



## Open Archive Toulouse Archive Ouverte (OATAO)

OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible.

This is an author-deposited version published in: <http://oatao.univ-toulouse.fr/>  
Eprints ID: 11442

**To cite this document:** Ponzoni Carvalho Chanel, Caroline *Planning for perception and perceiving for decision: POMDP-like online optimization in large complex robotics missions*. (2012) In: The International Conference on Automated Planning and Scheduling (ICAPS) Doctoral Consortium, 19 June 2012 - 25 June 2012 (Atibaia, São Paulo, Brazil). (Unpublished)

Any correspondence concerning this service should be sent to the repository administrator: [staff-oatao@inp-toulouse.fr](mailto:staff-oatao@inp-toulouse.fr)

# Planning for perception and perceiving for decision: POMDP-like online optimization in large complex robotics missions

**Caroline P. Carvalho Chanel**

Supervisors: Florent Teichteil-Königsbuch and Patrick Fabiani

Onera – The french aerospace lab

Université de Toulouse

ISAE – Institut Supérieur de l’Aéronautique et de l’Espace

2, avenue Edouard Belin

FR-31055 TOULOUSE

## Abstract

This ongoing PhD work aims at proposing a unified framework to optimize both perception and task planning using extended Partially Observable Markov Decision Processes (POMDPs). Targeted applications are large complex aerial robotics missions where the problem is too large to be solved off-line, and acquiring information about the environment is as important as achieving some symbolic goals. Challenges of this work include: (1) optimizing a dual objective in a single decision-theoretic framework, i.e. environment perception and goal achievement ; (2) properly dealing with action preconditions on belief states in order to guarantee safety constraints or physical limitations, what is crucial in aerial robotics ; (3) modeling the symbolic output of image processing algorithms as input of the POMDP’s observation function ; (4) parallel optimization and execution of POMDP policies in constrained time. A global view of each of these topics are presented, as well as some ongoing experimental results.

## Introduction

Many realistic applications of Artificial Intelligence require to plan actions with incomplete information of the state of the world. In this case, the agent may gather information as the same time as it performs actions to reach the mission goal. A natural formal framework for sequential decision-making under uncertainty on the result of actions and on observations is the Partially Observable Markov Decision Process (POMDP) (Kaelbling, Littman, and Cassandra 1998).

Modeling the double objective of perception and mission goal achievement as a POMDP remains a relative complex issue (Spaan 2008; Araya-López et al. 2010). The optimized policy tends to reach mission goals by maximizing rewards defined over the unobservable states of the system. Perception actions, which aim at gathering information about the environment, are implicitly optimized to minimize the uncertainty on the current state of the world. Yet, maximizing expected accumulated rewards does not necessarily maximize knowledge about hidden states. For us, perception is also an end in itself: in addition to plan for mission tasks, we also want to plan for perception action, e.g. optimizing

the sequence of actions required to analyze an object in the scene. We aim at proposing a single framework that maximizes both information and reward gathering. We believe that its is really important for some robotics applications like target tracking, target detection and identification, area surveillance. To this end, our PhD work has partly focused on dual criterion for POMDPs, which aggregates rewards associated with the hidden states of the world and entropy-like rewards that explicitly measure the knowledge of the agent about the system (Carvalho Chanel et al. 2010).

In other hand, acting in partially observable worlds while avoiding to apply dangerous actions is also a harsh issue. In some critical robotics missions, e.g. aerial robotics, the agent must apply only *safe* decisions whatever the hidden state of the world ; safety constraints are carefully taken into account by Unmanned Aerial Vehicle (UAV) system designers and airspace certifying authorities. As far as we know, there is no work that properly formalize action preconditions in the context of partial probabilistic observability. Thus, we have also studied a decoupled way of modeling strong action preconditions in a POMDP framework (Carvalho Chanel et al. 2011). First results show that the new proposed general model guarantees that forbidden actions will not be executed by the agent.

Another point frequently disregarded in POMDP, yet crucial in real applications, is the accuracy of the observation model. POMDP optimization assumes that the probabilistic transition and observation models are accurate, what is challenging in real-world applications for which probabilistic distributions are rarely available. Instead, we can learn the transition and observation models from real data (Spaan 2008). It could be interesting to study an optimization operator or criterion which provides policies that are robust to this kind of imprecision of the model .

