In *International Conference on Development and Learning* (ICDL-EpiRob),
San Diego, USA, November 7-9, 2012.

ASP+POMDP: Integrating Non-Monotonic Logic Programming and Probabilistic Planning on Robots

Shiqi Zhang and Mohan Sridharan
Stochastic Estimation and Autonomous Robotics Lab
Department of Computer Science
Texas Tech University, USA.
Email: {s.zhang, mohan.sridharan}@ttu.edu

Forrest Sheng Bao
Knowledge Representation Lab
Department of Computer Science
Texas Tech University, USA.
Email: forrest.bao@gmail.com

***Abstract*—Mobile robots equipped with multiple sensors and deployed in real-world domains frequently find it difficult to process all sensor inputs, or to operate without any human input and domain knowledge. At the same time, robots cannot be equipped with all relevant domain knowledge in advance, and humans are unlikely to have the time and expertise to provide elaborate and accurate feedback. This paper presents a novel framework that addresses these challenges by integrating high-level logical inference with low-level probabilistic sequential decision-making. Specifically, Answer Set Programming (ASP), a non-monotonic logic programming paradigm, is used to represent, reason with and revise domain knowledge obtained from sensor inputs and high-level human feedback, while hierarchical partially observable Markov decision processes (POMDPs) are used to automatically adapt visual sensing and information processing to the task at hand. Furthermore, a psychophysics-inspired strategy is used to merge the output of logical inference with probabilistic beliefs. All algorithms are evaluated in simulation and on wheeled robots localizing target objects in indoor domains.**

I. INTRODUCTION

Sophisticated learning, planning and control algorithms have enabled the use of mobile robots and agents in domains such as disaster rescue, reconnaissance and health care. Real-world domains characterized by partial observability, non-deterministic action outcomes and unforeseen dynamic changes frequently make it difficult for robots to process all sensor inputs, model the entire domain or operate without substantial domain knowledge and human feedback. At the same time, robots cannot be provided all relevant domain knowledge in advance. In addition, although human feedback can provide rich information about task and domain, humans are unlikely to have the time and expertise to provide elaborate and accurate feedback in complex domains. Information extracted from sensory cues and human feedback may also have different degrees of relevance to current or future tasks. Widespread deployment of intelligent robots and agents in real-world domains thus poses some formidable challenges—robots need to represent, reason with and revise domain knowledge; automatically adapt sensing and processing to the task at hand; and learn from high-level human feedback.

Partially observable Markov decision processes (POMDPs), an instance of probabilistic sequential decision-making, have been used to plan sensing and navigation on robots by modeling the associated uncertainty [10], [21]. However, it is a challenge to include common sense knowledge obtained from sensor inputs or human feedback in a POMDP. On the other hand, although non-monotonic logic programming is well-suited for knowledge representation and logical inference, it is not appropriate for modeling the uncertainty in real-world sensing and navigation [8]. This paper presents a novel framework that integrates Answer Set Programming (ASP), a non-monotonic logic programming paradigm, with hierarchical POMDPs to make the following contributions:

- ASP enables robots to represent, reason with and revise spatial knowledge of the application domain (and domain objects), using online repositories and information extracted from sensory cues and human feedback.
- Building on our prior work on hierarchical POMDPs, robots are enabled to adapt sensing and information processing to the task at hand [21]. The entropy of POMDP beliefs is used to identify the need for human feedback.
- A psychophysics-inspired strategy enables robots to use logical facts representing current domain knowledge to probabilistically initialize and revise POMDP beliefs.

The framework is evaluated in simulation and on wheeled robots that use visual inputs, high-level human feedback and laser range data to localize objects in complex indoor domains.

II. RELATED WORK

Mobile robots frequently have to plan a sequence of sensing and information processing actions, e.g., for locating objects and interacting with humans. Many POMDP-based algorithms have been developed to plan sensing, navigation and interaction on robots [10], [12], [16], [21]. Algorithms have also been developed for deriving preconditions and effects of actions in relational POMDPs [18], and for exploiting first-order reasoning in relationally-specified POMDPs [17]. In parallel, common sense reasoning using knowledge bases or human feedback has significantly improved the performance of robots [4], [9]. However, using logical inference *and* probabilistic modeling of uncertainty to exploit sensor inputs and human feedback, continues to be a challenge on robots.

