Universiteit
van
Amsterdam

IAS technical report IAS-UVA-15-01

An Analysis of Piecewise-Linear and Convex Value Functions for Active Perception POMDPs

Yash Satsangi[1], **Shimon Whiteson**[1], **and Matthijs T. J. Spaan**[2]

[1]Intelligent Systems Laboratory Amsterdam, University of Amsterdam, The Netherlands
[2]Delft University of Technology, The Netherlands

In *active perception* tasks, an agent aims to select actions that reduce its uncertainty about a hidden state. While *partially observable Markov decision processes* (POMDPs) are a natural model for such problems, reward functions that directly penalize uncertainty in the agent's belief can remove the *piecewise-linear and convex* (PWLC) property of the value function required by most POMDP planners. This paper analyses ρPOMDP and POMDP-IR, two frameworks that restore the PWLC property in active perception tasks. We establish the mathematical equivalence of the two frameworks and show that both admit a decomposition of the maximization performed in the Bellman backup, yielding substantial computational savings. We also present an empirical analysis on data from real multi-camera tracking systems that illustrates these savings and analyzes the critical factors in the performance of POMDP planners in such tasks.

IAS
intelligent autonomous systems

Contents

Intelligent Autonomous Systems
Informatics Institute, Faculty of Science
University of Amsterdam
Science Park 904, 1098 XH Amsterdam
The Netherlands
Tel (fax): +31 20 525 7463
http://isla.science.uva.nl/

Corresponding author:
Yash Satsangi
tel: +31 20 525 8516
y.satsangi@uva.nl

1 Introduction

In an *active perception* task [19, 20], an agent must decide what actions to take to efficiently reduce its uncertainty about one or more hidden state variables. For example, a mobile robot armed with a camera must decide where to go to find a particular person or object. Similarly, an agent controlling a network of cameras with computational, bandwidth, or energy constraints must decide which subset of the cameras to use at each timestep.

A natural decision-theoretic model for active perception is the *partially observable Markov decision process* (POMDP) [3, 10, 18]. However, in a typical POMDP, reducing uncertainty about the state is only a means to an end. For example, a robot whose goal is to reach a particular location may take sensing actions that reduce its uncertainty about its current location because doing so helps it determine what future actions will bring it closer to its goal. By contrast, in active perception POMDPs, reducing uncertainty is an end in itself. For example, a surveillance system's goal is typically just to ascertain the state of its environment, not use that knowledge to achieve another goal. While perception is arguably always performed to aid decision-making, in an active perception problem that decision is made by another agent such as a human, that is not modeled as part of the POMDP. For example, a surveillance system may be tasked with detecting suspicious activity but only the human users of the system may decide how react to such activity.

One way to formulate uncertainty reduction as an end in itself is to define a reward function whose additive inverse is some measure of the agent's uncertainty about the hidden state, e.g., the entropy of its belief [6]. However, this leads to a reward function that conditions on the belief, rather than the state, and thus can remove the *piecewise-linear and convex* (PWLC) property of the value function [2], which is exploited by most POMDP planners. Recently, two approaches have been proposed to address this problem.

ρPOMDP [2] extends the POMDP formalism to allow belief-dependent rewards. A PWLC approximation is then formed by selecting a set of vectors tangent to this reward. With minor modifications, existing POMDP planners that rely on the PWLC property of the value function can then be employed. By contrast, *POMDP with information rewards* (POMDP-IR) [21] works within a standard POMDP but adds *prediction actions* that allow the agent to make predictions about the hidden state. A state-based reward function rewards the agent for accurate predictions. Since the reward function does not directly depend on the belief, the PWLC property is preserved and standard POMDP planners can be applied.

To the best of our knowledge, no previous research has examined the relationship between these two approaches to active perception, their respective pros and cons, or their efficacy in realistic tasks. In this paper, we address this gap by presenting a theoretical and empirical analysis of ρPOMDP and POMDP-IR. In particular, we make the following three contributions.

First, we establish the mathematical equivalence between ρPOMDP and POMDP-IR. Specifically, we show that any ρPOMDP can be translated into a POMDP-IR (and vice-versa) that preserves the value function for equivalent policies. Our main insight is that each tangent in ρPOMDP can be viewed as a vector describing the value of a prediction action in POMDP-IR.

Second, we observe that selecting prediction actions in a POMDP-IR does not require lookahead planning. Consequently, the maximization performed during backups can be decomposed and, although the addition of prediction actions causes a blowup in the agent's action space, the additional computational costs those actions introduce can be controlled. In addition, thanks to the equivalence between POMDP-IR and ρPOMDP that we establish, this decomposition holds also for ρPOMDP.

Third, we present an empirical analysis conducted on multiple active perception POMDPs learned from datasets gathered on real multi-camera tracking systems. Our results confirm the computational benefits of decomposing the maximization, measure the effects on performance of

the choice of prediction actions/tangents, and compare the costs and benefits of myopic versus non-myopic planning. Finally, we identify and study critical factors relevant to the performance and behaviour of agent in active perception tasks.

2 POMDPs

A POMDP is a tuple $\langle S, A, \Omega, T, O, R, b_0, h\rangle$ [10]. At each timestep, the environment is in a state $s \in S$, the agent takes an action $a \in A$ and receives a reward whose expected value is $R(s,a)$, and the system transitions to a new state $s' \in S$ according to the transition function $T(s,a,s') = Pr(s'|s,a)$. Then, the agent receives an observation $z \in \Omega$ according to the observation function $O(s',a,z) = Pr(z|s',a)$. The agent maintains a *belief* $b(s)$ about the state using Bayes rule:

$$b^{a,z}(s') = \frac{O(s',a,z)}{Pr(z|a,b)} \sum_{s\in S} T(s,a,s')b(s), \tag{1}$$

where $Pr(z|a,b) = \sum_{s,s''\in S} O(s'',a,z)T(s,a,s'')b(s)$ and $b^{a,z}(s')$ is the agent's belief about s' given that it took action a and observed z. The agent's initial belief is b_0.

A policy π specifies for each belief how the agent will act. A POMDP planner aims to find a policy π^* that maximizes the expected cumulative reward: $\pi^* = \max_\pi E[\sum_{t=0}^{h-1} r_t \mid a_t = \pi(b_t)]$, where h is a finite time horizon and r_t, a_t, and b_t are the reward, action, and belief at time t. Given $b(s)$ and $R(s,a)$, the *belief-based* reward, $\rho(b,a)$ is:

$$\rho(b,a) = \sum_s b(s)R(s,a). \tag{2}$$

The t-step *value function* of a policy π can be calculated recursively using the *Bellman equation*:

$$V_t^\pi(b) = \left[\rho(b,a_\pi) + \sum_{z\in\Omega} Pr(z|a_\pi,b)V_{t-1}^\pi(b^{a_\pi,z})\right], \tag{3}$$

where $a_\pi = \pi(b)$. The *optimal value function* $V_t^*(b)$ can be computed recursively as:

$$V_t^*(b) = \max_a \left[\rho(b,a) + \sum_{z\in\Omega} Pr(z|a,b)V_{t-1}^*(b^{a,z})\right]. \tag{4}$$

