Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Collaborative human-autonomy semantic sensing through structured POMDP planning

Luke Burks, Nisar Ahmed [*], Ian Loefgren, Luke Barbier, Jeremy Muesing, Jamison McGinley, Sousheel Vunnam

Ann And H.J. Smead Aerospace Engineering Sciences Department, 429 UCB, University of Colorado Boulder, Boulder, CO 80309, USA

ARTICLE INFO

Article history:
Received 29 September 2020
Received in revised form 6 January 2021
Accepted 15 February 2021
Available online 4 March 2021

Keywords:
Human–robot interaction
Data fusion
Partially observable Markov decision processes
Planning
Bayesian methods
Target search
Target localization
Mobile robots
Autonomy

ABSTRACT

Autonomous unmanned systems and robots must be able to actively leverage all available information sources — including imprecise but readily available semantic observations provided by human collaborators. This work develops and validates a novel active collaborative human–machine sensing solution for robotic information gathering and optimal decision making problems, with an example implementation of a dynamic target search scenario. Our approach uses continuous partially observable Markov decision process (CPOMDP) planning to generate vehicle trajectories that optimally exploit imperfect detection data from onboard sensors, as well as semantic natural language observations that can be specifically requested from human sensors. The key innovations are a method for the inclusion of a human querying/sensing model in a CPOMDP based autonomous decision making process, as well as a scalable hierarchical Gaussian mixture model formulation for efficiently solving CPOMDPs with semantic observations in continuous dynamic state spaces. Unlike previous state-of-the-art approaches this allows planning in large, complex, highly segmented environments. Our solution is demonstrated and validated with a real human–robot team engaged in dynamic indoor target search and capture scenarios on a custom testbed.

1. Introduction

Dynamic target search and localization remains a very active research area for unmanned autonomous vehicle systems. Solutions typically leverage joint state space models of target dynamics, mobile sensor platform motion, and sensor observations to solve challenging combined optimal control and estimation problems. However, practical algorithms for data fusion and decision making can still be too computationally expensive and brittle to ensure full vehicle autonomy.

In many cases, human operators and users can act as ‘human sensors’ that contribute valuable information beyond the reach of autonomous vehicle sensors. For instance, operators in search and tracking missions using small unmanned aerial systems (UAS) can provide ‘soft data’ to narrow down possible survivor locations using semantic natural language observations (e.g. ‘Nothing is around the lake’; ‘Something is moving towards the fence’), or provide estimates of physical quantities (e.g. masses/sizes of obstacles, distances from landmarks) to help autonomous vehicles better understand search areas and improve online decision making with limited computational resources. This naturally raises the question of how autonomous reasoning can actively and opportunistically engage human reasoning to improve its own performance.

We present a rigorous framework for intelligent human-autonomy interaction that not only leverages combined robot–human sensing, but is also tightly integrated with dynamic platform decision making and planning. Our approach uses Bayesian data fusion to exploit soft data in such a way as to interface with existing autonomous sensing frameworks, while also enabling accessible and understandable interaction with the system on the part of the human. Such ‘plug and play’ human sensing for robot state estimation was explored in [1,2] for restricted types of human observations, and has received increased attention in recent years [3–6]. In this paper, we combine our recent work on Bayesian semantic natural language human data fusion [7,8] with concepts from optimal active sensing, in order to develop a new framework for *interactive* human–robot semantic sensing. Here we focus on the challenging problem of non-myopic decision making for *simultaneous (tightly coupled) vehicle motion planning and human sensor querying* in continuous dynamic search environments. As shown in Fig. 1, our approach leads to joint action–query *policies* (i.e. control laws) over a continuous target search space. A policy tells the robot how to respond to target location uncertainty, so that it simultaneously makes optimal decisions about how to move/sense on its own in the environment

[*] Corresponding author.
E-mail addresses: luke.burks@colorado.edu (L. Burks), nisar.ahmed@colorado.edu (N. Ahmed).

https://doi.org/10.1016/j.robot.2021.103753
0921-8890/© 2021 Elsevier B.V. All rights reserved.

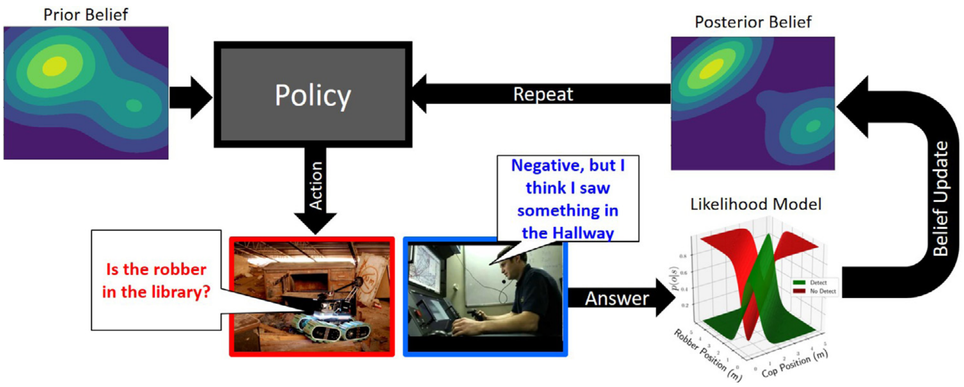


Fig. 1. Closed-loop collaborative Bayesian target search using a non-myopic policy for simultaneous semantic querying and sensor vehicle motion planning.

and about which semantic natural language questions it should ask human sensors in order to 'pull' useful information. The human only needs to act as a (voluntary) sensor; hence, the robot does not "depend" on the human but can opportunistically gain information or adapt to whatever soft information is provided by the human. Furthermore, the policy lets the robot conduct an optimal search with complex non-Gaussian uncertainties, even without human input.

Our technical approach builds on recent foundational work for efficiently finding policies based on continuous partially observable Markov decision process (CPOMDP) models [9]. While this CPOMDP approach provides several beneficial theoretical features for collaborative dynamic target search, this work also addresses technical challenges that arise for practical implementation for real human-autonomy teams. In particular, this work presents a scalable hierarchical CPOMDP solution that allows our framework to be deployed in arbitrarily complex environments, e.g. large indoor settings with multiple rooms/search areas, many possible semantic grounding references, and moving targets, which would otherwise be intractable for a single CPOMDP policy to handle. A novel formulation of human querying/sensing is also introduced within the context of the hierarchical CPOMDP, allowing information to be volunteered by and autonomously requested from the human with regard to each layer of the hierarchy. Finally, hardware demonstration results are presented which validate our approach with a real human–robot team engaged in dynamic indoor target search and capture scenarios. Varying levels and types of human-autonomy interaction are compared, showcasing the value of human information in a variety circumstances. While the presentation is grounded in dynamic target search problems, our human-autonomy collaboration framework can be applied to other interactive dynamic data fusion problems as well.

This paper is structured as follows. Section 2 reviews background and related work. Section 3 presents our new hierarchical CPOMDP framework for optimal search and interactive semantic soft data querying, in the context of indoor dynamic target search. Section 4 provides demonstration results on our custom human–robot team target search testbed, and Section 5 presents conclusions and future work.

2. Background and related work

While the novel framework debuted in this work is showcased on a dynamic target search problem, it is readily extensible to other scenarios involving human–robot collaborative decision making such as autonomous planning in human-collocated environments [10], robotic self-localization [11–13], autonomous driving [14,15], and inventory control [16].

In many applications, several fundamental difficulties arise, which lead to brittleness in practice. Firstly, autonomous robotic vehicles are subject to constraints on motion, size, weight, power, and cost; this limits their computing and sensing payloads as well as their operating time and range. Secondly, the sensing and planning horizons for approximate optimal search algorithms are inherently limited. This not only restricts the ability to correctly detect and sense targets, but also the ability to execute adaptive long term information gathering strategies in complex dynamic environments. Finally, sensing platforms may only have access to imperfect/highly uncertain target behavior models. This can lead to non-Gaussian probability distributions over target states, and make online planning and sensing/data fusion even more difficult.

Formal integration of robotic and human perception can greatly improve the efficiency and robustness of autonomous decision making, especially in situations where uncertainties cannot be well-characterized in advance and must be adapted on the fly. Soft data provided by humans can be broadly related to either 'abstract' phenomena that cannot be measured by robotic sensors (e.g. labels for occupied/unoccupied rooms, object categories and behaviors) or measurable dynamical physical states that must be monitored continuously (object position, velocity, attitude, temperature, size, mass, etc.) [17]. This work examines the problem of *active soft data fusion*, and builds on methods for addressing the following key issues: (i) soft semantic data modeling for recursive state estimation; and (ii) active semantic sensing for intelligent planning under uncertainty.

In the following, Section 2.1 defines the grounding dynamic target search problem used throughout the rest of the paper as a specific application of collaborative semantic human–robot sensing and planning. Section 2.2 then reviews related work for managing human–robot information exchange. In Section 2.3 a data fusion method leveraging Gaussian Mixture models for human–robot "hard–soft" data fusion is reviewed. Section 2.4 sets out the challenges and approaches for decision making under uncertainty as relates to the grounding problem. Finally, Sections 2.5 and 2.6 give a formal problem statement and general solution.

2.1. Motivating problem

This paper focuses on search and localization applications where the number and type of targets are known, and where mobile sensor platform dynamics, observation models, and environment maps are known. The motivating scenario is a dynamic target search problem, "Cops and Robbers" (CNR), which takes place in an indoor environment such as the one shown in Fig. 2. An autonomous robotic agent, referred to here as the cop, is tracking a robber within the confines of an indoor environment, with the help of an off-site human collaborator. The cop's goal is to localize, track, and intercept the robber as quickly as possible. The environment is segmented into interconnected "rooms", each

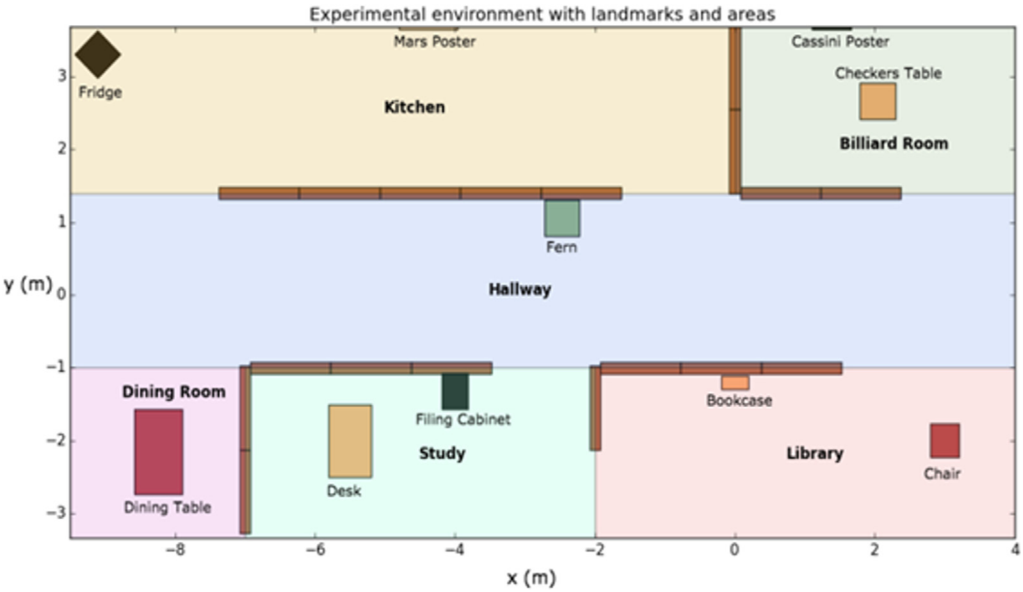


Fig. 2. An example cops and robots room layout. The global space is fully partitioned into labeled rooms, each of which contains objects of known semantic labels and positions.

of which carries a known semantic label typical to a domestic home environment, such as "hallway" or "kitchen". Within each room, distinct objects are placed in fixed locations, each with its own semantic label. For instance, in the room "kitchen", one of the labeled objects might be the "refrigerator", while in the "study" there might be a "desk". In a real-world scenario, each room would likely contain a multitude of non-unique objects in many possible configurations [18]. The robber moves randomly from room to room, independent from the movements of the cop and with no preference for a given room or sequence of rooms.

The human collaborator is able to imperfectly monitor the environment through the use of security cameras placed in the environment, and can also access the viewpoint of the cop through an on-board mounted camera. The human's role is to provide information relevant to the cop's task through two related methods, with the terminology for both adopted from previous work in [19]. The first is the "Robot-Pull" event, in which the cop requests information about the robber's position relative to a labeled object or room. For instance, the cop might ask "Is the robber in front of the chair?", to which the human can provide a binary answer of "yes" or "no". The second method is through "Human-Push" events, in which the human can volunteer information which they deem useful. As an example, after checking the security camera mounted in the kitchen, the human might push the statement, "The robber is not in the kitchen". In both cases the human is assumed to be imperfect but well-intentioned, and is capable of passing along false information by mistake. Note that the human is unable to directly influence the cop's movements, and at no point are commands or directions issued. However, the semantic information they provide directly influences the cop's understanding of the targets position, and therefore will have some effect on actions. For instance, repeated assertions by the human that the robber is in the study would reasonably lead to the cop investigating the study. In most cases however, due to the view of security cameras not spanning the entire environment, the human will not be able to immediately locate the target and will be limited to negative observations.

2.2. Algorithms for human–machine interaction

This section reviews existing frameworks and algorithms for human-autonomy interaction with emphasis on applicability to the motivating problem. The use of humans as sensors in the motivating problem provides unique challenges from both theoretical and implementation perspectives. Ultimately, the human must be modeled, and interacted with, as a multi-purpose semi-opaque probabilistic sensor. As an example, in a real world implementation a human might be able to reason about the future trajectory of the robber regarding which rooms have items of value, or be able to infer blind spots based on camera feeds. This imperfect natural inference can help bound the search area for the cop, using information the cop may not have thought to ask for. This form of reasoning both mirrors and complements robotic reasoning based on recursive Bayesian estimation (discussed in the next section). Thus, the methods by which the human and cop interact would ideally allow for both the useful but imprecise nature of human sensor observations, as well as their and capacity to offer surprising or unexpected information.