Furthermore, planning in probabilistic domains is known to be both time and memory consuming, due to the high complexity of exploring many belief states that are probably reachable by the current policy (Ross et al. 2008). Thus, when applied to autonomous systems subject to time and memory constraints, it is common to generate policies offline, then embedding and executing them on-board. However, this approach is not applicable if the policy is too big or complex to be embedded on-board, or if the planning problem to solve is not known before execution. All of these

restrictions hold for our robotics missions presented later. Therefore, we propose a continuous planning algorithm that optimizes and executes policies in parallel, dealing with time constraints due to the mission achievement's deadline and to the interaction with other functions (image processing, guidance laws, etc.) of the robot.

### Mission example

As an illustrating example, let us consider an autonomous UAV which must detect and recognize some targets under real world constraints. The mission consists in detecting and identifying a car of a particular model among several cars in the scene, and land next to this car. Due to the nature of the problem, it is modeled as a POMDP. The UAV can perform both high-level mission tasks (moving between zones, changing height level, land) and perception actions (change view angle in order to observe the cars). Cars can be in any of the zones beforehand extracted by image processing (no more than one car per zone). The total number of states depends on the number of zones, the height levels, the view angles, the number of targets and car models. In this test case, we consider 4 possible observations in each state: {*car not detected*, *car detected but not identified*, *car identified as target*, *car identified as non-target*}.

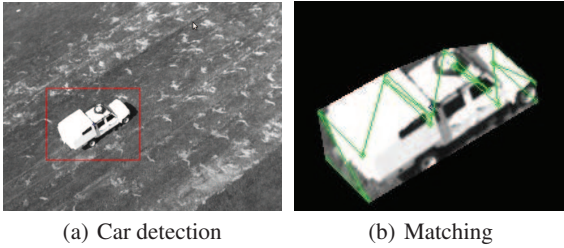


Figure 1: Target detection and recognition.

### Formal baseline framework: POMDP

A POMDP is a tuple  $\langle S, A, \Omega, T, O, R, b_0 \rangle$  where  $S$  is a set of states,  $A$  is a set of actions,  $\Omega$  is a set of observations,  $T : S \times A \times S \rightarrow [0; 1]$  is a transition function such that  $T(s_t, a, s_{t+1}) = P(s_{t+1} | a, s_t)$ ,  $O : \Omega \times S \rightarrow [0; 1]$  is an observation function such that  $O(o_t, s_t) = P(o_t | s_t)$ ,  $R : S \times A \rightarrow \mathbb{R}$  is a reward function associated with a pair state-action, and  $b_0$  is a probability distribution over initial states. We note  $\Delta$  the set of probability distributions over the states, named *belief state space*. At each time step  $t$ , the agent updates its *belief state* defined as an element  $b_t \in \Delta$ .

Solving POMDPs consists in constructing a policy function  $\pi : \Delta \rightarrow A$ , which maximizes some criterion generally based on rewards averaged over belief states. In robotics, where symbolic rewarded goals must be achieved, it is usually accepted to optimize the long-term average discounted accumulated rewards from any initial belief state (Cassandra, Kaelbling, and Kurien 1996; Spaan and Vlassis 2004):

$$V^\pi(b) = E_\pi \left[ \sum_{t=0}^{\infty} \gamma^t r(b_t, \pi(b_t)) \mid b_0 = b \right] \quad (1)$$

where  $\gamma$  is the actualization factor. The optimal value  $V^*$  of a optimal policy  $\pi^*$  is defined by the value function that satisfies the bellman's equation:

$$V^*(b) = \max_{a \in A} \left[ \sum_{s \in S} r(s, a) b(s) + \gamma \sum_{o \in O} p(o|a, b) V^*(b_a^o) \right] \quad (2)$$

Following from optimality theorems, the optimal value of belief states is piecewise linear and convex (Smallwood and Sondik 1973), i.e., at the step  $n \leq \infty$ , the value function can be represented by a set of hyperplanes over  $\Delta$ , known as  $\alpha$ -vectors. An action  $a(\alpha_n^i)$  is associated with each  $\alpha$ -vector, that defines a region in the belief state space for which this  $\alpha$ -vector maximizes  $V_n$ . Thus, the value of a belief state can be defined as  $V_n(b) = \max_{\alpha_n^i \in V_n} b \cdot \alpha_n^i$ . And the optimal policy in this step will be  $\pi_n(b) = a(\alpha_n^b)$ .