An important consequence of these equations is that the value function is *piecewise-linear and convex* (PWLC), a property exploited by most POMDP planners. Sondik [18] showed that a PWLC value function at any finite horizon t can be expressed as a set of vectors: $\Gamma_t = \{\alpha_0, \alpha_1, \ldots, \alpha_m\}$. Each α_i represents an $|S|$-dimensional hyperplane defining the value function over a bounded region of belief space. The value of a given belief point can be computed from the vectors as:

$$V_t^*(b) = \max_{\alpha_i\in\Gamma_t} \sum_s b(s)\alpha_i(s) \tag{5}$$

Exact POMDP solvers compute the value function for all possible belief points by computing the optimal Γ_t using the following recursive algorithm. For each action a and observation z, an intermediate $\Gamma_t^{a,z}$ is computed from Γ_{t-1}:

$$\Gamma_t^{a,z} = \{\alpha_i^{a,z} : \alpha_i \in \Gamma_{t-1}\}, \tag{6}$$

where, for all $s \in S$,

$$\alpha_i^{a,z}(s) = \sum_{s' \in S} T(s, a, s') O(s', a, z) \alpha_i(s'). \tag{7}$$

The next step is to take a cross-sum[1] over $\Gamma_t^{a,z}$ sets.

$$\Gamma_t^a = R(s, a) \oplus \Gamma_t^{a,z_1} \oplus \Gamma_t^{a,z_2} \oplus \dots \tag{8}$$

Then, we take the union of all the Γ_t^a-sets and prune any dominated α-vectors:

$$\mathbf{\Gamma}_t = \texttt{prune}(\cup_{a \in A} \Gamma_t^a). \tag{9}$$

For each α_i in the set, `prune` solves a linear program to determine whether it is dominated, i.e., whether for all b there exists an $\alpha_j \neq \alpha_i$ such that $\sum_s b(s)\alpha_j(s) \geq \sum_s b(s)\alpha_i(s)$.

Point-based planners [14, 17, 22] avoid the expense of solving for all belief points by computing Γ_t only for a set of sampled beliefs B. At each iteration, $\Gamma_t^{a,z}$ is generated from Γ_{t-1} for each a and z just as in (6) and (7). However, Γ_t^a is computed only for the sampled beliefs, i.e., $\Gamma_t^a = \{\alpha_b^a : b \in B\}$, where:

$$\alpha_b^a(s) = R(s, a) + \sum_{z \in \Omega} \arg\max_{\alpha \in \Gamma_t^{a,z}} \sum_s b(s)\alpha(s). \tag{10}$$

Finally, the best α-vector for each $b \in B$ is selected:

$$\begin{aligned} \alpha_b(s) &= \arg\max_{\alpha_b^a} \sum_s b(s)\alpha_b^a(s) \\ \Gamma_t &= \cup_{b \in B}\, \alpha_b. \end{aligned} \tag{11}$$

3 Active Perception POMDPs

The goal in an active perception POMDP is to reduce uncertainty about an *object of interest* that is not directly observable. In general, the object of interest may be only part of the state, e.g., if a surveillance system cares only about people's positions, not their velocities, or higher-level features derived from the state, e.g., that same surveillance system may care only how many people are in a given room. However, for simplicity, we focus on the case where the object of interest is simply the state s of the POMDP. Furthermore, we focus on pure active perception tasks in which the agent's only goal is to reduce uncertainty about the state, as opposed to hybrid tasks where the agent may also have other goals. However, extending our results to such hybrid tasks is straightforward.

A challenge in these settings is properly formalizing the reward function. Because the goal is to reduce uncertainty, reward is a direct function of the belief, not the state, i.e., the agent has no preference for one state over another, so long as it knows what that state is. Hence, there is no meaningful way to define a state-based reward function $R(s, a)$. Directly defining $\rho(b, a)$ using, e.g., negative *belief entropy*: $-H_b(s) = \sum_s b(s) \log(b(s))$, creates other problems, since $\rho(b, a)$ is no longer a convex combination of a state-based reward function, it is no longer guaranteed to be PWLC, a property both exact and point-based POMDP solvers rely on. In the following subsections, we describe two recently proposed frameworks designed to address this problem.

[1] The cross-sum of two sets A and B contains all values resulting from summing one element from each set:

$$A \oplus B = \{\mathbf{a} + \mathbf{b} : \mathbf{a} \in A \wedge \mathbf{b} \in B\}.$$

3.1 ρPOMDPs

Araya-López et al. [2] proposed the ρPOMDP framework for active perception tasks. A ρPOMDP, defined by the tuple $\langle S, A, T, \Omega, O, \Gamma_\rho, b_0, h\rangle$, is a normal POMDP except that the state-based reward function $R(s, a)$ has been omitted and the belief-based reward defined in the form of a set of vectors Γ_ρ has been added,

$$\rho(b) = \max_{\alpha_\rho \in \Gamma_\rho} \sum_s b(s)\alpha_\rho(s). \tag{12}$$

Since we consider only pure active perception tasks, ρ depends only on b, not on a and can thus be written $\rho(b)$.

Restricting ρ to be a Γ_ρ-set ensures that it is PWLC. If the "true" reward function is a non-PWLC function like negative belief entropy, then it can be approximated by defining Γ_ρ to be a set of vectors that are tangent to the true reward function. Figure 1 illustrates approximating negative belief entropy with different numbers of tangents.

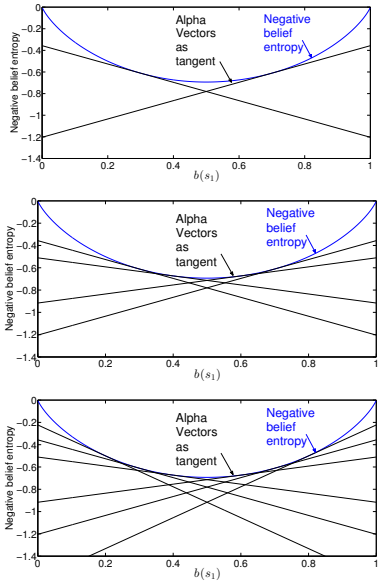


Figure 1: Defining Γ_ρ with different sets of tangents to the negative belief entropy curve in a 2-state POMDP.

Exactly solving a ρPOMDP requires a minor change to existing algorithms. In particular, since ρ now consists of a set of vectors for each a, as opposed to a single vector as in a standard POMDP, an additional cross-sum is required to compute Γ_t^a: $\Gamma_t^a = \Gamma_\rho \oplus \Gamma_t^{a,z_1} \oplus \Gamma_t^{a,z_2} \oplus \ldots$

Araya-López et al. [2] showed that the error in the value function computed by this approach, relative to the true reward function, whose tangents were used to define Γ_ρ, is bounded. However, their algorithm increases the computational complexity of solving the POMDP because it requires $|\Gamma_\rho|$ more cross-sums at each iteration in order to generate the Γ_t^a set.

3.2 POMDPs with Information Rewards

Spaan et al. [21] proposed *POMDPs with information rewards* (POMDP-IR) [21], an alternative framework for modeling active perception tasks that relies only on a standard POMDP. Instead of directly rewarding low uncertainty in the belief, the agent is given the chance to make predictions about the hidden state and rewarded, via a standard state-based reward function, for making accurate predictions. Formally, a POMDP-IR is a POMDP in which each action $a \in A$ is a tuple $\langle a_n, a_p\rangle$ where $a_n \in A_n$ is a *normal action*, e.g., moving a robot or turning on a camera, and $a_p \in A_p$ is a *prediction action*, which expresses predictions about the state. The joint action space is thus the Cartesian product of A_n and A_p, i.e., $A = A_n \times A_p$.