While existing literature has explored the idea of learning how individual humans offer information to machines in repeated tasks [20,21], these primarily focus on learning the parameters of a specific human collaborator. Tasks such as the motivating problem can also benefit from more generalized human models. These "plug and play" human models, as explored in [1] and [2], allow a system to understand information originating from a non-specific human. These ideas have been explored using varying mathematical frameworks [3], and adapted to account for human factors such as cognitive load and physiological state [4]. However, such frameworks have typically focused on the data fusion aspect of human sensing. In order to address the problem proposed in Section 2.1, a method of decision making which takes data fusion framework into account is also required.

A major component of the motivating problem is the ability of the cop to initiate a "Robot-Pull" event to query the human for specific information. This ability gives rise to the issue of deciding which question should be asked. Put another way, given a multitude of options regarding the various rooms, objects, and directions the cop can generate queries from, which combination leads to the greatest increase in utility? In general, this problem can be framed as a Value of Information (VOI) [19] problem. Previous work has grappled with this active sensing problem by positing a direct link between state uncertainty and utility. This leads to the implicit assumption that perfect knowledge of a robber's location will lead to maximal utility through rapid interception, and the correct question to ask is that which leads to the greatest expected decrease in uncertainty. Unfortunately, even in cases where the calculation of VOI can be accomplished easily for the current timestep, determining the optimal *sequence* of questions becomes intractable due to the combinatoric increase in questions trajectories over time. Thus VOI aware planning has generally been used in myopic implementations [8], though work has been done on training machine-learning algorithms to recognize non-myopic VOI from current uncertainty levels [22]. However, even these cases still calculate VOI according to expected changes in uncertainty rather than on expected changes in the probability of success. The framework proposed in this work instead directly links information gathering with a reward function representing desired behaviors. Thus active sensing explicitly becomes a matter of choosing information gathering actions that lead most certainly and quickly to success.

Solutions to the problem of integrating the knowledge of respective members of a human–robot team can take many forms. For many tasks, humans can serve as information providers not only about the state of the task, but also about potential policies or strategies to accomplish the task. Techniques such as Apprenticeship Learning [23] or Learning from Demonstration [24,25], allow the human to provide examples which allow the robot to learn an appropriate policy. These examples can be direct and initiated by the human [26], or a result of robot actions to evoke a response by the human [27]. A core assumption of these methods is that the human knows what they are doing, or at the least is capable of providing useful information about the optimal policy for the task. In cases where the human cannot serve as a reasonably effective teacher, the robot must rely on other means to calculate a policy.

In tasks allowing language based communication, natural language methods enable robust communication between human and robot team members, but introduces the problem of facilitating dialog on the part of the robot. Such dialog can handled within optimal planning frameworks to help accomplish shared goals [28], and account for language uncertainties [29]. However, many previous approaches applying natural language dialog to shared tasks assume the human as a direct collaborator, assuming they will perform their own actions cooperatively or independently from the robot, or that the human is in some way supervising a task and directing the robots actions. In cases where the human acts as a sensor, rather than an on-the-ground partner or commander, there arises a need to account for both temporary unavailability and inaccuracy on the part of human, such that a robot can accomplish a task fully independently in the worse case.

2.3. Mixture-based Bayesian soft data fusion

Ref. [7] showed how to model and fuse flexible semantic natural language data to provide a broad range of positive/negative information for Bayesian state estimation, e.g. 'The target is parked near the tree in front of you', 'Nothing is next to the truck heading North'. This fusion algorithm directly plugs into Gaussian mixture filters for robotic state estimation, which can accurately represent complex posterior pdfs while avoiding the curse of dimensionality. Suppose $s_k \in \mathbb{R}^n$ is a random vector representing some continuous state of interest at discrete time k (e.g. target location, velocity, heading) with prior pdf $p(s_k)$, which may already be conditioned on hard/soft sensor data and predicted forward in time from according to known a stochastic state transition pdf via the Chapman–Kolmogorov equation. Let D_k be a discrete random variable representing a human-generated semantic observation related to s_k. Bayes' rule gives the posterior pdf

$$p(s_k|D_k = i) = \frac{P(D_k = i|s_k)p(s_k)}{\int P(D_k = i|s_k)p(s_k)ds_k} \tag{1}$$

where the likelihood function $P(D_k|s_k)$ captures the human's semantic classification behavior conditioned on the true state s_k. If $D_k = i$ corresponds to one of m exclusive semantic categories for a known dictionary of state observations, then a *softmax function* (i.e. multinomial logistic function) can be used to model $P(D_k = i|s_k)$,

$$P(D_k = i|s_k) = \frac{e^{w_i^T s_k + b_i}}{\sum_{j=1}^m e^{w_j^T s_k + b_j}} \tag{2}$$

where w_j and b_j are vector weight and scalar bias for class label i. For a sufficiently rich dictionary of semantic observations D_k, multiple softmax models can be defined via Eq. (2) with $m = 2$ for different binary sets of semantically similar class labels ('nearby' vs. 'not nearby', 'next to' vs. 'not next to', 'close by' vs. 'not close by', etc.), so that they need not be treated as mutually exclusive labels within a single large softmax model. The likelihood parameters w_j and b_j can be learned from semantic human sensor calibration data [7] and algebraically manipulated to shift, dilate, rotate, and geometrically constrain semantic class boundaries in $\mathbb{R}^n$ [8,30].

Eq. (1) must be approximated for recursive Bayesian data fusion with softmax likelihoods, since the exact posterior pdf $p(s_k|D_k)$ cannot be obtained in closed-form for any $p(s_k)$. If $P(D_k = i|s_k)$ is generally given by a softmax model for observation label i and the prior is given by a finite Gaussian mixture (GM) with M prior components evaluated at s_k,

$$p(s_k) = \sum_{m}^{M} w_m \phi(s_k|\mu_m, \Sigma_m) \tag{3}$$

(where w_m, $\mu_m \in \mathbb{R}^n$, and $\Sigma_m \in \mathbb{R}^{n\times n}$ are the weights, mean vector and covariance for mixand m), then $p(s_k|D_k = i)$ can be well-approximated by an M component GM,

$$p(s_k|D_k = i) \approx \sum_{n}^{M} w_n \phi(s_k|\mu_n, \Sigma_n) \tag{4}$$

The weights, means and covariances of posterior component n can be determined by fast numerical approximations methods [7], and mixture compression methods can be used to manage the growth of mixture terms due to non-linear dynamics or application of non-convex 'multimodal' softmax models [31].

2.4. Active semantic sensing for planning under uncertainty

A major challenge for problems like target localization is that dynamics and uncertainties can quickly become quite non-linear and non-Gaussian, particularly given the types of semantic information available for fusion (e.g. negative information from 'no detection' readings [32]). As a result, typical stovepiped approaches to control/planning and sensing/estimation can lead to poor performance, since they rely on overly simplistic uncertainty assumptions. Constraints on human and robot performance also place premiums on when and how often collaborative data fusion can occur. For example, it is generally important to balance situational awareness and mental workload for a human sensor (who might also need to switch between tasks constantly). Likewise, it is important for the robot to know how and when a human sensor can be exploited for solving complex planning problems, which would otherwise be very inefficient to tackle using only its own local sensor data

Target search problems in uncertain environments can be cast as Partially Observable Markov Decision Processes (POMDPs) to non-myopically integrate VOI-based reasoning. In general, POMDPs solvers seek a *policy*, which maps a belief over the state space (i.e. a pdf) to a recommended action. These actions seek to maximize the expected time-discounted reward over time. Exact solutions to POMDPs are impractical in all but the most trivial of problems, and a variety of approximate solutions have been proposed.

One approach to POMDP approximation, the QMDP algorithm [33], attempts to use a fully observable MDP policy to compute the optimal action in a partially observed step. As QMDPs are only exactly optimal assuming the state will indeed become fully observable after a single timestep, they are generally unsuitable for information gathering dependent problems such as the one addressed in this work. However, the introduction of the oracular POMDP (OPOMDP) formulation [34,35] built on a QMDP policy and enabled the use of a human sensor to provide "perfect" state information at a fixed cost. Further work resulted in Human Observation Provider POMDPs (HOP-POMDPs) [36], which allow the consideration of oracular humans who are not always available to answer a robotic query. HOP-POMDPs calculate a cost of asking, which is then weighed against the potential information value, similar to VOI aware planning in [19]. When augmented with the Learning the Model of Humans as Observation Providers (LM-HOP) algorithm [21], HOP-POMDPs can estimate both the accuracy and availability of humans, thus treating them as probabilistic sensors. A primary drawback to using either OPOMDPs or HOP-POMDPs to address the motivating problem from Section 2.1 is that while they both enable a QMDP based policy to consider information gathering actions, these actions consist of a single self-localization query. Such formulations ignore the rich information set available in the motivating problem thanks to the presence of semantically labeled objects.

Another class of POMDP approximation known as Point-Based Value Iteration [37] and various related algorithms [38,39] rely on recursively solving the corresponding Bellman equations with known observation and transition models for some subset of possible beliefs that might be encountered during policy execution. When specified over discrete state, observation, and action spaces, the computational complexity explodes quickly with the number of joint configurations across each space.

The introduction of the CPOMDP method [40] showed that sets of Gaussian Mixture (GM) models can be used to approximate the policy over a continuous state space, while Switching-Mode POMDPs [41] further extended the CPOMDP framework to account for non-constant transition functions such as those caused by the presence of obstructions in the space. Finally, Variational Bayes POMDPs (VB-POMDPs) [9] were developed to handle non-Gaussian observation models in the form of softmax models. These easily model semantic observation statements and excel at parsing semantic input statements over a continuous space while significantly decreasing the computational cost of finding and implementing a policy. Unlike previous discrete state space POMDPs, which scaled in complexity with the number of discrete states, CPOMDP approaches scale with the complexity of the beliefs. This allows the use of POMDPs in much larger spaces, limited primarily by how precise the GM belief representation needs to be.

In many human-autonomy interaction applications, the introduction of common real world complications such as large semantic object dictionaries, walls, and expansive state spaces increase computational complexity beyond practicality when using a single monolithic POMDP policy. While the use of CPOMDP formulations such as VB-POMDP naturally handle large state spaces such a room, they do not naturally account for discontinuous transitions, e.g. such as those involving obstacles for mobile platform motion planning. While this was partially addressed by the Switching-Mode POMDP framework [41], the presence of many semantic labels can add prohibitively more computation to the policy solution. This occurs in two ways. For example, in the motivating problem, the cop is able to request information about the target's location with respect to a particular object, "Is the robber behind the Checkers Table". Larger semantic object dictionaries mean the cop must make a decision on which to ask about from among a larger set of possible queries. Having large numbers of objects which can act as anchors for semantic human sensor observations also leads to a large observation space, further increasing computational costs. Even with recent advances in scalability such as those showcased by VB-POMDP, existing policy approximation methods are ill-suited to handle the kind of complex and informationally dense settings found in real world applications. With its separate but connected rooms, and information-dense environment filled with objects that can be referenced for observations, the motivating problem presents a challenge for typical POMDP approaches. CPOMDP methods struggle with the number and complexity of switching-modes required to encapsulate objects and walls. Additionally, the introduction of "Human-Push" and "Robot-Pull" actions naturally lead to larger action spaces and dramatically larger observation spaces.

In principle, online POMDP solvers such as [42] or [43] could be used to come up with acceptable approximations for such problems using policy search techniques. But online solvers struggle with problems like CNR, since rewards can only be obtained at a single point in the state space, i.e. when the robber is caught. This causes problems for large state spaces as positive reward states will often lie beyond an online solver's effective planning horizon, and thus intermediate rewards cannot be obtained to promote adequate policy exploration. In principle, expanding the effective planning horizon negates these issues. However, results from [9] show that even in comparatively simple planning environments, online planners take significantly longer than offline planners at policy execution to achieve similar results. When applied to computationally constrained robotic platforms and larger/more complex spaces the advantage of offline planners becomes more pronounced if the aforementioned issues with information density can be addressed. In the specific case of the motivating problem, the continuous indoor environment is easily separable into connected regions, which suggests exploring the use of multiple connected CPOMDP policies that can be obtained offline.

The first major contribution of this work is the specification of CPOMDPs for collaborative robotic information gathering and optimal control where a human collaborator is formulated as a semantic sensor, alongside the ability to actively query this new sensor for information relevant to an autonomous agent's goal. The VB-POMDP [9] variant of the CPOMDP formulation is used to specify solve a target search problem with semantic sensor measurements. These measurements are described as softmax functions as in [7], and are used to fuse information in both "Human-Push" and "Robot-Pull" events, as well as an on-board robotic measurements. This results in the computation of an optimal POMDP planning and querying policy, where the actions recommended by the policy consist of both physical movements and queries for the human, and observations consist of both robotic and human semantic information. This pre-computed policy is able to run in real-time on a physical robotic platform, and actively accounts for collaborative information gathering in a human–robot team.

The second major contribution of this paper is a divide and conquer hierarchical POMDP-based querying strategy for large problem spaces. This approach exploits the natural segmentation of certain environments, such as a building, into multiple connected open spaces, such as rooms. Each of these rooms can be treated as a separate CPOMDP, while a PBVI-based discrete POMDP solver can be used to connect them at a higher level. This results in a hierarchical POMDP structure, with the discrete solver directing which room level CPOMDP policy should be followed at each time step.

