Collaborative Semantic Data Fusion with Dynamically Observable Decision Processes

Luke Burks and Nisar Ahmed*

***Abstract*— This work presents novel techniques for tightly integrated online information fusion and planning in human-autonomy teams operating in partially known environments. Motivated by dynamic target search problems, we present a new map-based sketch interface for online soft-hard data fusion. This interface lets human collaborators efficiently update map information and continuously build their own highly flexible ad hoc dictionaries for making language-based semantic observations, which can be actively exploited by autonomous agents in optimal search and information gathering problems. We formally link these capabilities to POMDP algorithms for optimal planning under uncertainty, and develop a new Dynamically Observable Monte Carlo planning (DOMCP) algorithm as an efficient means for updating online sampling-based planning policies for POMDPs with non-static observation models. DOMCP is validated on a small scale robot localization problem, and then demonstrated with our new user interface on a simulated dynamic target search scenario in a partially known outdoor environment.**

I. Introduction

Dynamic information gathering algorithms typically leverage well-defined environment and sensor models to solve challenging combined optimal control and estimation problems. While exact solutions are intractable, approximate algorithms for autonomous data fusion and decision making under uncertainty can be very brittle. This problem is exacerbated in unknown environments, where the demands of online perception and planning lead to even greater computing bottlenecks, uncertainties and risks.

To mitigate these issues, human teammates can act as 'sensors' that contribute valuable information beyond the reach of autonomous robots. For instance, vehicle operators in search and tracking missions can provide 'soft data', semantic categorical data often allowing less precision and broader uncertainty than precise numeric coordinates, 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 or location of obstacles, distances from landmarks) to help autonomous vehicles better judge and understand search areas – thus improving online decision making. Furthermore, human information can be leveraged to not only describe an environment but also define it. A human sensor might also act to indicate areas of interest and refer back to them in the course of providing semantic information, as in Fig. 1. But how can autonomous reasoning *actively and opportunistically* engage human sensing in unknown environments?

Fig. 1: Collaborative human-machine target search scenario.

This paper introduces a framework that enables humans and autonomous agents to work together to define and describe dynamic or unknown environments while simultaneously planning and acting within that environment to accomplish tasks. Our framework uses Bayesian data fusion to exploit human sensors and autonomous robotic sensor platforms in a 'plug and play' manner. This idea has gained increased attention in various contexts over the last decade [1], [2], [3], [4], [5], [6]. However previous work on 'soft' human data fusion with 'hard' robotic data has primarily focused on structured and known environments, and have largely ignored coupling to planning/control problems. In this work we apply techniques from our recent work on Bayesian semantic robot-human sensor data fusion in structured environments [7], [8], [9], [10] alongside concepts from optimal active sensing and online planning under uncertainty, in order to develop new methods for *interactive multi-level* human-robot sensing of dynamic states in unknown environments. We present methods for dynamic sensor model creation and adaptive online planning with changing sensor models, then combine these for collaborative human-robot search.

A. Motivating Concept and Problem Description

This paper focuses on dynamic target search problems, in which an autonomous robot attempts to track and intercept a moving target in a partially unknown environment. An outdated overhead map of the search area is available a priori (e.g. old satellite data for a large outdoor search region, or old floor plans for a building), and the true environment map and its features are gradually revealed to the search robot as it moves through the area. The robot carries a visual proximity sensor which allows it to identify the target at close range,

*Smead Aerospace Engineering Sciences Department, 429 UCB, University of Colorado Boulder, Boulder CO 80309, USA. E-mail: [luke.burks;nisar.ahmed]@colorado.edu. This work is funded by the Center for Unmanned Aircraft Systems (C-UAS), a National Science Foundation Industry/University Cooperative Research Center (I/UCRC) under NSF Award No. CNS-1650468 along with significant contributions from C-UAS industry members.

and relies on a human collaborator to provide additional soft observations about the target state. These observations are modeled by a codebook of possible semantic statements indicating both positivity and 2D direction with respect to the robot or landmarks, e.g. "The target is not West of you", "Target is next to the building, moving North".

The human collaborator can also define new semantic observational anchors other than the pursuer itself by adding and labeling fixed landmarks with a specified spatial extent to the environment map, using a sketch interface similar to the one in depicted Fig. 1. The new landmarks can then be used in future observations. The robot can also request specific semantic information by modeling the human as an on-demand soft sensor, i.e. a sensor able to provide categorical statements about the problem state. For example, after the human identifies and sketches a new landmark on an outdoor map and labels it as "Water", the robot can ask "Is anything near the Water?". This level and type of flexible interaction requires new algorithms for modeling ad hoc observations added by the human, autonomous planning under uncertainty, and dynamic integration of soft sensing into the search task.

II. Background

A. Model Representation and Human Interface

Our problem requires timely acquisition and suitable representation of ad hoc observation semantics. Free-form sketching provides a convenient way of indicating and constraining spatial objects for semantic reference in a 2D environment. In previous research for robotic target search and localization in uncertain environments, sketching has been used to indicate the positions of obstacles in a scene [11], to constrain the bounds of an operating area [12], or to give direct sensing inputs about the target locations [7]. In our application, the human operator can sketch boundaries of objects/areas of interest within the search environment and provide suitable labels for these. These sketches can then be interpreted and converted to new probabilistic sensing model components via softmax functions by leveraging the techniques of [9], to create structured natural language codebooks that permit the robot and human to exchange semantic observations describing the dynamical state space for the search problem. Each of these semantic observations is an instance of a 'soft data' update, which can be used along with the corresponding model via a Bayesian state posterior pdf update alongside 'hard data' from robot sensors.