Research in classical planning has resulted in many sophisticated algorithms for logical reasoning and knowledge representation [9]. However, many of these algorithms require a significant amount of prior knowledge, or are unable to

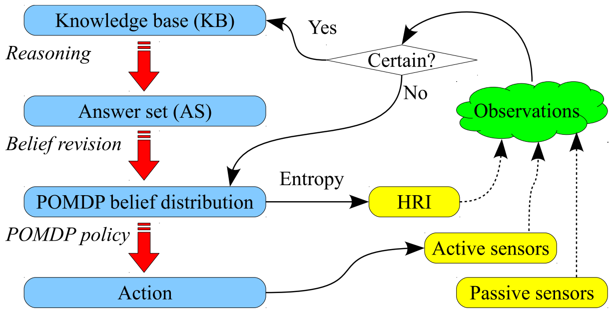


Fig. 1: Framework integrates probabilistic planning, non-monotonic logical reasoning and human-robot interaction.

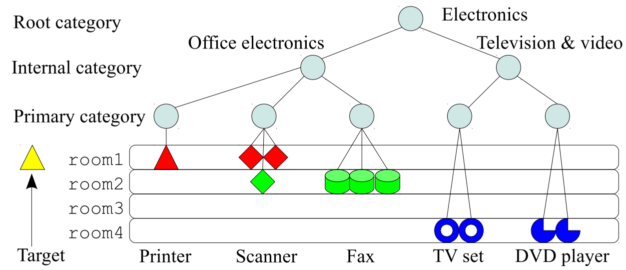


Fig. 2: Illustrative example of information about object categories stored in the knowledge base.

merge new (unreliable) information from sensors and humans with the current beliefs encoded in the knowledge base [6], [20]. Answer Set Programming (ASP), a non-monotonic logic programming paradigm, is well-suited for common sense knowledge representation and reasoning (especially *default reasoning*) [1], [8]. ASP has been used in different application domains and it has been integrated with natural language processing for service robots [4]. However, real-world sensing and navigation are non-deterministic, and humans are unlikely to provide elaborate and accurate feedback in complex domains. ASP is not well-equipped to model this uncertainty in sensing, navigation and interaction on mobile robots.

Many algorithms are being developed to integrate logical reasoning with probabilistic planning, e.g., the *switching planner* enables a robot to choose between logical reasoning and POMDPs for action selection [10]. Such a strategy, however, does not fully exploit the complementary properties of logical reasoning and POMDPs. Researchers have also combined deterministic and probabilistic algorithms for task and motion planning [13], while semantic maps and common sense knowledge about object positions have been used for target localization [11]. However, these algorithms use domain knowledge obtained from extensive human input or (generic) public resources (e.g., Internet [11]), which may not accurately reflect the specific task or domain. These algorithms are also typically unable to perform non-monotonic logical inference, where adding a new fact can reduce the set of (inferred) consequences. Therefore, integrating knowledge representation, (high-level) logical inference and probabilistic modeling of (low-level) uncertainty in sensing and navigation continues to be a formidable challenge for mobile robots. Our framework is a significant step towards addressing these challenges.

III. Problem Formulation

Figure 1 depicts our framework. The *Knowledge Base* (KB) in ASP contains causal rules and domain facts. Currently, rules are hand-coded and facts are learned from sensor inputs, human feedback and online repositories. For any specific query (or task), reasoning in the KB results in an *Answer Set* containing a set of grounded literals (Section III-A). The uncertainty in sensing and navigation is modeled using POMDP belief distributions (Section III-B). The answer sets from ASP initialize or revise POMDP belief distributions based on a psychophysics-inspired strategy (Section III-C1). Robots obtain observations from sensors activated when needed (e.g., cameras) and sensors that are always in operation (e.g., range finders). Observations made with high certainty update the KB, while other observations update POMDP belief distributions. Since human feedback is a valuable resource that is unreliable and not always available, human-robot interaction (HRI) is used when needed, e.g., if an object's location is known with considerable certainty, there is not much to gain by soliciting human help to locate the object. Robots therefore determine the need for human feedback based on entropy of POMDP belief distributions (Section III-C2). This paper illustrates the framework in the context of mobile robots localizing (i.e., computing the location of) target objects in indoor domains.

A. Knowledge Representation and Reasoning with ASP

Answer Set Programming (ASP) is a non-monotonic logic programming paradigm [1]. An ASP program is a collection of statements describing domain objects and relations between them [8]. An *answer set* is a set of ground literals that represent beliefs of an agent associated with the program. Program consequences are statements that are true in all such belief sets. ASP provides the ability to perform *default reasoning* using concepts such as *default negation* and *epistemic disjunction*, e.g., unlike "`¬ a`", "`not a`" only implies that "*a is not believed to be true*" and does not imply that "*a is believed to be false*". New information can hence be used to smoothly revise statements that are currently believed to be true.

