Auton Agent Multi-Agent Syst (2015) 29:1157–1185
DOI 10.1007/s10458-014-9279-8

Decision-theoretic planning under uncertainty with information rewards for active cooperative perception

Matthijs T. J. Spaan · Tiago S. Veiga · Pedro U. Lima

Published online: 23 December 2014

Abstract Partially observable Markov decision processes (POMDPs) provide a principled framework for modeling an agent's decision-making problem when the agent needs to consider noisy state estimates. POMDP policies take into account an action's influence on the environment as well as the potential information gain. This is a crucial feature for robotic agents which generally have to consider the effect of actions on sensing. However, building POMDP models which reward information gain directly is not straightforward, but is important in domains such as robot-assisted surveillance in which the value of information is hard to quantify. Common techniques for uncertainty reduction such as expected entropy minimization lead to non-standard POMDPs that are hard to solve. We present the POMDP with Information Rewards (POMDP-IR) modeling framework, which rewards an agent for reaching a certain level of belief regarding a state feature. By remaining in the standard POMDP setting we can exploit many known results as well as successful approximate algorithms. We demonstrate our ideas in a toy problem as well as in real robot-assisted surveillance, showcasing their use for active cooperative perception scenarios. Finally, our experiments show that the POMDP-IR framework compares favorably with a related approach on benchmark domains.

Keywords Active cooperative perception · Planning under uncertainty for robots · Partially observable Markov decision processes

1 Introduction

A system of networked mobile and static sensors can substantially improve situational awareness compared to a single sensor or a network of static sensors. However, the benefit of mobile

M. T. J. Spaan (✉)
Delft University of Technology, Delft, The Netherlands
e-mail: m.t.j.spaan@tudelft.nl

T. S. Veiga · P. U. Lima
Institute for Systems and Robotics, Instituto Superior Técnico, Universidade de Lisboa, Lisbon, Portugal

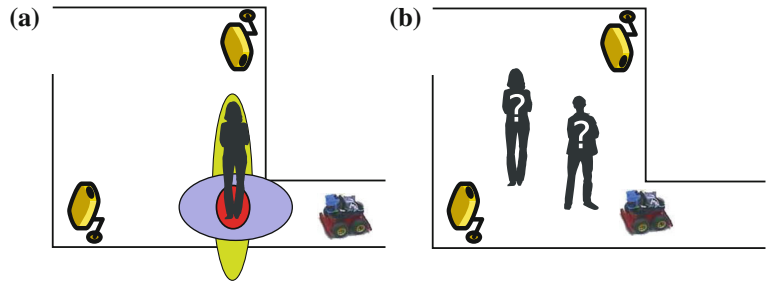

Fig. 1 Example of a robot cooperating with a camera network for **a** person localization and **b** identification

sensors is maximized if actions, such as positioning a mobile sensor, are carefully planned and executed [55]. In our work, we consider the problem of planning in networked robot systems (NRS), in which mobile robots carrying sensors interact with each other as well as with static sensors present in the environment to accomplish certain tasks [44] (as illustrated in Fig. 1). For instance, in a shopping mall, we can consider a NRS where cameras detect humans in need of help, but also detect a fire eruption or an abnormal activity which requires assistance. Robots might be used both to improve the confidence of event detection and to provide assistance for any of the above situations.

1.1 Problem description

We take a comprehensive approach to this problem, denoted here as *active cooperative perception* (ACP) [50]. In our context, *cooperative perception* refers to the fusion of sensory information between the fixed surveillance cameras and each robot, with the goal of maximizing the amount and quality of perceptual information available to the system. *Active* perception means that an agent considers the effects of its actions on its sensors, and in particular it tries to improve their performance. This can mean selecting sensory actions, for instance pointing a pan-and-tilt camera or choosing to execute a computationally expensive vision algorithm. Other effects might require reasoning about the future, such as adjusting a robot's path planning: given two routes to reach a goal location, take the more informative one, for instance. Combining the two concepts, active cooperative perception is the problem of active perception involving multiple sensors and potentially multiple cooperating decision makers.

There are many benefits of cooperation between sensors, in particular when some are mobile. An obvious advantage is that a mobile sensor can move to regions in which fixed sensors have no coverage. However, even when such coverage exists, it might not be sufficient, as illustrated in Fig. 1. First, while a person might be observed by a surveillance camera (Fig. 1a, where the uncertainty of the camera's measurements is indicated in green), additional sensor readings by the robot (blue) result in a more precise estimate of the person's location (red). Second, often not all relevant visual features required for person identification might be reliably detected by the fixed sensors (Fig. 1b), in which case the up-close and adjustable field of view of a mobile sensor can provide the required extra information.

1.2 A decision-theoretic approach

Our approach is based on decision-theoretic principles, in particular on discrete Partially Observable Markov Decision Processes (POMDPs) [23]. POMDPs form a general and pow-

erful mathematical basis for planning under uncertainty, and their use in mobile robotic applications has increased [13,14,55,57,62]. POMDPs provide a comprehensive approach for modeling the interaction of an active sensor with its environment. They model the uncertainty in action effects and state observations and express the (possibly multiple) goals of the system by rewards associated to (state, action) pairs. Based on prior knowledge of the stochastic sensor models and of the environment dynamics, we can compute policies that tell the active sensor how to act based on the observations it receives.

In active cooperative perception, the goal is typically to increase the available information by reducing the uncertainty regarding the state of environment. In a POMDP, if more information improves task performance, its policy will take information-gaining actions. For instance, a better self-localization estimate for a mobile sensor will improve the quality of fusing its information in the global frame with other sensors' information (Fig. 1a) or lowered detection uncertainty of an event reduces the risk of a false classification (Fig. 1b). However, optimal policies only include informative actions if those are beneficial to the task: if in the scenario of Fig. 1b the objective is to count the number of people in the room (instead of identifying them), the robot will not be dispatched to confirm their identity.

In a traditional POMDP model, state-based rewards allow for defining many tasks, but do not explicitly reward information gain. Directly rewarding information gain is often of interest in systems in which the POMDP's state estimate is used as input for a higher-level decision maker (which is not modeled as a POMDP), for instance a human supervisor whose preferences can be hard to quantify. One approach is consider non-standard POMDP models for this purpose, where the rewards are belief-based instead of state-based [27,29,56]. When rewards are defined for instance as the negative entropy of the belief state, the POMDP is non-standard and the optimal value function is no longer linear. The ρPOMDP framework [3,4] generalizes the POMDP framework and defines a reward function directly in terms of belief states and actions. It is based on the observation that, although many types of belief-based rewards are not PWLC and cannot be directly used in traditional POMDP solvers, if they are convex they can be approximated by PWLC functions. This makes it possible to extend traditional solvers with a bounded error, as long as the approximation of the reward function is bounded.

Considering finite and discrete models allows us to compute closed-loop non-myopic solutions. The difficulty of solving continuous-state POMDPs in closed form has obstructed their solution, leading for instance to open-loop feedback controllers [48], or requiring additional model assumptions [10,36]. MDP-based heuristic solutions will not work in ACP scenarios, as they do not reason about future belief states, which is crucial for ACP. For instance, one such heuristic, Q_{MDP} [28], which is popular in robotics, assumes that any uncertainty in the belief is resolved in a single time step. True POMDP methods, however, provide a principled approach to integrating value of information with other costs or rewards, optimizing task performance directly.

Decision theory also provides a good framework to model multi-objective problems [39]. The reward function can depend on different sets of state variables, which allows us to model different objectives in a single problem and assign each goal a different preference. Defining preferences in information-gain problems might be desirable, if the system designer gives more importance to information about some features in the environment than others, or if he needs more certainty about some features. In particular, we focus on scenarios in which the system has to balance regular task objectives with information objectives. In such cases, optimal policies have to trade off these multiple objectives. We assume that the multiple objectives can be scalarized by a linear combination of the reward functions associated with each objective.

1.3 Contributions

We present a POMDP-based framework, dubbed POMDP-IR (for POMDP with Information Rewards), which can serve for many ACP tasks in NRS. The core idea is that we can reward the agent for reaching a certain level of belief regarding a state feature, at the cost of extending the action space. Each time step the agent chooses not only a domain-level action to execute but also it can select a *commit* action asserting that it believes a state feature to be true (in case of binary state factors). These *commit* actions have no effect on the state of the environment whatsoever, but only serve to allow the system designer to reward the agent for reaching a particular level of knowledge regarding a state factor of interest. By carefully balancing the reward an agent receives for correctly and incorrectly selecting *commit* actions, we can induce the agent to exhibit information-seeking behavior. We show how these information rewards have to be set based on a desired level of relative belief entropy (for instance).

The POMDP model has been used before to address information-gain tasks such as sensing [22] or spoken dialog systems [61]. However, we provide a framework to balance information-gain objectives with other tasks. In particular, we provide a principled way for the user to specify the degree of certainty regarding a state feature the agent should aim to reach. Furthermore, we show the effect on the behavior that such choices have, providing insight into the tradeoff between seeking information and completing concurrent tasks.

The benefit of the POMDP-IR framework is that we can reason about beliefs over certain features in the environment without leaving the classic POMDP framework. By remaining in the standard POMDP setting we can exploit many results that are known as well as successful approximate algorithms that have been developed. In this sense, our contributions are independent of the particular POMDP planner used. Furthermore, existing POMDP models can be easily augmented with information rewards to capture information-gain objectives.

We demonstrate the POMDP-IR framework in several experimental settings. First, we use a toy problem to illustrate the effect of choosing different parameter settings. Second, we compare our modeling framework with the ρPOMDP framework which also targets information gain in POMDPs. We show that on their benchmark problems we obtain better information-gain results. Third, we perform simulation experiments in a larger robot-assisted surveillance setting, in which a robot assists a network of surveillance cameras to identify a person, as illustrated in Fig. 1b. Fourth, we illustrate the applicability of a slightly modified framework on a real robotic system by showcasing it on a mobile robot and 5 cameras.

1.4 Outline

The paper is organized as follows. In Sect. 2 we provide some background on POMDPs and their solution methods, with a focus on factored models, which are crucial to ensure scalability. Section 3 introduces the active cooperative perception problem, followed by the introduction of the POMDP-IR framework in Sect. 4. Section 5.1 shows how the POMDP-IR framework can be applied to a simple toy problem and Sect. 5.2 provides experimental results comparing with ρPOMDPs [3]. Next, in Sect. 6 we present a case study with simulated and real-robot results in a robot-assisted surveillance setting. Section 7 discusses related work, and finally, in Sect. 8 we draw our conclusions and present future work.

2 Background

We provide the required background on the POMDP model and solution concepts as well as a short introduction to factored POMDP models.

2.1 Partially observable Markov decision processes

We introduce the POMDP model [23] on which our work is based. A POMDP models the interaction of an agent with a stochastic and partially observable environment, and it provides a rich mathematical framework for acting optimally in such environments. A more detailed overview of POMDPs and their solution algorithms is provided by Spaan [51].