2.5. Formal problem definition

Formally, the generalized human-autonomy collaborative sensing and planning problem addressed in this chapter is stated as an infinite horizon POMDP, represented by a 7-tuple $\{\mathcal{S},\mathcal{A},\mathcal{T},\mathcal{R},\Omega,\mathcal{O},\gamma\}$. States $s \in \mathcal{S}$ are assumed to be continuous in $\mathbb{R}^n$. The action space $\mathcal{A}$ consists of a Cartesian product of the set of robotic actions $\mathcal{A}_m$, such as movements or other such decisions that purely affect the autonomy, and query actions $\mathcal{A}_q$, which denote a decision to request information from the human collaborator, such that

$$\mathcal{A} = \mathcal{A}_m \times \mathcal{A}_q, (a \in \mathcal{A}) = [a_m, a_q] \tag{5}$$

The probabilistic transition function $\mathcal{T}$, in this application, is specified as an n-dimensional Gaussian, with mean $s + \Delta a$, and variance Σ_a, where Δa is the n-dimensional vector of the expected resulting change occurring due to action a, and Σ_a is an $n \times n$ covariance matrix of transition noise resulting from action a. Executing action a results in a probabilistic transition from state $s \in \mathcal{S}$ to state $s' \in \mathcal{S}$.

$$\mathcal{T} = p(s'|s,a) = \phi(s'|s+\Delta a, \Sigma_a) \tag{6}$$

While this Chapter only considers transition functions of this unimodal Gaussian structure, they could also be constructed from Gaussian Mixtures as shown in prior work [9], in order to capture multi-modal uncertainties. This would require no significant algorithmic or theoretical changes to the methods described below. Furthermore, non-Gaussian models could in theory be applied in so much as their product with either softmax functions or Gaussian mixtures can be approximated as a Gaussian Mixture after the manner of the VB-POMDP algorithm described in [9].

The observation space $\mathcal{O}$ is decomposed similarly to the action space in that it consists of a Cartesian product of observations resulting from the robot's onboard sensors $o_v \in \mathcal{O}_v$ and the human's responses to the robot's query actions $o_q \in \mathcal{O}_q$, such that:

$$\mathcal{O} = \mathcal{O}_v \times \mathcal{O}_q, (o \in \mathcal{O}) = [o_v, o_q] \tag{7}$$

The observation model Ω relates the probability of receiving a particular observation o to the state s through the use of an n-dimensional softmax function, which can be constructed as per [7,8].

$$\Omega = p(o|s,a) = \frac{e^{w_o^T s + b_o}}{\sum_{j=1}^{|\mathcal{O}|} e^{w_j^T s + b_j}} \tag{8}$$

The probability distribution over the current state s at time t, referred to in this work as the belief b, is represented as a Gaussian Mixture pdf consisting of M weighted mixands,

$$b(s) = p(s_t|a_{0:t}, o_{0:t}) = \sum_m^M w_m \phi(s|\mu_m, \Sigma_m) \tag{9}$$

$$1 = \sum_m^M w_m \tag{10}$$

2.6. General POMDP solution

The general solution to the POMDP is a policy $\pi(b)$, which maps from a belief to an action $\pi(b) \rightarrow a$, such that the expected discounted reward function $\mathcal{R}(s,a)$ over time is maximized. Thus the value function V under a particular policy π and starting belief b_0 is

$$V^{\pi}(b_0) = \sum_{t=0}^{\infty} \gamma^t E[\mathcal{R}(s_t,a_t)|b_0,\pi] \tag{11}$$

for a given time discount factor γ.

As shown in [44], the policy $\pi(b)$ can be represented as a set of piece-wise continuous functions, each of which correspond to a non-exclusive action. These piece-wise functions, when in a continuous state space as in [40], are known as α-functions, and are constructed as Gaussian Mixture models over $\mathcal{S}$. A given action may correspond to multiple α-functions, in a one-to-many fashion, while each function can be associated with only one action. These α-functions are collected in the set Γ, which is used to find the policy $\pi(b)$ for a given belief as the action attached to the α-function maximizing its continuous dot product with the belief,

$$\pi(b) = \arg\max_{\alpha\in\Gamma} \langle \alpha(s), b(s) \rangle \tag{12}$$

Such a policy is obtained in the manner of the VB-POMDP algorithm [9], using dynamic programming to solve the Bellman backup equations with a Variational Bayesian approximation to fuse the softmax observation functions with Gaussian Mixture α-functions.

3. Hierarchical CPOMDP collaborative semantic sensing

This section proposes a hierarchical CPOMDP solution to the formal problem statement in Section 2.5. First, the problem is cast into a target search and localization application in Section 3.1, and addressed in a single-policy fashion in Section 3.2. Sections 3.3–3.8 detail arguments for and implementation of a hierarchical structure which more scales more effectively and incorporates human information input.

3.1. Application to dynamic target search and localization

The grounding CNR problem is now formulated in the CPOMDP setting. As stated above in the general formulation, it is an infinite horizon POMDP 7-tuple $\{\mathcal{S},\mathcal{A},\mathcal{T},\mathcal{R},\Omega,\mathcal{O},\gamma\}$. The autonomous agent referred to as the "cop" has state $s_c \in S_c$ while the target "robber" has state $s_r \in S_r$ such that $S = S_c \times S_r \in \mathbb{R}^4$. The cop may initiate movement actions a_m and human query actions a_q such that the full discrete action space is $A = a_m \times a_q$. Movement actions a_m dictate changes from the cop's current state s_c to its next s'_c, modeled alongside the robber's motion through a conditional Gaussian transition model.

Query actions a_q have no effect on the state, therefore the transition model is held independent of them.

The cop is rewarded only for being co-located with the robber, such that a robber in the same position as a cop is held to be captured. This reward is expressed in the reward model $\mathcal{R}$, such that for some distance threshold τ,

$$\mathcal{R} = \begin{cases} 100 & dist(s_c,s_r) \leq \tau \\ -1 & dist(s_c,s_r) > \tau \end{cases} \tag{13}$$

Note that the reward function in this case is not explicitly dependent on actions, but rather implicitly through the state transitions actions cause.

The cop receives observations from two sources, first being those from its on-board camera view o_v, which can take values $o_v \in \mathcal{O}_v = \{Detect, No\ Detect\}$. The viewable area of this cone is assumed to have angle θ projected to a forward flat leading edge of length L_v in the direction of movement. The results of human query actions a_q are modeled as additional action dependent observations $o_q \in \mathcal{O}_q = \{Yes, No\}$, such that $\mathcal{O} = \mathcal{O}_v \times \mathcal{O}_q$ with a set size $|O| = 4$. All observations are generated from corresponding softmax observation models Ω for some known fixed dictionary, and the time discount $\gamma \in [0,1]$ is here set to 0.9. While a $\gamma = 1$ would correspond to a truly infinite horizon policy, most practical problems require some temporal discounting for policy convergence [45].

3.2. Single policy implementation

The implementation of the policy described above results in a single-policy, or monolithic, POMDP, wherein a single belief $b(s)$ is maintained which spans the state space $\mathcal{S}$ and a single policy maps said belief to an action. Obtaining such a policy for an active human sensing solution to a robotic search problem follows the four primary steps below, and is summarized in Algorithm 1:

(1) Identify and label salient semantic features in the state space. These could be objects, such as chairs and buildings, spatial areas such as rooms or neighborhoods, or categorical identifiers such as high, medium, and low on a continuous scale. Note that semantic features need not exist exclusively in Cartesian space. However, their ability to be interpreted by a human collaborator should be considered.

(2) For each semantic feature, identify and collect a set of semantic relational indicators with respect to the feature. These indicators can include global bearings such as North and East, local bearings such as “In front of” and “Behind”, or binary existence indicators such as “Inside” and “Outside”, as well as any other types of relation relevant to the problem at hand. Note, while not mathematically required, relational indicators for human collaboration problems should generally be intuitive or understandable to the human, as queries from the autonomous system will be drawn from this set. Including “Inside” as an indicator for the semantic feature “Chair” may not make sense in most problems, even if chair takes up a defined spatial area which a point could technically be inside of. The total collection of semantic labels and relational indicators make up the semantic dictionary for the problem.

(3) Using the softmax synthesis methods drawn from previous work [8,30], geometrically construct softmax functions around the spatial extent of each feature, labeling the resulting classes from the semantic dictionary. This could be in a one-to-one fashion, or a multi-modal softmax representation in a one-to-many approach where each label combines multiple classes. The query action set A_q should consist of each member of the semantic dictionary, while the query observation function Ω should now contains the labeled softmax class weights and biases.

(4) Having combined A_q with a problem appropriate movement set A_m as in the previous section, apply the VB-POMDP algorithm from Algorithm 2 to the 7-tuple $\{\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \Omega, \mathcal{O}, \gamma\}$ with problem dependent choices for $\mathcal{T}, \mathcal{R}$, to obtain the policy π.

Algorithm 1 Active Human Collaboration POMDP Construction

```
Input: States $\mathcal{S}$, Transitions $\mathcal{T}$, Rewards $\mathcal{R}$, Discount $\gamma$
Hand select semantic feature set $\{f\}$ from state space
Assign semantic label set $f_l$ to each feature
$A_q = \sum_f \{f\}_l$
$\Omega = p(o|s,a) =$ Softmax Synthesis($f$), $\forall a \in A_q$
$\pi$ = VB-POMDP($\{\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \Omega, \mathcal{O}, \gamma\}$) #Algorithm 2
return $\pi$
```

Actions chosen during policy execution $\pi(b)$ will now be drawn from the set $A = A_m \times A_q$, which can then be disambiguated and used appropriately. The queries A_q and their resulting human observations have the effect of actively pointing a human sensor with respect to the chosen semantic dictionary, while the movement actions A_m effect the state and potential rewards more directly. It is imperative to note that while here actions A_m are referred to as movements, that does not constrain the frameworks discussed here to only physically embodied mobile robots. Rather, movement references a change in abstract state affected from an autonomous agent. Indeed, even the “human” part of the human query actions A_q is not strictly necessary. Any agent, physical or virtual, need only be capable of influencing the state through decisions and querying an external information source for probabilistically modeled semantic observations in order to implement this framework.

3.3. Formal hierarchical solution

In this section a general hierarchical framework is proposed for human-autonomy sensing and planning problems. The continuous state space $\mathcal{S}$ is fully partitioned into a set of non-overlapping lower level state subspaces $\mathcal{S}_l$, such that for each label (l), $\mathcal{S}_l$ is a proper subset of $\mathcal{S}$ which shares exactly zero states with other subspaces.

$$\forall l,\ \mathcal{S}_l \subset \mathcal{S} \tag{14}$$

$$\mathcal{S} = \cup_l \mathcal{S}_l \tag{15}$$

$$\mathcal{S}_a \cap \mathcal{S}_b = \emptyset,\ \forall (a \neq b) \tag{16}$$

This partitioning occurs along discontinuous transition boundaries in $\mathcal{S}$, where the Gaussian transition model outlined in Section 3.1 fails to hold. Partitions need not be of equal size, nor are they necessarily restricted to any uniform regularity conditions such as shape or dimensionality. Each subspace S_l is then treated as a separate CPOMDP, and a policy π_l is found, with actions A_l and observations O_l. In general, partitions are predetermined by hand, and transitions between partitions should be representative of state space transitions in the unpartitioned space. For example, a neighborhood might “naturally” decompose into blocks, while transitions between blocks occur through a 4-connected or 8 connected grid. Alternatively, an indoor environment might be partitioned as a set of rooms, where the locations of doorways govern the transition between rooms.

A meta-space S_h is defined as the set of labels (l), and consists of a state space for a higher level discrete POMDP. The action space A_h of this POMDP corresponds to a Cartesian product of movements between partitions $A_{m,h}$ and human queries regarding each partition $A_{q,h}$,

$$A_h = A_{m,h} \times A_{q,h} \tag{17}$$

Similarly, observations O_h are given with respect to each partition as $O_{v,h}$ and queries regarding partitions $O_{q,h}$,

$$O_h = O_{v,h} \times O_{q,h} \tag{18}$$

The Gaussian Mixture (GM) belief $b(s)$ over the non-partitioned state space $\mathcal{S}$ is broken up into a set of conditional beliefs $b_l(s) = p(s|l)$. This is accomplished with a hard classification of each mixand to the partition containing its mean, creating the set $\{\omega_l\}$. Each mixand's mean in $\{\omega_l\}$ is constrained to stay within its assigned partition under dynamics updates, and any probability density originating from mixands outside the partition is ignored. The belief for the discrete meta-space S_h then becomes the sum of the weights for each element of $\{\omega_l\}$,

$$b(l \in S_h) = \sum w_m \mathbb{1}(w_m \in \{\omega_l\}) \tag{19}$$

$$\sum_l b(l \in S_h) = 1, \tag{20}$$

where $\mathbb{1}(w_m \in \{\omega_l\})$ denotes the indicator function with respect to the spatial position of the mean within $\{\omega_l\}$. As the original belief $b(s)$ over $\mathcal{S}$ is required to be a proper pdf, this ensures that $b(l)$ is a proper pmf. The conditional beliefs for each partition can then be weighted by their respective probability $b(l)$, and the belief $b(s)$ over the full state space can be extracted from a sum over subspaces,

$$b_l(s) = p(s|l) = \frac{1}{b(l)} \sum_{m}^{M} w_m \phi(s|\mu_m, \Sigma_m) \mathbb{1}(w_m \in \{w_l\}) \tag{21}$$

$$b(s) = \sum_{l} b(l) b_l(s) = \sum_{l} p(l) p(s|l) \tag{22}$$

Of note, this leads to the slightly paradoxical assumption that a mixand mean transitioning towards a partitions boundary which would otherwise be traversable, such a doorway between rooms, will remain bound in its original partition. Rather than govern the flow of probability density between rooms in such a manner at the partition level, such flow is mediated at the meta state space level S_h, with probability mass being allocated among discrete states by a uniform change to each partitions contained mixand weights.

The observation models in the lower level CPOMDPs O_l are specified as softmax functions. These can be fused directly into the full space belief $b(s)$ by way of the Variational Bayes algorithm described in [7],

$$p(s|O_l) \approx \frac{p(s)p(O_l|s)}{p(O_l)} \tag{23}$$

Observations in the higher level discrete POMDP can be fused directly into $b(l)$ using a discrete Bayes Filter.