The illustrative example of a robot localizing target objects can be reduced to finding answer sets for queries. The semantic (2D) domain description has the following elements: `room/1`, a space bounded by walls and doors that can be occupied by robots and objects; `object/1`, a visually identifiable element in a room; and `category/1`, a set of objects or sub-categories. Categories with objects as children are *primary categories*. A tree of object categories is created automatically from the KB—Figure 2 is an example for electronics. Information is extracted automatically from online repositories (e.g., *Amazon*) to identify some of the relationships between object categories. These relationships are used to create a subset of the tree, e.g., some of the nodes and links from root node to primary categories in Figure 2. Robots use sensor inputs and human feedback to add instances of objects in the KB and revise the tree, e.g., `room1` has a printer (shown as red triangle) and two scanners (red diamonds). All objects are visually distinguishable, including targets such as a new printer (yellow triangle) that is in `room1` unknown to the robot.

The following predicates represent some relations between elements: (1) `is(X,C)` implies category `C` is an ancestor of object or category `X`, e.g., `is(tv, electronics)`; (2) `observed(O,R,S)` implies that object `O` is observed in room `R` at timestep `S`; (3) `located(C,R,S)` implies that object(s) of category `C` can be (inferred) in room `R` at timestep `S`; and (4) `location(R,X,Y)` provides the `(X,Y)` coordinates of the center of room `R` in a learned domain map.

The following rules are used for reasoning: (1) if object `O` of category `C` is observed in room `R`, it is believed that objects of category `C` can be in `R`; (2) if objects of category `C` can be in room `R`, objects of the parent category (and all ancestor categories) of `C` can be in `R`; and (3) (rules of inertia) an object retains its location until it is known to be elsewhere and a room remains accessible until it is known to inaccessible.

```
   located(C,R,S) :- observed(O,R,S), is(O,C)
  located(C1,R,S) :- located(C2,R,S), is(C2,C1)
observed(O,R1,S+1):- observed(O,R1,S),
             not observed(O,R2,S+1), R1 != R2
 accessible(R,S+1):- accessible(R,S),
                    not ¬ accessible(R,S+1)
```

Consider the following illustrative example of non-monotonic reasoning in ASP:

- *Test-case 1* has the following facts:
  ```
  step(1..2). observed(printer1, lab, 1).
  is(printer1, printer).
  ```
  Reasoning in ASP produces the following answer set (existing facts are not repeated):
  ```
  observed(printer1, lab, 2).
  located(printer, lab, 1).
  located(printer, lab, 2).
  ```
- Now consider *Test-case 2* that has a new fact about an object's current location:
  ```
  step(1..2). observed(printer1, lab, 1).
  is(printer1, printer).
  observed(printer1, office, 2).  % new fact
  ```
  Reasoning in ASP now produces the following new answer set (existing facts not repeated):
  ```
  located(printer, lab, 1).
  located(printer, office, 2).
  ```

Adding a new fact has thus reduced the set of consequences and revised the outcome of the previous inference step—see Baral [1] for more details on ASP. We use the Clingo grounder and solver [7] to solve ASP programs.

B. Uncertainty Modeling with POMDP

Let us assume that ASP has provided candidate locations for a target object in a learned map of an office. The robot now has to move and analyze a sequence of images of a sequence of scenes. This objective is posed as a planning task and addressed using our prior work on hierarchical POMDPs for reliable and efficient visual sensing and information processing on robots [21]. This hierarchy is briefly summarized below.

The high-level (HL) POMDP determines the sequence of 3D scenes to process to locate the target. The 3D area is represented as a discrete 2D *occupancy grid*. Each entry of the state vector corresponds to the event that the target is in the corresponding grid cell. To estimate the state, a probability distribution of target occurrence is maintained over the states, called the *belief state*. Uncertainty in belief is measured by computing the entropy:

$$\mathscr{H}(B_t) = -\sum_{i=1}^{N} b_{i,t} log(b_{i,t}) \tag{1}$$

where $b_{i,t}$ is the i^{th} entry of belief state at time t. The reward of action a_t is defined as the reduction in entropy between belief state B_{t-1} and the resultant belief state B_t. The robot learns an observation function to model the probability of target detection as a function of robot position, target position, camera's field of view and lower levels of the hierarchy. A policy gradient solver [2] is then used to compute a *policy* that maps belief states to actions by minimizing entropy over a planning horizon. The number of grid cells can increase exponentially and change arbitrarily in real-world domains, making real-time solutions difficult. The robot hence learns a *convolutional policy kernel* from the policy for a small region, exploiting the rotation and shift invariance properties of visual search [3]. This kernel is used to automatically and efficiently generate policies for larger maps. Since movement between grid cells expends time and introduces errors, movement is associated with a cost proportional to the distance to be traveled. The robot also improves computational efficiency by planning a path through grids that have a significantly higher probability than their immediate neighbors.