A POMDP can be represented by a tuple $\langle S, A, O, T, \Omega, R, h, \gamma \rangle$. At any time step the environment is in a state $s \in S$, the agent takes an action $a \in A$ and receives a reward $R(s, a)$ from the environment as a result of this action, while the environment switches to a new state s' according to a known stochastic transition model $T : p(s'|s, a)$. After transitioning to a new state, the agent perceives an observation $o \in O$, that may be conditional on its action, which provides information about the state s' through a known stochastic observation model $\Omega : p(o|s', a)$. The agent's task is defined by the reward it receives at each time step t and its goal is to maximize its expected long-term reward $E[\sum_{t=0}^{h-1} \gamma^t R(s_t, a_t)]$, where h is the planning horizon, and γ is a discount rate, $0 \leq \gamma < 1$.

Given the transition and observation model the POMDP can be transformed to a belief-state MDP: the agent summarizes all information about its past using a belief vector $b(s)$. The initial state of the system is drawn from the initial belief b_0, and every time the agent takes an action a and observes o, its belief is updated by Bayes' rule:

$$b^{ao}(s') = \frac{p(o|s', a)}{p(o|a, b)} \sum_{s \in S} p(s'|s, a) b(s), \tag{1}$$

where

$$p(o|a, b) = \sum_{s' \in S} p(o|s', a) \sum_{s \in S} p(s'|s, a) b(s) \tag{2}$$

is a normalizing constant.

2.2 POMDP policies

In a POMDP, a policy π can be characterized by a value function $V^\pi : \Delta(S) \to \mathbb{R}$ which is defined as the expected future discounted reward $V^\pi(b)$ the agent can gather by following π starting from belief b:

$$V^\pi(b) = E_\pi \left[\sum_{t=0}^{h-1} \gamma^t R(b^t, \pi(b^t)) \middle| b^0 = b \right], \tag{3}$$

where $R(b^t, \pi(b^t)) = \sum_{s \in S} R(s, \pi(b^t)) b^t(s)$. A policy π which maximizes V^π is called an optimal policy π^*, the value of which is defined by the optimal value function V^*. The optimal value function satisfies the Bellman optimality equation $V^* = HV^*$, where H is the Bellman backup operator for POMDPs:

$$V^*(b) = \max_{a \in A} \left[\sum_{s \in S} R(s, a) b(s) + \gamma \sum_{o \in O} p(o|b, a) V^*(b^{ao}) \right], \tag{4}$$

with b^{ao} given by (1), and $p(o|b, a)$ as defined in (2). Solving POMDPs optimally is hard, and thus algorithms that compute approximate solutions are often used [21,37,53].

2.3 PWLC value functions

Many of the exact and approximate algorithms exploit the fact that the finite-horizon POMDP value function is piecewise linear and convex (PWLC), which allows for its compact representation. Next we provide the intuition why a POMDP value function has a PWLC shape. If the agent has only one time step left to act, we only have to consider the immediate reward and (4) reduces to:

$$V_0^*(b) = \max_a \Big[\sum_s R(s,a)b(s) \Big]. \tag{5}$$

We can view $R(s,a)$ as a set of $|A|$ vectors α_0^a, where $\alpha_0^a(s) = R(s,a)$. Now we can rewrite (5) as follows:

$$V_0^*(b) = \max_a \sum_s \alpha_0^a(s)b(s), \tag{6}$$

$$= \max_{\{\alpha_0^a\}_a} b \cdot \alpha_0^a, \tag{7}$$

where $(\cdot)$ denotes inner product. In summary, we view the immediate reward as a vector α_0^a for each state and (7) averages α_0^a with respect to the belief b. As averaging is a linear operator and V_0^* consists of $|A|$ linear vectors α_0^a, it is piecewise linear. Since the value function is defined as the upper surface of these vectors, due to the max operator, V_0^* is also convex. In the general case, we parameterize a value function V_n at stage n by a finite set of vectors or hyperplanes $\{\alpha_n^k\}$. The value of a belief b is given by

$$V_n(b) = \max_{\{\alpha_n^k\}_k} b \cdot \alpha_n^k. \tag{8}$$

2.4 Factored POMDP models

As is common in the literature on real-world POMDPs [7,55], we considered factored POMDP models, which allow for exploiting structure by representing models in a two-stage dynamic Bayesian network (DBN) representation [8,20]. In this case the state space is factored as follows (and the observation space can be factored in a similar fashion):

$$S = X_1 \times X_2 \times \cdots \times X_i \times \cdots \times X_k, \tag{9}$$

where X_i denotes a state factor. For instance, the person localization problem illustrated by Fig. 1a could be modeled using 2 state factors: the person's location and the robot's location. When the robot also needs to keep track of the identity of a person (Fig. 1b), a state factor modeling the person's features should be added. Using a factored representation is computationally convenient, but is often a simplification of the environment. For instance, in our robot experiments we assume that the transitions of robot and people are independent, but the robot's trajectories depend on obstacles it encounters, including people. Generally speaking, in the domains that we consider the ability to solve larger problem domains outweighs the loss of modeling accuracy.

Subsequently, rewards can depend on subsets of state factors, and the reward can be defined as

$$R(S,A) = \sum_j R(Y_j, A), \tag{10}$$

where Y_j is a subset of the state factors that make up S. As defined in Sect. 2.1, in a standard POMDP rewards depend on state-action pairs. Assuming for the moment that each reward

component depends on a single state factor, this translates to pairs of state factors and actions in the factored case:

$$R : X_i \times A \to \mathbb{R}, \tag{11}$$

and for a given belief b_i over a state factor X_i,[1] the POMDP considers the expected reward:

$$R(b_i, a) = \sum_{x_i \in X_i} b_i(x_i) R(x_i, a). \tag{12}$$

This linear reward function leads to linear value functions, as described in Sect. 2.3.

3 Active cooperative perception

Next, we introduce the intuition behind our new approach to active cooperative perception with POMDPs. We start by discussing how information gain is used in classic POMDPs, followed by an example of how we envision active cooperative perception in our framework.

3.1 Information gain in POMDPs

Regular state-based rewards allow for defining many tasks, but do not explicitly reward information gain [50]. However, if more information improves task performance, a POMDP policy will take information-gaining actions. For instance, a better self-localization estimate can help a robot to navigate faster to a certain destination. Indeed, a key point of the POMDP framework is that it will only reduce uncertainty when it is beneficial for task performance, but not for information gain per se. Araya-López et al. [4] note that "One can argue that acquiring information is always a means, not an end, and thus, a "well-defined" sequential-decision making problem with partial observability must always be modeled as a normal POMDP." They observe that in some applications, such as surveillance tasks or the exploration of a particular area, it is not clear how the information will be used by whoever is designing the system.

For instance, we may explicitly model the classification of a particular target as an objective, which induces a well-defined state-based task [19]. If we consider a more general surveillance setting, considering non-standard POMDP models could be beneficial. In particular, often no model will be available of how human operators evaluate and act upon the output of a surveillance system. Without a detailed description of all the surveillance objectives the problem cannot be cast as a well-defined POMDP. In such cases, the proposed methodology can be used to actively reduce uncertainty regarding certain important state features.

It is important to note that in many applications the user of the system might not be interested in reducing uncertainty in general, but only with respect to some features in the environment. For instance, consider the examples presented in Fig. 1, in which a robot provides information to either localize a target more accurately or to identify a person. In both cases, the robot's own location might be uncertain as well, but that is irrelevant to the user of the system.

[1] Maintaining a factorized but exact belief state is typically not desirable for reasons of tractability, but bounded approximations are possible [9] in which the exact belief b is approximated by the product of marginals over individual state factors b_i. This is a common approach in factorized POMDP solving [37].

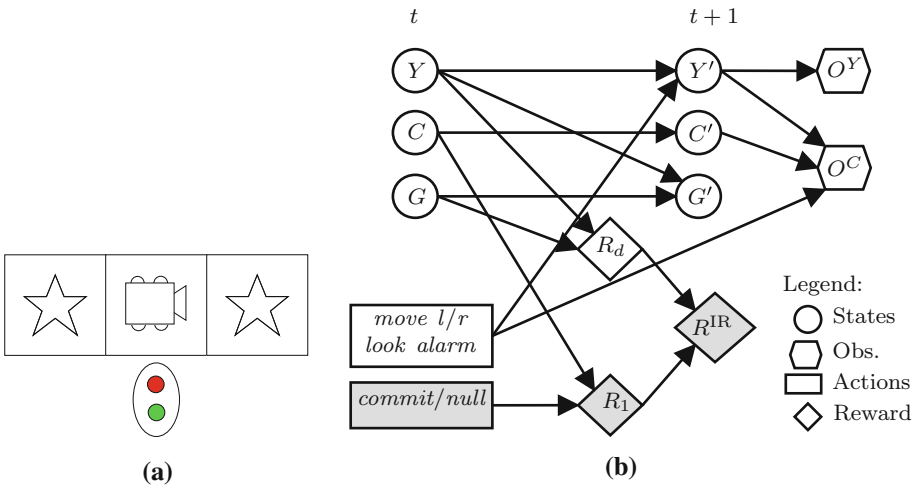

Fig. 2 PATROL example. **a** Illustration of the problem, showing a corridor of length 3, a robot which has to travel between two goal nodes (marked by *stars*) and an alarm device in the middle. **b** Dynamic Bayesian Network representation of the problem with state factors $Y = \textit{robot position}$, $C = \textit{alarm color}$ and $G = \textit{current goal}$. The *shaded nodes* indicate the POMDP-IR additions to the model

3.2 Running example: PATROL

To clarify the concepts presented in this section we include a small illustrative example, shown in Fig. 2a, dubbed PATROL. Imagine that we have a corridor environment in which a surveillance robot has to patrol between the two ends of the space (represented by stars). However, in the middle of the corridor an alarm device is present, which can be in two different configurations, *red* and *green*, where a red color means that the attention of a human operator is required.

This can be seen as a multi-objective problem, where the task of the robot is to patrol the environment while keeping track of the state of the alarm. The latter can be formalized as maintaining a low-uncertainty belief regarding the state of the alarm device. The robot can pause to observe the alarm, which delays the original patrol task. Each task is valued by its reward function and the system designer defines the balance between these objectives.

Figure 2b presents a DBN representation of this problem (where the shaded nodes indicate the POMDP-IR extensions discussed in Sect. 4). This model is based on a simple robot navigation model, with a state factor $Y = \{1, 2, 3\}$ denoting the robot position in order from left to right, and a state factor $G = \{\textit{left}, \textit{right}\}$ representing the current goal. The latter represents the left or the right end of the corridor (see Fig. 2a), and flips when the current goal is reached. To account for the alarm, we introduce a state factor C, which models the alarm color and hence has two possible values, *red* and *green*. The initial position of the robot is the leftmost end of the environment and the initial belief regarding alarm color is uniform. The model assumes that the alarm turns *red* with probability 0.2, and once it does, it returns to *green* with probability 0.1. The action space is defined as {*move left*, *move right*, *look alarm*} and when moving the probability of reaching the target location is 0.8 while the robot stays in the same location with probability 0.2. The reward for reaching the current goal is 0.3 for a corridor length of 3. It is not straightforward how to implement a reward function in order to perform information gain. As discussed in Sect. 3.1, state-based rewards do not allow for

explicitly rewarding information gain. We return to this problem in Sect. 5.1 where we show how to model it in the POMDP-IR framework, which we introduce next.

4 POMDPs with Information Rewards

In this section we present the main contribution of this work, a framework for rewarding low-uncertainty beliefs without leaving the classic POMDP framework for efficiency reasons.

