Information Seeking Macro-Actions for POMDPs

Max Merlin
Brown University

Thomas Ottaway
Brown University

Vadim Kudlay
Brown University

Calvin Luo
Brown University

Abstract

Modeling and understanding complex, real-world environments remains a difficult challenge in the design of intelligent agents due to imperfect information and uncertainty. Partially observable Markov decision processes (POMDPs) are a widely-studied and general way to represent such environments and perform planning operations. Unfortunately, they quickly become intractable to solve in domains with large amounts of uncertainty, limiting their viability in many real-world scenarios. In this work, we propose a method to disentangle high-level state uncertainty from low-level sensor uncertainty through macro-actions under the options framework. We devise a skill that can understand and refine the confidence in a low-level sensory observation, before passing it to the POMDP to use for planning. This reduction in uncertainty enables the POMDP to successfully model complex real-world interactions, which we demonstrate on a Boston Dynamics Spot robot that can automatically locate and fetch a desired object in a room.

1. Introduction

Designing robots that can autonomously understand, interact with, and plan in the world around them is a long-standing challenge, and designing a principled way to model and resolve sources of uncertainty is crucial in tackling it. A partially observable Markov decision process (POMDP) [3] is a natural way to represent an agent interacting with a complex environment. Fundamentally, in complex environments, an agent is unable to directly observe the complete underlying system state that it currently inhabits. However it can use its observations, which can potentially be noisy, in order to infer what state the agent is currently in. Learning a POMDP then amounts to modeling belief over the possible states, as well as learning a policy to perform the optimal actions from these belief states. While the POMDP is a general, powerful framework that can theoretically represent complex real-world environments, in practice it is rather intractable to learn and plan with in all but the smallest domains. This is because POMDPs, in their default formulation, capture many possible sources of state uncertainty together that compound with each other, resulting in an overly large branching factor that severely limits the effective horizon that can be reasoned over. In this work, we distinguish between and disentangle two key sources of uncertainty that we refer to as **high-level state uncertainty** and **low-level sensor uncertainty**.

We use **high-level state uncertainty** to refer to uncertainty that the agent has due to a lack of global knowledge. With high-level uncertainty, there is generally a lack of information or observations about the object of uncertainty, which is more impactful on higher level task goals; if you don't know which room a jar is in, you don't know which doors to open or which actions to take to help find the jar. In contrast, we use **low-level state uncertainty** to refer to the uncertainty caused by noisy or imperfect observations. Low-level uncertainty is generally less impactful on those task goals, and is inherently local and tied to observations that have already been gathered. When a POMDP is resolving low-level uncertainty it is often attempting to gather information as opposed to interacting with the environment. It needs to balance the reward it can gain from interaction against the likelihood that it is wrong about the reward it will receive due to local state uncertainty.

When applied to real-world robotic tasks, sensor noise (a low-level uncertainty) is unavoidable due to the realities of sensing technology. Using such noisy observations propagates low-level uncertainty into the agents model of the high-level state, increasing the complexity of learning and using POMDPs. A common approach to making POMDPs more tractable despite such high complexity is to utilize some form of abstraction. Some examples of this are include grouping sets of states together as **symbols** or **predicates** [5, 4], modeling independence of parts of the state by defining them as **objects** [1, 7], and defining abstracted skills called **options** that can chain actions together to reach a certain predefined state or sub-goal [6]. In this work, we propose a new type of skill that we call an Information Seeking Macro-Action (ISMA). This new type of action abstraction specializes in resolving low-level observation confidence estimation and refinement, thereby increasing the

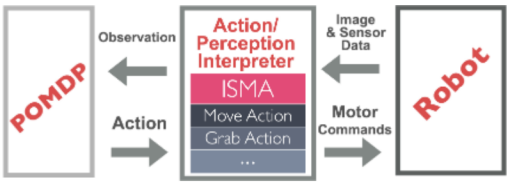


Figure 1. POMDPs all but require both state and action abstraction to be feasible in a real world robotic task. Our action perception interpreter helps us convert back and forth between our abstracted domain and the actual robot environment.