In terms of policy execution, action output $a_{m,h}$ from the discrete POMDP indicates which CPOMDP policy π_l to query. The action generated by $\pi_l(b_l(s))$ is then taken. Human queries are then generated from both $a_{q,h}$ and $a_{q,l}$. Responses to these queries, as well as robotic observations $o_{v,h}$ and $o_{v,l}$ are then fused back into the belief before requesting another action. This approach results in a set of CPOMDP policies, governed by a single discrete POMDP. Each policy can be solved independently and combined during runtime.

3.4. Application to dynamic target search and localization

Here the general hierarchical POMDP formulation from the previous section is applied to the motivating CNR problem for target search in complex indoor search spaces, e.g. see Fig. 2. A separate CPOMDP policy is found for each distinct room in a particular map, where obstacles are sparse enough not to necessitate the switching modes used in [41]. Each of these room level policies is then treated as an action selection by a discrete POMDP policy over the rooms. Transitions between rooms in S_h are governed by a discrete transition model, as shown in Fig. 5. This leads to a novel hierarchical CPOMDP policy that can not only take fuse low-level semantic soft information about target locations in metric physical space (e.g. ‘next to the chair’; ‘not by the refrigerator’), but also exploit higher-level semantic data about target locations in abstract label spaces, i.e. room designations (‘in the kitchen’; ‘not in the dining room’). By accounting for the dependencies between these different types of high-level and low-level semantic data, we arrive at an intelligent hierarchical decision making policy that enables top-down motion planning (i.e. determine which areas to search, and then how to search them), as well as determination of the best set of high-level and low-level semantic queries for a human sensor that will ensure rapid capture of the robber. While this approach carries additional approximations, such as shifting consideration of the case where the cop and robber are not in the same room to the higher-level discrete policy, it allows the consideration of problems at the scale of CNR.

3.5. Lower level CPOMDP

The lower level CPOMDP for each room is specified in the same manner as the formal problem statement in the previous section, with state $S_l \in \mathbb{R}^4$ for room (l). The cop's state variables are changed deterministically with actions while the robber's are assumed to be drawn from a high variance Gaussian random walk,

$$p(s_r') = \phi(s_r'|s_r, \Sigma_r) \tag{24}$$

The vector Δa is necessarily zero for the robber states s_r, as the cop's actions do not directly effect the robber's movements.

This Nearly Constant Position (NCP) model introduces an approximation when dealing with most real systems. While a drifting or unpowered target in a dynamic environment might adhere well to a Brownian motion scheme, intentional agents rarely do. Instead they tend to operate according to trajectories or patterns. The CPOMDP framework shown in [40] is equipped to handle transition functions which can be modeled as conditional Gaussian distributions with their mean shifted by the actions Δa expected effect on the state s:

$$p(s'|s,a) = \phi(s'|s + \Delta a, \Sigma_a) \tag{25}$$

When only a limited number of state components are directly controllable, the others are forced to execute a Gaussian random walk with the given variance. This prevents the use of target trajectories in search problems such as the one described here. In order to incorporate target dynamics, it is desirable to have a transition function of the form,

$$p(s'|s,a) = \phi(s'|Fs + \Delta a, \Sigma_a) \tag{26}$$

where F is the state transition matrix which encapsulates changes in the state independent of actions. Bellman backups can be easily resolved with this alteration in the CPOMDP framework, thus permitting solutions to a broader class of continuous space planning problems, as shown in [9]. As the focus of this work is the hierarchical CPOMDP structure, and in an attempt to minimize the dimensionality, velocities are not included here.

The cop can choose from among 5 noisy movements $A_m = \{East, West, North, South, Stay\}$, and can ask questions about the robber's spatial relation to each object in the room such that $A_q = \{Objects\} \times \{Left, Right, Front, Behind\}$. Of note, while the question space could have been based in global compass coordinates, they are instead represented in local body coordinates. This is due to the fact that the cameras through which the human views the space, both the cop's on-board view and security cameras, contain no explicit global coordinate reference. While the provided map in the interface could allow the human operator to reason about global coordinates by knowing the camera placements, cognitive load is decreased by instead assigning orientations to each object and having the human refer to them each in their local frame. For instance, the chair object is set up with a 90 degree counterclockwise rotation, so the observation “Front” with respect to the chair indicates the area to its west. The full discrete action space is then $A = A_m \times A_q$. An example action might be “Move East and ask ‘Is the robber in front of the fern?’”.

In addition to human information resulting from its queries, the cop receives viewcone observations through its onboard camera, which is capable of visually detecting the robber at close range. Given that the cop's viewcone is fixed to the direction of travel along one of four directions, and has a leading edge of length L_v, the area of the box will correspond exactly to the area swept out by the leading edge of the viewcone during the preceding action. This can be shown by integrating the length of the leading edge of the view cone in the direction of travel. So for any movement, the magnitude of the approximated area A_a swept out is:

$$|A_a| = \int_{s_c}^{s_c'} L_v ds_c = \int_{0}^{\Delta s_c} L_v ds_c = L_v \Delta s_c \tag{27}$$

Therefore any movement of m meters ($\Delta s_c = m$) behind a leading edge L_v meters long will produce an area of approximately $L_v \Delta s_c$ square meters which would have triggered the viewcone observation o_v during the time in which the movement was executed. From Eq. (27), it is clear that for different cop displacement or viewcone parameters the box approximation can be modified appropriately. The box model remains an approximation however, due to the fact that most of the area swept out by the viewcone will be in front of the cop, whereas we assume the cop to be centered in that area. Furthermore, the box approximation does not perfectly account for area swept out without contact with the leading edge of the viewcone, and sacrifices distant coverage in the direction of movement in favor of coverage near to the cop, as shown in Fig. 3. Thus while the area of the box represents a shifted area of similar size to that swept out by the viewcone, its size serves as a lower bound for that reached by the entirety of the viewcone. The approximation improves as the viewcone angle θ is increased while holding L_v constant, such that the area of the viewcone A_v becomes an infinitesimal area along L_v. Thus the approximation area magnitude $|A_a|$, and true area coverage magnitude $|A_t|$ approach each other in the limit of the angle as:

$$|A_v| = \frac{1}{2} L_v \tan\left(\frac{180 - \theta}{2}\right) \tag{28}$$

$$|A_t| = |A_v| + \int_{s_c}^{s_c'} L_v ds_c \tag{29}$$

$$\lim_{\theta \to 180} |A_v| = \lim_{\theta \to 180} \frac{1}{2} L_v \tan\left(\frac{180 - \theta}{2}\right) = 0 \tag{30}$$

$$\lim_{\theta \to 180} |A_t| = |A_a| = L_v \Delta s_c \tag{31}$$

This limit, corresponding to a coverage area of a thin line immediately leading the robot, results in an approximation area which leads the true area, rather than the opposite which occurs with our hardware. Including orientation into the state vector would remove the need for the box approximation altogether, but further increase the dimensionality of the state. Thus to reduce the size of the state space for this implementation, the box approximation was used. Alternatively, the viewcone observations could carry a dependence on the movement action taken, transforming the likelihood model $p(o_v|s)$ to $p(o_v|s, a_m)$ where a_m acts as a switch between a number of likelihood models corresponding to the correct orientation for that action. While this modification increases the complexity of implementation somewhat, it is valid within the CPOMDP framework and will be explored in future work.

It is important to note that the action and observation spaces in the problem could be implemented without recognition of their ability to factor into distinct sets $\{A_m, A_q\}$ and $\{O_v, O_q\}$. However, this increases the difficulty of implementation when trying to account for the diverse results of the combined action/observation. In this case, the cop's movement actions A_m primarily effect the state without changing the observations, while the cop's 'question actions' A_q have no effect on the state at the current time, and fully dictate the meaning of the observations O_q. Similarly, the viewcone observations O_v are only state dependent and thus independent of either action, while O_q depends on both state and action. Factoring each space into its constituent parts allows for simpler handling of these dependencies, and increases the explainability of the cop's actions and the changes in its beliefs. The differences in the two approaches are summed up in Fig. 4. Each room's lower level continuous POMDP policy is found using the VB-POMDP algorithm detailed in [9].

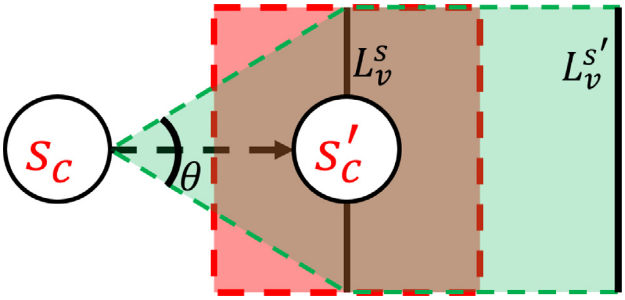


Fig. 3. A comparison of the area swept out by the cop's viewcone in one timestep (green) vs. the area of the box approximation (red) for a viewcone with angle θ when moving from position s_c to s_c'. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3.6. Higher level discrete POMDP

The higher level discrete POMDP is specified on a state vector consisting of all rooms. For example, the room configuration shown in Fig. 2 corresponds to the state space:

$$S_h = \{\textit{Billiard Room}, \textit{Hallway}, \textit{Kitchen}, \textit{Dining Room}, \textit{Study}, \textit{Library}\} \tag{32}$$

as shown in Fig. 5. The state s_r represents the robber's current position, and the robber randomly transitions according to the particular connections between rooms in the map being used with probability $p(l'|l)$ independent of actions. The location of the cop is left out of the high level POMDP, instead being accounted for by a combination of the high level actions and low level state. The cop can choose movement actions $a_{m,h}$ corresponding to each room, which will deterministically move the cop to that room according to

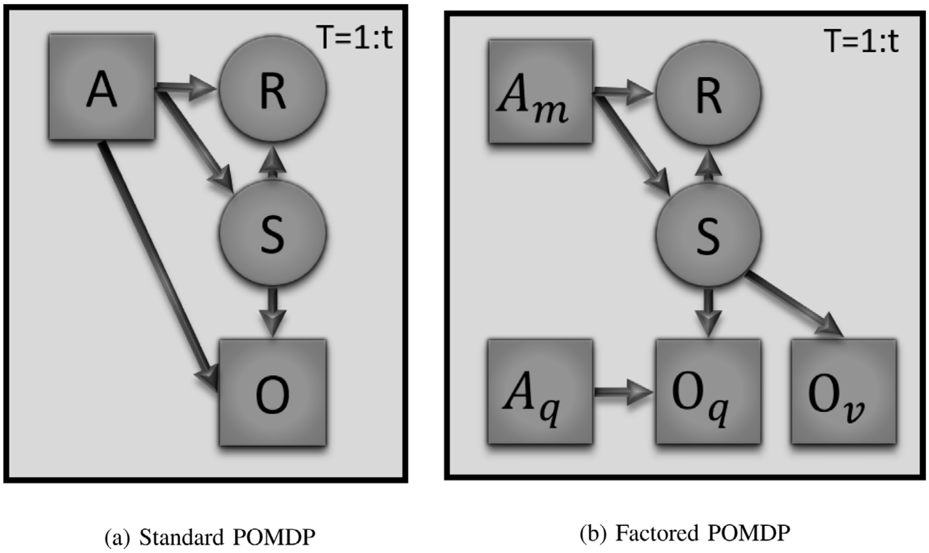
$$p(s_c' = l | a_{m,h} = l) = 1 \tag{33}$$
$$p(s_c' \neq l | a_{m,h} = l) = 0$$

as well as questions actions $a_{q,h}$, which will ask the human if the robber is in a particular room. Thus for the higher level policy, both $A_{m,h} = S_h$ and $A_{q,h} = S_h$. If the $a_{m,h}$ indicates the room the cop currently occupies, the movement action of the lower POMDP policy is respected within that room as per Algorithm 2. In general, as a product of the PBVI [37] roots of the CPOMDP family of algorithms, similar beliefs will lead to similar actions. Therefore in practice movement between rooms tends to only occur after either a thorough search of a room or significant shift in belief, and the cop avoids bouncing back and forth between rooms without searching either. This behavior is confirmed in the experimental results below. As with lower level policies, the full action space is $A_l = A_{m,l} \times A_{q,l}$. An example action would be "Search the Library and ask 'Is the robber in the Kitchen?'".

In the higher level discrete POMDP, the cop is rewarded for choosing to move to the room the containing the robber, and penalized for choosing the wrong room according to

$$\mathcal{R}(s_h, a_{m,h}) = \begin{cases} 100 & a_{m,h} = s_h \\ -1 & a_{m,h} \neq s_h > \tau \end{cases} \tag{34}$$

The cop receives viewcone observations $O_{v,h}$ at each time step, similar to the lower level CPOMDP. Given that being in the same room as the robber does not guarantee a viewcone detection, likelihoods for $O_{v,h} = \textit{Detect}$ are fairly low for any given time step even if the cop and robber are in the same room. With respect to



(a) Standard POMDP

(b) Factored POMDP

Fig. 4. POMDP graphical models with different action/observation factorizations.

Algorithm 2 Hierarchical CPOMDP Evaluation

```
  Input: Discrete Policy $\pi_h$, Continuous Policy Set $\{\pi_l\}$
  With Belief $b(s)$ and Cop State $s_c$
  while $o_v \neq Detect$ do
      $l = a_m^h = \pi_h(b)$
      $a_m^l = \pi_l(b)$
      if $s_c \in l$ then
          $s_c' \sim p(s_c'|s_c, a_m^l)$
      else
          $s_c' \to l$
      end if
      $a_q = N\_Actions()$ #Algorithm 3
      $o_q =$ Human Responses
      $b = Belief\_Update(b, a_m^l, a_m^h, o_v, o_q)$
  end while
```

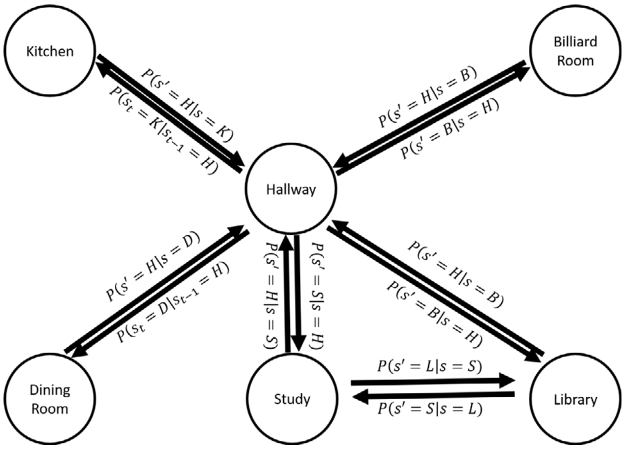


Fig. 5. The states and transitions for the higher level discrete POMDP corresponding to the room layout in Fig. 2.