4.1 Models for active perception

From the perspective of active perception, as the belief is a probability distribution over the state space, it is natural to define the quality of information based on it. Now, for belief-based rewards one can define the reward ρ directly over the belief space:

$$\rho : \Delta(S) \times A \rightarrow \mathbb{R}. \tag{13}$$

Araya-López et al. [4] mention several convex functions that are typically used for maximizing information, such as the Kullback–Leibler divergence (also known as relative entropy) with respect to the simplex center c, i.e., a uniform belief ($D_{KL}(b||c)$),

$$\rho_{KL}(b, a) = \sum_s b(s) \log\left(\frac{b(s)}{c(s)}\right) = \log(|S|) + \sum_{s \in S} b(s) \log(b(s)), \tag{14}$$

or the distance from the simplex center (DSC)

$$\rho_{dsc}(b, a) = \|b - c\|_e, \tag{15}$$

where e indicates the order of the metric space.

We could use the belief to define a measurement of the expected information gain when executing an action. For instance, a common technique is to compare the entropy of a belief b^t at time step t with the entropy of future beliefs, for instance at $t+1$. If the entropy of a future belief b^{t+1} is lower than b^t, the robot has less uncertainty regarding the true state of the environment [11]. Assuming that the observation models are correct (e.g., unbiased) and observations are independent, this would mean we gained information. Given the models, we can predict the set of beliefs $\{b^{t+1}\}$ we could have at $t+1$, conditional on the robot's action a. If we adjust the POMDP model to allow for reward models that define rewards based on beliefs instead of states, i.e., $\rho(b, a)$, we can define a reward model based on the belief entropy.

However, a non-linear reward model defined on beliefs like the entropy significantly raises the complexity of planning, as the value function will no longer be piecewise linear and convex, as Eq. (7) no longer holds, and therefore we are no longer able to apply classic solvers to such problems. Moreover, if we want to model problems with two different kinds of goals, both information gain and task performance, reward models defined only on beliefs are not convenient.

4.2 Rewarding low-uncertainty beliefs

We introduce a different way to reward information gain, while remaining in the classic POMDP framework with PWLC value functions, as discussed in Sect. 2. Instead of directly rewarding beliefs, we introduce the addition of "information-reward" actions to the problem definition, which allow for rewarding the system to obtain a certain level of knowledge

regarding particular features of the environment. In this way, we achieve a similar objective as defining reward functions based on negative belief entropy, while remaining in the classic POMDP framework.

Hence, the goal of our work is to reward the system to have beliefs that have low uncertainty with respect to particular state factors of interest. For convenience, we assume these are the first l state factors (with $l \leq k$), and hence can be denoted $X_1, X_2, \ldots, X_i, \ldots X_l$. For simplicity, for the moment we assume that each X_i is binary, having values x_i and $\overline{x}_i$. The idea is that we can expand the action space in such a way to allow for rewarding reaching or maintaining a particular low-uncertainty belief over each X_i. In the binary case, we can model this by considering actions that assess whether $X_i = x_i$.

We now define our POMDP-IR model by extending the standard POMDP definition (as provided in Sect. 2.1) with information-reward actions.

Definition 1 (*Action space*) We call the original, domain-level action space of the agent A_d. For each $X_i, i \leq l$, we define

$$A_i = \{commit, null\}.$$

The action space of the POMDP-IR is

$$A^{\text{IR}} = A_d \times A_1 \times A_2 \times \cdots \times A_l.$$

Given this definition, at each time step the agent *simultaneously* chooses a regular domain-level action and an action for each state factor of interest. These actions have no effect on the state transitions nor on the observations, but do affect rewards. The *null* action is added to give the agent the option to not make any assertions regarding the information objectives. It is provided for modeling convenience, as it allows the system designer to consider the agent's task without information objectives.

In the PATROL example, as we are interested in the state of the alarm, C, we denote $X_1 = C$ and set $l = 1$. The action space is now defined as $\{move\ left, move\ right, look\ alarm\} \times \{commit, null\}$.

Definition 2 (*Information rewards*) We call the original reward function of the POMDP R_d. The POMDP-IR's reward function R^{IR} is the sum of R_d and a reward R_i for each $X_i, i \leq l$:

$$R^{\text{IR}}(X, A) = R_d(X, A_d) + \sum_{i=1} R_i(X_i, A_i).$$

Each $R_i(X_i, A_i)$ is defined as

$$\begin{aligned} R_i(x_i, commit) &= r_i^{\text{correct}}, \\ R_i(x_i, null) &= 0, \\ R_i(\overline{x}_i, commit) &= -r_i^{\text{incorrect}}, \\ R_i(\overline{x}_i, null) &= 0, \end{aligned}$$

with $r_i^{\text{correct}}, r_i^{\text{incorrect}} > 0$.

The upshot of this reward function is that at every time step, the agent can choose to either execute only a domain-level action ($A_i = null$), or in addition also receive reward for its belief over X_i ($A_i = commit$). Intuitively, r_i^{correct} is the reward the agent can obtain for guessing the state of X_i correctly, and $r_i^{\text{incorrect}}$ is the penalty for an incorrect guess. We choose r_i^{correct} and $r_i^{\text{incorrect}}$ in such a way that the agent only benefits from guessing when it is certain enough

about the state of X_i. When the agent is not certain enough, it can simply choose the *null* action (in combination with any domain-level action). However, the possibility of obtaining information rewards by having a low-uncertainty belief over X_i will steer the agent's policy towards such beliefs.

4.3 Choosing the information-reward parameters

From Definition 2 we can see that the expected reward for each information-reward action is:

$$R_i(b, commit) = b_i(x_i) r_i^{\text{correct}} - (1 - b_i(x_i)) r_i^{\text{incorrect}}, \tag{16}$$

$$R_i(b, null) = 0, \tag{17}$$

using a short-hand notation $b_i(x_i)$ to indicate $b_i(X_i = x_i)$. Thus, the expected reward of choosing *commit* is only higher than the *null* action when

$$b_i(x_i) r_i^{\text{correct}} - (1 - b_i(x_i)) r_i^{\text{incorrect}} > 0, \tag{18}$$

which, by rearranging, becomes:

$$b_i(x_i) > \frac{r_i^{\text{incorrect}}}{r_i^{\text{correct}} + r_i^{\text{incorrect}}}. \tag{19}$$

This inequality indicates the range over which the agent will execute a *commit* action, and we can see that the values of r_i^{correct} and $r_i^{\text{incorrect}}$ are decisive to determine this range.

If we assume that we want to reward the agent for having a degree of belief of at least β regarding a particular X_i, i.e., $b_i(x_i) > \beta > 0$, β is computed according to Eq. (19), and therefore:

$$\beta = \frac{r_i^{\text{incorrect}}}{r_i^{\text{correct}} + r_i^{\text{incorrect}}}, \tag{20}$$

$$\Leftrightarrow r_i^{\text{correct}} = \frac{(1 - \beta)}{\beta} r_i^{\text{incorrect}}. \tag{21}$$

Equation (20) tells us the relation between r_i^{correct} and $r_i^{\text{incorrect}}$. Their precise values depend on each problem, and on the insight of the system designer (just as the domain-level rewards), taking into account his knowledge of the models and the environment and calibrating it with the original reward model R_d. For instance, in the PATROL example, if the robot is rewarded too much for maintaining a low-uncertainty belief regarding the alarm, it will prefer to look at the alarm constantly, ignoring its patrol task.

The effectiveness of this scheme depends on whether in the particular POMDP reaching a particular β is possible at all, due to sensory limitations. For instance, if an agent's sensors do not observe X_i at all, or provide too noisy information, beliefs in which $b_i(x_i) > \beta$ can be unreachable given an initial belief state. In point-based POMDP methods that operate on a pre-defined belief set, this condition can be checked easily, and β be adjusted accordingly.

A second option is to define several β levels for an X_i, which reward the agent for reaching different levels of certainty. Care needs to be taken, however, to ensure that the agent will try to reach the highest β possible. For instance, Table 1 defines possible values for r_i^{correct} and $r_i^{\text{incorrect}}$ for different values of β, computed using different criteria. As the potential reward for higher β is higher as well, the agent is guided to reach the highest certainty level possible.

Table 1 Example rewards for varying β

	$D_{KL}(b\|\|c)$ (14)		DSC, $m = 1$ (15)		DSC, $m = 2$ (15)		DSC, $m = \infty$ (15)	
β	r_i^{correct}	$r_i^{\text{incorrect}}$	r_i^{correct}	$r_i^{\text{incorrect}}$	r_i^{correct}	$r_i^{\text{incorrect}}$	r_i^{correct}	$r_i^{\text{incorrect}}$
0.60	0.03	0.04	0.20	0.30	0.02	0.03	0.10	0.15
0.75	0.19	0.57	0.50	1.50	0.12	0.38	0.25	0.75
0.90	0.53	4.78	0.80	7.20	0.32	2.88	0.40	3.60
0.99	0.92	91.00	0.98	97.02	0.48	47.54	0.49	48.51

4.4 Extensions and variations

When defining the POMDP-IR framework in Sect. 4.2, we assumed binary state factors for simplicity. Our framework can be easily generalized to multi-valued state factors, by defining one or multiple *commit* actions for a state factor X_i. In the binary case, we assign a positive reward to one element in the domain of X_i, and a negative one to the other. In the multi-valued case, a single *commit* action can result in positive reward for multiple values of X_i when each of those values is equally desirable, by trivial generalization of Definition 2. In the case that distinct values of X_i differ in desired reward, the definition of the information-reward action can be extended to

$$A_i = \{commit_1, commit_2, \ldots, commit_j, null\}.$$

In this way, certain values of X_i can be rewarded differently, and sets of values that share the same reward can be grouped under the same $commit_j$ action.

In our framework, we consider the general case in which an agent at each time step can opt for a *commit* action or not: information rewards are obtainable at each time step. However, possible use cases of the POMDP-IR framework might consider other scenarios. For instance, it might be desirable to reward the agent only a single time for each *commit* action. Such a scenario is easily implemented by adding a Boolean bookkeeping variable that keeps track whether a *commit* action has been performed or not. By making the information reward dependent on the value of this variable, the agent will optimize its course of action to execute *commit* only once.

In a similar way, in certain active-sensing or sensor-management scenarios an agent might have to decide whether to execute *commit* after a predefined number of steps. By encoding the time step as an extra state factor or by defining time-dependent action spaces, the POMDP-IR framework can be used in these types of scenarios as well.

5 Experiments

We perform an extensive experimental evaluation of the POMDP-IR framework. First, we illustrate how information rewards influence agent behavior in the PATROL example. Second, we compare the performance of POMDP-IR against ρPOMDPs [3] on the ROCK DIAGNOSIS problems.

5.1 Information rewards in the PATROL example

At this point we return to the PATROL example presented in Sect. 3.2. In this example we want to solve a problem where an agent needs to patrol an environment while at the same

time considering the uncertainty regarding the state of an alarm device. This illustrates the tradeoff between task performance and information gain that we want to tackle.