POMDPs applicability in complex real-world environments by reducing the overall problem complexity caused by low level uncertainty and planning horizon length.

2. Method

We address POMDP planning on a robot in a real-world environment, so we need to make certain assumptions about our model and our environment to bridge the conceptual gap and maintain feasibility. We assume that our state $S(i_1...i_n)$ is processed and factored, as opposed to raw sensor input. We also assume that our action set is made up of options, so that we do not need to plan at the level of individual motors. We introduce an action perception interpreter that takes in the high-level action names from the POMDP and executes their policies by sending the appropriate motor commands to the robot. It also converts the raw sensor data from the robot into a state that is sent to the POMDP as observation data or used within each options policy.

2.1. ISMA Definition

We draw inspiration from the options framework [6] to formally define an Information Seeking Macro-Action (ISMA) as follows:

- I: The initiation set of the ISMA. Defines the conditions required to execute the skill.
- i: The target object we are trying to observe
- θ: the confidence threshold that needs to be exceeded in order to terminate
- π: The policy of the ISMA. Determines the actions performed during skill execution
- β: The termination condition. The skill will terminate by returning an observation over specifically the target object i such that:

$$o_i = \begin{cases} s[i] & \text{if belief that } i \text{ is present} > \theta \\ None & \text{if belief that } i \text{ is not present} < 1 - \theta \end{cases} \quad (1)$$

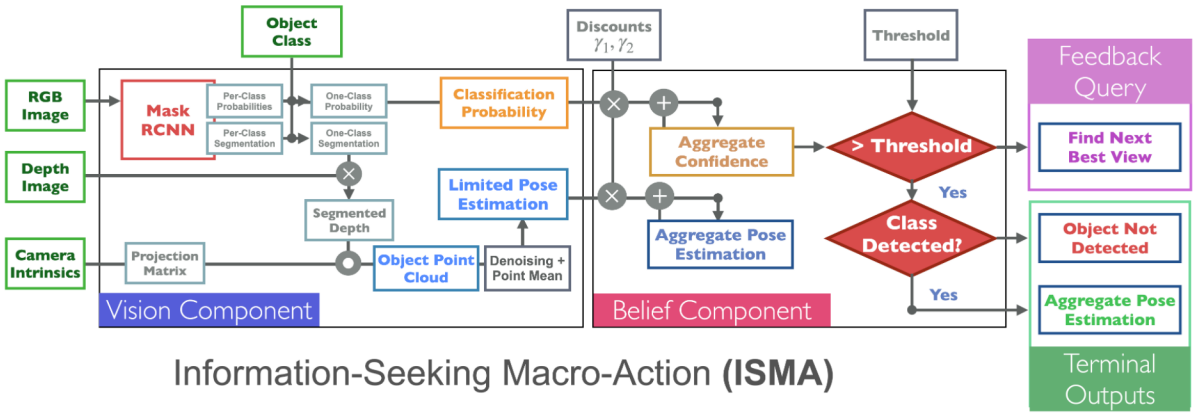
The ISMA is added to the POMDP as an additional action that the agent can perform. Because the ISMA returns an aggregate of many observations with guarantees about the end confidence, we can consider the output as an **abstract observation**. In addition, as long as the θ threshold is tuned appropriately, we can treat that abstract observation as if it were a direct observation of the true world state. This gives the ISMA a trait that is very beneficial to POMDP planning: executing an ISMA is *guaranteed* to reduce (or in redundant cases not affect) overall state uncertainty.