$O_{v,h}$ for the high level POMDP, rooms are treated as identical, such that each has an identical detection rate ν

$$p(o_{v,h} = Detect|a_{m,h} = l, s_h = l) = \nu, \forall l \tag{35}$$

This approach could be refined by considering the ratio of the area covered by the viewcone and the total area of the room, such that each rooms detection likelihood reflects its size, or by leveraging prior knowledge about likely robber movements and positions within each room.

It is important to note that during implementation of a policy, the observations $o_{v,h}$ and $o_{v,l}$ are only derived from the actual visual system, and are not double counted as separate observations from the high and low level systems. Viewcone observations are only modeled as above when finding a high level policy, to approximate the response of the low level viewcone. Also, as the policy is solved over the state space S_h, each additional spatial area considered adds only 1 additional state. This additional lower level policy does not contribute any complexity to the other room policies, allowing the hierarchical structure to scale well to larger numbers of spatial areas.

We use the Point-Based Value Iteration (PBVI) algorithm from [37] to find the policy π_h for the discrete layer.

3.7. Hierarchy and question lists

At each time step, the policy chooses an action consisting of a room to search and a room to query. If the cop is outside the search room, it is directed to go there. Otherwise, if the cop is already in the search room, the lower level CPOMDP policy is queried to provide a movement action. The query room is asked about in the form "Is the robber in (room)?", and the low level CPOMDP policy for that room gives an additional question about the robber's relation to an object in that room.

In this implementation the human sensor receives a question from the cop at every time step. Because the policy was trained to expect responses from the human sensor, steps where the human fails to answer are unknown events from the system's standpoint, i.e. they are not accounted for when solving for the policy. One method for handling these failures would be to include a "Null" observation with a uniform likelihood across states to represent a lack of human observation. Further steps could be taken by incorporating a form of human attention model into the state vector and an option to ask a "Null" question when the policy believes the human would not be able to answer. Each of these methods increases the problems complexity, either by enlarging the observation space for the first or by enlarging the state, action, and observation spaces for the second. Also, it should be noted that the policies for both the Higher and Lower level POMDPs map from *any* belief to an action. Both PBVI and VB-POMDP solve policies from a set of example beliefs that the system may encounter, and interpolate between them when an unseen belief is encountered during runtime. A belief resulting from an unmodeled human failure would in most cases not have been explicitly explored during policy solution. Yet it is likely sufficiently close to a belief that was such that a suitable action can be found. Therefore any effects of this discrepancy can be minimized, if not entirely negated, by solving over a sufficient set of example beliefs.

In most applications, the desired output of a POMDP policy is the action with the highest value for the current belief. This makes sense in most contexts as only one action can be taken at a time. However, in our problem multiple questions could be displayed to the human at each time step, and so we want to ask the N most valuable questions. In both discrete and continuous policies using PBVI-type approximations each policy element, or α-element, contained in Γ corresponds to an action and encodes part of the approximate value function over beliefs. As each α-element is specified over the entire belief space, it can provide a value for its action at any belief, even were it does not provide the maximum value. Therefore, the α-elements with the top N values can be said to correspond to the top N actions, as shown in Fig. 6. As multiple α-elements might correspond to the same action, this does not guarantee N unique actions. However, as all α-elements must be evaluated to choose the correct action for a belief, the top N unique actions can still be chosen. This also implies that choosing a list of actions requires only the minimal extra computation of a sorting function substituted for an argmax, as in Algorithm 3. This method of choosing the top N actions is ultimately a heuristic, equivalent to asking “Given the policy available right now, if the existence of the top $N-1$ actions were ignored, what would the best action be?”. This disregards the fact that most PBVI-type algorithms prune away a large portion of α-elements which do not maximize the value of any particular belief, and are not actively collecting second-best or third-best actions. Therefore the only actions identified by this heuristic will be ones associated with α-elements which maximize the value of a different belief. While at the core of the PBVI approach rests the assumption that similar beliefs will generally have similar actions, the different beliefs maximized by a second or third best α-element are not guaranteed to be similar to the current belief.

While Algorithm 3 can identify a ranked set of action preferences, it is important to note that most applications can only execute a single action for a given timestep. In such cases the output of Algorithm 3 can be used to adapt to unexpectedly forbidden actions or recover gracefully from system failures which disallow actions. The application detailed in this work varies slightly, in that while a single movement action a_m can be taken, multiple query actions a_q can be posed simultaneously without conflict. Thus the list returned from the algorithm can be filtered for only actions pertaining to the highest value movement action. This filtered list then makes up a ranked set of preferred questions, all corresponding to the same robotic movement.

Algorithm 3 Choose Top N Actions

```
Function: $N\_Actions$
Input: Policy $\Gamma$, Belief $b(s)$, N
for $\forall\alpha \in \Gamma$: do
    V($\alpha$) = $\int \alpha(s)b(s)ds$
end for
list = sort(V)
return list[0:N]
```

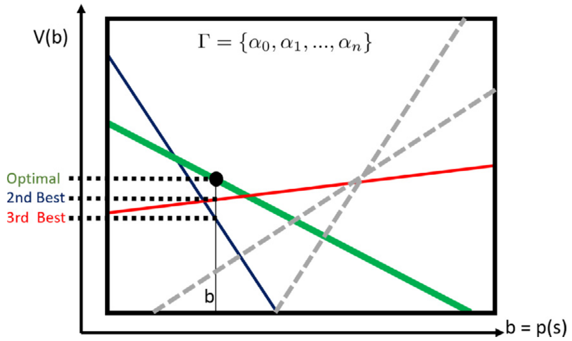


Fig. 6. Application of the N_Actions() algorithm for belief b and $N = 3$. Policy elements α who have the greatest value at b are chosen, and their questions presented to the human collaborator.

3.8. Hierarchical policy implementation

The implementation of the framework described above results in a hierarchical POMDP policy, composed of multiple monolithic continuous state policies governed by a single discrete space policy. Obtaining such a structure for an active human sensing solution to a robotic search problem follows the four primary steps below, and is summarized in Algorithm 4:

(1) Partition the state space. In the case of the CNR problem, this is done according to rooms, with each partition's boundaries matching precisely with the rectangular walls of each room. Transitions between rooms are mapped out according to doorways, wherein rooms without doorways have probability zero of transition. In the general case, depending on the appropriate target model, transition probabilities should either be distributed randomly between adjacent states or according to inferred target intent. Assign each partition a semantic label (l), creating a labeled continuous state space S_l, keeping in mind that these labels should be intelligible to a human collaborator. The set of labeled partitions becomes S_h, while the transition model becomes $\mathcal{T}_h$.

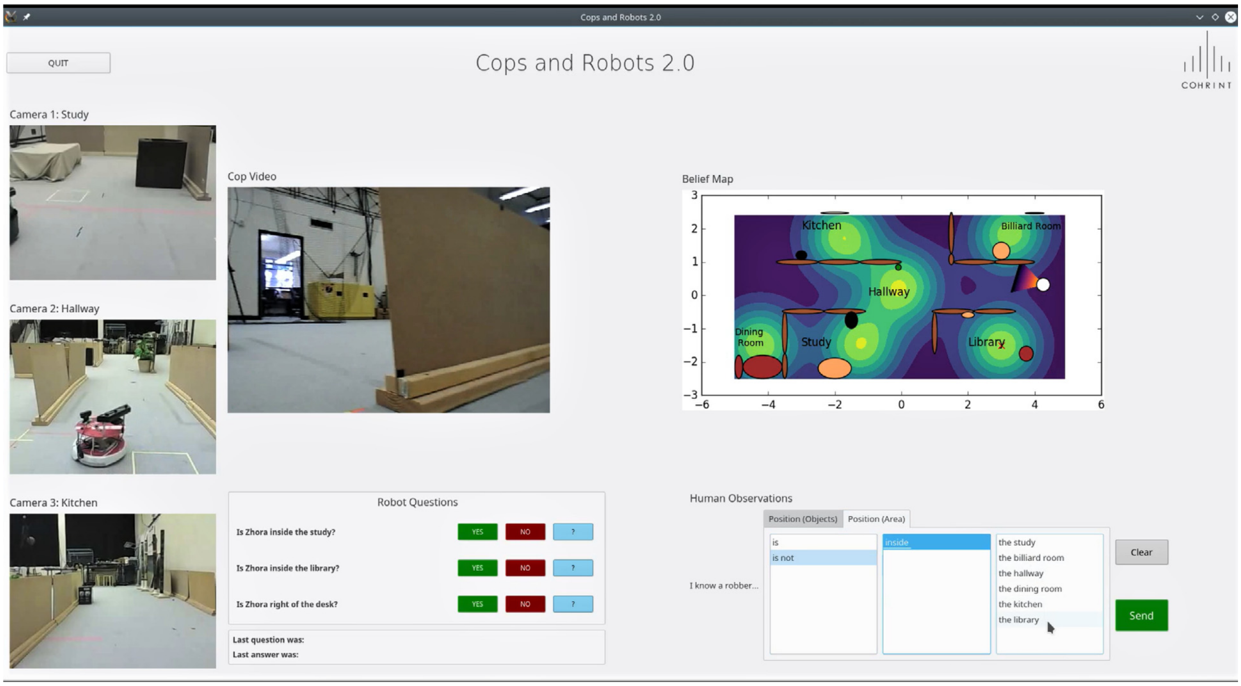
(2) Following the process laid out in Algorithm 1, create a semantic dictionary from labeled features within each partition. Construct the action set $A_{q,l}$ for each partition, containing relational indicators attached to each semantic feature, along with a query observation function Ω_l

(3) Solve each partition's associated POMDP using the VB-POMDP Algorithm [9], and store each resulting policy π_l in a lower level policy set.

(4) Using the PBVI algorithm [37] and the higher level S_h and $\mathcal{T}_h$, solve for the high level discrete policy across partitions π_h. This policy, along with the lower level set π_l can then be implemented using Algorithm 2.

Algorithm 4 Hierarchical Human Collaborative POMDP Construction

```
1: Partition and label state space $\mathcal{S}$ into set of $S_l$
2: Construct semantic dictionary and models $A_{q,l}$, $\Omega_l$, $\forall l$ using Algorithm 1
3: $\pi_l$ = VB-POMDP($S_l$) $\forall l$
4: $\pi_h$ = PBVI($S_h$)
```



Is Zhora inside the study?

Is Zhora inside the library?

Is Zhora right of the desk?

Last question was:

Last answer was:

I know a robber...

is

is not

inside

the study

the billiard room

the hallway

the dining room

the kitchen

the library

Fig. 7. Cops and Robots user interface: 'robot pull' queries are answered in the lower middle panel with 'Yes/No' buttons; voluntary 'human push' sensor inputs are provided with the structured text input on the lower right panel.

4. Results on the CNR experimental testbed

Hierarchical CPOMDPs were implemented and tested on the Cops and Robots (CNR) hardware platform at the University of Colorado at Boulder's Research and Engineering Center for Unmanned Vehicles. Cops and Robots is a physical simulation of a home environment, as described in Section 2.1. For the data presented here, a single human participant played the role of the deputy assigned to assist the cop robot. Semantic labels assigned to the rooms on a given map and objects within each room. These labels are known to both the cop and human, which allows for communication of information through a fixed codebook of possible observations. Individual robotic agents playing the part of the cop and robber were instantiated on Turtlebots running from an Odroid U3 microcontroller on iRobot Create platforms. The cop was equipped with an Xbox Kinect to both relay video to the human and visually detect the robber. Given goal positions from the Hierarchical CPOMDP, the cop implemented low level navigation using the A* algorithm and an occupancy grid representation of object and wall locations. A VICON motion tracking system was used to both provide the cop's fully observable data during runtime and track the ground-truth robber location for post-run data analysis. All elements of the hardware system are networked using a Robot Operating System (ROS) [46] layer, which provides a node based publisher/subscriber interface inter-component communication.

In this work a maximum of three exclusively unique objects per room are considered, each of which has a fixed position and orientation as in Fig. 2. In Fig. 2, each room has either 1 or 2 objects, each of which carries 4 ego-centric relational indicators such as "In front of (Object)" or "To the left of (Object)". This results in a semantic dictionary, and corresponding query action set A_q, of size 4 or 8 depending on the room. With the movement action set defined in Section 3.3, where $|A_m| = 5$, this results in $|A| = 20$–40 per room. Given the arrangement of six rooms, with 10 objects total, framing the CNR problem as a monolithic POMDP via Section 3.2 would result in an action set of size $|A| = |A_m| \times |A_q| = 5 \times 50 = 250$. Given the one-to-many nature of action to α function assignments discussed in Section 2.6 this implies an optimal monolithic policy would likely be described by significantly more alpha functions than actions. Each of these α functions must be regularly condensed both when finding and executing a VB-POMDP policy under the logic of Section 3.2, which when combined with the additional processing for action selection during policy execution results in a significant computational burden during runtime. This limits the ability of such a policy to be used on compute limited platforms, such as those used in CNR.

