize the fact of keeping tracking the same target while there are others not being monitored. To highlight that advantage of our method, we run simulations where two targets need to be observed by a single UAV (Table 2). In the case of static targets, where the UAV can easily track them in turns once they have been detected, we notice the difference clearly. With our approach the UAV travels significantly more to cover both targets, and achieves homogeneous entropies and estimation errors for both targets. Other methods can get better results for a specific target, but do not compensate both targets. For instance, in the case of a fixed assignment, the UAV focuses on one of the targets, not gathering a lot of information about the other.

The sample experiment in Fig. 4 depicts the effect of switching between targets during the mission. When the

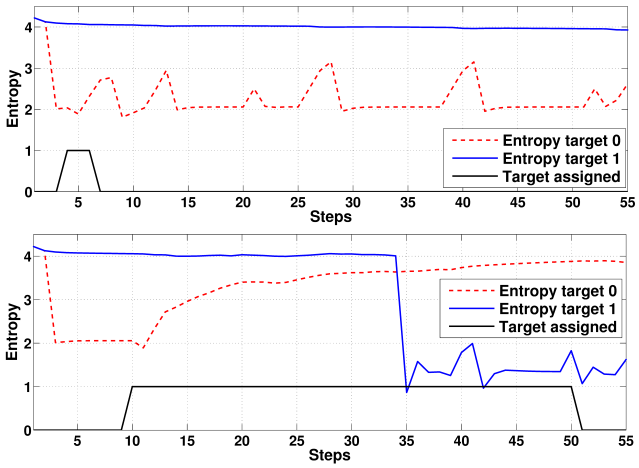

Fig. 4: Evolution of the entropy of the target beliefs for a short experiment with a UAV tracking two targets. Top: the UAV selects the target reasoning about distances. Bottom: the UAV selects the target reasoning about expected values of the POMDP policy.

UAV selects targets with the distance-based method, it sticks to target 0, not penalizing the fact of leaving the other *free*. With our value-based method for assignment, once target 0 is detected, the UAV switches to target 1. It can be seen how entropies of both targets are reduced in turns.

5.2 Testbed experiments

We implemented our approach for a real team of quadcopters (AscTec Hummingbird) and run some preliminary experiments in an indoor testbed. We emulated in the testbed a small part of a city and used two quadcopters to track two radio-control cars that were moving around the city. Figure 5 shows the scenario and the discretization employed for the POMDP models. Different sets of states are considered for targets and UAVs, since cars can only move along roads. The transition functions assume that UAVs and cars can move to any of their neighboring cells (U-turns are considered for the cars). Moreover, the flying vehicles can move through zones that cars cannot traverse. Finally, the observation model for the UAVs is depicted in Fig. 5, right. UAVs can detect cars that are in their 9-connected cells, except for a tunnel in the city where cars move without being noticed. Apart from that, the POMDP model used for computing a UAV-to-target policy is that in Section 4.2. That policy is used to reproduce two behaviors (one to track each car).

Figure 6 shows the reconstruction of the city scenario in the indoor testbed during some experiments. In the experiments, the two quadcopters were executing

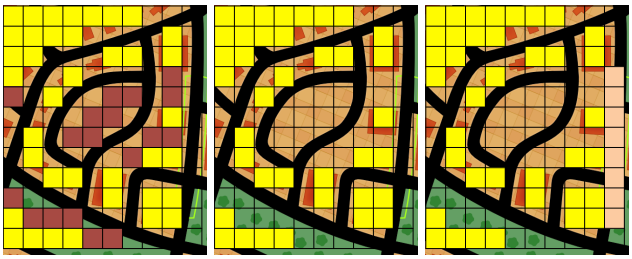

Fig. 5: Discretization of the scenario: yellow cells are unattainable for both, targets and quadcopters. Left: non-shadowed cells are the possible states for targets (cars have to move along the roads). Middle: non-shadowed cells are the possible states for quadcopters (they can traverse more places). Right: non-shadowed cells are those where cars can be detected (there is a tunnel in the right where quadcopters cannot see).