Prediction actions have no effect on states or observations but can trigger rewards via the standard state-based reward function $R(s, a)$. While there are many ways to define A_p and R, a simple approach is to create one prediction action for each state, i.e., $A_p = S$, and give the agent positive reward if and only if it correctly predicts the true state:

$$R(s, \langle a_n, a_p\rangle) = \begin{cases} 1, & \text{if } s = a_p \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

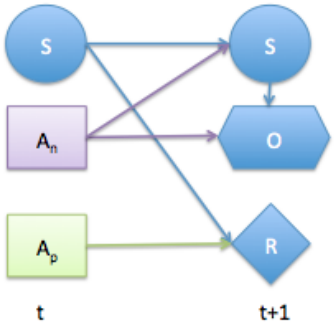


Figure 2: Influence diagram for POMDP-IR.

Thus, POMDP-IR indirectly rewards beliefs with low uncertainty, since these enable more accurate predictions and thus more expected reward. Furthermore, since a state-based reward function is explicitly defined, ρ can be defined as a convex combination of R, as in (2), guaranteeing that the value function is PWLC, as in a regular POMDP. Thus, a POMDP-IR can be solved with standard POMDP planners. However, the introduction of prediction actions leads to a blowup in the size of the joint action space $|A| = |A_n||A_p|$ of POMDP-IR.

Note that, though not made explicit in [21], several independence properties are inherent to the POMDP-IR framework, as shown in Figure 2. In particular, because we focus on pure active perception, the reward function R is independent of normal actions. Furthermore, state transitions and observations are independent of prediction actions. In the rest of this paper, we employ terminology for the reward, transition, and observation functions in a POMDP-IR that reflect this independence, i.e., we write $R(s, a_p)$, $T(s, a_n, s')$, and $O(s', a_n, z)$. In addition, we show in Section 5 how to exploit this independence to speed up planning.

4 ρPOMDP & POMDP-IR Equivalence

In this section we show the relationship between these two frameworks by proving mathematical equivalence of ρPOMDP and POMDP-IR. In particular, we show that solving a ρPOMDP is equivalent to solving a translated POMDP-IR and vice-versa. We show this equivalence by starting with a ρPOMDP and then translating it to a POMDP-IR. We then show that the value function, V_t^π for ρPOMDP we started with and the translated POMDP-IR are same. To complete our proof, we repeat the same process by starting with a POMDP-IR and then translating it to a ρPOMDP. We show that the value function V_t^π for the POMDP-IR and the corresponding ρPOMDP are same.

Definition 1. *Given a ρPOMDP $\mathbf{M}_\rho = \langle S, A_\rho, \Omega, T_\rho, O_\rho, \Gamma_\rho, b_0, h\rangle$ the* TRANSLATE-POMDP-ρ-IR*$(\mathbf{M}_\rho)$ produces a POMDP-IR $\mathbf{M}_{IR} = \langle S, A_{\text{IR}}, \Omega, T_{\text{IR}}, O_{\text{IR}}, R_{\text{IR}}, b_0, h\rangle$ via the following procedure.*

- *The set of states, set of observations, initial belief and horizon remain unchanged.*
- *The set of normal actions in $\mathbf{M}_{\text{IR}}$ is equal to the set of actions in $\mathbf{M}_\rho$, i.e., $A_{n,\text{IR}} = A_\rho$;*
- *The set of prediction actions $A_{p,IR}$ in $\mathbf{M}_{\text{IR}}$ contains one prediction action for each $\alpha_\rho(s) \in \Gamma_\rho$.*
- *The transition and observation functions in $\mathbf{M}_{\text{IR}}$ behave the same as in $\mathbf{M}_\rho$ for each a_n and ignore the a_p, i.e., for all $a_n \in A_{n,IR}$: $T_{\text{IR}}(s, a_n, s') = T_\rho(s, a, s')$ and $O_{\text{IR}}(s', a_n, z) = O_\rho(s', a, z)$, where $a \in A_\rho$ corresponds to a_n.*

- *The reward function in* $\mathbf{M}_{\text{IR}}$ *is defined such that* $R_{\text{IR}}(s, a_p) = \alpha_\rho(s)$*, where* α_ρ *is the* α*-vector corresponding to* a_p*.*

For example, consider a ρPOMDP with 2 states, if ρ is defined using tangents to belief entropy at $b(s_1) = 0.3$ and $b(s_1) = 0.7$. When translated to a POMDP-IR, the resulting reward function gives a small negative reward for correct predictions and a larger one for incorrect predictions, with the magnitudes determined by the value of the tangents when $b(s_1) = 0$ and $b(s_1) = 1$:

$$R_{\text{IR}}(s, a_p) = \begin{cases} -0.35, & \text{if } s = a_p \\ -1.12, & \text{otherwise.} \end{cases} \tag{14}$$

Definition 2. *Given a policy* π_ρ *for a* ρ*POMDP,* $\mathbf{M}_\rho$*, the* TRANSLATE-POLICY-ρ-IR*(*π_ρ*) procedure produces a policy* π_{IR} *for a POMDP-IR as follows. For all* b*,*

$$\pi_{\text{IR}}(b) = \langle \pi_\rho(b), \arg\max_{a_p} \sum_s b(s) R(s, a_p) \rangle. \tag{15}$$

That is, π_{IR} *selects the same normal action as* π_ρ *and the prediction action that maximizes expected immediate reward.*

Using these definitions, we prove that solving $\mathbf{M}_\rho$ is the same as solving $\mathbf{M}_{IR}$.

Theorem 1. *Let* $\mathbf{M}_\rho$ *be a* ρ*POMDP and* π_ρ *an arbitrary policy for* $\mathbf{M}_\rho$*. Furthermore let* $\mathbf{M}_{\text{IR}}$ = TRANSLATE-POMDP-ρ-IR($\mathbf{M}_\rho$) *and* π_{IR} = TRANSLATE-POLICY-ρ-IR(π_ρ)*. Then, for all* b*,*

$$V_t^{\text{IR}}(b) = V_t^\rho(b), \tag{16}$$

where V_t^{IR} *is the* t*-step value function for* π_{IR} *and* V_t^ρ *is the* t*-step value function for* π_ρ*.*

Proof. By induction on t. To prove the base case, we observe that, from the definition of $\rho(b)$,

$$V_0^\rho(b) = \rho(b) = \max_{\alpha_\rho \in \Gamma_\rho} \sum_s b(s) \alpha_\rho(s). \tag{17}$$

Since $\mathbf{M}_{\text{IR}}$ has a prediction action corresponding to each α_ρ, thus the a_p corresponding to $\alpha = \arg\max_{\alpha_\rho \in \Gamma_\rho} \sum_s b(s)\alpha_\rho(s)$, must also maximize $\sum_s b(s) R(s, a_p)$. Then,

$$\begin{aligned} V_0^\rho(b) &= \max_{a_p} \sum_s b(s) R_{\text{IR}}(s, a_p) \\ &= V_0^{\text{IR}}(b). \end{aligned} \tag{18}$$

For the inductive step, we assume that $V_{t-1}^{\text{IR}}(b) = V_{t-1}^\rho(b)$ and must show that $V_t^{\text{IR}}(b) = V_t^\rho(b)$. Starting with $V_t^{\text{IR}}(b)$,

$$V_t^{\text{IR}}(b) = \max_{a_p} \sum_s b(s) R(s, a_p) + \sum_z \Pr(z|b, \pi_{\text{IR}}^n(b)) V_{t-1}^{\text{IR}}(b^{\pi_{\text{IR}}^n(b), z}), \tag{19}$$

where $\pi_{\text{IR}}^n(b)$ denotes the normal action of the tuple specified by $\pi_{\text{IR}}(b)$ and:

$$\Pr(z|b, \pi_{\text{IR}}^n(b)) = \sum_s \sum_{s''} O_{\text{IR}}(s'', \pi_{\text{IR}}^n(b), z) T_{\text{IR}}(s, \pi_{\text{IR}}^n(b), s'') b(s).$$