Recent offline solving algorithms, e.g. PBVI (Pineau, Gordon, and Thrun 2003), HSVI2 (Smith and Simmons 2005), SARSOP (Kurniawati, Hsu, and Lee 2008) and symbolic PERSEUS (Poupart 2005), approximate the value function with a bounded set of belief states  $B$ , where  $B \subset \Delta$ . These algorithms implement different heuristics to explore the belief state space, and update the value of  $V$ , which is represented by a set of  $\alpha$ -vectors, by a backup operator for each  $b \in B$  explored or relevant. Therefore,  $V$  is reduced and contains a limited number  $|B|$  of  $\alpha$ -vectors.

We claim that optimizing belief state values, which are piecewise linear, as in eq. 1 provides a relatively too simple mathematical model for POMDP reasoning on perception applications (Spaan 2008; Araya-López et al. 2010). Indeed, linearizing belief states' average value comes back to flatten observations and to finally loose distinctive information about them. Thus, the optimized policy does not lead to acquire sufficient information about the environment before acting to gather rewards when perception actions are at stake: as discussed later, such a strategy unfortunately results in less reward gathered at execution than expected.

### Active Perception

Looking at the literature about active perception, which aims at maximizing the information gain gathered from environment (Deinzer, Denzler, and Niemann 2003; Eidenberger et al. 2008), we note that the optimization criterion often used relies on Shannon's entropy. The latter represents the amount of information contained in the belief. An example of criterion, where Shannon's entropy is accumulated along expected trajectories, is:

$$H^\pi(b) = E_\pi \left[ \sum_{t=0}^{\infty} \gamma^t \sum_{s \in S} b_t(s) \log(b_t(s)) \mid b_0 = b \right] \quad (3)$$

On the contrary to the criterion of eq. 1, the one of eq. 3 is non linear over belief states, what offers the possibility to make a clear distinction between observations that reduce uncertainty and others. Note that this criterion does not take into account standard POMDP rewards defined over state-action pairs, for instance associated with mission goals.

### Mixed criterion

(Mihaylova et al. 2002) formalizes the active perception problem as a weighted sum of uncertainty measures and costs. Based on this work, we proposed a new optimization criterion (Carvalho Chanel et al. 2010), which aggregates averaged rewards and Shannon's entropies of the belief

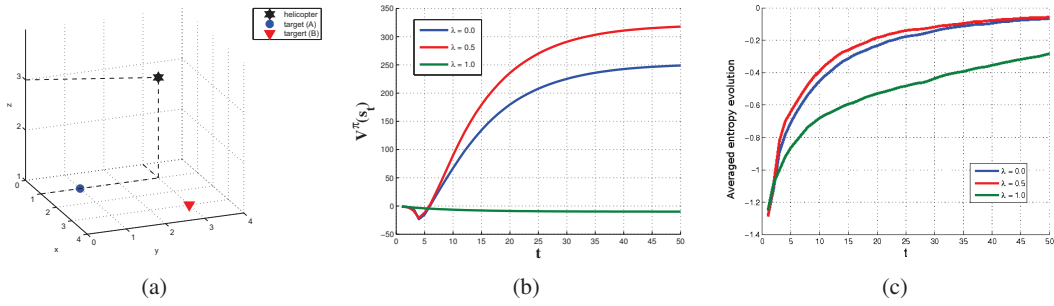


Figure 2: (a) Initial position of the agent, target 1 (A) and target 2 (B).; (b) Averaged value function of the current state  $V^\pi(s_t) = \frac{1}{500} \sum_{k=0}^t \gamma^k r^\pi(s_k | s_t, b_0)$ . (c) Evolution of averaged weighted sum of entropies to the different criteria.

state:  $J_\lambda(V^\pi(b_0), H^\pi(b_0))$ , where  $\lambda \in [0, 1]$ . In this way, optimized policies would sufficiently often execute information acquisition actions in order to reach the mission goal with minimal guarantee on the knowledge of the world. The entropy represents a penalty for bigger uncertainties (supposing that the two criteria are well balanced), what urges the agent on quickly acquiring information.