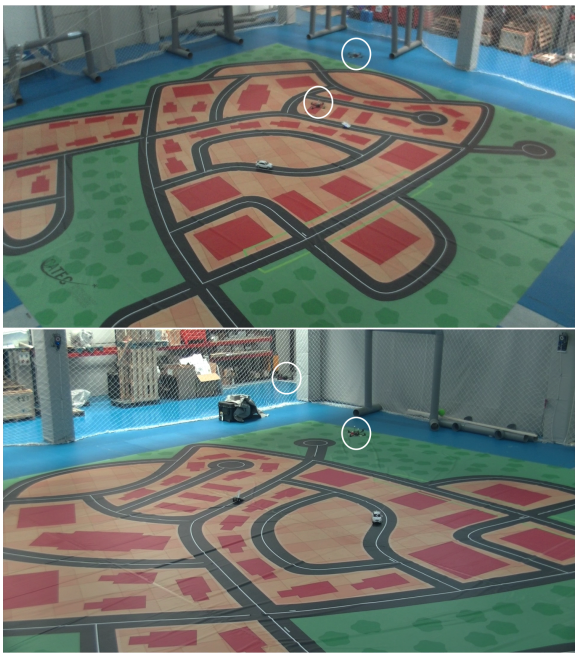

Fig. 6: Two views of the experiments performed in the testbed. A city is emulated and two radio-control cars used as targets. The two quadcopters (in circles) are controlled with our proposed method.

in real time our proposed method. First, a decentralized data fusion filter was used to maintain a multi-target belief over the two cars. In this particular experiment, both quadcopters were within communication range all the time. Second, at each step, the quadcopters selected a behavior to monitor one of the cars, and then selected the next cell to move depending on the action provided by the corresponding policy.

Regarding the implementation in the testbed system, our POMDP planners send waypoints to the UAV controllers depending on their next cell to go (or command them to hover at their current cell). The UAVs move between waypoints following straight lines, and in the current implementation, they fly at different heights to avoid collisions (a path planner could be used to navigate between waypoints). There is a VICON system (motion capture system) which provides positions of the quadcopters and cars with millimeter accuracy. The poses of the UAVs are used for flight stabilization and control, as well as part of the observable state of the POMDP. The poses of the cars are used in conjunction with the poses of the UAVs to emulate measurements from the onboard sensors, taking into account the limited fields of view and probabilities of detection (UAVs cannot observe directly the real positions of the cars). All the modules, except for the low-level UAV controllers, are implemented using the Robotic Operating System (ROS).

Figure 7 shows our visualizer based on RViz during an experiment[1]. That visualizer depicts the actual positions of the quadcopters and the belief over the targets positions (maintained by the decentralized data fusion filter). The belief of each target is shown with a different color map, where brighter cells indicate higher probabilities. Furthermore, the POMDP planners reason about negative information, and thus many times the vehicles move to places to discard possibilities. They also reason about the lack of information in the tunnel, and thus Fig. 7 illustrates how one of the UAVs waits at one of the entrances of the tunnel while receiving at the same time information from the other UAV.

6 Conclusions

We have presented a method for multi-target surveillance with multiple UAVs. The method uses POMDPs in order to reason about the uncertainties present in the application, both in the dynamic and observation models (due to occlusions, noise, etc). In order to scale POMDPs to real-time missions for teams of multiple UAVs, we propose a behavior-based auction. Instead of solving a joint multi-UAV POMDP, which is usually intractable, a policy for a factored POMDP with a single UAV and target is computed. Then, the factored policy is used to emulate different behaviors that are allocated dynamically to the UAVs, fostering cooperation between them. Furthermore, a decentralized data fusion system is used to obtain a belief of the joint factored state integrating observation from all team-members.