Using the translation procedure, we can replace T_{IR} and O_{IR} and $\pi_{\text{IR}}^n(b)$ with their ρPOMDP counterparts on right hand side of the above equation:

$$\begin{aligned} \Pr(z|b, \pi_{\text{IR}}^n(b)) &= \sum_s \sum_{s''} O_\rho(s'', \pi_\rho(b), z) T_\rho(s, \pi_\rho(b), s'') b(s) \\ &= \Pr(z|b, \pi_\rho(b)). \end{aligned} \tag{20}$$

Similary, for the belief update equation,

$$\begin{aligned} b^{\pi_{\text{IR}}^n(b),z} &= \frac{O_{\text{IR}}(s', \pi_{\text{IR}}^n(b), z)}{\Pr(z|\pi_{\text{IR}}^n(b), b)} \sum_s b(s) T_{\text{IR}}(s, \pi_{\text{IR}}^n(b), s') \\ &= \frac{O_\rho(s', \pi_\rho(b), z)}{\Pr(z|\pi_\rho(b), b)} \sum_s b(s) T_\rho(s, \pi_\rho(b), s') \\ &= b^{\pi_\rho(b),z}. \end{aligned} \tag{21}$$

Substituting the above result in (19) yields:

$$V_t^{\text{IR}}(b) = \max_{a_p} \sum_s b(s) R(s, a_p) + \sum_z Pr(z|b, \pi_\rho(b)) V_{t-1}^{\text{IR}}(b^{\pi_\rho(b),z}). \tag{22}$$

Since the inductive assumption tells us that $V_{t-1}^{\text{IR}}(b) = V_{t-1}^\rho(b)$ and (18) shows that $\rho(b) = \max_{a_p} \sum_s b(s) R(s, a_p)$:

$$\begin{aligned} V_t^{\text{IR}}(b) &= [\rho(b) + \sum_z Pr(z|b, \pi_\rho(b)) V_{t-1}^\rho(b^{\pi_\rho(b),z})] \\ &= V_t^\rho(b). \end{aligned} \tag{23}$$

□

Definition 3. *Given a POMDP-IR* $\mathbf{M}_{IR} = \langle S, A_{\text{IR}}, \Omega, T_{\text{IR}}, O_{\text{IR}}, R_{\text{IR}}, b_0, h \rangle$ *the* TRANSLATE-POMDP-IR-ρ*(*$\mathbf{M}_{IR}$*) produces a* ρ*POMDP* $\mathbf{M}_\rho = \langle S, A_\rho, \Omega, T_\rho, O_\rho, \Gamma_\rho, b_0, h \rangle$ *via the following procedure.*

- *The set of states, set of observations, initial belief and horizon remain unchanged.*
- *The set of actions in* $\mathbf{M}_\rho$ *is equal to the set of normal actions in* $\mathbf{M}_{\text{IR}}$*, i.e.,* $A_\rho = A_{n,\text{IR}}$*.*
- *The transition and observation functions in* $\mathbf{M}_\rho$ *behave the same as in* $\mathbf{M}_{IR}$ *for each* a_n *and ignore the* a_p*, i.e., for all* $a \in A_\rho$*:* $T_\rho(s, a, s') = T_{\text{IR}}(s, a_n, s')$ *and* $O_\rho(s', a, z) = O_{\text{IR}}(s', a_n, z)$ *where* $a_n \in A_{n,IR}$ *is the action corresponding to* $a \in A_\rho$*.*
- *The* Γ_ρ *in* $\mathbf{M}_\rho$ *is defined such that, for each prediction action in* $A_{p,IR}$*, there is a corresponding* α *vector in* Γ_ρ*, i.e.,* $\Gamma_\rho = \{\alpha_\rho(s) : \alpha_\rho(s) = R(s, a_p)$ *for each* $a_p \in A_{p,IR}\}$*. Consequently, by definition,* ρ *is defined as:* $\rho(b) = \max_{\alpha_\rho} [\sum_s b(s) \alpha_\rho(s)]$*.*

Definition 4. *Given a policy* $\pi_{IR} = \langle a_n, a_p \rangle$*. for a POMDP-IR,* $\mathbf{M}_{IR}$*, the* TRANSLATE-POLICY-IR-ρ*(*π_{IR}*) procedure produces a policy* π_ρ *for a POMDP-IR as follows. For all b,*

$$\pi_\rho(b) = \pi_{IR}^n(b), \tag{24}$$

Theorem 2. *Let* $\mathbf{M}_{IR}$ *be a POMDP-IR and* $\pi_{IR} = \langle a_n, a_p \rangle$ *an policy for* $\mathbf{M}_{IR}$*, such that* $a_p = \max_{a'_p} b(s) R(s, a'_p)$*. Furthermore let* $\mathbf{M}_\rho =$ TRANSLATE-POMDP-IR-ρ*(*$\mathbf{M}_{IR}$*) and* $\pi_\rho =$ TRANSLATE-POLICY-IR-ρ*(*π_{IR}*). Then, for all b,*

$$V_t^\rho(b) = V_t^{IR}(b), \tag{25}$$

where V_t^{IR} *is the value of following* π_{IR} *in* $\mathbf{M}_{\text{IR}}$ *and* V_t^ρ *is the value of following* π_ρ *in* $\mathbf{M}_\rho$*.*

Proof. By induction on t. To prove the base case, we observe that, from the definition of $\rho(b)$,

$$\begin{aligned} V_0^{IR}(b) &= \max_{a_p} \sum_s b(s)R(s,a_p) \\ &= \sum_s b(s)\alpha(s) \text{ \{where } \alpha(s) \text{ is the } \alpha(s) \text{ corresponding to} \\ &\qquad\qquad a_p = \max_{a'_p} \textstyle\sum_s b(s)R(s,a'_p).\} \\ &= \rho(b) \\ &= V_0^{\rho}(b) \end{aligned} \tag{26}$$

For the inductive step, we assume that $V_{t-1}^{\rho}(b) = V_{t-1}^{IR}(b)$ and must show that $V_t^{\rho}(b) = V_t^{IR}(b)$. Starting with $V_t^{\rho}(b)$,

$$V_t^{\rho}(b) = \rho(b) + \sum_z Pr(z|b,\pi_\rho(b)) V_{t-1}^{\rho}(b^{\pi_\rho(b),z}), \tag{27}$$

where $\pi_{\text{IR}}^n(b)$ denotes the normal action of the tuple specified by $\pi_{\text{IR}}(b)$ and:

$$Pr(z|b,\pi_\rho(b)) = \sum_s \sum_{s''} O_\rho(s'',\pi_\rho(b),z) T_\rho(s,\pi_\rho(b),s'') b(s).$$

From the translation procedure, we can replace T_ρ and O_ρ and $\pi_\rho(b)$ with their POMDP-IR counterparts:

$$\begin{aligned} Pr(z|b,\pi_\rho(b)) &= \sum_s \sum_{s''} O_{IR}(s'',\pi_{IR}^n(b),z) T_{IR}(s,\pi_{IR}^n(b),s'') b(s) \\ &= Pr(z|b,\pi_{IR}(b)). \end{aligned} \tag{28}$$