5.1.1 Model

The POMDP model is an extension of the model presented in Sect. 3.2 and is depicted in Fig. 2b. The domain-level action space is $A_d = \{move\ left, move\ right, look\ alarm\}$ and we extend it with an information-reward action regarding the alarm state factor C, $A_1 = \{commit, null\}$. The reward function is a sum of rewards for both types of actions, $R^{\text{IR}} = R_d + R_1$, according to Definition 2. R_1 rewards the robot if its belief whether the alarm is red exceeds a threshold β. The reward values will depend on each particular problem, namely the environment size and the threshold defined in each case. Information reward values depend on the value of β used in each experiment, and is up to the system designer to choose a criteria for those values. In our particular case, values are taken from the first column of Table 1. The patrol reward encoded in R_d is 0.3 for corridor length 3 and 0.5 for corridors of length 5. We keep initial conditions and models constant for all experiments to be able to showcase the effect of the belief threshold(s).

5.1.2 Experiments

To showcase the POMDP-IR framework, we present simulations of the PATROL problem using policies computed by Symbolic Perseus [37]. We consider different cases, namely corridor lengths of 3 and 5, several threshold values, and more than one threshold. Furthermore, we study the scalability of our approach.

Single threshold Considering that the original behavior of the robot would be to patrol the corridor up and down, we note a change in its behavior in Fig. 3a, b, as it continues to perform the patrolling task, but with some intermediate stops (as seen in the 2nd row of plots). The bottom row details the patrol (R_d) and information reward (R_1) the robot receives. Figure 3a shows results where $\beta = 0.75$ and the robot clearly tries to maintain up-to-date information on the alarm, by looking at it every time it passes by. In this case the threshold is such that the robot has enough time to continue patrolling and still return to the alarm position before its belief falls below the threshold. On the contrary, we note that with $\beta = 0.9$ (Fig. 3b), the threshold is higher than in the first case and thus the behavior changes slightly. Every time the robot observes a green alarm it continues performing the patrolling task. However, if it observes a red alarm, it decides to stay and keep looking at it. On the other hand, with $\beta = 0.99$ (Fig. 3c) the threshold is so high that even if the robot stays looking at the alarm, it will never get to levels above the threshold. Therefore, it will only focus on patrolling, illustrating the need to choose the thresholds carefully. In Fig. 3d we extend the simulation to a corridor length of 5, with $\beta = 0.75$. In this setting the behavior is similar to the previous results, except for the fact that when the robot decides to look at the alarm it spends 2 time steps.

Multiple thresholds Next, we test a model with two information-reward actions, each corresponding to a different threshold. We see in Fig. 3b, e that here the behavior of the robot is equivalent to the maximum threshold it can reach. For instance, in Fig. 3e with $\beta_1 = 0.75$ and $\beta_2 = 0.9$ the robot's behavior is equivalent to Fig. 3b ($\beta = 0.9$). On the other hand, in Fig. 3f the thresholds $\beta_1 = 0.75$ and $\beta_2 = 0.99$ are set, but as seen before the system will

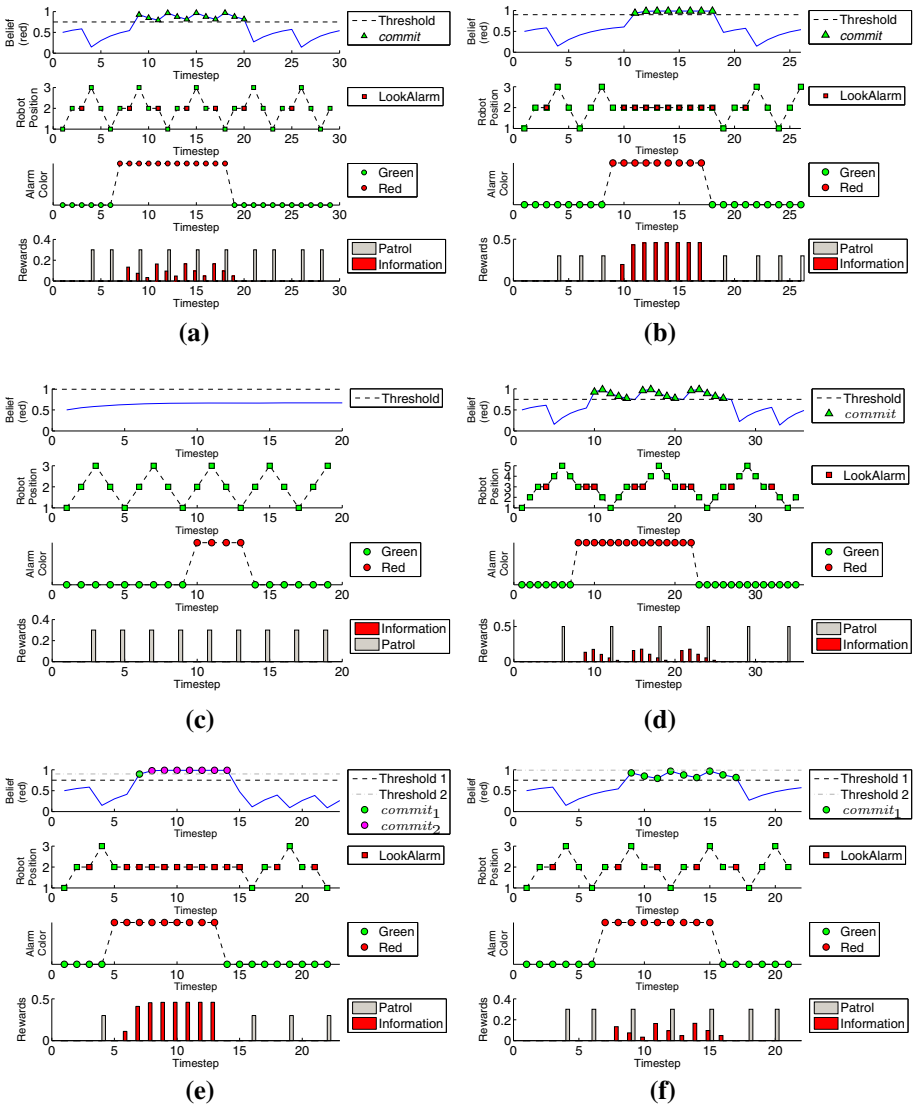

Fig. 3 PATROL problem results. Each *figure* shows belief evolution (*top row*), robot position (*2nd row*), true alarm color (*3rd row*) and received rewards (*bottom row*). Corridor length is 3 except for **d**. **a** Threshold $\beta = 0.75$, **b** Threshold $\beta = 0.9$, **c** Threshold $\beta = 0.99$, **d** $\beta = 0.75$, corridor of length 5, **e** Threshold $\beta_1 = 0.75$ and $\beta_2 = 0.9$, **f** Threshold $\beta_1 = 0.75$ and $\beta_2 = 0.99$

ignore the second threshold as the belief will never reach it, and therefore its behavior will be as if there were only one threshold $\beta = 0.75$.

Scalability Our approach inflates the action space, which will lead to increased computation time. Therefore, it is also important to measure the computational impact of adding information-reward actions to problems, shown in Fig. 4. Note, however, that POMDP solving scales linearly in the number of actions, compared to exponentially in the number of

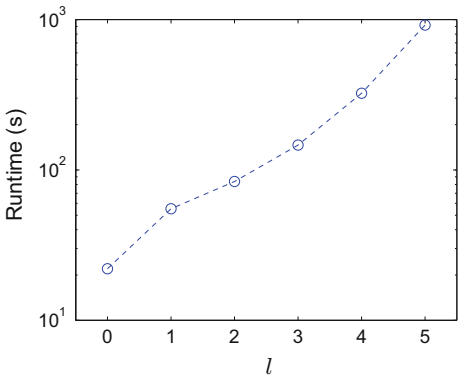

Fig. 4 Performance measures for PATROL problem with 5 alarms, and variable number of *commit* actions with $\beta = 0.9$

observations. To allow for a fair comparison we must compare models with the same domain-level actions and a variable number of information-reward actions. We create a PATROL model which includes 5 alarms and test the planner's performance when including a different number of *commit* actions. We note that the planner's performance degrades exponentially, as expected, although with more *commit* actions the system will receive higher value (not shown).

5.1.3 Discussion

With this toy problem we showed the advantages of our approach, by directly incorporating the concept of information gain in a classical POMDP setup. We can observe that although information-reward actions do not directly affect state transitions they have a clear effect on the robot's general behavior, as it will change its physical actions according to not only the original task to solve, but also to the uncertainty level in the system and the thresholds.

As expected, the robot faces a tradeoff between patrol task completion and information gain regarding the alarm. However, this is dependent on a number of different parameters, the desired certainty level in particular. In general when it is possible to reach the desired threshold the robot will interrupt patrolling to observe the alarm and get information. On the contrary, if the desired level is unreachable or if it is so low that the belief converges to a value above the threshold without the need for direct observation, the robot does not need to waste time looking at the alarm and only performs the patrol task.

5.2 Comparison with ρPOMDPs on the ROCK DIAGNOSIS problem

Next, we compare our work to a POMDP-based framework for information gain, the ρPOMDP framework [3]. We present results in a scenario where information gain plays an important role: the ROCK DIAGNOSIS problem [3], which is a variation of the ROCK SAMPLING problem [49].

The approach in ρPOMDPs extends POMDPs to allow for direct belief-based rewards. It is noted that the belief-MDP formulation already accounts for a reward based on beliefs (12), but formulated in terms of state-based reward, in order to maintain the PWLC property. Therefore, ρPOMDPs generalize POMDPs to handle other types of belief-based rewards, under the assumption that those rewards are convex. Algorithms are modified in order to deal with rewards represented as a set of vectors. For PWLC reward functions, this is an exact

representation. For non-PWLC reward functions it is an approximation which improves as the number of vectors used to represent them increases. As we saw in Sect. 4 our approach also considers the belief-MDP formulation, but in a way that represents rewards for information gain in terms of state-based rewards, allowing us to use existing methods without changes.

In their experiments [3] an extension of point-based methods is used to accommodate this generalization. For comparison we present here their results with two different reward functions, entropy (PB-Entropy) and linear (PB-Linear). The entropy-based reward is the Kullback–Leibler divergence with the uniform distribution as reference, while the linear-based reward is an approximation of the entropy-based reward which corresponds to the L_∞-norm of the belief.

5.2.1 ROCK DIAGNOSIS *problem definition*

The ROCK DIAGNOSIS is an information-gathering problem, in which a rover receives a set of rock positions and must perform sampling procedures in order to reduce uncertainty regarding the type of several rocks spread in an environment. Each rock can have a *good* or a *bad* value and we want the system to be capable of returning low-uncertainty information regarding each rock's type. Hence, our objective is to produce policies which guide the rover through the environment, allowing it to observe each rock's type. The map of the environment is considered to be a square grid of size p and there are q rocks in the environment to analyze. We refer to Araya-López [3] for further details on the ROCK DIAGNOSIS problem.

In the original formulation $A_d = \{north, east, south, west, check_1, check_2, \ldots, check_q\}$ and in our POMDP-IR formalization we use the extension described in Sect. 4.4 to include a set of information-reward actions for each rock: $A_i = \{commit_1, commit_2, null\}$, with $1 \leq i \leq q$. In this case, $commit_1$ and $commit_2$ encode the action that a rock is *good* or *bad*, respectively. The intuition is that the policy will try to reduce the uncertainty regarding each rock's type in order to get a higher value, thereby achieving the problem's final objective.