For any chosen scene, the remaining layers of the hierarchy plan the sequence of algorithms to be applied on a sequence of regions of interest (ROIs) in a sequence of images. Salient ROIs are extracted from each image of the scene and each ROI is modeled as a lower-level (LL) POMDP. Each LL policy provides the sequence of information operators (e.g., detect color) to apply on a specific ROI to detect the target object. LL policies of all image ROIs are used to automatically create an intermediate-level (IL) POMDP. Executing an action in the IL policy directs attention to a specific ROI. Executing the corresponding LL policy (until termination) provides an observation that causes an IL belief update and action choice until presence or absence of the target in the image is determined. This provides an HL observation and belief update, resulting in the robot choosing a scene for subsequent analysis. This process continues until the object is found or the belief does not converge over a period of time. The entire hierarchy adapts automatically to the task at hand—see [19], [21] for details.

C. Integrating ASP and POMDP

The ASP formulation (Section III-A) models domain knowledge and provides an answer set that represents the result of non-monotonic logical inference. The POMDP formulation models the uncertainty in sensing and navigation to adapt sensing and processing to any given task. This section describes a psychophysics-inspired strategy to convert answer sets to beliefs that initialize or revise POMDP beliefs. The entropy of POMDP beliefs is then used to identify the need for high-level human feedback, using information extracted from sensor inputs and human feedback to augment and revise the KB.

1) **Bias Generation and Belief Merging:** Merging the beliefs encoded by an answer set and a POMDP belief distribution proceeds in two steps: (1) a bias distribution is generated using *literals* in the answer set relevant to the current task; and (2) the bias distribution is merged with the POMDP belief distribution.

Bias Generation*:* The bias distribution is computed using the object categories in the KB and the following hypotheses that capture co-occurrence relationships between objects:

1. An object is more likely to be co-located with *close* "relatives", where closeness is defined as the distance to the lowest common ancestor in the tree of object categories. E.g., in Figure 2, a printer is more likely to be co-located with scanners than DVD players.
2. For any category, the influence of "siblings", i.e., of categories with a common parent, increases as the number of "siblings" decreases. The influence of a "sibling" category increases when there is sufficient support for the sibling's existence (predicate `observed/3`).

These hypotheses enable robust evidence propagation. The relationship between object occurrence probabilities (i.e., belief state entries) and evidence provided by categories (and siblings) is inspired by **Fechner's law**[1]. For ease of explanation, consider the bias distribution in the context of locating a specific target in a set of rooms:

$$b_i^A = \alpha \ln \left(1 + \sum_{m=1}^{M_i} \frac{N_{i,m}^F}{\prod_{k=0}^{K_{i,m}-1} N_{i,k,m}^S} \right) \tag{2}$$

where b_i^A, the probability that the target is in room i, is a logarithmic function (inspired by Fechner's law) of the evidence from the current answer set, and α is a normalizer. The parameter m is the index of primary category C_m, ranging from 1 to the total number of primary categories with leaf objects known to be in room i (i.e., M_i)—$N_{i,m}^F$ counts the number of objects of C_m known in room i. Values of M_i and $N_{i,m}^F$ are obtained by counting the number of relevant `located/3` and `observed/3` literals (respectively) in the answer set. $K_{i,m}$ is the height (in object category tree) of the lowest common ancestor of C_m and the target object. The product in the denominator accounts for category nodes along the path from C_m to the lowest common ancestor. Variable k represents the height of nodes along this path, ranging from 0 (object level) to $K_{i,m}-1$, one level less than the lowest common ancestor. $N_{i,k,m}^S$ is the number of siblings of the node (including itself) on the path at height k, and $N_{i,0,m}^S = 1$.

[1]Fechner's law was introduced in 1860 and serves as the basis of modern Psychophysics. It states that subjective sensation is proportional to the logarithm of stimulus intensity.

Belief Merging*:* Since the KB (and hence the answer set) can contain incomplete or outdated information, the answer set-based bias distribution and POMDP beliefs are merged using relative *trust factors*, resulting in a r-norm probability that is a generalized form of linear and logarithmic averaging methods [5], e.g., it computes the arithmetic average for $r = 1$.

$$b_i' = \beta \left\{ (1-\Omega)(b_i)^r + \Omega(b_i^A)^r \right\}^{1/r} \tag{3}$$

where b_i^A is the answer set-based belief of target occurrence in room i (Equation 2), while b_i and b_i' are the beliefs of target occurrence in room i before and after belief merging (respectively), and β is a normalizer. The parameter $\Omega \in [0,1]$ represents the relative trust in the beliefs encoded by the answer set. The effects of Ω and r on accuracy and computational efficiency are analyzed experimentally in Section IV.