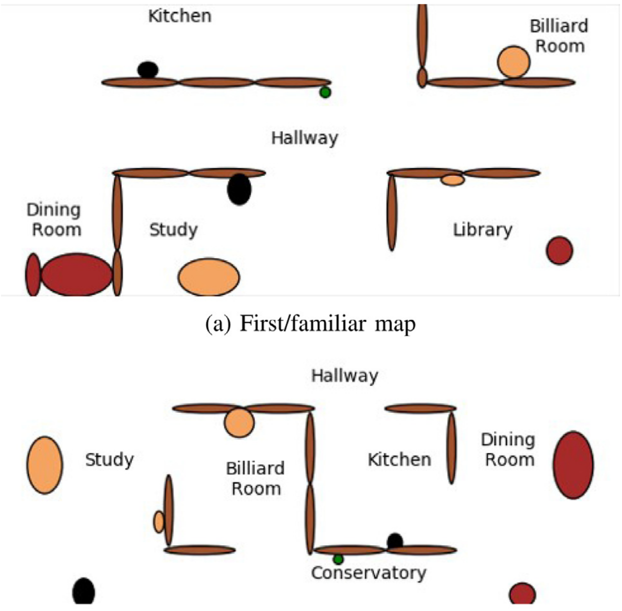
Furthermore, monolithic VB-POMDP policies explicitly assume Gaussian state transition functions. While this is inevitably only an approximation in physical hardware (such as at the bounds of a state space), such models are more dramatically inaccurate when dealing with the discontinuous transition environments in CNR, e.g. due to walls. Even if the policy could be constructed to prevent the cop from trying to transition through walls, the robber is assumed to be uncontrollable by the policy. This can easily result in policies solved accidentally assuming an intangible robber, the ghost of a robot perhaps, which further increase the unavoidable error between the policy model and the true hardware capabilities. Approaches such as Switching-Mode POMDPs [41] might alleviate this specific concern, but apply an additional complexity and computation layer on top of the already nearly intractable problem resulting from the large action set. Thus, a monolithic POMDP approach is impractical at best for this problem. However, the hierarchical techniques developed in Section 3.3 specifically allow large problems to be broken up into more manageable sizes, as well as allow discontinuous transition features to serve as subspace boundaries rather than mid-space obstacles.

To give insight into the practicality these more manageable problems, all lower level policies were solved on the time scale of a few hours, and the higher level discrete policy required a trivial solution time. For policy execution, previous work [9] has found the policy query time, or time required to extract an action from a policy, in VB-POMDP based algorithms to require only fractions of a second. Even given the additional computational overhead of the hierarchical policy structure in this work, this still compares favorably with the action execution time of the Turtlebot platform in use.

The human interface, shown in Fig. 7, visualizes the cop's belief about the robber's position as a heatmap, as well as the cop's position and viewcone detection range (See Section 3.5). The interface also displays a real-time feed from the cop's camera and various security cameras placed throughout the space. Each security camera is connected to a Raspberry Pi microcomputer, which relays imagery over ROS to the computer running the human interface. The security cameras each allow the human a fixed view of a room. The cop's camera facilitates observations in the cop's immediate area as well as a visual robber detection system. This system is implemented as an OpenCV blob detection algorithm [47], where a sufficient number of contiguous pixels in a given color range triggers a detection event. The robber robot is color coded bright red which is distinct from other colors in the environment, and the minimum threshold of the pixel count can be tuned to allow capture over a range of distances. While in this case a blob detection algorithm is used due to the simplicity of implementation and compatibility with available sensors, other proximity sensor modalities could be substituted without affecting planning or decision making, provided they can be modeled with similar types of likelihood models as described in Section 2.5.

The human plays the role of a sensor, voluntarily passing information to the cop through the semantic codebook embedded in the interface and answering binary 'yes/no' questions passed from the cop (e.g. 'Is the robber in the kitchen?'; 'Is robber in front of fern?'). The set of semantic statements constructable using the codebook represents a one-to-one correspondence with the classes of the cop's softmax function observation models. Therefore each observation which the human can volunteer has a corresponding form which can be used a robot query action. To further distinguish these forms of human statement, the following terminology from Section 2.1 and existing literature [19] is used. Robotic query actions in the following text are referred to as 'pull' or 'robot pull' actions, while volunteered human information is labeled as 'push' or 'human push' observations. Human statements generally followed the template form, "The robber (is/isn't) (relation) of (object)". For instance, the statement "The robber is in front of the dining table" initiates the same Bayesian belief update as an affirmative answer to the question "Is the robber in front of the dining table?". The human is also required to validate visual detections of the robber made by the cop, where a positive validation leads to successful capture of the robber and the end of the experiment run. Visual detection instances are suggested by the robot, eliminating the possibility of human error instigating false positives (where the human falsely indicated a successful capture). Also, in the data collected for this work, there were no instances of false negatives, where the human incorrectly rejected a successful capture. Thus the data analysis to follow need not account for such events, although other applications leveraging broader data sets or more complex settings may need to account for this possibility.

The Hierarchical CPOMDP policy approximation method was tested on two CNR maps, each with a different rooms structure. The first map, shown in Fig. 8a, consisted primarily of a hallway running the length of the space, with rooms branching off on both sides. The second map, shown in Fig. 8b, had the rooms in a semi-bipartite arrangement, with two sets connected through a long hallway and conservatory on the margins. In data collection, the human participant was fully familiarized with the first map beforehand, while the second map was presented as a previously unknown environment.



(a) First/familiar map

(b) Second/unfamiliar map

Fig. 8. Layouts for first (above) and second (below) maps.

Table 1
Times required for the cop to capture the robber in each scenario posed for the first map.

First map: All runtimes (s)				
Room\Case	NonHuman	HumanPush	RobotPull	Both
Library	307	80	51	69
Study	191	132	42	61
Kitchen	123	87	75	40

Each map was tested under 4 input conditions. As a baseline, the Hierarchical CPOMDP policy was implemented without human input, with the cop relying only on its visual sensor to gather information about the world. Second, the policy was implemented with a human who did not respond to the 'robot pull questions', but only provided 'human push' statements at their own discretion. Third, the policy received a human who only responded to 'robot pull' questions, and ignored 'human push'. Finally, the policy was implemented with a human who used both the 'robot pull' questions and 'human push statements' to give information. The resulting times required to catch the robber are summarized in Tables 1 and 2. The data shows that introducing human information, whether through 'push' or 'pull' data, shortens the time needed to capture the robber. Intriguingly, the case using both 'push' and 'pull' performs better than either singularly, implying that the robot is obtaining useful information the human did not volunteer through queries, while the human is able to push information the robot was not aware it needed, thus neatly complementing the strengths of each team member.

4.1. The familiar map

Across each input condition, tests were run with the robber's initial position in 3 different rooms: the Library, the Study, and the Kitchen. The cop's initial position was constant throughout all tests as the far right end of the hallway. The cop also held an identical initial belief for the robber state for each test, with belief dispersed equally between rooms. Each room's belief was initialized with a single Gaussian, with mean located at the rooms

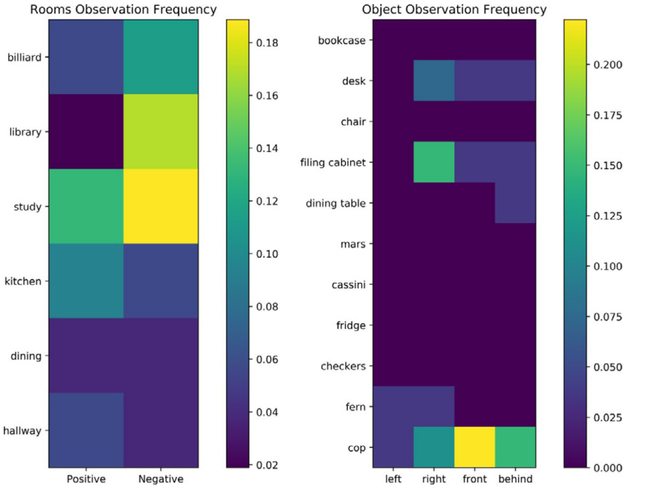


Fig. 9. Heatmap of the relative frequency of each human observation in the first map.

centroid and covariance chosen to extend probability density throughout the room.

The results show that across starting positions, the "Only Robot Pull" and "Robot Pull and Human Push" input conditions tended to require less time to catch the robber than either the "Only Human Push" or "No Human" input conditions. This is expected as the policy was computed assuming the robot would be able to pull information from the human, and thus the resulting information is accounted for in the robot's planning, while the 'push' information is helpful yet unexpected. Furthermore, the "Only Human Push" condition improved on the times for the "No Human" input condition over all cases, demonstrating the utility of unexpected human semantic sensor data.

Over all tests in the familiar map, 79 observations were given, averaging approximately 9 human inputs per test excluding the "No Human" condition. In terms of the propensity for the human deputy to provide negative information (e.g. "I know the Robber is not in the Study") vs. positive information (e.g. e.g. "I know the Robber is in the Study"), about 53% of all statements were positive relations. Limited to observations about rooms, the human only provided positive observations 40% of the time. When referencing objects in each room, 78% of observations were positive. The human referenced rooms about twice as often as they did objects, as shown in Fig. 9. Furthermore, the sparse nature of the right hand side of Fig. 9 indicates that many objects were rarely talked about by the human. While this peculiarity may subside given a larger dataset, it is possible that certain objects were not found as useful (explicitly or otherwise) by the human for their task. This implies the existence of a subset of salient features to which the full semantic dictionary could be reduced without substantially reducing the efficacy of the human–robot team. Ideally,

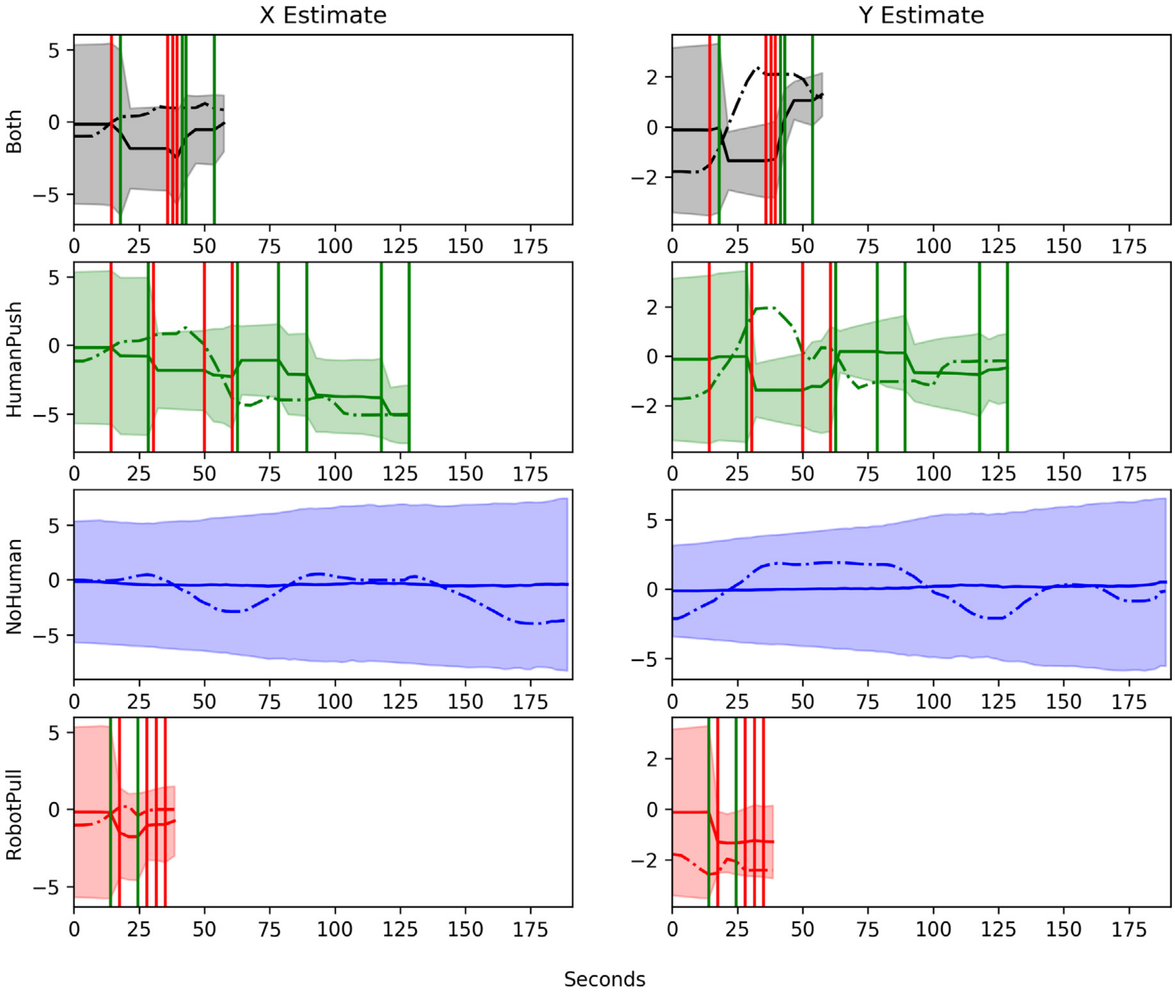


Fig. 10. Summary of cop's beliefs for the first map. Gaussian mixture mean and 2-sigma bounds of the cop's belief pdf for robber state are plotted against robber's true position (dashed line). Vertical lines are color coded for positive (green) and negative (red) human statements. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 2
Times required for the cop to capture the robber in each scenario posed for the second map.

Second map: All runtimes (s)				
Room\Case	NonHuman	HumanPush	RobotPull	Both
Billiard	176	86	87	46
Study	214	183	99	35

this insight could be used to dynamically construct ‘appropriate’ dictionaries for the task at hand, ensuring that planning efforts on the part of the autonomy are directed towards information the human is likely to engage with.

The cop’s beliefs are summarized in Fig. 10 for the 4 test runs with the robber starting in the Study. For each input type, the mixture mean and 2-sigma bounds are plotted along with the robber’s actual position. Belief compression, using the hybrid pre-clustering technique of prior work [9], was uniformly applied across rooms to maintain a maximum of 10 mixands per room. The “No Human” input condition sees the belief expanding faster than the cop can gather visual sensor data, as the robber is out of view and moving with unknown direction and velocity. Human semantic observations, shown as vertical lines in the plot, can cause dramatic belief shifts. The robber’s position is can be seen to be generally well bounded by the cop’s belief, which can correct for errors through additional human observations.