Some authors studied applications with a fixed value of  $\lambda$  and no long-term accumulation of entropy (Burgard, Fox, and Thrun 1997). Others defined a new model  $\rho$ POMDP (Araya-López et al. 2010), but without providing solving frameworks. Our approach proposes a mixed optimization criterion and corresponding solving algorithms: the expected sum of rewards is added to the expected sum of belief states' entropies. These values are weighted by a constant  $\lambda$ , in order to well balanced the influence of these two criteria:

$$J^\pi(b) = (1 - \lambda)V^\pi(b) + \lambda H^\pi(b) \quad (4)$$

where  $V^\pi(b)$  is defined as in eq. 1 and  $H(b_t)$  given by eq. 3 (same discount factor  $\gamma$ ). Note that the addition of these two heterogeneous criteria does not a priori invalidate the optimization scheme used, because the value only depends on  $b \in \Delta$ . Actually, Bellman's equation with  $\alpha$ -vectors are still valid, because the mixed criterion remains convex. However, it is no more linear in  $\Delta$ , and some modifications are needed so that standard POMDP algorithms can be used, as proposed by (Araya-López et al. 2010) in a more general case.

**Results for the mixed criterion** The model studied is close to the mission example presented in introduction section. The objective is to identify mobile targets and land next to the target of interest. Targets may be of two types A or B. The agent, which is an autonomous helicopter, must land next to the target of type A, without initially knowing the real nature of each target. This scenario combines perception and mission goal: it is necessary to reduce the uncertainty about targets to identify them with sufficient confidence (perception) and land next to the desired target (mission goal).

In this model, the agent can perform 7 actions in an environment sketched in Fig. 2(a): go ahead or go to  $x$ ,  $y$  and  $z$  axis (cost of 1), and land (cost or reward of 100 according to the target). Moving actions are probabilistic, except landing. The position of the agent and of the targets are completely observable. Yet, the nature of the targets is partially observ-

able and its observation model depends on the distance between the helicopter and the targets.

We optimized policies for different values of  $\lambda = \{0, 0.5, 1\}$  using Symbolic PERSEUS (Poupart 2005), which we adapted for our new mixed criterion.  $\lambda = 0$  represents the classical total discounted reward criterion used in POMDPs.  $\lambda = 0.5$  equally takes into account the classical criterion and the weighted sum of entropies.  $\lambda = 1.0$  comes back to the criterion often used in active perception, based on Shannon's entropy. We compare these policies over  $V^\pi(s_t)$ , which, independently of the criterion used, represents the rewards actually won at execution from the hidden initial state. In Fig. 2(b),  $V^\pi(s_t)$  is shown for the three cases.

In the first case,  $\lambda = 0$ , the agent tends to early land (within 3 or 4 steps) close to the target that it *believes* to be the good one, often wrongly, because landing is the mission goal optimized via reward maximization. However, in simulations where it could acquire more information about the environment, the autonomous agent could land close to the actually good target. For  $\lambda = 0.5$ , one can verify that the criterion value is higher at execution: landing actions applied sooner than for the classical criterion impact the hidden value function  $V^\pi(s_t)$ . Explicitly acquiring more information about the environment allows the agent to earlier reduce uncertainty on its belief state, and finally often land next to the good target (364 versus 326). The non linearity of this criterion enables to better evaluate the value of  $b(s)$ , giving more weight to more accurate beliefs (in the sense of the entropy). For  $\lambda = 1$ , the criterion only optimizes the information gain, and so the averaged  $V^\pi(s)$  stays over zero: the agent does not land, because the gain associated with the land action is not taken into account. Therefore, our results show that accumulated entropies and rewards should be optimized hand in hand to maximize the value function from the hidden initial state, which represents the rewards actually gathered at execution (but not the value function averaged over *belief* states).

### Action preconditions for safe policy execution

In classical planning, preconditions are widely used to model environment properties required to perform an action. Preconditions are boolean-valued formulas that represent the definition domain of an action, i.e. the set of states on which this action can be applied (Ghallab, Nau, and Traverso

2004). In real-world applications, securing such guarantees is mandatory in order, for instance, to protect a robot against physical damage or to put its environment in jeopardy.

To our knowledge, proper use of preconditions has never been adapted for POMDPs, despite identical theoretical and practical needs. Research in POMDPs is still more focused on improving the efficiency of general algorithms tackling the complexity of general POMDP models, rather than on real world applications. Technically speaking, precondition checking is not straightforward when working on probabilistic partially observable domains, because the current state is not known precisely and it is replaced by a probability distribution over a set of states.