Consider the illustrative example in Figure 2. The corresponding answer set is used to compute the ASP-based bias distribution $b^A = [0.3890, 0.3361, 0.0000, 0.2749]$. The initial POMDP belief distribution (uniform in the absence of knowledge) is then revised as described in Equation 3, with $r = 1$ (arithmetic average) and the trust factor Ω set such that POMDP and ASP are trusted equally. The revised belief vector for the target is $[0.3195, 0.2931, 0.1250, 0.2625]$. The belief for each room is spread over grid cells in the room using a large-variance Gaussian centered in the middle of the room to induce the robot to move to a central location. Prior knowledge about likely locations of objects within rooms suitably revises the mean and variance of the Gaussian. The updated beliefs are used in the learned HL-POMDP policy to choose an action, resulting in the robot moving to analyze a specific scene.

2) **Knowledge acquisition:** The final component of the framework (in Figure 1) is the knowledge acquisition from sensor inputs and human-robot interaction (HRI). To simulate high-level feedback from non-expert humans with limited time, human feedback is limited to simplistic verbal inputs.

As the robot moves in the application domain, images are processed periodically to detect humans (specific humans are *not* modeled separately). When a human is detected nearby, the robot computes the need for human feedback based on entropy of the belief distribution for the object being localized. A low entropy implies that the robot is confident of the target object's location—the human is then ignored (except for safe navigation). If the entropy is high, the robot draws the human's attention, followed by a query about a room's accessibility or the target object's location. These queries and responses are based on simplistic templates such as:

```
Robot: Where is the [object]?
Human: In [room]./I do not know.
Robot: Is [room] accessible?
Human: Yes./No./I do not know.
```

In addition to human feedback, the robot processes images at specific locations in the domain and low-resolution images as it moves between locations, detecting objects using learned object models. An object detected with high certainty is added to the knowledge base, using the detected position to form a suitable fact. This piece of information may be relevant to the current task and/or to future tasks. In addition to domain objects of interest, robot may observe unforeseen changes in object configurations and obstacle locations, e.g., a door that was open may now be closed. The robot can confirm such changes using human feedback, and changes detected with high certainty also update the KB. These updates and additions to the KB occur incrementally and continuously, adding and eliminating areas for subsequent analysis.

IV. Experimental Evaluation

Experimental trials were conducted in simulation and on wheeled robots visually identifying the locations of target objects in indoor domains. The following hypotheses were evaluated: (I) integrating ASP and POMDP enables reliable target localization while significantly reducing target localization time in comparison with using ASP or POMDP individually; and (II) entropy-based strategy enables the robot to make best use of human feedback to localize targets.

A. Experiments in Simulated Domains

A realistic simulated domain was designed to extensively evaluate the framework, using learned object models and observation models to simulate motion and perception. Figure 3(a) shows an instance where four rooms are connected by a surrounding hallway in a 15×15 grid. Fifty stationary objects in 10 primary categories are simulated, and one or more of these objects are randomly selected as targets whose positions are unknown to the robot. The robot automatically creates the corresponding category tree from the KB. Each data point in the results described below is the average of 5000 simulated trials. In each trial, the robot's location, target object(s) and location(s) of target object(s) are chosen randomly. Unless stated otherwise, a trial ends when the belief in a grid cell exceeds a threshold (e.g., 0.90).

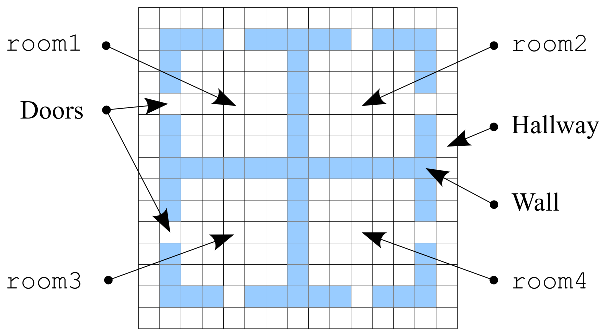
Hypothesis I is evaluated using three measures: accuracy, localization time and the ratio of these values. The accuracy is maximum when reported position and ground truth position of an object are identical (e.g., same grid cell), and drops off exponentially as the distance between reported position and ground truth position increases. Figures 4(a)-4(c) summarize experimental results, with the x-axis depicting the extent to which ASP beliefs are trusted (Ω)—*all results in these figures are statistically significant*. Figure 4(a) shows that when ASP beliefs are not considered (0 along the x-axis), the accuracy is high (≈ 0.95) irrespective of the value of r (Equation 3). Even the few errors correspond to objects close to the edge of a grid cell being localized in one of the neighboring cells. However, the corresponding target localization time is large, as shown in Figure 4(b). As the robot starts considering ASP-based beliefs, i.e., Ω grows from 0 to 1, the target localization time decreases substantially. The effect of ASP-based beliefs on accuracy also depends on the value of r, e.g., a decrease in accuracy is observed very soon for $r = 0.05$ but not for $r = 0.2$. Target localization accuracy and time have different relative importance in different situations. The trade-off between these two measures is modeled by computing their ratio. Figure 4(c) displays the value of this third measure as a function of the value of Ω. We observe that irrespective of the value of r, the best accuracy-time balance occurs when the value of Ω (i.e., trust in ASP-based beliefs) is neither too high nor too low. We therefore conclude that combining answer sets and POMDP beliefs exploits their complementary properties, resulting in high accuracy while reducing the target localization time.