While the method and architecture are general, we apply them to the case of multi-target surveillance. Our main contributions are showing that POMDPs are an

[1] A sample video can also be found at http://personal.us.es/jcapitan/capitan_jint.mp4

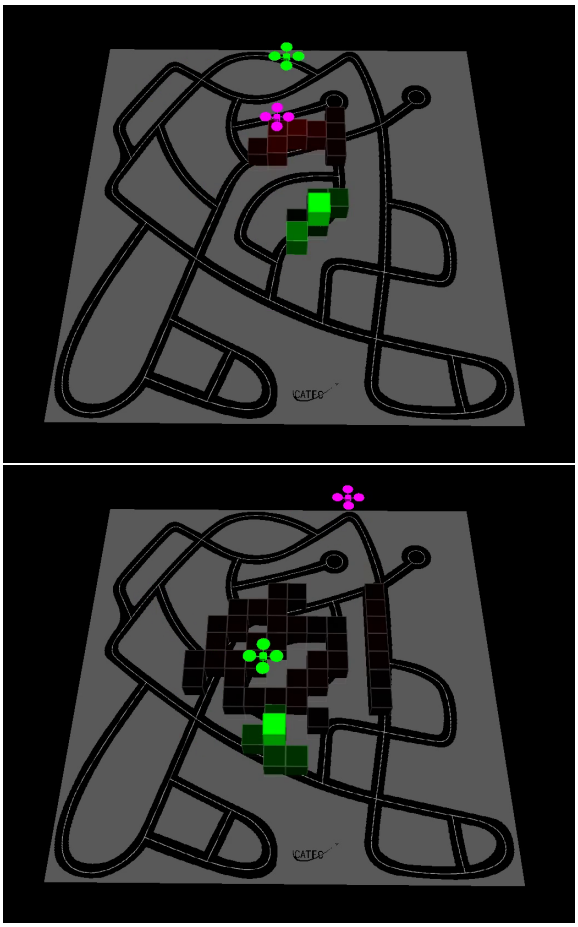

Fig. 7: Two snapshots of our visualizer during an experiment. A map of the city is shown. The beliefs of the positions of the two targets are represented in red and green, respectively. Top: the estimations of the two targets are quite informative, and both quadcopters follow different cars. Bottom: one of the quadcopters is tracking the green target, while the other is searching for the red one. The magenta quadcopter is waiting in one of the endings of the tunnel. The information provided by the green quadcopter about the green target is also fused by the magenta quadcopter.

adequate model for multi-UAV planning and making the system scalable with the number of targets and UAVs. The computational complexity to compute the policies does not grow with more UAVs or targets, and the complexity of the auction and the decentralized data fusion only depend on the size of the local communication neighborhood of each UAV. Moreover, the methods have been evaluated using simulations, both from a quantitative and qualitative points of view. We have also implemented our system in a testbed with real quadcopters tracking moving cars in a city-like scenario.

As future work, instead of predefining them, the dynamic and observation models could be learned from data from previous missions. A key issue in our approach is also a proper definition of the reward (cost) functions. An interesting line of research is learning those cost functions from observing human experts determining the actions to be carried out. Also, experiments with larger teams and more complex scenarios will be performed to highlight the benefits of the method presented.