Given our mission example, even if the desired target is found in a particular zone, maybe this zone is actually not landable. To properly model this kind of information in a POMDP framework, it would be necessary to add a state variable that indicates if it is possible to land or not in a particular zone. Moreover, it would be necessary to add a corresponding observation variable *and* a very high cost for the land action if it is not applicable in the states coherent with the observation received (Pineau, Gordon, and Thrun 2003; Smith and Simmons 2005; Poupart 2005). People use to associate a finite value that represents a high cost to the state-action pair. This value aims at guaranteeing that no infeasible action will be performed. Actually, it is hard to properly set this value beforehand, which depends on the unknown optimal value of states. This solution is not satisfactory for our UAV applications. Moreover, the more observation variables are added to the model, the more complex is the optimization. Given that the number of  $\alpha$ -vectors grows exponentially with the number of observations (Pineau, Gordon, and Thrun 2003).

An alternative solution to the hazardous adjustment of costs consists in defining an indicative function that informs about the feasible actions given a particular state:  $\mathbb{I}(a, s)$ . We define  $\mathcal{A}_f(s)$  as the set of feasible actions in  $s$ . Thus, preconditions based on states are defined by  $\mathbb{I}(a, s) = \mathbf{1}_{a \in \mathcal{A}_f(s)}$  where  $\mathbf{1}_{cond}$  is 1 if the condition *cond* is true, or 0 if not.  $\mathbb{I}(a, s)$  can be seen as the probability 1 or 0 of the applicability of an action  $a$  given a state  $s$ , i.e.  $\mathbb{I}(a, s) = Pr(a \in \mathcal{A}_f(s) | s_t = s)$ . At each optimization step, we apply the max operator only over the set of actions defined by:

$$\mathcal{A}_f(b) = \{a \in A | \exists s \in S, b(s) \neq 0 \wedge \mathbb{I}(a, s) = 1\} \quad (5)$$

which results in the following optimization equation:

$$V_{n+1}(b) = \max_{a \in \mathcal{A}_f(b)} \sum_{s \in S} r(s, a) b(s) + \gamma \sum_{o \in \Omega} p(o|a, b) V(b_a^o) \quad (6)$$

This optimization framework would ensure that no infeasible action will be performed by the optimized policy. But, to guarantee it, we actually need to add to the model the state and observation variables informing about the possibility of landing in a particular zone, what increase the complexity of the problem as discussed earlier.

### Decoupled approach

Properly dealing with preconditions in a POMDP framework almost without increasing the optimization complex-

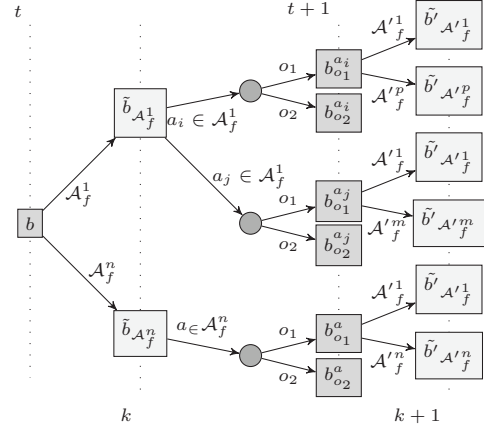


Figure 3: Proposed decision schema.

ity, requires add an additional information step to the classical POMDP model, and changing the set of states without changing the set of observations (Carvalho Chanel et al. 2011). This additional information step takes place before decision, and informs the agent about the set of applicable actions. It allows to reduce the probability distribution over states that have the same set of applicable actions. The set of applicable actions could be seen as a new kind of observation, but it is decoupled from standard POMDP observations, in the sense that it is received independently from standard observations. Thus, the optimization complexity admittedly increases, but not as much as if new observations were added to the set of standard observations.

As shown in Fig. 3, the decision schema changes, what implies to redefine the belief state update step, and the  $\max_a$  optimization operator of Bellman's equation. Foremost, the joint indicative function of a set of applicable actions  $\mathcal{U} \subset A$  given  $s$  is formalized as:

$$\mathbb{I}(\mathcal{U}, s) = \prod_{a_i \in \mathcal{U}} \mathbb{I}(a_i, s) \prod_{a_j \notin \mathcal{U}} (1 - \mathbb{I}(a_j, s)) \quad (7)$$

We directly have:  $\mathbb{I}(\mathcal{A}_f(s), s) = 1$ . Furthermore,  $\mathbb{I}(\mathcal{U}, s) = P(\{\mathcal{A}_f(s) = \mathcal{U} | s\})$  is the probability 1 or 0 that a set of applicable actions  $\mathcal{U}$  conditioned on  $s$  is equal to the set  $\mathcal{A}_f(s)$ .