5.2.2 Experimental setup

We computed policies for this problem using Perseus [53] and Symbolic Perseus [37], with $\gamma = 0.95$ and $\epsilon = 10^{-3}$, where ϵ is the convergence criterion used both by Perseus and Symbolic Perseus. We ran a set of 10 repetitions of 100 trajectories of 100 steps, re-sampling the belief set containing 5000 beliefs at each repetition. Given the information-gathering nature of this problem, the performance criterion used to evaluate the agent's behavior should also be an information measure. In particular, the Kullback–Leibler divergence between the belief distribution and the uniform distribution is used (14).

In this problem the performance criterion only considers the available information at the end of the trajectory, as the objective is to disambiguate the state of all the rocks' types. Therefore, we present results as the average of the Kullback–Leibler divergence of the belief distribution in the last time step at each trajectory. The unit used to measure information is *nat* (natural unit for information entropy), since we use natural logarithms. Also note that the target variable is the rock type, thus our information measure considers only the belief over state factors which represent each rock's type. Therefore, the maximum possible final reward is $q \log(2)$, precisely when there is no uncertainty regarding any rocks' type.

Table 2 ROCK DIAGNOSIS results, comparing POMDP-IR with ρPOMDP [3]

ρPOMDP		POMDP-IR (Perseus)		POMDP-IR (Symbolic Perseus)	
Method	Return [*nats*]	β	Return [*nats*]	β	Return [*nats*]
$q = 3; p = 3$					
PB-Entropy	1.58 ± 0.25	0.6	2.065 ± 0.099	0.6	1.978 ± 0.092
PB-Linear	2.06 ± 0.03	0.9	2.079 ± 0.000	0.9	1.988 ± 0.223
		0.99	1.815 ± 0.334	0.99	2.035 ± 0.124
$q = 3; p = 6$					
PB-Entropy	0.76 ± 0.09	0.6	1.763 ± 0.463	0.6	1.386 ± 0.000
PB-Linear	0.79 ± 0.08	0.9	1.790 ± 0.417	0.9	1.580 ± 0.213
		0.99	1.662 ± 0.390	0.99	1.156 ± 0.114
$q = 5; p = 7$					
PB-Entropy	0.37 ± 0.09	0.6	—	0.6	1.580 ± 0.313
PB-Linear	0.53 ± 0.03	0.9	—	0.9	1.553 ± 0.298
		0.99	—	0.99	1.737 ± 0.456

5.2.3 Results

We present in Table 2 a set of results for different maps with varying values of β, and also, for comparison, results presented by Araya-López [3]. The implementation of POMDP-IR with flat Perseus could not handle the larger map ($q = 5$; $p = 7$).

We can see that, in general, POMDP-IR achieves better results than the two ρPOMDP variations. The total return with ρPOMDP deteriorates with increasing problem size due to sampling a larger state space with the same number of belief points. However, we note that POMDP-IR results worsen less, which may be explained by the fact that ρPOMDPs approximate the reward function with linear vectors, imposing some error on the original reward function. This results in a loss of quality in larger problems, since the same number of points is used to approximate a higher dimensional belief space. Instead we directly consider reward depending on states, thus staying in the traditional POMDP formulation. Using fewer approximations we can achieve better performance.

Our results show better performance in the smaller map, with similar results for Perseus and Symbolic Perseus. Note the particular case of Perseus with $q = 3, p = 3, \beta = 0.9$ where we always achieve perfect information about all rocks in the environment. There is a trend for Perseus to perform slightly better than Symbolic Perseus. However, we note that Symbolic Perseus implements an additional approximation by maintaining a factored belief representation which may explain the difference in results.

We may then state that, although our focus is on adding information rewards to regular POMDPs, we also improve performance on these pure information-gathering problems. The problem size can be an issue, as we see in the largest scenario that could not be handled using a flat method like Perseus. However, our focus is on factored representations and we showed that we can perform well using Symbolic Perseus.

We also include a comparison between average computation times in Fig. 5, where we included ρPOMDP timings reported by Araya-López [3] which precludes direct comparisons. The increase in computation time in POMDP-IR implementations is a natural consequence of the increase in the action space size (as the growth is exponential in the number of action factors), but we present a reasonable tradeoff between computation time and average value in this particular case.

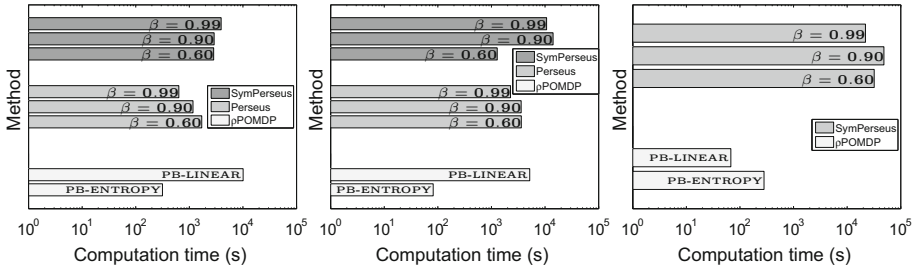

Fig. 5 ROCK DIAGNOSIS computation times. ρPOMDP timings are taken from [3] and hence cannot be compared directly. **a** Map with $q = 3$; $p = 3$, **b** Map with $q = 3$; $p = 6$, **c** Map with $q = 5$; $p = 7$

6 Case study: robot-assisted surveillance

The main motivating application for our research is robot-assisted surveillance. In such scenarios, a robot can be seen as a mobile extension of a camera network, helping to improve confidence of detections. For instance, due to several reasons (lighting, distance to camera, capture quality, etc.) the image quality may vary from camera to camera, leading to different uncertainty rates when detecting robots, persons or other features in the environment. Also, in a surveillance system the camera network will typically not cover all the environment, leaving some blind spots. Such problems can be alleviated by using mobile sensors, which can move to provide (improved) sensing in particular areas.

In our case study we are interested in a surveillance system which detects persons with a particular feature. Feature detection is achieved through image processing methods that detect persons moving in the environment. In our case, for simplicity, we consider as feature whether a person is wearing red upper body clothing. Note that our methods are rather independent of the actual feature detector used, as long as false positive and false negative rates can be estimated.

6.1 Model and experimental setup

We implemented our case study in a testbed for NRS [5], which consists of a camera network mounted on the ceiling of the lab (Fig. 6b), and mobile robots (Pioneer 3-AT) with onboard camera and laser range finder (Fig. 6a). The POMDP controllers are computed using Symbolic Perseus [37].

Figure 7 depicts a graphical representation of the model for time steps t and $t + 1$. As before, we encode the environment in several state variables, depending on how many people and features the system needs to handle. The locations of people and robot are represented by a discretization of the environment, for instance a topological map. Graphs can be used to describe topological maps with stochastic transitions and hence sets of nodes represent the robot location Y and k people locations P_1 through P_k. In particular, we run our experiments in the lab shown in Fig. 6, building a discretized 8-node topological map for navigation and position identification. Besides a person's location, we represent a set of m features where each feature f is associated with a person, for instance whether it matches a visual feature. We assume each person has at least one feature, hence $m \geq k$, and features are represented by variables F_1 through F_m.

Fig. 6 Experimental setup. **a** Robot Pioneer 3-AT with camera and laser range finder onboard. **b** Example of an image captured by the camera network. **c** Feature detection by the robot

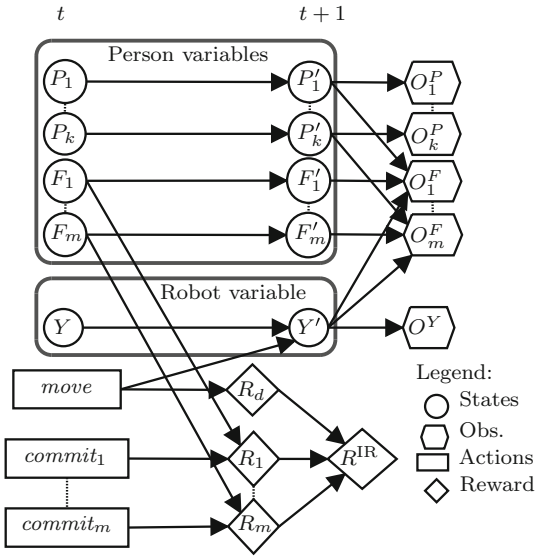

Fig. 7 Two-stage dynamic Bayesian network representation of the proposed POMDP-IR model. In this figure we assume $m \equiv k$ for simplicity, i.e., each person has only one feature

We assume a random motion pattern for each person, as we do not have prior knowledge about the person's intended path, in which the person can either stay in its current node with probability 0.6, or move to neighboring nodes with the remaining probability mass split equally among them. Such a representation allows us to model movement constraints posed by the environment (for instance, corridors, walls or other obstacles). We also need to take into account the uncertainty in the robot movement due to possible errors during navigation or unexpected obstacles present in the environment. In this model when the robot moves either it arrives at its destination with probability 0.6 or stays in the same node with probability 0.4. The value of the feature nodes $F_1, \ldots, F_m$ have a low probability of change (0.01), as it is unlikely that a particular person's characteristics changes. For each state variable we define a set of observations. Each observation $o_1^p \in O_1^P$ through $o_k^p \in O_k^P$ and $o^y \in O^Y$ indicates an observation of a person or robot close to a corresponding $p_k \in P_K$ resp. $y \in Y$. The binary observations $o_1^f \in O_1^F$ through $o_m^f \in O_m^F$ indicate whether a particular feature is observed.

The key to cooperative perception lies in the observation model for detecting features. The false negative and false positive rates are different at each location, depending on conditions

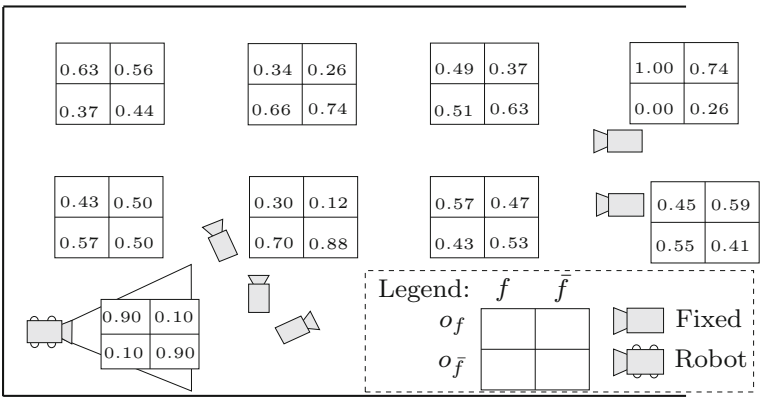

Fig. 8 Learned observation model for fixed cameras and predefined model for robot camera. Each *matrix* represents $p(O^F|F)$ at a node in the map, except for the robot camera

such as the position of sensors, their field of view, lighting, etc. For detecting certain features, mobile sensors have a higher accuracy than fixed sensors, although with a smaller field of view. Therefore, the observation model differs with respect to person and robot location. In particular, if a person is observed by the robot (illustrated in Fig. 6c) the probability of false negatives $P(O^F = o^{\bar{f}}|F = f)$ or false positives $P(O^F = o^f|F = \bar{f})$ is low. This is an important issue in decision making, as the presence of a mobile sensor will be more valuable in areas where fixed sensors cannot provide high accuracy. In Fig. 8 we show the uncertainty related to observations for each node considered. Note that the observation uncertainty has been estimated for the fixed sensors but we assume a constant observation model for the mobile sensor.