Some errors in the experimental trials are due to the incorrect organization of the categories (extracted from online repositories), and the robot not receiving sufficient observations to correct these KB errors. Another reason is that the evidence from "related" objects can sometimes overwhelm certain facts. For instance, when the `scanner` in `room2` is selected as the target in Figure 2, `room1` has the highest initial belief based on the answer set. It is a challenge for robots to recover from such situations if ASP-based beliefs are trusted substantially, especially when this trust is combined with false positive observations of target(s).

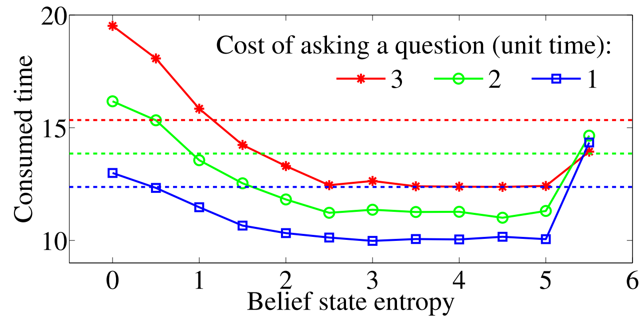
Next, to evaluate hypothesis II, human feedback is considered in addition to sensor inputs. The simulator uses known ground truth to simulate human feedback that is *available* to the robot approximately once every five actions. In addition, there is a 20% likelihood of the feedback being incorrect. The results in Figure 3(b) are for the domain in Figure 3(a). Humans can help identify the room containing the target (*but not the exact location*) and comment on accessibility of rooms, as described in Section III-C2. The x-axis shows the belief entropy threshold above which the robot seeks human input. The three solid lines correspond to different costs associated with human feedback (in units of time). As a baseline for comparison, the three dashed lines (different colors correspond to different costs) represent the random acquisition of human feedback without considering the entropy. The trust factor for ASP is chosen in the range ($\approx 0.2 - 0.6$) that results in good performance in Figures 4(a)–4(c) and r is 1. When the threshold equals the maximum entropy (≈ 5.4), the robot never asks for human feedback, whereas the robot always solicits human feedback (when available) when the threshold is 0. Since human feedback can be unreliable, acquiring and using a lot of human feedback increases target localization time. At the same time, if the robot rarely solicits human feedback (high entropy threshold), target localization takes more time. For any entropy threshold between $2.5 - 5.0$, time taken by the robot to localize targets is minimum. Human feedback thus helps significantly if used when needed. Furthermore, as cost of interacting with humans increases, feedback should be acquired more judiciously.

B. Experiments on Physical Robots

Experiments were also conducted on physical robots operating on two floors of the Computer Science department at our University. The second floor, for instance, has three classrooms, a conference room, eight offices, a research lab, a kitchen and a common area—see Figure 5(a). The test platform was a wheeled robot (inset in Figure 5(a)) equipped

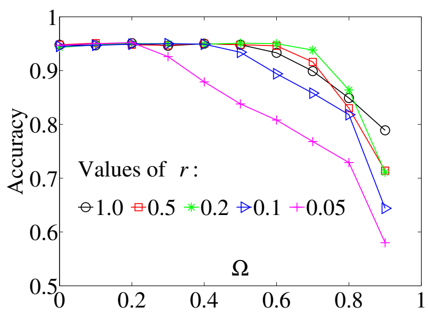


(a) Simulated domain.

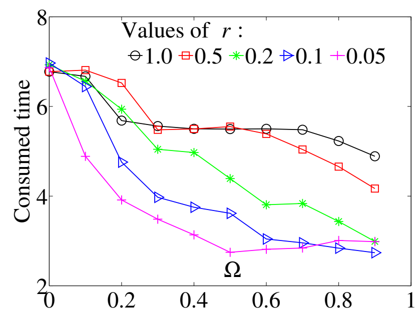


(b) Effect of human input.