4.2. The unfamiliar map

For the second set of test scenarios, the human observer was familiar with the task and platform, but not with the map itself, shown in Fig. 8b. The locations of the rooms and positions of the objects within remained unknown to the human until the start of testing, in order to explore whether the human’s ability to communicate effectively with the robot was primarily an artifact of their ability to communicate about the current map. If the robot is similarly able to utilize both ‘push’ and ‘pull’ human information in an unknown (to the human, not the robot) environment, it implies that the advantage of human information is not diluted by the human’s preconceived biases or experience, but rather a result of optimal use of the human sensor in a general sense. Furthermore, the transition structure of the higher level discrete POMDP in the unfamiliar map varies significantly when compared to the known first map, providing additional insight into the ability of hierarchical human-collaborative POMDPs to plan in a variety of structured environments. As in the first map, 4 input conditions were tested over multiple initial robber positions, in this case the Billiard Room and the Study. Across all tests, the cop’s initial position was set in the Kitchen, and the belief was evenly distributed between rooms. The timing results from the test, summarized in Table 2, are generally comparable with those of the first map, taking an additional 11 s to catch the robber on average. The comparison between input conditions also remains consistent, with the unfamiliar map results even suggesting an additional advantage for the “Both” condition over “Robot Pull Only”.

For all tests in the second map there were a total of 66 observations, with an average of 11 human inputs per test excluding the “No Human” condition. In this case about 47% of all statements were positive relations, with 42% positives for room observations and 58% positive for objects. Rooms were referenced almost 3 times as much as objects, with frequencies for each statement shown in Fig. 11. Interestingly, the human discussed a broader variety of objects in the unfamiliar map, implying a predilection for certain salient features in the first map arose partly from familiarity. However, for each semantic object used by the human, a single relational indicator tended to dominate the relative frequency of observations for that object. This fact might have a geometric interpretation, such that for certain objects the robber was far more often ‘in front’ due to their placement in the room, or another facet of the human’s dynamic understanding of the map. In either case, the semantic dictionary used by the human ended up being much sparser than what was available to them, further indicating the opportunity to focus robotic planning around a smaller number of more salient features.

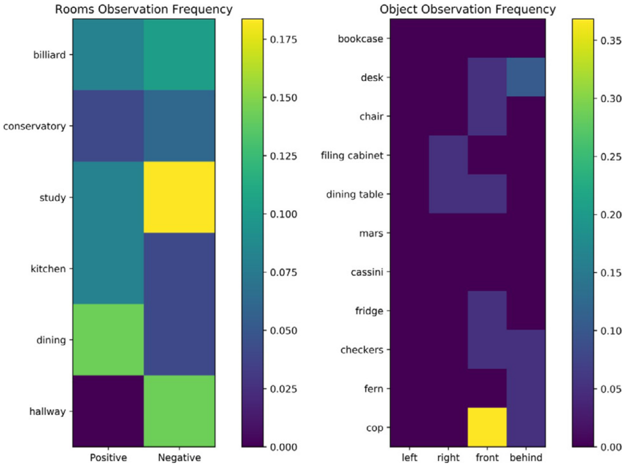


Fig. 11. Heatmap of the relative frequency of each human observation in the second map.

The cop’s beliefs, summarized in Fig. 12, are once again a reasonable estimate of the robber’s position despite slightly more errors.

4.3. Discussion

The cop using the Hierarchical CPOMDP approach succeeded in all cases at catching the robber, and was demonstrably quicker in cases where it received and fused human information. Of particular note is the improvement of the “Robot Pull Only” input condition over the “Human Push Only” condition. This implies that information delivered at the policy’s request was more valuable than that which the human decided to volunteer. As the policy is meant to approximate the optimal value function for the problem, this serves as evidence of its efficacy.

The system was also able to adapt to false information from the human sensor, as displayed in Fig. 13. In Fig. 13a, after the robber passed in front of the security camera in the Study while moving into the Kitchen, the human unintentionally gave a series of false observations, rapidly shifting the belief from an uncertain but reasonable one to one that was decidedly inaccurate. Later in the same run, the human was able to combine visual information from both the Hallway camera and the cop’s viewcone to indicate correctly that the robber had moved into the Dining Room, as shown in Fig. 13b.

With human information, the policy was able to direct the cop more efficiently. As shown in Fig. 14a, without any human sensor data the policy primarily directs the cop to patrol the Hallway, popping in and out of individual rooms along the way. This behavior is reasonable considering the Hallway’s position as a hub room, where the cop could expect to eventually stumble upon the robber as it moves from room to room. This displays the robustness of the policy’s action selection in the absence of expected information. However, when a human operator is able to provide information as in Fig. 14b, the policy chooses a path

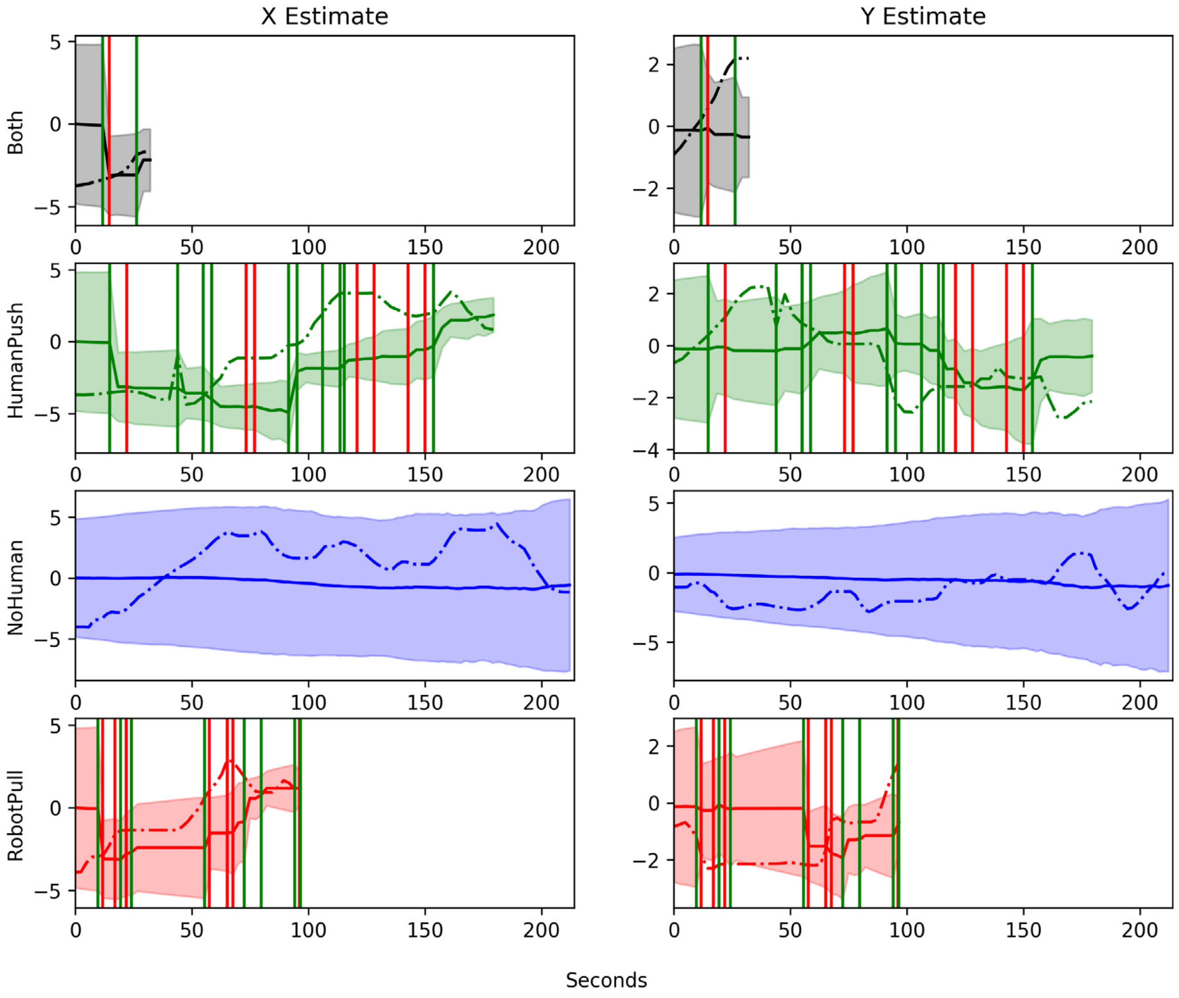
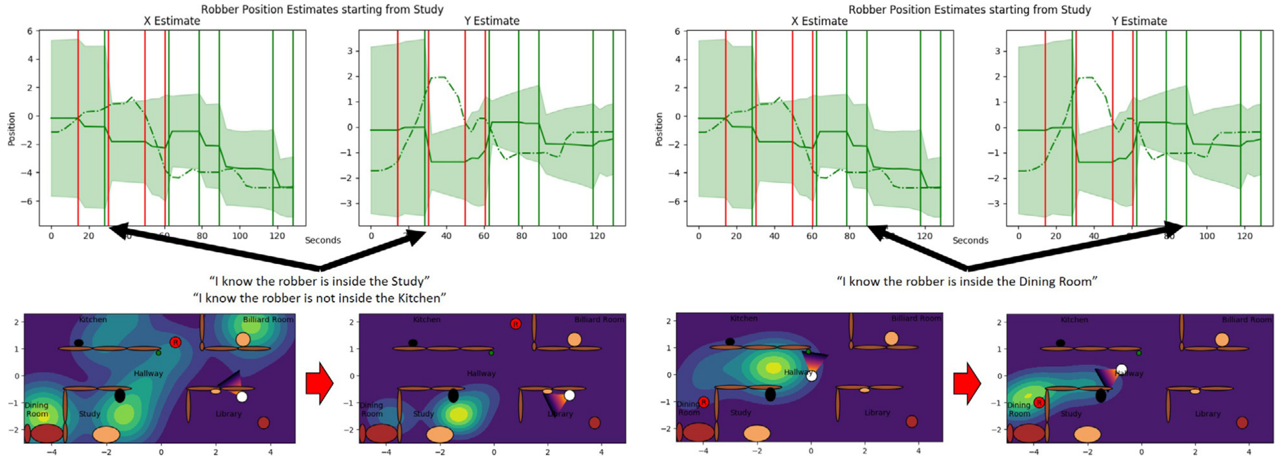


Fig. 12. Summary of cop's beliefs for the second map. Mean and 2-sigma bounds of the cop's belief are plotted against robber's true position (dashed line). Vertical lines are color coded for positive (green) and negative (red) human statements. As expected, the unfamiliar environment leads to less accurate beliefs in the Human Push scenario. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



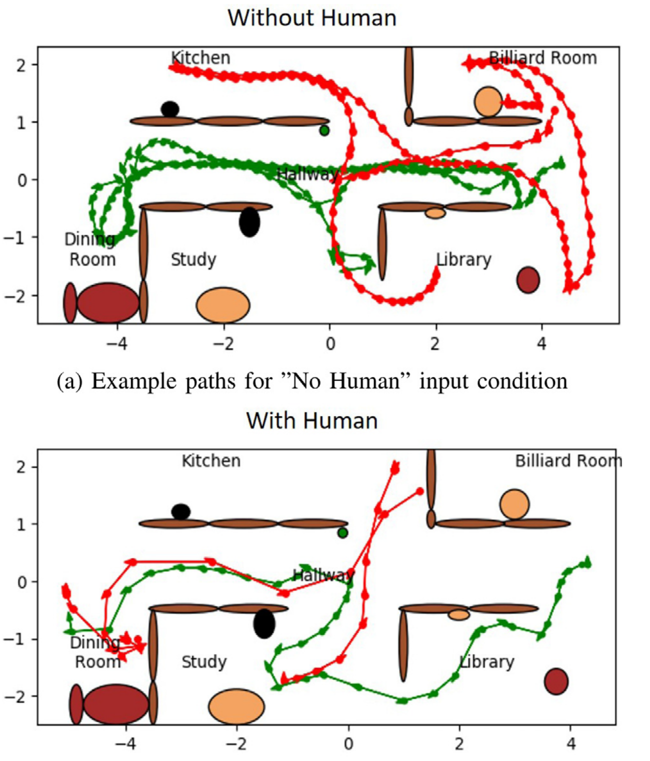
(a) Human gives inaccurate series of observations

(b) Human shifts belief with information unavailable to the cop

Fig. 13. Top: The human gives a series of mistaken observations. Bottom: The human gives a helpful statement.

through the Library, and ends up tracking the robber directly through the Study, and into the Hallway, finally cornering it in the Dining Room.

The observations given in each scenario show interesting differences between each map, as shown in Figs. 9 and 11. In the first map, where the human was familiar with the map layout and object placement, the observations were sparse. While room observations were dispersed, the human tended to focus on a few key objects, where observations would have a well known and predicable effect on the belief. In the unfamiliar case, the operator



(a) Example paths for ”No Human” input condition

(b) Example paths for ”Human Push Only” input condition

Fig. 14. Cop (green) and robber (red) paths without vs. with human sensor input. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

gave observations about a broader range of objects, often using the objects in the cops camera feed for reference. In both maps, the object most referenced was the cop itself. In particular, the human operator used the observation “The Robber is in front of the Cop” more than any other across all tests, possibly in attempts to urge the cop forward when the robber was in view but distant.