Similarly, for the belief update equation,

$$\begin{aligned} b^{\pi_\rho(b),z} &= \frac{O_\rho(s',\pi_\rho(b),z)}{Pr(z|\pi_\rho(b),b)} \sum_s b(s) T_\rho(s,\pi_\rho(b),s') \\ &= \frac{O_{\text{IR}}(s',\pi_{\text{IR}}^n(b),z)}{Pr(z|\pi_{\text{IR}}^n(b),b)} \sum_s b(s) T_{\text{IR}}(s,\pi_{\text{IR}}^n(b),s') \\ &= b^{\pi_{IR}(b),z}. \end{aligned} \tag{29}$$

Substituting the above result in (27) yields:

$$V_t^{\rho}(b) = \rho(b) + \sum_z Pr(z|b,\pi_{IR}(b)) V_{t-1}^{\text{IR}}(b^{\pi_{IR}(b),z}). \tag{30}$$

Since the inductive assumption tells us that $V_{t-1}^{\rho}(b) = V_{t-1}^{\text{IR}}(b)$ and (26) shows that $\rho(b) = \max_{a_p} \sum_s b(s)R(s,a_p)$,

$$\begin{aligned} V_t^{\rho}(b) &= [\max_{a_p} \sum_s b(s)R(s,a_p) + \sum_z Pr(z|b,\pi_{IR}(b)) V_{t-1}^{IR}(b^{\pi_{IR}(b),z})] \\ &= V_t^{IR}(b). \end{aligned} \tag{31}$$

□

The main implication of the theorem 1 and 2 is that any result that holds for either ρPOMDP or POMDP-IR also holds for the other framework. For example, the results presented in theorem 4.3 in Araya-López et al. [2] that bound the error in the value function of ρPOMDP also hold for POMDP-IRs. Thus, there is no significant difference between the two frameworks and both can be used with equal efficacy to model active perception.

5 Decomposed Maximization

As mentioned in Section 3.2, the addition of prediction actions leads to a blowup in the size of the joint action space. However, the transition and observation functions are independent of these prediction actions. Furthermore, reward is independent of normal actions. A consequence of these independence properties is that the maximization over actions performed in (4) can, in a POMDP-IR, be decomposed into two simpler maximizations, one over prediction actions and one over normal actions:

$$V_t^*(b) = \max_{a_p} \sum_s b(s)R(s,a_p) \; + \max_{a_n} \sum_z Pr(z|a_n,b)V_{t-1}^*(b^{a_n,z}). \tag{32}$$

In other words, maximization of immediate reward need only consider prediction actions and maxmization over future reward need only consider normal actions. Note that, in the special case where the POMDP-IR reward function is defined as in (13), the first term in (32) is simply the max of the belief, $\max_s b(s)$.

In this section, we show how to exploit this decomposition in exact and point-based methods.

5.1 Exact Methods

Exact methods cannot directly exploit the decomposition as they do not perform an explicit maximization. However, they can be made faster by separating the pruning steps that they employ. First, we generate a set of vectors just for the prediction actions: $\Gamma^R = \{\alpha^{a_p} : a_p \in A_p\}$, where for all $s \in S$, $\alpha^{a_p}(s) = R(s,a_p)$. Then, we generate another set of vectors for the normal actions, as in a standard exact solver:

$$\begin{aligned} \Gamma_t^{a_n,z} &= \{\alpha_i^{a_n,z}(s) : \alpha_i \in \Gamma_{t-1}\}, \\ \alpha_i^{a_n,z}(s) &= \sum_{s' \in S} T(s,a_n,s')O(s',a_n,z)\alpha_i(s'), \\ \Gamma_t^{a_n} &= \Gamma^{a_n,z_1} \oplus \Gamma^{a_n,z_2} \ldots \end{aligned} \tag{33}$$

Finally, we can compute Γ_t as follows:

$$\mathbf{\Gamma}_t = \texttt{prune}(\texttt{prune}(\Gamma^R) \oplus \texttt{prune}(\cup_{a_n \in A_p} \Gamma_t^{a_n})). \tag{34}$$

This is essentially a special case of *incremental pruning* [5], made possible by the special structure of the POMDP-IR. The independence properties enable the normal and prediction actions to be treated separately. This, in turn allows us to prune Γ^R and $\Gamma_t^{a_n}$ separately resulting in faster pruning (because of smaller size) and hence faster computation of the final $\mathbf{\Gamma}$-set.

5.2 Point-Based Methods

Point-based methods do explicitly maximize for sampled beliefs in B. Thus, we can construct a point-based method that exploits this decomposed maximization to solve POMDP-IRs more efficiently.

Having computed Γ^R and $\Gamma_t^{a_n,z}$ as above, we can compute each element of $\Gamma_t^{a_n} = \{\alpha_b^{a_n} : b \in B\}$ using decomposed maximization. For all $s \in S$,

$$\alpha_b^{a_n}(s) = \arg\max_{\alpha \in \Gamma^R} \sum_s b(s)\alpha(s) + \sum_z \arg\max_{\alpha \in \Gamma_t^{a_n,z}} \sum_s b(s)\alpha(s). \tag{35}$$

As before, we can then select the best α-vector for each $b \in B$, but now we only have to maximize across the $\alpha_b^{a_n}$'s:

$$\begin{aligned} \alpha_b(s) &= \arg\max_{\alpha_b^{a_n}} \left(\sum_s \alpha_b^{a_n}(s) b(s) \right) \\ \Gamma_t &= \cup_{b \in B} \alpha_b. \end{aligned} \tag{36}$$

By decomposing the maximization, this approach avoids iterating over all $|A_n||A_p|$ joint actions. At each timestep t, this approach generates $|A_n||\Omega||\Gamma_{t-1}|+|A_p|$ backprojections and then prunes them to $|B|$ vectors, yielding a computational complexity of $O(|S||B|(|A_p| + |A_n||\Omega||\Gamma_{t-1}|))$. By contrast, a naive application of point-based methods in POMDP-IR has a complexity of $O(|S||B||A_p||A_n||\Omega||\Gamma_{t-1}|)$. Hence, the advantages of the POMDP-IR framework can be achieved without incurring significant additional computational costs due to the blowup in the size of the joint action space.

6 Experiments

In this section, we present the results of experiments designed to confirm the computational benefits of decomposing the maximization, measure the effects on performance of the choice of prediction actions/tangents, and compare the costs and benefits of myopic versus non-myopic planning. We consider the task of tracking people in a surveillance area with a multi-camera tracking system. The goal of the system is to select a subset of cameras, to correctly predict the position of people in the surveillance area, based on the observations received from the selected cameras.

We compare the performance of POMDP-IR with decomposed maximization to a naive POMDP-IR that does not decompose the maximization. Thanks to Theorem 1 and 2, these approaches have performance equivalent to their ρPOMDP counterparts. We also compare against two baselines. The first is a weak baseline we call the *rotate policy* in which the agent simply keeps switching between cameras on a turn-by-turn basis. The second is a stronger baseline we call the *coverage policy*, which was developed in earlier work on active perception [19, 20]. As in POMDP-IR, cameras are selected according to a policy computed by a POMDP planner. However, instead of using prediction actions, the state-based reward function simply rewards the agent for observing the person, i.e., the agent is encouraged to select the cameras that are most likely to generate positive observations.

6.1 Simulated Setting

We start with experiments conducted in a simulated setting, first considering the task of tracking a single person with a multi-camera system and then considering the more challenging task of tracking multiple people.