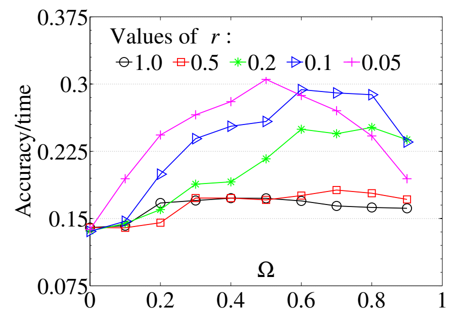
Fig. 3: (a) Illustrative example of simulated domain; (b) target localization time when robot solicits human feedback based on belief state *entropy*—target localization takes longer with too little or too much use of human feedback.



(a) Accuracy.



(b) Localization time.



(c) Accuracy/time.

Fig. 4: Performance measures for our framework that integrates ASP and POMDP: accuracy of target localization, target localization time, and the ratio of these measures, as a function of trust in ASP (Ω). *Results are statistically significant.*

with cameras, range finder, microphones and on-board $2GHz$ processor. Algorithms were implemented on the robot using the Robot Operating System [15].

Figure 5(b) shows examples of target objects in this domain. Objects are characterized using visual features such as color and local image gradients. The robot uses our visual learning algorithm to autonomously learn object models as a combination of models for these individual features [14]. Inputs from sensors and humans are processed to populate the KB. Plan execution in the lowest level of hierarchical POMDPs causes the robot to apply a sequence of actions, i.e., operators based on individual feature models in the learned object models, on input images, merging evidence to identify target objects.

We conducted 30 experimental trials—in each trial, the robot's starting location, targets (e.g., a coffee maker or a printer) and target locations were chosen randomly. The robot starts with learned object models, learned domain map and some domain knowledge, which are revised incrementally. In all experimental trials, the robot successfully localized target objects in the appropriate positions. The results were similar to the simulated trials summarized in Figure 3(b) and Figure 4. In these trials, target localization times vary substantially depending on the initial positions of robot and targets. We therefore do not report the actual target localization times measured in the individual trials. However, using ASP-based beliefs and POMDP beliefs significantly reduces the target localization time by a factor of ≈ 0.6 (on average, with $\Omega = 0.4$) compared with just using POMDP beliefs. Trusting ASP beliefs a lot more than POMDP beliefs reduces localization accuracy—just using ASP beliefs results in trials where the robot does not find the targets even after a long period of time. Furthermore, judicious use of human feedback enables the robot to interact with different humans and further reduce target localization time.

Consider a trial where the robot knows the presence of a refrigerator and a microwave in the "kitchen" and has to localize a coffee maker. Based on the object category tree of the current knowledge base, the robot concludes that the coffee maker is highly likely to occur in the same room with other kitchenware, resulting in high initial belief (of target occurrence) in the kitchen after merging the answer set-based bias distribution with the POMDP beliefs. As the robot moves to the kitchen, it meets a human but does not ask for input because the belief entropy is not high. In the main office outside the kitchen, the robot detects an HP printer that had recently been moved from the floor above, and the door to an instructor's office that was closed recently. These pieces of information, though not relevant to the current task, revise the KB for later use. When the robot reaches the kitchen, it processes images of different scenes and localizes the coffee

(a) Domain map and wheeled robot platform.

(b) Examples of target objects in domain.

Fig. 5: Domain description for experiments on physical robots.

maker. If the robot has to enter the instructor's office or find the (recently moved) HP printer in subsequent trials, it uses the existing knowledge to automatically generate suitable initial belief distributions and solicits human input appropriately. The video of an experimental trial is available online: www.cs.ttu.edu/~smohan/Movies/Planning/aspPomdp.mp4

V. Conclusions

This paper presented a novel framework that integrates answer set programming, hierarchical POMDPs and a psychophysics-inspired strategy to enable a mobile robot to: represent, reason with and revise domain knowledge, automatically adapt sensing and information processing to the task at hand, merge non-monotonic logical inference with probabilistic beliefs, and acquire and use high-level human feedback when such feedback is available and necessary. Experimental results show that the framework enables a robot to localize objects in complex indoor domains, making best use of domain knowledge, sensor inputs and human feedback.

The framework opens many directions of future research. We will explore a tighter coupling between logical inference and probabilistic planning for intelligent robots and agents. We will also investigate other algorithms for bias generation from answer sets, and consider other tasks such as information gathering and area coverage for evaluating the framework. Another research direction is the choice of questions for human feedback to enable more realistic human-robot interaction. The ultimate goal is to enable widespread deployment of mobile robots that can interact and collaborate with humans in complex real-world domains.

Acknowledgment

The authors thank Michael Gelfond and Yuanlin Zhang for their feedback on the work described in this paper. This work was supported in part by the ONR Science of Autonomy award N00014-09-1-0658. Opinions and conclusions expressed in this paper are those of the authors.