This concludes the basis for all models used in our case study. In the following, we will present two different ways of formalizing information-gain actions. First we present simulation results using the POMDP-IR framework, followed by real-robot results in a slightly modified model.

6.2 Simulation experiments with information rewards

Here we apply the POMDP-IR framework to the robot-assisted surveillance problem, showcasing its behavior in larger and more realistic problem domains. In this problem there are no concurrent goals, unlike the PATROL problem, where we had to perform information gain while performing other tasks. Therefore, in this model we include two possible sets of information-reward actions, $A_1 = \{commit_1, null\}$, $A_2 = \{commit_2, null\}$.[2] A_1 encodes the action that the person is wearing red, while A_2 relates to the opposite (asserting that state factor F_1 is false).

We present in Fig. 9 some experiments with threshold $\beta = 0.75$ (Exp. A and Exp. B) and $\beta = 0.6$ (Exp. C and Exp. D). For each experiment, we show the robot's path, initial position of robot and person, cumulative reward, and the evolution of the belief over the feature of interest.

Exp. A presents a case in which a person wearing red is detected in an area with low uncertainty. We note that the belief increases rapidly, and the robot does not need to approach

[2] An alternative option would be to implement the extension described in Sect. 4.4.

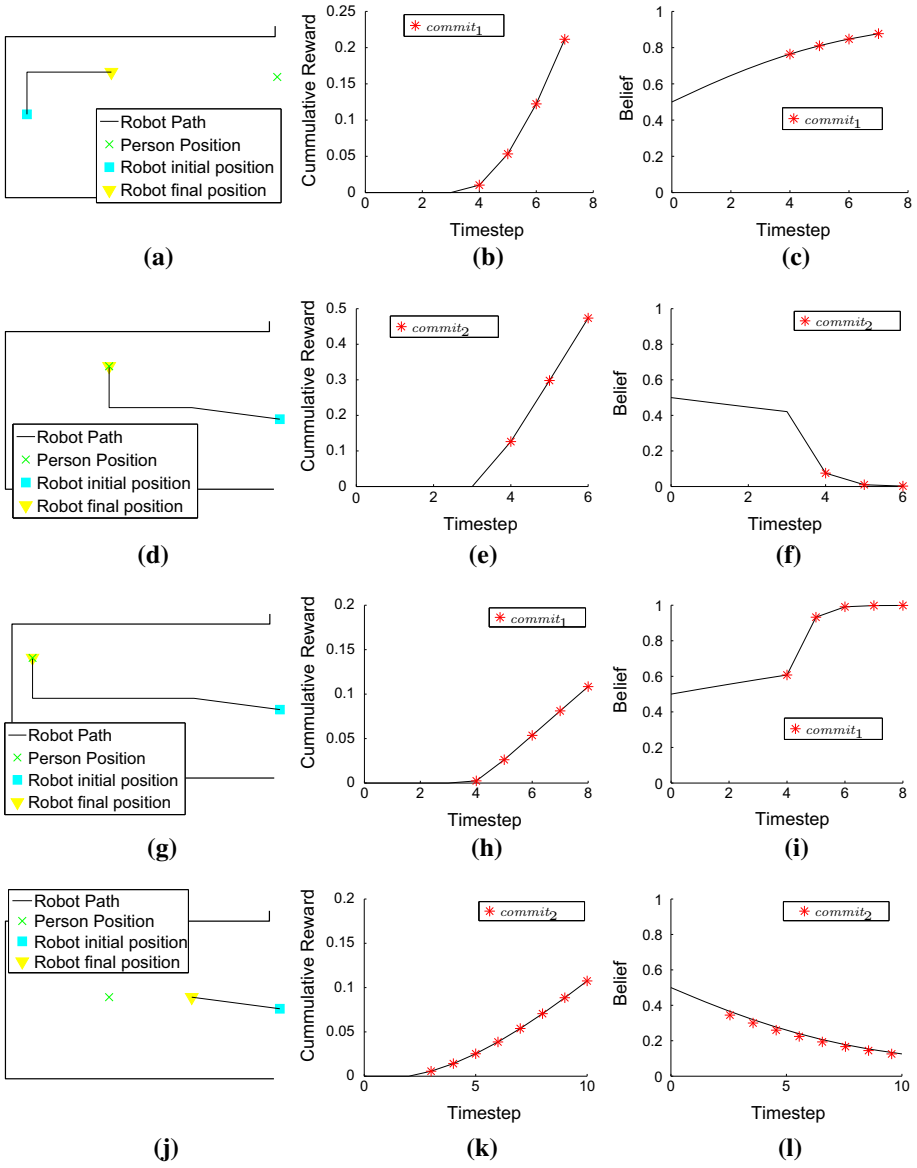

Fig. 9 POMDP-IR simulation results for the case study. Experiments A and B: $\beta = 0.75$. Experiments C and D: $\beta = 0.6$. Action $commit_1$ corresponds to asserting that the person is wearing *red* while action $commit_2$ asserts the opposite. **a** Exp. A: Robot's path, **b** Exp. A: $\sum_t \gamma^t r_t$, **c** Exp. A: b_i^t $(F_1 = red)$, **d** Exp. B: Robot's path, **e** Exp. B: $\sum_t \gamma^t r_t$, **f** Exp. B: b_i^t $(F_1 = red)$, **g** Exp. C: Robot's path, **h** Exp. C: $\sum_t \gamma^t r_t$, **i** Exp. C: b_i^t $(F_1 = red)$, **j** Exp. D: Robot's path, **k** Exp. D: $\sum_t \gamma^t r_t$, **l** Exp. D: b_i^t $(F_1 = red)$ (Color figure online)

the person for the system to have enough information to start applying information-reward actions. The belief crosses the threshold at time step 4, and from that moment on the $commit_1$ action is applied, indicating that the system decides to classify this person as wearing red. The cumulative reward does not have a linear shape as the reward for the information-reward action is higher as the belief increases.

Exp. B shows a case where a person is detected not wearing red in an area with higher uncertainty. Therefore, the system decides to move the robot near the person to observe what is happening before executing $commit_2$ actions. In both experiments with $\beta = 0.75$ we see that, in fact, the threshold is respected, and *commit* actions are taken only when the belief exceeds the specified value.

In the case in which $\beta = 0.6$ the system takes into account the different threshold. In Exp. C a person wearing red is detected in a higher uncertainty area. The robot moves towards the person for better identification, but from the moment when the belief crosses the threshold, the action $commit_1$ is chosen. In Exp. D the system detects the person without red clothing (action $commit_2$) and the robot moves but stops half way (c.f. Exp. B).

6.3 Real-robot experiments

Next, in order to show the applicability of our ideas to real-world applications, we present a set of real-robot experiments. They are based on a model that is similar to the POMDP-IR framework in spirit [55]. The main difference is that instead of considering multiple action factors, the model extends the domain-level action space with classification actions, which fulfill the same role as the *commit* actions in the POMDP-IR model. This means that the robot has to choose between moving and announcing that an event has been classified. The POMDP-IR framework, on the other hand, clearly separates domain-level actions from information-reward actions. Also, the model includes m bookkeeping variables, keeping track of which features have already been classified or not, and a movement cost is included. Figure 10 shows some results obtained in this context.

In Exp. E, a person is detected by the camera network as wearing red. Since the observation model in this particular area is error prone, the system sends the robot near the person to more rapidly gain information. This happens at time step 20 when the robot arrives and observes the person's features. On the other hand, in Exp. F, we see what happens when the person is not wearing red. Again, the robot approaches the person, but since the observations received in the meantime are consistent the robot stops so as not to waste resources.

We also consider more complex scenarios with 2 persons instead of a single one. The system must reason whether to classify each one, and if the uncertainty is high and the robot needs to check on them, in which order to do so. In Exp. G, one person is detected in a low-uncertainty area and the other one in a region where uncertainty is higher. Note that $b_i^t(F_2 = red)$ increases faster than $b_i^t(F_1 = red)$. Therefore, it is not necessary to check on person 2, but rather on person 1, and the robot navigates in its direction and uncertainty decreases when the robot confirms the person's feature. Meanwhile, the system has received enough information to classify the other person as well.

So far, we have assumed that all locations and people have equal priorities. However, another interesting scenario is when an area in the environment requires special attention. In Exp. H, persons are detected in the same areas as in Exp. G but each location has a different priority, as encoded in the reward function: the reward for classifying person 2 is double the reward of person 1. Although person 2 is in a low uncertainty area, the robot goes to check on it first.

6.4 Discussion

The case study presented shows an example of a real-world problem where information gain plays an important role. In general, the behavior of the system is as expected, as both in simulation as well as in reality system tries to optimize the robot behavior in order to detect