6.1.1 Single-Person Tracking

We start by considering the task of tracking one person walking in a grid-world composed of $|S|$ cells and N cameras. At each timestep, the agent can select only K cameras, where $K \leq N$. Each selected camera generates a noisy observation of the person's state. The agent's goal is to minimize its uncertainty about the person's state. In the experiments in this section, we fixed $K = 1$ and $N = 10$.

We model this task as a POMDP with one state for each grid cell. A normal action is a vector of N binary *action features* indicating whether the given camera is selected. Unless

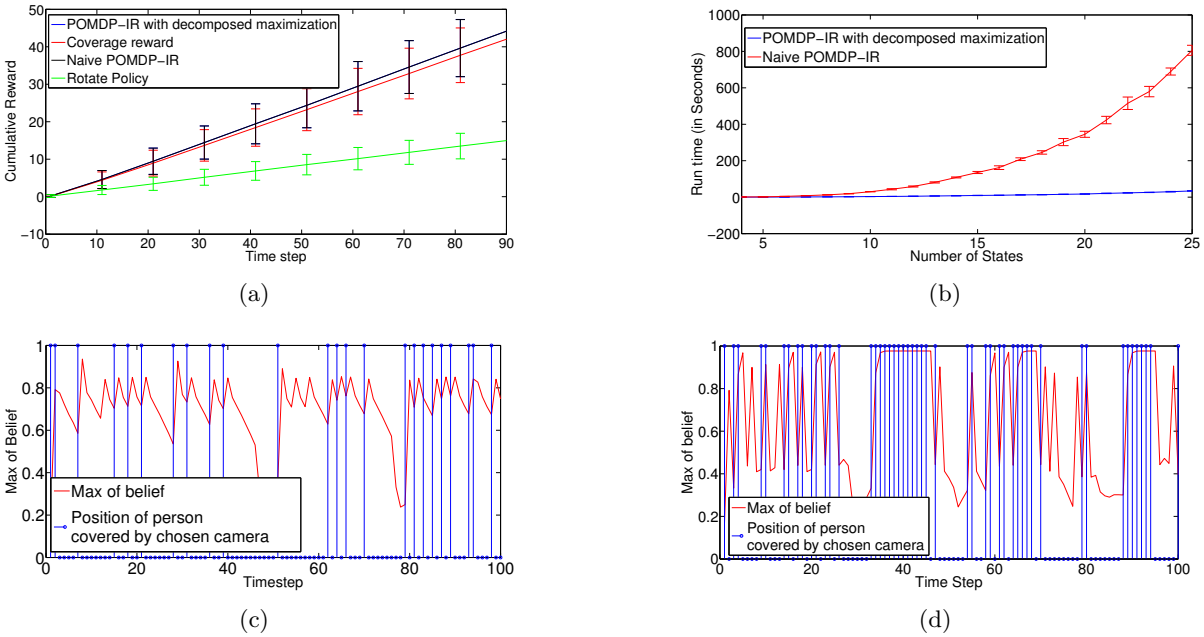


Figure 3: (a) Performance comparison between POMDP-IR with decomposed maximization, naive POMDP-IR, coverage policy, and rotate policy; (b) Runtime comparison between POMDP-IR with decomposed maximization and naive POMDP-IR; (c) Behaviour of POMDP-IR policy; (d) Behaviour of coverage policy.

stated otherwise, there is one prediction action for each state and the agent gets a reward of +1 if it correctly predicts the state and 0 otherwise. An observation is a vector of N *observation features*, each of which specifies the person's position as estimated by the given camera. If a camera is not selected, then the corresponding observation feature has a value of null. The transition function $T(s, s') = Pr(s'|s)$ is independent of actions as the agent's role is purely observational. It specifies a uniform probability of staying in the same cell or transitioning to a neighboring cell.

To compare the performance of POMDP-IR to the baselines, 100 trajectories were simulated from the POMDP. The agent was asked to guess the person's position at each time step. Figure 3(a) shows the cumulative reward collected by all four methods. As expected, POMDP-IR with decomposed maximization and naive POMDP-IR perform identically. However, Figure 3(b), which compares the runtimes of POMDP-IR with decomposed maximization and naive POMDP-IR, shows that decomposed maximization yields a large computational savings.

Figure 3(a) also shows that POMDP-IR greatly outperforms the rotate policy and modestly outperforms the coverage policy. Figures 3(c) and 3(d) illustrate the qualitative difference between POMDP-IR and the coverage policy. The blue lines mark the point when the agent chose to observe the cell occupied by the person and the red lines plot the max of the agent's belief. The main difference between the two policies is that once POMDP-IR gets a good estimate of the state, it proactively observes neighboring cells to which the person might transition. This helps it to more quickly find the person when she moves. By contrast, the coverage policy always looks at the cell where it believes the person to be. Hence, it takes longer to find her again when she moves. This is evidenced by the fluctuations in the max of the belief, often drops below 0.5 for the coverage policy, while it rarely does so for POMDP-IR.

Next, we examine the effect of approximating a true reward function like belief entropy with more and more tangents. Figure 1 illustrates how adding more tangents can better approximate negative belief entropy. To test the effects of this, we measured the cumulative negative belief

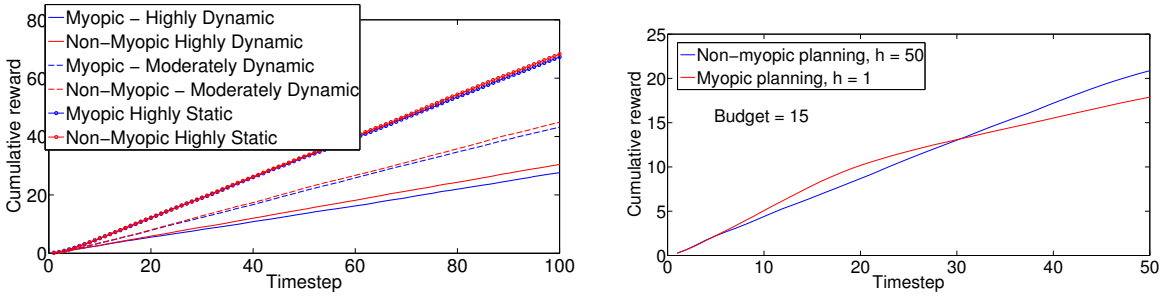


Figure 5: (a) Performance comparison for myopic vs. non myopic policies; (b) Performance comparison for myopic vs non myopic policies in budget-based setting.

entropy when using between one and four tangents per state. Figure 4 shows the results and demonstrates that, as more tangents are added, the performance in terms of the true reward function improves. However, performance also quickly saturates, as four tangents perform no better than three.

Next, we compare the performance of POMDP-IR to a myopic variant that seeks only to maximize immediate reward, i.e., $h = 1$. We perform this comparison in three variants of the task. In the *highly static* variant, the state changes very slowly: the probability of staying is the same state is 0.9. In the *moderately dynamic* variant, the state changes more frequently, with a same-state transition probability of 0.7. In the *highly dynamic* variant, the state changes rapidly (with a same-state transition probability of 0.5). Figure 5(a) shows the results of these comparisons. In each setting, non-myopic POMDP-IR outperforms myopic POMDP-IR. In the highly static variant, the difference is marginal. However, as the task becomes more dynamic, the importance of look-ahead planning grows. Because the myopic planner focuses only on immediate reward, it ignores what might happen to its belief when the state changes, which happens more often in dynamic settings.