### Belief state update steps.

1. A projection of  $b$  is created for each possible set of applicable actions  $\mathcal{A}_f^i$ .<sup>1</sup>

$$\tilde{b}_t(s)_{\mathcal{A}_f^i} = \frac{\mathbb{I}(\mathcal{A}_f^i | s_t = s) b_t(s)}{\sum_{s'' \in S} \mathbb{I}(\mathcal{A}_f^i | s_t = s'') b_t(s'')} \quad (8)$$

2. An action  $a \in \mathcal{A}_f^i$  is chosen and an observation  $o$  is received, bringing the agent to a belief state  $b_a^o$ .

$$b_{a \in \mathcal{A}_f^i}^o(s') = \frac{p(o|s') \sum_{s \in S} p(s'|s, a) \tilde{b}_t(s)}{\sum_{s \in S} \sum_{s' \in S} p(o|s') p(s'|s, a) \tilde{b}_t(s)} \quad (9)$$

<sup>1</sup>If  $b$  is incompatible with the set of applicable actions  $\mathcal{A}_f^i$ , the denominator becomes zero, what invalidates Bayes' rule. In this case, we can choose to initialize the belief state with  $b(s) = 1/N$ , where  $N$  represents the number of states for which the set of applicable actions is the same  $Pr(\mathcal{A}_f^i | s_t = s) = 1$ .

The new information step does not invalidate the assumption that the belief state is a complete information state, because it still fulfils Markov's property. Moreover, the agent must take its decision based on  $\tilde{b}$ 's in steps  $k$  and  $k + 1$  (see Fig. 3), after the information about the set of applicable actions is received. To handle it, the optimization algorithm needs to determine a policy for the  $\tilde{b}$ 's.

**Optimization operator.** We define the optimization operator that computes the expected averaged value of a belief state  $\tilde{b}$  via the value of belief states  $b_{a \in \mathcal{A}_f}^o$ :

$$V_{k+1}(\tilde{b}_{\mathcal{A}_f}) = \max_{a \in \mathcal{A}_f} \{r(\tilde{b}, a) + \sum_o p(o|a, \tilde{b}) V_n(b_a^o)\} \quad (10)$$

where  $\mathcal{A}_f$  represents the set of applicable actions associated with  $\tilde{b}$ . We note that the value  $V_{k+1}(\tilde{b}_{\mathcal{A}_f})$  is linear in  $\tilde{b}$ , and depends on the value of future belief states  $b_{a \in \mathcal{A}_f}^o$ . We compute  $V_n(b_{a \in \mathcal{A}_f}^o)$  as averaged over future  $\tilde{b}_{\mathcal{A}'_f}$  values:

$$V_n(b) = \sum_{i=1}^C Pr(\mathcal{A}_f^i) V_k(\tilde{b}_{\mathcal{A}'_f^i}(b)) \quad (11)$$

where  $C$  represents the number of  $\tilde{b}$  successors of  $b$ .

We have worked on this approach and first results were presented in (Carvalho Chanel et al. 2011), which have been skipped here because of lack of space. They demonstrated that our decoupled approach produces policies that never perform illegal actions, and that the experimental computational complexity is competitive with standard "penalizing" approaches.

### Learning the real observation model

POMDP models require a proper probabilistic description of actions' effects and observations, what is difficult to obtain in practice for real complex applications. For our target detection and recognition missions, we automatically learned from real data the observation model, which relies on image processing. We recall that we consider 4 possible observations in each state:  $\{car \text{ not detected}, car \text{ detected but not identified}, car \text{ identified as target}, car \text{ identified as non-target}\}$ . The key issue is to assign a prior probability on the possible semantic outputs of image processing given a particular scene.