We propose two possible methods for policies that can be used within the ISMA. Method one is to model a reduced POMDP that is a strict subset of the existing POMDP. State space, action sets, and observation space is strictly reduced to whichever subsets of each is feasible within the local region that is being observed. The sub-POMDP receives positive reward for reducing uncertainty beyond the threshold in either direction (observing the object or observing the distinct lack of object), and terminates once that threshold is passed. Method two is to create the internal policy of the ISMA more directly using the existing state of the art computer vision methods. Belief is still tracked internally but the programmer has the freedom to integrate other models that are more tuned to the task of observation and mapping, but which may otherwise be difficult to use in a POMDP setting. In this paper we primarily pursued this method for policy generation, and the details of which are as follows.

2.2. ISMA Model

Our model is composed of two major parts: the vision component and the belief component. The vision component processes the raw sensor observations, including the RGB and depth data of the scene. The belief component represents the current belief of the abstracted state and determines whether the ISMA can terminate or if it needs to gather more data. Together, the ISMA is able to continuously retrieve, aggregate, and process low-level information until a confident understanding of the sensor observations is achieved. This can then be passed into a POMDP for further planning and acting in the environment.

The vision component uses a trained Mask R-CNN [2] model to produce both a segmentation of the image as well as the per-class probabilities. Given a desired object class from the categories trained on my by the model, we can get a mask segmentation of where the objects are in the image. From there, we can project the image into 3D space based on the segmented depth map approximation from the robot alongside the camera intrinsics. From there, we can forward the specific object class probability to the belief component to update its belief on whether the object is present in our local view or not. Furthermore, if the object exists, we can utilize its RGB segmentation by combining it with the depth image to get a depth segmentation of the object. This seg-



(a) A visual depiction of our proposed framework, which consists of a vision component and a belief component.

mented depth can be normalized adjusted for the camera intrinsics using a projection matrix to create a canonical object point cloud. After denoising this point cloud, a pose of the object can be inferred and transferred to the belief component. Overall, the vision component processes the raw sensor input and outputs both the probability of the object presence in the local scene as well as its predicted pose.

The belief component represents the local belief in the context of the current skill being taken. The local belief aggregates the outputs of multiple different processed views from the vision component by compiling the output classification probabilities and pose estimates. Because the object point clouds are standardized by projecting to a common space and denoising, the output point clouds can be aggregated with ease. By integrating the probabilities from multiple different views, our agent becomes more sure about whether or not the object exists while also becoming more confident regarding the location and pose. Once the aggregated confidence of the object's presence in the scene passes a set threshold, a final output is returned. This is either the fact that no object was detected, or the aggregated pose estimate of the object if it was detected. This information can guide the agent to perform its next action, such as moving to inspect another location for the the desired object, or to grab the object if it is currently present.

3. Experiments and Results

We deploy our framework on the Boston Dynamics Spot robot, on an object-retrieval task. Specifically, we would like the robot to automatically locate an object in the room around it and retrieve it. We model this domain as a POMDP using the POMDP-Py library, with uncertainty as to which table the desired object is on. At each timestep the POMDP agent can move to any table, move back to the home pose, run an ISMA at any table, or attempt to pick up the object. The agent receives reward for picking up the right object and further positive reward for bringing it back to the home position, with a negative reward at all other timesteps. Planning is done using the existing implementation of POMCP.

For the sake of demonstration, the locations of interest (different movable tables) are denoted by AR tags, and both their poses and robot's pose is considered fully observable. The desired object is a bottle of Mr. Planters peanuts, which will be placed randomly at a location of interest. Furthermore, the locations of the tables are not fixed, and can be moved around the room as long as the AR tags are visible.

We implement the skills as described in the POMDP: one to navigate to a certain table (identifiable by its AR tag), another to determine if and where the desired object is at the robot's current location, and a third to retrieve the object.

We successfully demonstrate the Spot's ability to retrieve the desired bottle when placed on an arbitrary surface, even when the surfaces are shuffled. In our implementation, the Spot will first select a table on which to search for the object and move to (with the help of AR tags). After positioning itself close to the table, it cycles through a set of camera poses that inspect the surface of the table. Each view in which Mask RCNN identifies the object and updates the agents belief of the objects presence and pose. Finally, if the object is believed to be present with high confidence, the median of the aggregated point clouds is returned as the location of the object. The Spot robot then decides to execute a grab command, using the returned parameters, to successfully secure the object.