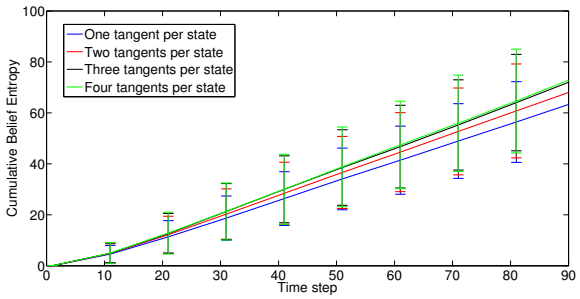
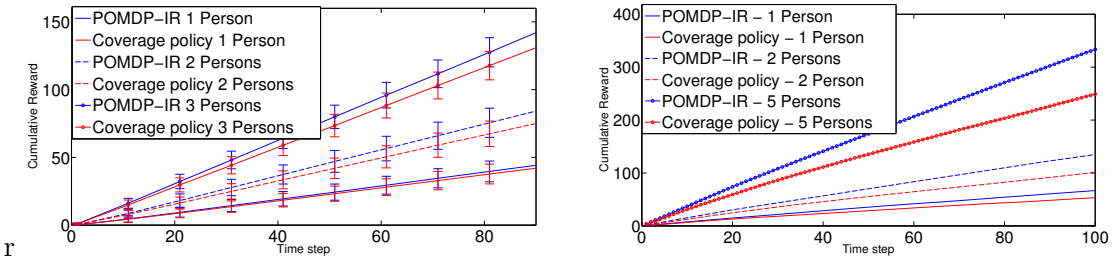


Figure 4: Performance comparison as negative belief entropy is better approximated.

Finally, we compare the performance of myopic and non-myopic planning in a *budget-constrained* environment. This specifically corresponds to energy-constrained environment, where it is not possible to keep the cameras on for all the time but the camera can be employed only a few times over the entire trajectory. This is augmented with resource-constraints, so that the agent has to plan not only when to use the camera, but also decide which camera to select. Specifically, the agent can only employ the camera a total of 15 times across all 50 timesteps. On the other timesteps, it must select an action that generates only a null observation. Figure 5(b) shows that non-myopic planning is of critical importance in this setting. Whereas myopic planning greedily consumes the budget as quickly as possible, non-myopic planning saves the budget for situations in which it is highly uncertain about the state.

6.1.2 Multi-Person Tracking

To extend our analysis to a more challenging problem, we consider a simulated setting in which multiple people must be tracked simultaneously. Since $|S|$ grows exponentially in the number



r

Figure 6: (a) Multi-person tracking performance for POMDP-IR and coverage policy. (b)Performance of POMDP-IR and coverage policy when only important cells must be tracked.

of people, the resulting POMDP quickly becomes intractable. Therefore, we compute instead a factored value function $V_t(b) = \sum V_t^i(b^i)$ where $V_t^i(b^i)$ is the value of the agent's current belief b^i about the i-th person. Thus, $V_t^i(b^i)$ needs to be computed only once, by solving a POMDP of the same size as that in the single-person setting. During action selection, $V_t(b)$ is computed using the current b_i for each person. This kind of factorization corresponds to the assumption that each person's movement is independent of that of other people. Although violated in practice, such an assumption can nonetheless yield good approximations.

Figure 6 (a), which compares POMDP-IR to the coverage policy with one, two, and three people, shows that the advantage of POMDP-IR grows substantially as the number of people increases. Whereas POMDP-IR tries to maintain a good estimate of everyone's position, the coverage policy just tries to look at the cells where the maximum number of people might be present, ignoring other cells completely.

Finally, we compare POMDP-IR and the coverage policy in a setting in which the goal is only to reduce uncertainty about a set of "important cells" that are a subset of the whole state space. For POMDP-IR, we prune the set of prediction actions to allow predictions only about important cells. For the coverage policy, we reward the agent only for observing people in important cells. The results, shown in Figure 6 (b), demonstrate that the advantage of POMDP-IR over the coverage policy is even larger in this variant of the task. POMDP-IR makes use of information coming from cells that neighbor important cells (which is of critical importance if the important cells do not have good observability), while the coverage policy does not. As before, the difference gets larger as the number of people increases.

6.2 Hallway Dataset

To extend our analysis to a more realistic setting, we used a dataset collected by four stereo overhead cameras mounted in a hallway. Tracks were generated from the recorded images using a proprietary software package [1]. For each person recorded, one track is generated after processing observations coming from all four cameras. The dataset consists of a recording of 30 tracks, specifying the x-y position of a person through time.

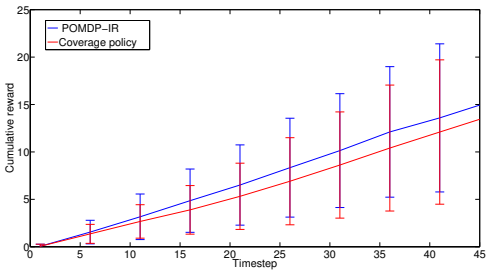


Figure 7: Performance of POMDP-IR and the coverage policy on the hallway dataset.

To learn a POMDP model from the dataset, we divided the continuous space into 32 cells ($|S| = 33$: 32 cells plus an external state indicating the person is no longer in the hallway). Using the data, we learned a maximum-

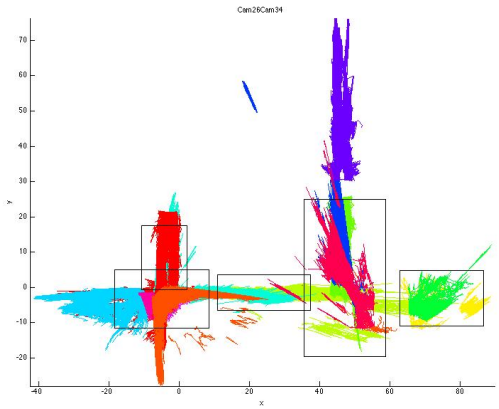


Figure 8: Sample tracks for all cameras; each color denotes all tracks observed by a given camera; boxes denote regions of high overlap between cameras.

likelihood tabular transition function. Since we do not have ground truth about people's locations, we introduced random noise into the observations. For each camera and each cell in that camera's region, the probability of a false positive and false negative were set by uniformly sampling a number from the interval $[0, 0.25]$. Figure 7 shows that POMDP-IR again substantially outperforms the coverage policy, for the same reasons mentioned before.

6.3 Shopping Mall Dataset

Finally, we extended our analysis to a real-life dataset collected in a shopping mall. This dataset was gathered over 4 hours using 13 CCTV cameras located in a shopping mall [4]. Each camera uses a FPDW [7] pedestrian detector to detect people in each camera image and in-camera tracking [4] to generate tracks of the detected people's movements over time. The dataset consists of 9915 tracks each specifying one person's x-y position over time. Figure 8 shows the sample tracks from all of the cameras.

To learn a POMDP model from the dataset, we divided the continuous space into 20 cells ($|S| = 21$: 20 cells plus an external state indicating the person has left the shopping mall). As before, we learned a maximum-likelihood tabular transition function. However, in this case, we were able to learn a more realistic observation. Because the cameras have many overlapping regions (see Figure 8), we were able to manually match tracks of the same person recorded individually by each camera. The "ground truth" was then constructed by taking a weighted mean of the matched tracks. Finally, this ground truth was used to estimate noise parameters for each cell (assuming zero-mean Gaussian noise), which was used as the observation function. Figure 9 shows that, as before, POMDP-IR substantially outperforms the coverage policy for various numbers of cameras. In addition to the reasons mentioned before, the high overlap between cameras contributes to POMDP-IR's superior performance. The coverage policy has difficulty ascertaining people's exact locations because it is rewarded only for observing them somewhere in a camera's large overlapping region, whereas POMDP-IR is rewarded for deducing their exact locations.