References

[1] Chitta Baral. *Knowledge Representation, Reasoning and Declarative Problem Solving*. Cambridge University Press, 2003.

[2] O. Buffet and D. Aberdeen. The Factored Policy-Gradient Planner. *Artificial Intelligence*, 173(5-6):722–747, 2009.

[3] N. J. Butko and J. R. Movellan. I-POMDP: An Infomax Model of Eye Movement. In *International Conference on Development and Learning*, Monterey, USA, August 2008.

[4] Xiaoping Chen, Jianmin Ji, Jiehui Jiang, Guoqiang Jin, Feng Wang, and Jiongkun Xie. Developing High-Level Cognitive Functions for Service Robots. In *International Conference on Autonomous Agents and Multiagent Systems*, May 2010.

[5] Roger M. Cooke. *Experts in Uncertainty: Opinion and Subjective Probability in Science*. Oxford University Press, USA, 1991.

[6] C. Galindo, J. A. Fernandez-Madrigal, J. Gonzalez, and A. Saffiotti. Robot task planning using semantic maps. *Robotics and Autonomous Systems*, 56(11):955–966, 2008.

[7] M. Gebser, R. Kaminski, B. Kaufmann, M. Ostrowski, T. Schaub, and M. Schneider. Potassco: The Potsdam Answer Set Solving Collection. *AI Communications*, 24(2):105–124, 2011.

[8] Michael Gelfond. Answer Sets. In Frank van Harmelen and Vladimir Lifschitz and Bruce Porter, editor, *Handbook of Knowledge Representation*, pages 285–316. Elsevier Science, 2008.

[9] Malik Ghallab, Dana Nau, and Paolo Traverso. *Automated Planning: Theory and Practice*. Morgan Kaufmann, San Francisco, USA, 2004.

[10] Moritz Göbelbecker, Charles Gretton, and Richard Dearden. A Switching Planner for Combined Task and Observation Planning. In *National Conference on Artificial Intelligence (AAAI)*, 2011.

[11] Marc Hanheide, Charles Gretton, Richard Dearden, Nick Hawes, Jeremy Wyatt, Andrzej Pronobis, Alper Aydemir, Moritz Gobelbecker, and Hendrik Zender. Exploiting Probabilistic Knowledge under Uncertain Sensing for Efficient Robot Behaviour. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2011.

[12] Jesse Hoey, Pascal Poupart, Axel Bertoldi, Tammy Craig, Craig Boutilier, and Alex Mihailidis. Automated Handwashing Assistance for Persons with Dementia using Video and a Partially Observable Markov Decision Process. *Computer Vision and Image Understanding*, 114(5):503–519, 2010.

[13] Leslie Kaelbling and Tomas Lozano-Perez. Domain and Plan Representation for Task and Motion Planning in Uncertain Domains. In *IROS Knowledge Representation for Autonomous Robots Workshop*, 2011.

[14] Xiang Li and Mohan Sridharan. Vision-based Autonomous Learning of Object Models on a Mobile Robot. In *AAMAS-2012 Workshop on Autonomous Robots and Multirobot Systems*, Valencia, Spain, 2012.

[15] Morgan Quigley, Ken Conley, Brian P. Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y. Ng. ROS: An Open-Source Robot Operating System. In *Open Source Software Workshop*, 2009.

[16] Stephanie Rosenthal, Manuela Veloso, and Anind Dey. Learning Accuracy and Availability of Humans who Help Mobile Robots. In *National Conference on Artificial Intelligence*, San Francisco, 2011.

[17] Scott Sanner and Kristian Kersting. Symbolic Dynamic Programming for First-order POMDPs. In *National Conference on Artificial Intelligence (AAAI)*, Atlanta, USA, July 11-15, 2010.

[18] Dafna Shahaf and Eyal Amir. Learning Partially Observable Action Schemas. In *National Conference on Artificial Intelligence (AAAI)*, Boston, USA, July 16-20, 2006.

[19] Mohan Sridharan, Jeremy Wyatt, and Richard Dearden. Planning to See: A Hierarchical Aprroach to Planning Visual Actions on a Robot using POMDPs. *Artificial Intelligence*, 174:704–725, July 2010.

[20] Karthik Varadarajan and Marcus Vincze. Ontological Knowledge Management Framework for Grasping and Manipulation. In *IROS Workshop on Knowledge Representation for Autonomous Robots*, 2011.

[21] Shiqi Zhang and Mohan Sridharan. Active visual sensing and collaboration on mobile robots using hierarchical POMDPs. In *International Conference on Autonomous Agents and Multiagent Systems*, 2012.