References

1. Anderson, R., Milutinovic, D.: Anticipating stochastic observation loss during optimal target tracking by a small aerial vehicle. In: Unmanned Aircraft Systems (ICUAS), 2013 International Conference on, pp. 278–287. IEEE (2013)
2. Bar-Shalom, Y., Li, X., Kirubarajan, T.: Estimation with Applications to Tracking and Navigation. Wiley Interscience (2001)
3. Beard, R., McLain, T., Nelson, D., Kingston, D., Johanson, D.: Decentralized cooperative aerial surveillance using fixed-wing miniature uavs. Proceedings of the IEEE **94**(7), 1306–1324 (2006)
4. Bernstein, D.S., Givan, R., Immerman, N., Zilberstein, S.: The complexity of decentralized control of Markov decision processes. Mathematics of Operations Research **27**(4), 819–840 (2002)
5. Bourgault, F., Furukawa, T., Durrant-Whyte, H.: Decentralized bayesian negotiation for cooperative search. In: Proceedings of International Conference on Intelligent Robots and Systems, pp. 2681–2686 (2004)
6. Burdakov, O., Doherty, P., Holmberg, K., Kvarnstrom, J., Olson, P.M.: Relay positioning for unmanned aerial vehicle surveillance. International Journal of Robotics Research **29**(8), 1069–1087 (2010)
7. Capitan, J., Merino, L., Caballero, F., Ollero, A.: Decentralized delayed-state information filter (DDSIF): A new approach for cooperative decentralized tracking. Robotics and Autonomous Systems **59**, 376–388 (2011). DOI DOI:10.1016/j.robot.2011.02.001. URL `http://www.sciencedirect.com/science/article/B6V16-526DWNX-1/2/dd823dbe262732b7d71fecdeb8cd1354`
8. Capitan, J., Spaan, M., Merino, L., Ollero, A.: Decentralized multi-robot cooperation with auctioned POMDPs. International Journal of Robotics Research **32**(6), 650–671 (2013). DOI DOI:10.1177/0278364913483345
9. Cook, K., Bryan, E., Yu, H., Bai, H., Seppi, K., Beard, R.: Intelligent cooperative control for urban tracking with unmanned air vehicles. In: Unmanned Aircraft Systems (ICUAS), 2013 International Conference on, pp. 1–7. IEEE (2013)
10. Grocholsky, B., Makarenko, A., Kaupp, T., Durrant-Whyte, H.: Scalable control of decentralised sensor platforms. In: F. Zhao, L. Guibas (eds.) Information Processing in Sensor Networks, *Lecture Notes in Computer Science*, vol. 2634, pp. 551–551. Springer Berlin / Heidelberg (2003)
11. He, R., Bachrach, A., Roy, N.: Efficient Planning under Uncertainty for a Target-Tracking Micro-Aerial Vehicle. In: IEEE International Conference on Robotics and Automation, 2010 (2010)
12. Hsieh, M.A., Cowley, A., Keller, J.F., Chaimowicz, L., andCamillo J. Taylor, B.G.V.K., End, Y., Arkin, R.C., Jung, B., Wolf, D.F., Sukhatme, G.S., MacKenzie, D.C.: Adaptive teams of autonomous aerial and ground robots

for situational awareness. Journal of Field Robotics **24**, 991–1014 (2007)
13. Hsu, D., Lee, W., Rong, N.: A point-based POMDP planner for target tracking. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 2644–2650 (2008)
14. Kaelbling, L.P., Littman, M.L., Cassandra, A.R.: Planning and acting in partially observable stochastic domains. Artificial Intelligence **101**, 99–134 (1998)
15. Matignon, L., Jeanpierre, L., Mouaddib, A.: Distributed Value Functions for Multi-Robot Exploration. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 1544–1550. St. Paul, USA (2012)
16. Morbidi, F., Mariottini, G.L.: On active target tracking and cooperative localization for multiple aerial vehicles. In: Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on, pp. 2229–2234. IEEE (2011)
17. Nair, R., Varakantham, P., Tambe, M., Yokoo, M.: Networked distributed POMDPs: A synthesis of distributed constraint optimization and POMDPs. In: Proceedings of the National Conference on Artificial Intelligence, pp. 133–139 (2005)
18. Ong, L., Upcroft, B., Bailey, T., Ridley, M., Sukkarieh, S., Durrant-Whyte, H.: A decentralised particle filtering algorithm for multi-target tracking across multiple flight vehicles. In: IROS, pp. 4539–4544 (2006)
19. Ong, S.C., Png, S.W., Hsu, D., Lee, W.S.: POMDPs for Robotic Tasks with Mixed Observability. In: Proceedings of the Robotics: Science and Systems Conference. Seattle, USA (2009)
20. Papadimitriou, C.H., Tsitsiklis, J.N.: The complexity of markov decision processes. Mathematics of operations research **12**(3), 441–450 (1987)
21. Poupart, P.: Exploiting Structure to Efficiently Solve Large Scale Partially Observable Markov Decision Processes. Ph.D. thesis, University of Toronto (2005)
22. Pynadath, D.V., Tambe, M.: The communicative multiagent team decision problem: Analyzing teamwork theories and models. Journal of Artificial Intelligence Research **16**, 389–423 (2002)
23. Skoglar, P., Orguner, U., Gustafsson, F.: On information measures based on particle mixture for optimal bearings-only tracking. In: Aerospace conference, 2009 IEEE, pp. 1–14. IEEE (2009)
24. Wong, E.M., Bourgault, F., Furukawa, T.: Multi-vehicle Bayesian search for multiple lost targets. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 3169–3174 (2005)