Some campaigns were performed and led to an observation model learned via a statistical analysis of the image processing algorithm's answers (see Fig. 1). Image processing is described in (Saux and Sanfourche 2011), and is already embedded on autonomous UAVs. More precisely, we count the number of times that one of the four observations was answered by the image processing algorithm in a given state  $s$ . So, we compute  $p(o_i|s)$  by:  $p(o_i|s) \simeq$

$$\frac{1}{N_{exp}} \sum_{n=1}^{N_{exp}} \mathbb{I}_{\{o_n=o_i|s\}}, \quad N_{exp} \gg 1.$$

An important point is that the observation model is not really accurate. In other words, one can compute a confidence interval for the average probability calculated  $p(o_i|s)$ . Classical POMDP frameworks assume that the observation model is exact, but this condition rarely holds in real applications. A possible solution would consist in extending the POMDP optimization framework in order to take into account this confidence interval on the observation model. Some authors have been working on this subject (Itoh and

Nakamura 2007; Ni and Liu 2008). Nevertheless, many challenges still remain, especially for the determination of the worst model when the objective is to provide a robust policy to face up the inaccuracy of models.

### Parallel optimization and execution under time constraints

Large and complex POMDP problems can be rarely optimized off-line, because of lack of sufficient computation means. Moreover, the problem to solve is not always known in advance, e.g. our target detection and recognition missions where the POMDP problem is based on zones that are automatically extracted from online images of the environment. Such applications require an efficient online framework for solving POMDPs and executing policies, before the mission's deadline. We have proposed a versatile optimize-while-execute framework to on-line solve large POMDPs under time constraints, as part of a generic meta (PO)MDP planner (Teichteil-Konigsbuch, Lesire, and Infantes 2011). The meta planner relies on standard POMDP planners like PBVI, HSVI, PERSEUS, etc., which are called from possible future execution states while executing the current optimized action in the current execution state, in anticipation of the probabilistic evolution of the system and its environment. This framework is different from real-time algorithms like RTDP-Bel (Bonet and Geffner 2009) that solve the POMDP only from the current execution state (but not future possible ones).

### Work in progress

We have implemented our meta planner with the anytime POMDP algorithms PBVI (Pineau, Gordon, and Thrun 2003) and AEMS (Ross and Chaib-Draa 2007). AEMS is particularly useful for our optimize-while-execute framework with time constraints, since we can explicitly control the time spent by AEMS to optimize an action in a given belief state. The meta planner handles planning and execution requests in parallel, as shown in Fig. 4. At a glance, it works as (for details, see (Teichteil-Konigsbuch, Lesire, and Infantes 2011)):

1. Initially, our meta-planner plans for an initial belief state  $b$  during a certain amount of time (bootstrap), using AEMS from  $b$ .
2. After bootstrap, our meta-planner receives an execution request, to which it returns back the action optimized by AEMS for  $b$ .
3. The approximated execution time of the returned action is estimated, for instance 6 seconds, so that the meta planner will plan from some next possible belief states using AEMS during a portion of this time (e.g. 2 seconds each for 3 possible future belief states), while executing the returned action.
4. After action execution, an observation is received and the belief state is updated to a new  $b'$ , for which the current optimized action is sent by the meta-planner to the execution engine.

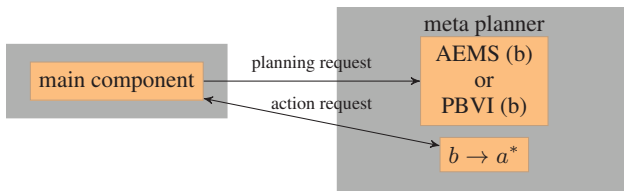


Figure 4: Meta planner planning / execution schema.

## Perspectives and future work

We are going to perform in the next weeks a first outdoor experimentation of our complete system (mixed criterion, safe action preconditions, online meta-planner) for real target detection and recognition missions. Next, we hope to enrich the scenario with multi-sensors, e.g. multiple cameras, lasers, ultrasonic sound.

As previously highlighted, a very interesting research point would be to consider the impact of an inaccurate or imprecise observation model on the optimized policy. The decision maker would provide a set of possible observation models for each state-observation pairs, and optimize the POMDP for the worst observation model for instance. But, nowadays, many difficulties have been raised in the community, especially concerning the choice of the worst model, or the correct update of the belief state.