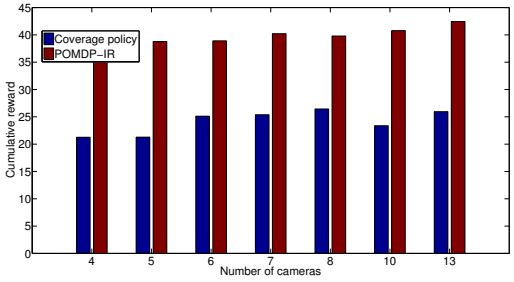


Figure 9: Performance of POMDP-IR and the coverage policy on the shopping mall dataset.

7 Conclusions & Future Work

This paper presented a detailed analysis of ρPOMDP and POMDP-IR, two frameworks for modeling active perception tasks while preserving the PWLC property of value functions. We established the mathematical equivalence of the two frameworks and showed that both admit a decomposition of the maximization performed in the Bellman optimality equation, yielding substantial computational savings. We also presented an empirical analysis on data from both simulated and real multi-camera tracking systems that illustrates these savings and analyzes the critical factors in the performance of POMDP planners in such tasks. In future work, we aim to develop richer POMDP models that can represent continuous state features and dynamic numbers of people to be tracked. In addition, we aim to consider hybrid tasks, perhaps modeled in a multi-objective way, in which active perception must be balanced with other goals.

References

[1] Eagle Vision. www.eaglevision.nl.

[2] Mauricio Araya-López, Olivier Buffet, Vincent Thomas, and François Charpillet. A POMDP extension with belief-dependent rewards. In *Advances in Neural Information Processing Systems*, pages 64–72, 2010.

[3] K. J. Astrom. Optimal control of Markov decision processes with incomplete state estimation. *Journal of Mathematical Analysis and Applications*, pages 174–205, 1965.

[4] Henri Bouma, Jan Baan, Sander Landsmeer, Chris Kruszynski, Gert van Antwerpen, and Judith Dijk. Real-time tracking and fast retrieval of persons in multiple surveillance cameras of a shopping mall. In *SPIE Defense, Security, and Sensing*, pages 87560A–87560A–13, 2013.

[5] Anthony Cassandra, Michael L. Littman, and Nevin L. Zhang. Incremental pruning: A simple, fast, exact method for partially observable Markov decision processes. In *In Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence*, pages 54–61, 1997.

[6] Thomas M Cover and Joy A Thomas. Entropy, relative entropy and mutual information. *Elements of Information Theory*, 1991.

[7] Piotr Dollár, Serge Belongie, and Pietro Perona. The fastest pedestrian detector in the west. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 2903–2910, 2010.

[8] Adam Eck and Leen-Kiat Soh. Evaluating POMDP rewards for active perception. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*, pages 1221–1222, 2012.

[9] Shihao Ji, R. Parr, and L. Carin. Nonmyopic multiaspect sensing with partially observable Markov decision processes. *Signal Processing, IEEE Transactions on*, pages 2720–2730, 2007.

[10] Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1-2):99–134, 1998.

[11] Chris Kreucher, Keith Kastella, and Alfred O. Hero, III. Sensor management using an active sensing approach. *Signal Processing*, 85(3):607–624, 2005.

[12] Vikram Krishnamurthy and Dejan V Djonin. Structured threshold policies for dynamic sensor scheduling: A partially observed markov decision process approach. *Signal Processing, IEEE Transactions on*, 55(10):4938–4957, 2007.

[13] Prabhu Natarajan, Trong Nghia Hoang, Kian Hsiang Low, and Mohan Kankanhalli. Decision-theoretic approach to maximizing observation of multiple targets in multi-camera surveillance. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*, pages 155–162, 2012.

[14] Joelle Pineau, Geoffrey J Gordon, and Sebastian Thrun. Anytime point-based approximations for large POMDPs. *Journal of Artificial Intelligence Research (JAIR)*, 27:335–380, 2006.

[15] Stephane Ross, Joelle Pineau, Sebastien Paquet, and Brahim Chaib-draa. Online planning algorithms for POMDPs. *Journal of Artificial Intelligence Research*, pages 663–704, 2008.

[16] Yash Satsangi, Shimon Whiteson, and Frans Oliehoek. Exploiting submodular value functions for faster dynamic sensor selection. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 3356–3363, January 2015.

[17] Guy Shani, Joelle Pineau, and Robert Kaplow. A survey of point-based POMDP solvers. *Autonomous Agents and Multi-Agent Systems*, 27(1):1–51, 2013.

[18] Edward J. Sondik. *The Optimal Control of Partially Observable Markov Processes.* PhD thesis, Stanford University, United States – California, 1971.

[19] Matthijs T. J. Spaan. Cooperative active perception using POMDPs. In *AAAI 2008 Workshop on Advancements in POMDP Solvers*, 2008.

[20] Matthijs T. J. Spaan and Pedro U. Lima. A decision-theoretic approach to dynamic sensor selection in camera networks. In *International Conference on Automated Planning and Scheduling*, pages 279–304, 2009.

[21] Matthijs T. J. Spaan, Tiago S. Veiga, and Pedro U. Lima. Decision-theoretic planning under uncertainty with information rewards for active cooperative perception. *Autonomous Agents and Multi-Agent Systems*, 29(6):1157–1185, 2015.

[22] Matthijs T. J. Spaan and Nikos Vlassis. Perseus: Randomized point-based value iteration for POMDPs. *Journal of Artificial Intelligence Research*, 24:195–220, 2005.

[23] J.L. Williams, J.W. Fisher, and A.S. Willsky. Approximate dynamic programming for communication-constrained sensor network management. *Signal Processing, IEEE Transactions on*, 55:4300–4311, 2007.

Acknowledgements

We thank Henri Bouma and TNO for providing us with the dataset used in our experiments. We also thank the STW User Committee for its advice regarding active perception for multi-camera tracking systems. This research is supported by the Dutch Technology Foundation STW (project #12622), which is part of the Netherlands Organisation for Scientific Research (NWO), and which is partly funded by the Ministry of Economic Affairs.

IAS reports

This report is in the series of IAS technical reports. The series editor is Bas Terwijn (B.Terwijn@uva.nl). Within this series the following titles appeared:

[Satsangi(2014)] Y.Satsangi, S. Whiteson and F.A.Oliehoek *Exploiting Submodular Value Functions for Dynamic Sensor Selection* Technical Report IAS-UVA-14-02, Informatics Institute, University of Amsterdam, The Netherlands, November 2014

[Oliehoek(2014)] F.A. Oliehoek and C. Amato *Dec-POMDPs as Non-Observable MDPs* Technical Report IAS-UVA-14-01, Informatics Institute, University of Amsterdam, The Netherlands, November 2014.

[Visser(2012)] A. Visser *UvA Rescue Technical Report: a description of the methods and algorithms implemented in the UvA Rescue code release* Technical Report IAS-UVA-12-02, Informatics Institute, University of Amsterdam, The Netherlands, September 2012.

[Visser(2012)] A. Visser *A survey of the architecture of the communication library LCM for the monitoring and control of autonomous mobile robots* Technical Report IAS-UVA-12-01, Informatics Institute, University of Amsterdam, The Netherlands, September 2012.

All IAS technical reports are available for download at the ISLA website:
http://isla.science.uva.nl/node/85