Currently, because the object point cloud depths are all calculated with respect to the surface of the bottle, we manually coded an object-specific offset that we can use to target the grasp skill. We note that this is a proof-of-concept, and a more general algorithm can be applied to estimate the center of an object from an accurate or partial point cloud of its surface.

Figure 3. An image of Spot having retrieved a bottle of peanuts from its surroundings automatically. Behind it is an example surface, denoted by an AR tag, of where the bottle could have been placed.

4. Conclusion and Future Work

In this work we propose ISMA, a general framework that can improve planning by disentangling high-level state and low-level state uncertainty. This is done by defining a skill that utilizes state-of-the-art computer vision methods to perform low level uncertainty resolution, thus extracting the problem from the POMDP. This enables a POMDP to successfully model and perform planning in complex, real world, environments, which we demonstrate by getting a Spot robot to fetch a desired object a selection of locations. We ensure that our approach remains general so that there are multiple directions in which future work can iterate or fine-tune the method.

4.1. Successes and Shortcomings

We were able to successfully implement the full pipeline as intended from the start. Our navigation and picking skills were extremely reliable, the implementation and processing of Mask R-CNN was consistent and robust, and the POMDP model integrated with the rest and generated reasonable plans. However, there are several points in the pipeline where components became over-simplified for the sake of completion. We did not end up fully tracking belief internally to the ISMA, and it followed a fixed policy taking two preset camera views instead of calculating intelligent camera poses. This worked because our object identification system was so robust, but it reduced our prioritization to create a more interactive and adaptive policy. We also simplified the POMDP significantly in order to get it to perform as intended, and we only modeled the existence of the singe object we were attempting to retrieve. Despite the shortcuts taken in certain areas due to time constraints, we are confident that the method is sound and intend to develop each of those sections further in future work. We also intend to explore other base computer vision methods, other forms of aggregating/updating object pose belief, and more complex task plans and scenes. Eventually, we hope to apply this method to all objects in a local scene instead of just a single object.

5. Author Contributions

- **Max Merlin** provided the initial concept for the project and developed the theoretical groundings and contributions. He developed/debugged the planning side of the code with Thomas Ottaway, and contributed to creating the poster content for the presentation and writing the final report.
- **Thomas Ottaway** developed code to perform perception and manipulation tasks on the spot robot. He also developed/debugged the planning code along with Max Merlin.
- **Vadim Kudlay** investigated vision-side problem and formulated the vision aspects of the project along with Thomas Ottaway. Developed pipeline visualizations with Max Merlin and helped with experimentation/debugging.
- **Calvin Luo** investigated next best view techniques for determining how to best visually process the local scene. Furthermore he organized a substantial portion of the poster content in terms of its writing and figures. Lastly he summarized the project findings by writing a draft of the final report.

We would also like to thank Eric Rosen, Kaiyu Zheng, and Nick DeMarinis for their help in working with the existing codebases and systems required to make this work possible.

References

[1] Carlos Diuk, Andre Cohen, and Michael L Littman. An object-oriented representation for efficient reinforcement learning. In *Proceedings of the 25th international conference on Machine learning*, pages 240–247, 2008.

[2] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

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

[4] George Konidaris. On the necessity of abstraction. *Current opinion in behavioral sciences*, 29:1–7, 2019.

[5] Lihong Li, Thomas J Walsh, and Michael L Littman. Towards a unified theory of state abstraction for mdps. *ISAIM*, 4(5):9, 2006.

[6] Richard S Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211, 1999.

[7] Arthur Wandzel, Yoonseon Oh, Michael Fishman, Nishanth Kumar, Lawson LS Wong, and Stefanie Tellex. Multi-object search using object-oriented pomdps. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 7194–7200. IEEE, 2019.