## References

- Araya-López, M.; Buffet, O.; Thomas, V.; and Charpillet, F. 2010. A POMDP Extension with Belief-dependent Rewards. *Advances in Neural Information Processing Systems* 23.
- Bonet, B., and Geffner, H. 2009. Solving POMDPs: RTDP-bel vs. point-based algorithms. In *Proceedings of the 21st international joint conference on Artificial intelligence, IJCAI'09*, 1641–1646. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
- Burgard, W.; Fox, D.; and Thrun, S. 1997. Active mobile robot localization. In *Proc. of IJCAI-97*. Morgan Kaufmann.
- Carvalho Chaneil, C.; Farges, J.; Teichteil-Königsbuch, F.; and G. Infantes. 2010. POMDP solving: what rewards do you really expect at execution? In *Proc. of the 5th Starting AI Researchers' Symposium*.
- Carvalho Chaneil, C.; Teichteil-Königsbuch, F.; Infantes, G.; and Fabiani, P. 2011. Modeling action feasibility in pomdps with boolean-valued preconditions. In *IJCAI Workshop on Decision Making in Partially Observable, Uncertain Worlds: Exploring Insights from Multiple Communities*.
- Cassandra, A.; Kaelbling, L.; and Kurien, J. 1996. Acting under uncertainty: Discrete Bayesian models for mobile-robot navigation. In *Proceedings of IEEE/RSJ*.
- Deinzer, F.; Denzler, J.; and Niemann, H. 2003. Viewpoint selection-planning optimal sequences of views for object recognition. *Lecture notes in computer science* 65–73.
- Eidenberger, R.; Grundmann, T.; Feiten, W.; and Zoellner, R. 2008. Fast parametric viewpoint estimation for active object detection. In *Proc. of the IEEE International Conference on Multisensor of Fusion and Integration for Intelligent Systems (MFI 2008)*, Seoul, Korea.
- Ghallab, M.; Nau, D.; and Traverso, P. 2004. *Automated Planning: theory and practice*. Morgan Kaufmann.
- Itoh, H., and Nakamura, K. 2007. Partially observable markov decision processes with imprecise parameters. *Artificial Intelligence* 171(8-9):453–490.
- Kaelbling, L.; Littman, M.; and Cassandra, A. 1998. Planning and acting in partially observable stochastic domains. *AIJ* 101(1-2).
- Kurniawati, H.; Hsu, D.; and Lee, W. 2008. SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces. In *Proc. RSS*.
- Mihaylova, L.; Lefebvre, T.; Bruyninckx, H.; Gadeyne, K.; and Schutter, J. D. 2002. Active sensing for robotics – a survey. In *5th Intl Conf. On Numerical Methods and Applications*, 316–324.
- Ni, Y., and Liu, Z. 2008. Bounded-parameter partially observable markov decision processes. In *Proc. of the 18th International Conference on Automated Planning and Scheduling (ICAPS)*, 240–247.
- Pineau, J.; Gordon, G.; and Thrun, S. 2003. Point-based value iteration: An anytime algorithm for POMDPs. In *Proc. of IJCAI*.
- Poupart, P. 2005. *Exploiting structure to efficiently solve large scale partially observable Markov decision processes*. Ph.D. Dissertation, University of Toronto.
- Ross, S., and Chaib-Draa, B. 2007. Aems: An anytime online search algorithm for approximate policy refinement in large pomdps. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07)*, 2592–2598.
- Ross, S.; Pineau, J.; Paquet, S.; and Chaib-Draa, B. 2008. Online planning algorithms for pomdps. *Journal of Artificial Intelligence Research* 32(1):663–704.
- Saux, B., and Sanfourche, M. 2011. Robust vehicle categorization from aerial images by 3d-template matching and multiple classifier system. In *Image and Signal Processing and Analysis (ISPA), 7th International Symposium on*, 466–470. IEEE.
- Smallwood, R., and Sondik, E. 1973. The optimal control of partially observable Markov processes over a finite horizon. *Operations Research* 1071–1088.
- Smith, T., and Simmons, R. 2005. Point-based POMDP algorithms: Improved analysis and implementation. In *Proc. UAI*.
- Spaan, M., and Vlassis, N. 2004. A point-based POMDP algorithm for robot planning. In *ICRA*.
- Spaan, M. 2008. Cooperative Active Perception using POMDPs. *Association for the Advancement of Artificial Intelligence - AAAI*.
- Teichteil-Königsbuch, F.; Lesire, C.; and Infantes, G. 2011. A generic framework for anytime execution-driven planning in robotics. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 299–304. IEEE.