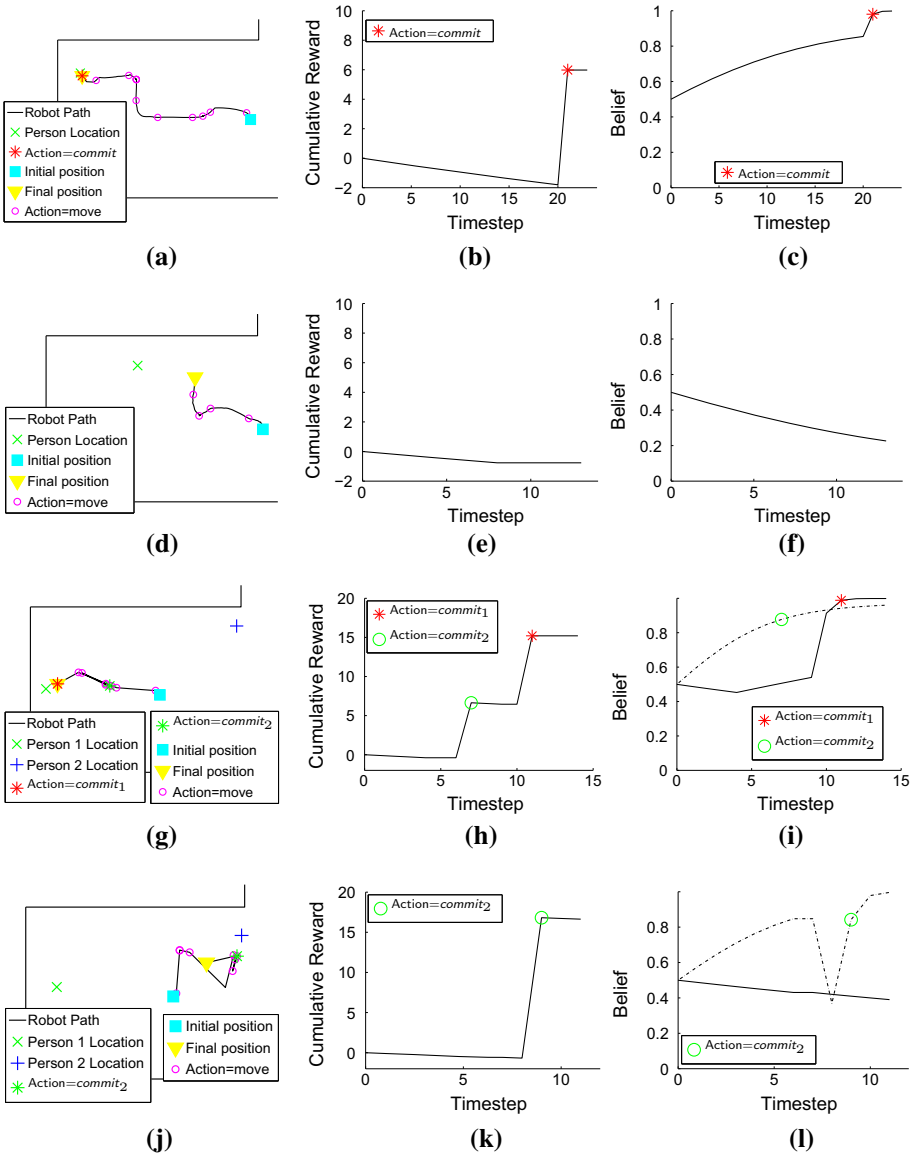

Fig. 10 Real-robot experimental results. Experiments E and F: 1 person. Experiments G and H: 2 persons. Actions $commit_1$ and $commit_2$ correspond to asserting that person 1 resp. 2 is wearing *red*. **a** Exp. E: Robot's path, **b** Exp. E: $\sum_t \gamma^t r_t$, **c** Exp. E: b_i^t $(F_1 = red)$, **d** Exp. F: Robot's path, **e** Exp. F: $\sum_t \gamma^t r_t$, **f** Exp. F: b_i^t $(F_1 = red)$, **g** Exp. G: Robot's path, **h** Exp. G: $\sum_t \gamma^t r_t$, **i** Exp. G: b_i^t $(F_1 = red)$, (*solid*) and b_i^t $(F_2 = red)$ (*dashed*), **j** Exp. H: Robot's path, **k** Exp. H: $\sum_t \gamma^t r_t$, **l** Exp. H: b_i^t $(F_1 = red)$, (*solid*) and b_i^t $(F_2 = red)$ (*dashed*) (Color figure online)

information in the system. When the networked cameras do not provide enough information the controller asks the robot to move towards the person.

However, we note some advantages of the POMDP-IR approach compared to the model in Sect. 6.3. The robot is free to keep moving, independently of whatever information-reward

action the system decides to take, which may be useful in problems where person or object tracking is crucial. Furthermore, we are able to better define the level of uncertainty we want to tolerate by explicitly computing the information rewards needed to reach a particular belief threshold.

7 Related work

In this section, we present an overview of the relevant literature. We discuss related work in applying POMDPs to robotic applications, followed by a discussion of how active sensing relates to our contributions. Finally, we discuss how the POMDP-IR framework compares to related POMDP models for rewarding information gain.

7.1 POMDPs in a robotic context

Techniques for single-agent decision-theoretic planning under uncertainty such as POMDPs are being applied more and more to robotics [60]. Over the years, there have been numerous examples demonstrating how POMDPs can be used for mobile robot localization and navigation [43,47]. Emery-Montemerlo et al. [16] demonstrate the viability of approximate multiagent POMDP techniques for controlling a small group of robots to catch an intruder, while more recently Amato et al. [2] use similar frameworks in larger multi-robot domains to perform cooperative tasks. Capitán et al. [13] show how POMDP task auctions can be used for multi-robot target tracking applications. These types of applications are potentially suitable for the POMDP-IR framework, as they typically involve reasoning about the belief regarding the target's location.

A relevant body of work exists on systems interacting with humans driven by POMDP-based controllers. Fern et al. [17] propose a POMDP model for providing assistance to users, in which the goal of the user is a hidden variable which needs to be inferred. POMDP-based models have been applied to a real-world domain to assist people with dementia, in which users receive verbal assistance while washing their hands [7], and to high-level control of a robotic assistant designed to interact with elderly people [35,42]. Merino et al. [30] develop a real-time robust person guidance framework using POMDP policies which includes social behaviors and robot adaptation by integrating social feedback. Interacting with humans whose goals and objectives might need to be estimated provides opportunities for the POMDP-IR model.

7.2 Active sensing

Information gain has been studied in the active sensing framework [31], which can be formalized as acting so as to acquire knowledge about certain state variables.

A large amount of research effort has been put in approaches to robot localization using active methods. Burgard et al. [11] propose an active localization approach providing rational criteria for setting the robot's motion direction and determining the pointing direction of the sensors so as to most efficiently localize the robot, while Roy et al. [41] use environment information to minimize uncertainty in navigation. Velez et al. [59] add informative views of objects to regular navigation and task-completion objectives. Another aspect which may be included in active sensing is where to position sensors to maximize the level of information of observations. Krause and Guestrin [24] and Krause et al. [26] exploit submodularity to efficiently trade off observation informativeness and cost of acquiring information, while

Krause et al. [25] apply such methods to sensor placement in large water distribution networks. More recently, Natarajan et al. [32] maximize observing multiple targets in a multi-camera surveillance scenario.

Active sensing problems typically consider acting and sensing under uncertainty, hence POMDPs offer a natural solution to model such problems. Work on active sensing using POMDPs includes collaboration between mobile robots to detect static objects [62] and combining a POMDP approach with information-theoretic heuristics to reduce uncertainty on goal position and probability of collision in robot navigation [12]. Finally, Spaan and Lima [52] consider objectives such as maximizing coverage or improving localization uncertainty when dynamically selecting a subset of image streams to be processed simultaneously.

7.3 POMDP frameworks for rewarding information gain

Multi-objective POMDP formulations which include information gain have been proposed. Mihaylova et al. [31] review some criteria for uncertainty minimization in the context of active sensing. They consider a multi-objective setting, linearly trading off expected information extraction with expected costs and utilities. Similar to our setting, the system designer is responsible for balancing the two objectives.

Using the belief-MDP formulation, it is possible to directly define a reward on beliefs. While we may assume this function to be convex, which is a natural property for information measures, there is still the need to approximate it by a PWLC function [4]. Thus, we cannot directly model non-linear rewards as belief-based without approximating such functions.

Eck and Soh [15] introduce hybrid rewards, which combine the advantages of both state and belief-based rewards, using state-based rewards to encode the costs of sensing actions and belief-based rewards to encode the benefits of sensing. Such a formulation lies out of a traditional solver's scope. While we also consider multi-objective problems, solving our models is independent of which solver is used. We are interested in an approach that while staying in the standard POMDP framework, and thus with PWLC value functions, is still able to perform multi-objective problems, where only some tasks are related to information gain. We do that without using purely belief-based rewards, such that our framework can be plugged in any traditional solver. Furthermore we are able to impose thresholds on the amount of uncertainty we desire for particular features in the environment.

Similar to our *commit* actions, Williams and Young [61] propose the use of *submit* actions in slot-filling POMDPs, in which a spoken dialog system must disambiguate the internal state of the user it is interacting with. In this framework there is no clear separation between domain-level actions and information-reward actions like in the POMDP-IR framework. The *submit* actions lead to an absorbing state, which indicates a final goal for the system. Our system is intended to be parallel to normal task execution, hence the *commit* actions do not have any effect on the state of the environment. Furthermore, in contrast to Williams and Young [61], we provide guidance on how such actions should be rewarded.

8 Conclusions

In this paper we presented a modeling approach to deal with information gain in active cooperative perception problems, which involve cooperation between several sensors and a decision maker. We based our approach on POMDPs due to their natural way of dealing with uncertainty, but we faced the problem of how to build

a POMDP which explicitly performs information gain. There are several ways to reward information gain, such as using the negative belief entropy. Although resulting value functions are shown to remain PWLC if the reward is PWLC [4], that is not generally the case with belief-based rewards such as negative entropy. Moreover, we are interested in modeling multi-objective problems where the system must complete a set of different tasks, only part of which might be related to direct information gain. However, that can be cumbersome with a belief-dependent reward implementation.

We proposed a new framework, POMDP-IR, which builds on the classic POMDP framework, and extends its action space with actions that return information rewards. These rewards depend only on a particular state factor, and are defined on states, not beliefs. However, their definition ensures that the reward for information-reward actions is positive only if the belief regarding the state factor is above a threshold. In this way, we stay inside the classic POMDP framework and can use solvers that rely on PWLC value functions, but we are still able to model certain information-gain tasks.

We illustrated our framework on a toy problem and we showed that it compares favorably with the ρPOMDP model on benchmark domains. Furthermore, we included a case study on robot surveillance, demonstrating how our approach behaves in such environments. Besides the fact that they represent more realistic scenarios, they are challenging as they include multiple sensors, both active and passive ones, in a multi-objective scenario.

8.1 Future work

As two main avenues of future work we discuss tailoring POMDP solvers to our framework as well as considering multiple decision makers.

In the current experiments, we used an off-the-shelf factorized point-based solver to compute POMDP policies [37]. A particular point of interest is improving scalability with respect to the number of observations, which is the traditional bottleneck in POMDP solving. It is particularly pressing in ACP scenarios given the large number of sensors and we could extend value function approximation for factored POMDPs to factored observation spaces [18,58]. Furthermore, in our approach we increase the action space size and methods that address this particular issue could be adapted to our context. For instance, Agrawal et al. [1] compute vector updates in point-based methods using mixed linear integer programs when the problem has large or continuous action and observation spaces and Scharpff et al. [45] avoid exponentially sized action spaces by dividing each time step into multiple stages. It is also promising to explore other types of solution techniques, for instance ones that do not (approximately) solve models off-line. By defining POMDP-IR models dynamically and by using on-line POMDP solvers [40,46], we can overcome the inherent limitations of off-line solvers.

In this work we assumed only one active sensor in the environment. However, we can easily imagine problems which extend to multiple decision makers, e.g., surveillance scenarios with several robots or active cameras. A requirement for treating (parts of) the system as a centralized POMDP is fast and reliable communication, as cameras and robots need to share local observations [38]. When communication delays are limited and potentially stochastic, the problem can be modeled as a multiagent POMDP with delayed communication [33,54]. Finally, when no communication channel is present, the problem can be modeled as a decentralized POMDP [6,34]. Extending the POMDP-IR framework to any of these multiagent models is promising.

Acknowledgments This work was partially supported by Fundação para a Ciência e a Tecnologia (FCT) through grant SFRH/BD/70559/2010 (T.V.), as well as by FCT ISR/LARSyS strategic funding PEst-OE/EEI/LA0009/2013, and the FCT project CMU-PT/SIA/0023/2009 under the Carnegie Mellon-Portugal Program. We thank Shimon Whiteson for useful discussions.