In summary, the robot was able to actively use human information within a hierarchical POMDP framework to improve its own effectiveness. Such information led to distinct behavioral difference as in Fig. 14, and shorter capture times. The human, even as an imperfect sensor, was able to actively recognize mistakes and error correct. However, though the POMDP successfully advantage of the structure of the problem, it relied on perfect a prior knowledge of this structure. The techniques used throughout this paper address models of uncertainty but fail in the face of models which are themselves incomplete or uncertain. Furthermore, Figs. 9 and 11 show large parts of the semantic dictionary went unused, while others were heavily exploited by the human. Ideally, the dictionary would contain primarily useful entries, and respond to new information about the problem by expanding appropriately to allow communication about new semantic features. Such an approach is explored in [48].

5. Conclusion

We developed and validated a novel collaborative human–machine sensing solution for dynamic target search. Our approach used continuous partially observable Markov decision process (CPOMDP) planning to generate vehicle trajectories that optimally exploit imperfect detection data from onboard sensors and semantic natural language observations that can be requested from human sensors. The main innovation was a scalable hierarchical Gaussian mixture model formulation for efficiently solving CPOMDPs with semantic observations in continuous dynamic state spaces. The approach was demonstrated with a real human–robot team engaged in dynamic indoor target search and capture scenarios on a custom testbed. The results showed that combined human–robot sensing not only enhances target localization quality (as expected), but that the resulting CPOMDP policies provide sensible simultaneous search movements and semantic human sensor queries that allow the search vehicle to intercept the target more efficiently. The resulting CPOMDP policies are robust and effective even with irregular/unpredictable inputs and occasional errors from the human sensor.

Ongoing and future research will focus on semantic data fusion in problems where we relax our assumptions of: known number of targets; known search environment/map and semantic reference objects; known search vehicle states; and known human sensor parameters. These problems are significantly more challenging to solve, but also have important practical implications for applications involving target search in highly uncertain environments, e.g. search and rescue or disaster relief. Building on the work here and in [9], it is of interest to investigate how semantic human sensor data can be actively leveraged for online interactive learning and planning, as well as online state estimation and perception.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work has been funded by the Center for Unmanned Aircraft Systems (C-UAS), USA, a National Science Foundation Industry/University Cooperative Research Center (I/UCRC) under NSF, USA Award No. CNS-1650468 along with significant contributions from C-UAS industry members.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.robot.2021.103753.

References

[1] T. Kaupp, B. Douillard, F. Ramos, A. Makarenko, B. Upcroft, Shared environment representation for a human–robot team performing information fusion, J. Field Robotics 24 (11–12) (2007) 911–942.

[2] M. Bourgault, N. Drouin, E. Hamel, Decision making within distributed project teams: An exploration of formalization and autonomy as determinants of success, Proj. Manag. J. 39 (1_suppl) (2008) S97–S110.

[3] B. Khaleghi, A. Khamis, F. Karray, Random finite set theoretic based soft/hard data fusion with application for target tracking, in: 2010 IEEE Conference on Multisensor Fusion and Integration, IEEE, 2010, pp. 50–55.

[4] A. Dani, M. McCourt, J.W. Curtis, S. and Mehta, Information fusion in human–robot collaboration using neural network representation, in: 2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC), IEEE, 2014, pp. 2114–2120.

[5] J. Frost, A. Harrison, S. Pulman, P. Newman, A probabilistic approach to modelling spatial language with its application to sensor models, in: Proceedings of the Workshop on Computational Models of Spatial Language Interpretation at Spatial Cognition (COSLI), Citeseer, 2010.

[6] S.S. Mehta, M. McCourt, E.A. Doucette, J.W. Curtis, A touch interface for soft data modeling in bayesian estimation, in: 2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC), IEEE, 2014, pp. 3732–3737.

[7] N.R. Ahmed, E.M. Sample, M. Campbell, Bayesian multicategorical soft data fusion for human–robot collaboration, IEEE Trans. Robot. 29 (1) (2013) 189–206.

[8] N. Sweet, N. Ahmed, Structured synthesis and compression of semantic human sensor models for Bayesian estimation, in: 2016 American Control Conference (ACC), IEEE, 2016, pp. 5479–5485.

[9] L. Burks, I. Loefgren, N.R. Ahmed, Optimal continuous state pomdp planning with semantic observations: A variational approach, IEEE Trans. Robot. (2019).

[10] M.E. Walker, H. Hedayati, D. Szafir, Robot teleoperation with augmented reality virtual surrogates, in: 2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI), IEEE, 2019, pp. 202–210.

[11] A. Brooks, A. Makarenko, S. Williams, H. Durrant-Whyte, Parametric POMDPs for planning in continuous state spaces, Robot. Auton. Syst. 54 (11) (2006) 887–897.

[12] H. Bai, D. Hsu, W.S. Lee, V.A. Ngo, Monte Carlo value iteration for continuous-state POMDPs, in: Algorithmic Foundations of Robotics IX, Springer, 2010, pp. 175–191.

[13] J. Van Den Berg, P. Abbeel, K. Goldberg, LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information, Int. J. Robot. Res. 30 (7) (2011) 895–913.

[14] H. Bai, D. Hsu, W.S. Lee, Integrated perception and planning in the continuous space: A POMDP approach, Int. J. Robot. Res. 33 (9) (2014) 1288–1302.

[15] S. Brechtel, T. Gindele, R. Dillmann, Probabilistic decision-making under uncertainty for autonomous driving using continuous POMDPs, in: 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), IEEE, 2014, pp. 392–399.

[16] E. Zhou, M.C. Fu, S.I. Marcus, Solving continuous-state POMDPs via density projection, IEEE Trans. Automat. Control 55 (5) (2010) 1101–1116.

[17] D.L. Hall, J.M. Jordan, Human-Centered Information Fusion, Artech House, 2010.

[18] T. Southey, J.J. Little, Learning qualitative spatial relations for object classification, in: IROS 2007 Workshop: From Sensors to Human Spatial Concepts, 2007.

[19] T. Kaupp, A. Makarenko, H. Durrant-Whyte, Human–robot communication for collaborative decision making—A probabilistic approach, Robot. Auton. Syst. 58 (5) (2010) 444–456.

[20] J. Muesing, L. Burks, M. Iuzzolino, D. Szafir, N.R. Ahmed, Fully bayesian human-machine data fusion for robust dynamic target surveillance and characterization, in: AIAA Scitech 2019 Forum, 2019, p. 2208.

[21] S. Rosenthal, M. Veloso, A.K. Dey, Learning accuracy and availability of humans who help mobile robots, in: Twenty-Fifth AAAI Conference on Artificial Intelligence, 2011.

[22] K.G. Lore, N. Sweet, K. Kumar, N. Ahmed, S. Sarkar, Deep value of information estimators for collaborative human-machine information gathering, in: Proceedings of the 7th International Conference on Cyber-Physical Systems (ICCPS 2016), IEEE Press, 2016, pp. 3–12.

[23] P. Abbeel, A.Y. Ng, Apprenticeship learning via inverse reinforcement learning, in: Proceedings of the twenty-first international conference on Machine learning, 2004, p. 1.

[24] B. Hayes, B. Scassellati, Discovering task constraints through observation and active learning, in: 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2014, pp. 4442–4449.

[25] S. Chernova, A.L. Thomaz, Robot learning from human teachers, Synth. Lect. Artif. Intell. Mach. Learn. 8 (3) (2014) 1–121.

[26] C. Mueller, J. Venicx, B. Hayes, Robust robot learning from demonstration and skill repair using conceptual constraints, in: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2018, pp. 6029–6036.

[27] A.D. Dragan, S. Bauman, J. Forlizzi, S.S. Srinivasa, Effects of robot motion on human–robot collaboration, in: 2015 10th ACM/IEEE International Conference on Human-Robot Interaction (HRI), IEEE, 2015, pp. 51–58.

[28] N. Gopalan, S. Tellex, Modeling and solving human–robot collaborative tasks using pomdps, in: RSS Workshop on Model Learning for Human-Robot Communication, 2015.

[29] F. Doshi, N. Roy, Spoken language interaction with model uncertainty: an adaptive human–robot interaction system, Connect. Sci. 20 (4) (2008) 299–318.

[30] N.R. Ahmed, Data-free/data-sparse softmax parameter estimation with structured class geometries, IEEE Signal Process. Lett. 25 (9) (2018) 1408–1412.

[31] A.R. Runnalls, Kullback-Leibler approach to Gaussian mixture reduction, IEEE Trans. Aerosp. Electron. Syst. 43 (3) (2007) 989–999.

[32] W. Koch, On ‘negative’information in tracking and sensor data fusion: Discussion of selected examples, in: Proceedings of the Seventh International Conference on Information Fusion, Vol. 1, IEEE Publ., Piscataway, NJ, 2004, pp. 91–98.

[33] M.L. Littman, A.R. Cassandra, L.P. Kaelbling, Learning policies for partially observable environments: Scaling up, in: Machine Learning Proceedings 1995, Elsevier, 1995, pp. 362–370.

[34] N. Armstrong-Crews, M. Veloso, Oracular partially observable markov decision processes: A very special case, in: Proceedings 2007 IEEE International Conference on Robotics and Automation, IEEE, 2007, pp. 2477–2482.

[35] N. Armstrong-Crews, M. Veloso, An approximate algorithm for solving oracular pomdps, in: 2008 IEEE International Conference on Robotics and Automation, IEEE, 2008, pp. 3346–3352.

[36] S. Rosenthal, M. Veloso, Modeling humans as observation providers using pomdps, in: 2011 RO-MAN, IEEE, 2011, pp. 53–58.

[37] J. Pineau, G. Gordon, S. Thrun et al, Point-based value iteration: An anytime algorithm for POMDPs, in: 2003 International Joint Conference on Artificial Intelligence (IJCAI 2003), Vol. 3, 2003, pp. 1025–1032.

[38] M.T. Spaan, N. Vlassis, Perseus: Randomized point-based value iteration for POMDPs, J. Artificial Intelligence Res. 24 (2005) 195–220.

[39] H. Kurniawati, D. Hsu, W.S. Lee, SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces. in: Robotics: Science and Systems IV. Zurich, Switzerland. 2008, pp. 336–343..

[40] J.M. Porta, N. Vlassis, M.T. Spaan, P. Poupart, Point-based value iteration for continuous POMDPs, J. Mach. Learn. Res. 7 (Nov) (2006) 2329–2367.

[41] E. Brunskill, L.P. Kaelbling, T. Lozano-Pérez, N. Roy, Planning in partially-observable switching-mode continuous domains, Ann. Math. Artif. Intell. 58 (3–4) (2010) 185–216.

[42] D. Silver, J. Veness, Monte-Carlo planning in large POMDPs, in: Advances in Neural Information Processing Systems, 2010, pp. 2164–2172.

[43] A. Somani, N. Ye, D. Hsu, W.S. and Lee, DESPOT: Online POMDP planning with regularization, in: Advances in Neural Information Processing Systems, 2013, pp. 1772–1780.

[44] E.J. Sondik, The Optimal Control of Partially Observable Markov Decision Processes (Ph.D. thesis), Stanford University, 1971.

[45] L.P. Kaelbling, M.L. Littman, A.R. Cassandra, Planning and acting in partially observable stochastic domains, Artificial Intelligence 101 (1–2) (1998) 99–134.

[46] Stanford Artificial Intelligence Laboratory, and others, Robotic operating system, [Online]. Available: https://www.ros.org.

[47] G. Bradski, The opencv library, Dr. Dobb's J. Softw. Tools (2000).

[48] L. Burks, N. Ahmed, Collaborative semantic data fusion with dynamically observable decision processes, in: 2019 22nd International Conference on Information Fusion (FUSION), IEEE, 2019, pp. 1–8.

Luke Burks is a Research Scientist at Optimus Ride in Boston, MA. He obtained his Ph.D. in Aerospace Engineering in 2020 from the University of Colorado Boulder, and his BS in Physics from University of Arkansas. His Ph.D. thesis developed and validated probabilistic planning algorithms for active semantic sensing for collaborative teams of humans and autonomous robots. His research interests are in autonomous planning under uncertainty, data fusion, and artificial intelligence for mobile robotics.

Nisar Ahmed is an assistant professor in the Smead Department of Aerospace Engineering Sciences at the University of Colorado Boulder, where he also holds a courtesy appointment in Computer Science. He is a member of the Research and Engineering Center for Unmanned Vehicles (RECUV) and directs the Cooperative Human–Robot Intelligence (COHRINT) Lab. His research interests are in the modeling and estimation for intelligent control of dynamical systems, especially for applications involving human–robot interaction, distributed sensor networks, and information fusion. He received his B.S. in Engineering from Cooper Union, and Ph.D. in Mechanical Engineering from Cornell University in 2012. He was a postdoctoral research associate in the Cornell Autonomous Systems Lab from 2012 to 2014. He was awarded the 2011 AIAA Guidance, Navigation, and Control Conference Best Paper Award; an ASEE Air Force Summer Faculty Fellowship in 2014; and recently received the 2018 ACGSC Dave Ward Memorial Lecture Award. His work is supported by the Army, Air Force, Navy, NASA, DARPA, and industry, and he has also organized several workshops and symposia on autonomous robotics, sensor fusion, and human-autonomy interaction. He is a Member of IEEE and the AIAA Intelligent Systems Technical Committee.

Ian Loefgren is a research engineer with the Perception and Autonomy Group at the Charles Stark Draper Laboratory in Cambridge, MA. He earned his MS in Aerospace Engineering in 2020 and his BS in Aerospace Engineering in 2018, both from the University of Colorado Boulder.

Luke Barbier is an MS student in Computer Science at the University of Colorado Boulder, where he also earned his BS in ECEE in 2020 from the University of Colorado Boulder.

Jeremy Muesing earned his MS in Aerospace Engineering in 2019 and BS in Aerospace Engineering in 2017 from the University of Colorado Boulder, where he performed research on collaborative human-autonomy interaction for information fusion. He is currently a research and development analyst for the San Diego Padres Major League Baseball team.

Jamison McGinley is an MS student in Aerospace Engineering at the University of Colorado Boulder, where he also earned his BS in Aerospace Engineering in 2020.

Sousheel Vunnam is a software engineer at Amazon in Seattle, WA. He earned his BS in Computer Science from the University of Colorado Boulder in 2020.