References

1. Agrawal, R., Realff, M. J., & Lee, J. H. (2013). MILP based value backups in partially observed Markov decision processes (POMDPs) with very large or continuous action and observation spaces. *Computers & Chemical Engineering*, *56*, 101–113.
2. Amato, C., Konidaris, G., Cruz, G., Maynor, C. A., How, J. P., & Kaelbling, L. P. (2014). Planning for decentralized control of multiple robots under uncertainty. In *ICAPS-14 Workshop on Planning and Robotics*.
3. Araya-López, M. (2013). *Des algorithmes presque optimaux pour les problémes de décisione séquentielle à des fins de collecte d'information*. PhD thesis, University of Lorraine.
4. Araya-López, M., Buffet, O., Thomas, V., & Charpillet, F. (2010). A POMDP extension with belief-dependent rewards. In *Advances in Neural Information Processing Systems*, Vol. 23.
5. Barbosa, M., Bernardino, A., Figueira, D., Gaspar, J., Gonçalves, N., Lima, P. U., Moreno, P., Pahliani, A., Santos-Victor, J., Spaan, M. T. J., & Sequeira, J. (2009). ISRobotNet: A testbed for sensor and robot network systems. In *Proceedings of International Conference on Intelligent Robots and Systems*.
6. Bernstein, D. S., Givan, R., Immerman, N., & Zilberstein, S. (2002). The complexity of decentralized control of Markov decision processes. *Mathematics of Operations Research*, *27*(4), 819–840.
7. Boger, J., Poupart, P., Hoey, J., Boutilier, C., Fernie, G., & Mihailidis, A. (2005). A decision-theoretic approach to task assistance for persons with dementia. In*Proceedings of International Joint Conference on Artificial Intelligence*.
8. Boutilier, C., & Poole, D. (1996). Computing optimal policies for partially observable decision processes using compact representations. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*.
9. Boyen, X., & Koller, D. (1998). Tractable inference for complex stochastic processes. In *Proceedings of Uncertainty in Artificial Intelligence*.
10. Brunskill, E., Kaelbling, L., Lozano-Perez, T., & Roy, N. (2008). Continuous-state POMDPs with hybrid dynamics. In *Proceedings of the International Symposium on Artificial Intelligence and Mathematics*.
11. Burgard, W., Fox, D., & Thrun, S. (1997). Active mobile robot localization by entropy minimization. In *Proceedings of the Second Euromicro Workshop on Advanced Mobile Robots*.
12. Candido, S., & Hutchinson, S. (2011). Minimum uncertainty robot navigation using information-guided POMDP planning. In *Proceedings of the International Conference on Robotics and Automation*.
13. Capitán, J., Spaan, M. T. J., Merino, L., & Ollero, A. (2013). Decentralized multi-robot cooperation with auctioned POMDPs. *International Journal of Robotics Research*, *32*(6), 650–671.
14. Doshi, F., & Roy, N. (2008). The permutable POMDP: Fast solutions to POMDPs for preference elicitation. In *Proceedings of International Conference on Autonomous Agents and Multi Agent Systems*.
15. Eck, A., & Soh, L.-K. (2012). Evaluating POMDP rewards for active perception. In *Proceedings of International Conference on Autonomous Agents and Multi Agent Systems*.
16. Emery-Montemerlo, R., Gordon, G., Schneider, J., & Thrun, S. (2005). Game theoretic control for robot teams. In *Proceedings of the International Conference on Robotics and Automation*.
17. Fern, A., Natarajan, S., Judah, K., & Tadepalli, P. (2007). A decision-theoretic model of assistance. In *Proceedings of the International Conference on Artificial Intelligence*.
18. Guestrin, C., Koller, D., & Parr, R. (2001). Solving factored POMDPs with linear value functions. In *IJCAI-01 Workshop on Planning under Uncertainty and Incomplete Information*.
19. Guo, A. (2003). Decision-theoretic active sensing for autonomous agents. In *Proceedings of the International Conference on Computational Intelligence, Robotics and Autonomous Systems*.
20. Hansen, E. A., & Feng, Z. (2000). Dynamic programming for POMDPs using a factored state representation. In *International Conference on Artificial Intelligence Planning and Scheduling*.
21. Hsu, D., Lee, W., & Rong, N. (2008). A point-based POMDP planner for target tracking. In *Proceedings of the International Conference on Robotics and Automation*.
22. Ji, S., Parr, R., & Carin, L. (2007). Non-myopic multi-aspect sensing with partially observable Markov decision processes. *IEEE Transactions on Signal Processing*, *55*(6), 2720–2730.
23. Kaelbling, L. P., Littman, M. L., & Cassandra, A. R. (1998). Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, *101*, 99–134.

24. Krause, A., & Guestrin, C. (2007). Near-optimal observation selection using submodular functions. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence*.
25. Krause, A., Leskovec, J., Guestrin, C., Vanbriesen, J., & Faloutsos, C. (2008). Efficient sensor placement optimization for securing large water distribution networks. *Journal of Water Resources Planning and Management*, *134*(6), 516–526.
26. Krause, A., Singh, A., & Guestrin, C. (2008). Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, *9*, 235–284.
27. Krishnamurthy, V., & Djonin, D. (2007). Structured threshold policies for dynamic sensor scheduling—A partially observed Markov decision process approach. *IEEE Transactions on Signal Processing*, *55*(10), 4938–4957.
28. Littman, M. L., Cassandra, A. R., & Kaelbling, L. P. (1995). Learning policies for partially observable environments: Scaling up. In *International Conference on Machine Learning*.
29. Martinez-Cantin, R., de Freitas, N., Brochu, E., Castellanos, J., & Doucet, A. (2009). A Bayesian exploration–exploitation approach for optimal online sensing and planning with a visually guided mobile robot. *Autonomous Robots*, *27*, 93–103.
30. Merino, L., Ballesteros, J., Pérez-Higueras, N., Ramón-Vigo, R., Pérez-Lara, J., & Caballero, F. (2014). Robust person guidance by using online POMDPs. In *ROBOT2013: First Iberian Robotics Conference. Advances in intelligent systems and computing* (Vol. 253). Springer.
31. Mihaylova, L., Lefebvre, T., Bruyninckx, H., Gadeyne, K., & De Schutter, J. (2003). A comparison of decision making criteria and optimization methods for active robotic sensing. In *Numerical methods and applications. LNCS* (Vol. 2543). Springer.
32. Natarajan, P., Hoang, T. N., Low, K. H., & Kankanhalli, M. (2012). Decision-theoretic approach to maximizing observation of multiple targets in multi-camera surveillance. In *Proceedings of International Conference on Autonomous Agents and Multi Agent Systems*.
33. Oliehoek, F. A., & Spaan, M. T. J. (2012). Tree-based pruning for multiagent POMDPs with delayed communication. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*.
34. Oliehoek, F. A., Spaan, M. T. J., & Vlassis, N. (2008). Optimal and approximate Q-value functions for decentralized POMDPs. *Journal of Artificial Intelligence Research*, *32*, 289–353.
35. Pineau, J., Montemerlo, M., Pollack, M., Roy, N., & Thrun, S. (2003). Towards robotic assistants in nursing homes: Challenges and results. *Robotics and Autonomous Systems*, *42*(3–4), 271–281.
36. Porta, J. M., Vlassis, N., Spaan, M. T. J., & Poupart, P. (2006). Point-based value iteration for continuous POMDPs. *Journal of Machine Learning Research*, *7*, 2329–2367.
37. Poupart, P. (2005). *Exploiting structure to efficiently solve large scale partially observable Markov decision processes*. PhD thesis, University of Toronto.
38. Pynadath, D. V., & Tambe, M. (2002). The communicative multiagent team decision problem: Analyzing teamwork theories and models. *Journal of Artificial Intelligence Research*, *16*, 389–423.
39. Roijers, D., Vamplew, P., Whiteson, S., & Dazeley, R. (2013). A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research*, *48*, 67–113.
40. Ross, S., Pineau, J., Paquet, S., & Chaib-draa, B. (2008). Online planning algorithms for POMDPs. *Journal of Artificial Intelligence Research*, *32*, 664–704.
41. Roy, N., Burgard, W., Fox, D., & Thrun, S. (1999). Coastal navigation—Mobile robot navigation with uncertainty in dynamic environments. In *Proceedings of the International Conference on Robotics and Automation*.
42. Roy, N., Gordon, G., & Thrun, S. (2003). Planning under uncertainty for reliable health care robotics. In *Proceedings of the International Conference on Field and Service Robotics*.
43. Roy, N., Gordon, G., & Thrun, S. (2005). Finding approximate POMDP solutions through belief compression. *Journal of Artificial Intelligence Research*, *23*, 1–40.
44. Sanfeliu, A., Andrade-Cetto, J., Barbosa, M., Bowden, R., Capitán, J., Corominas, A., et al. (2010). Decentralized sensor fusion for ubiquitous networking robotics in urban areas. *Sensors*, *10*(3), 2274–2314.
45. Scharpff, J., Spaan, M. T. J., Volker, L., & de Weerdt, M. M. (2013). Planning under uncertainty for coordinating infrastructural maintenance. In *Proceedings of the International Conference on Automated Planning and Scheduling*.
46. Silver, D., & Veness, J. (2010). Monte-Carlo planning in large POMDPs. In *Advances in Neural Information Processing Systems*, Vol. 23.
47. Simmons, R., & Koenig, S. (1995). Probabilistic robot navigation in partially observable environments. In *Proceedings of the International Joint Conference on Artificial Intelligence*.
48. Singh, S. S., Kantas, N., Vo, B.-N., Doucet, A., & Evans, R. J. (2007). Simulation-based optimal sensor scheduling with application to observer trajectory planning. *Automatica*, *43*(5), 817–830.

49. Smith, T., & Simmons, R. (2004). Heuristic search value iteration for POMDPs. In *Proceedings of Uncertainty in Artificial Intelligence*.
50. Spaan, M. T. J. (2008). Cooperative active perception using POMDPs. In *AAAI 2008 Workshop on Advancements in POMDP Solvers*.
51. Spaan, M. T. J. (2012). Partially observable Markov decision processes. In M. Wiering & M. van Otterlo (Eds.), *Reinforcement learning: State of the art*. Berlin: Springer.
52. Spaan, M. T. J., & Lima, P. U. (2009). A decision-theoretic approach to dynamic sensor selection in camera networks. In *Proceedings of International Conference on Automated Planning and Scheduling*.
53. Spaan, M. T. J., & Vlassis, N. (2005). Perseus: Randomized point-based value iteration for POMDPs. *Journal of Artificial Intelligence Research*, *24*, 195–220.
54. Spaan, M. T. J., Oliehoek, F. A., & Vlassis, N. (2008). Multiagent planning under uncertainty with stochastic communication delays. In *Proceedings of International Conference on Automated Planning and Scheduling*.
55. Spaan, M. T. J., Veiga, T. S., & Lima, P. U. (2010). Active cooperative perception in network robot systems using POMDPs. In *Proceedings of International Conference on Intelligent Robots and Systems*.
56. Stachniss, C., Grisetti, G., & Burgard, W. (2005). Information gain-based exploration using Rao-Blackwellized particle filters. In *Proceedings of Robotics: Science and Systems*.
57. Thrun, S., Burgard, W., & Fox, D. (2005). *Probabilistic robotics*. Cambridge: MIT Press.
58. Veiga, T. S., Spaan, M. T. J., & Lima, P. U. (2014). Point-based POMDP solving with factored value function approximation. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*.
59. Velez, J., Hemann, G., Huang, A. S., Posner, I., & Roy, N. (2011). Planning to perceive: Exploiting mobility for robust object detection. In *Proceedings of International Conference on Automated Planning and Scheduling*.
60. Vlassis, N., Gordon, G., & Pineau, J. (2006). Planning under uncertainty in robotics. *Robotics and Autonomous Systems, 54*(11). Special issue.
61. Williams, J. D., & Young, S. (2007). Partially observable Markov decision processes for spoken dialog systems. *Computer Speech and Language*, *21*(2), 393–422.
62. Zhang, S., & Sridharan, M. (2012). Active visual sensing and collaboration on mobile robots using hierarchical POMDPs. In *Proceedings of International Conference on Autonomous Agents and Multi Agent Systems*.