The semantic human-robot interface considered here builds significantly on the interfaces developed and used in [7], [13], [14]. These earlier interfaces restricted the possible set of human observations to a pre-determined codebook in a small perfectly mapped indoor search environment, where the human could be located either remotely (e.g. with access to the vehicle's visual sensing feed) or in situ in the search environment. As in previous work, the human gathers information about the problem independently from their robotic teammate. This is then fused with other information available to the robot during either a "Human-Push" event (i.e. the human opportunistically provides information without prompting) or "Robot Pull" event (i.e. the robot opportunistically prompting the human for information). Any approach used by the robot to plan future actions and "Robot Pull" prompts must account for Bayesian data fusion of both soft semantic data and hard sensor measurements, as well as process and model uncertainties for state belief evolution.

The prior work in [7], [13], [14] relied on the availability of known environment and reference object models for offline codebook generation. This severely limits the applicability of semantic human-robot data fusion in outdoor, dynamic, and/or partially known environments, where flexible human-level sensing, perception, and reasoning is generally quite valuable for decision-making to cope with complex uncertainties. Other work addresses the dynamic models using online solvers by assuming slowly changing models [15] or using a representative belief sample from a static offline policy that is then adapted during runtime [16]. Neither assumption holds here, as the model and set of reachable beliefs can change dramatically during runtime. The interface developed here allows the human to spontaneously create ad-hoc semantic observation models online, to tailor codebooks to problem needs and user preferences.

B. Optimal Planning under Uncertainty

Dynamic target search problems feature complex stochastic model uncertainties, process disturbances, and observation errors, which make autonomous optimal planning challenging. Such problems can be formulated as Partially Observable Markov Decision Processes (POMDPs) to arrive at principled optimal stochastic robot guidance and control policies. However, POMDPs are impractical to solve exactly in all but the simplest problems [17]. Various POMDP approximations have been developed to exploit properties inherent to different dynamics models [18], state space representations [19], and observation models [10].

Formally, a POMDP is specified as a 7-tuple $(S, A, T, R, \Omega, O, \gamma)$. The goal of POMDP planning is to identify a policy π which maps Bayesian beliefs (posterior pdfs) $b = p(s)$ over the state space S onto a set A of discrete actions a. The belief contains all available information about the unknown state up to the current timestep (via the Markov assumption). Actions lead to state transitions according to a discrete time probabilistic transition function T, which maps from s to s' given action a via $p(s'|s, a)$. The agent carrying out the policy π receives observations $o \in \Omega$ which depend on s according to the observation likelihood $O = p(o|s)$ and rewards r according to the reward function $R(s, a)$. For infinite horizon planning, a discount factor $\gamma \in [0, 1)$ models diminishing future returns. An optimal policy $\pi[b(s)] \rightarrow a$ is one which maximizes the expected discounted reward gained by the agent executing the policy, as represented by the Value Function V, where $V^{\pi}(b) = E\left[\sum_{t=0}^{\infty} \gamma^t R_t(b, \pi(b))\right]$. Our motivating problem is a POMDP where V is to be maximized by a policy which recommends both movement actions for the robot *and* questions to be asked of the human, such that the expected time to target interception is

minimized. Thus a "Robot Pull" event can be considered as a special action to ask the human a question, such that the target can be captured more quickly by modifying b. The inclusion of semantic queries in A brings the human collaborator into the closed loop planning and sensing process, treating them as a queryable sensor. These active sensing/querying actions rely heavily on the observation model O to anticipate expected changes in b based on the human's reply to questions. The main challenge is that the set of possible observations Ω changes unpredictably over time as the human provides new semantic models, whereas existing methods largely require O to be static to find π.

In contrast to Bayes-adaptive methods [20], such model alterations are not merely a refinement or discovery of an underlying true model, but rather the introduction of new elements by an external information source (the human collaborator). Thus the class of "full-width" offline point-based POMDP planners [21] is generally unusable for this problem, since this requires knowledge of the full POMDP model,or at the least a model of how O and Ω change. The information-gathering nature of the problem similarly rules out approximation methods that rely on the combination of fully observable policies with state uncertainty or single use observation models [22]. An additional challenge here is that the number of observations $|\Omega|$ not only changes, but can grow arbitrarily large as new observational anchors are defined online. Online POMDP approaches [23] have had success recently for large problems with various observation [24] and state space formulations [25]. Their use of generative 'black box' process models and interwoven planning/execution steps make them good candidates for our application, if they are combined with a method for adapting to dynamically changing semantic observation models.

C. Formal Problem Statement

We now formalize the problem of collaborative human-robot dynamic target search in uncertain partially known environments. An autonomous robot (the 'pursuer') with continuous dynamical states s_p attempts to localize, track, and intercept a target with continuous dynamical states s_t, where $[s_p, s_t]^T = s \in S = \mathcal{R}^N$ is the joint state space. The human collaborator interacts with the pursuer through an interactive data fusion interface which displays s_p as well as a pdf $b(s_t) = p(s_t|o^r_{1:t}, o^h_{1:t}, a_{1:t-1})$ for the current Bayesian posterior belief in s_t given all robot observations $o^r_{1:t}$ and human observations $o^h_{1:t}$ available up through time t. These states and beliefs are displayed over an geo-referenced map, which contains some incomplete/outdated information and is progressively updated in proximity to the pursuer by its sensors. The human can sketch and label new map elements $o' \in \Omega$, which are automatically converted to probabilistic likelihood models $O' = p(o'|s)$ to support active online querying and reporting of semantic data with respect to s.

The first problem addressed here is the process by which an ad-hoc human sketch is converted into an new observation model O'. This new O' can then be used by an autonomous robot to find a POMDP policy π that allows the robot to select actions a that determine trajectories/motion plans in the search environment and make specific queries to human collaborators that actively request semantic observations o^h. To support long duration search missions in unknown environments, the second problem is to then adapt an existing policy π online to a new optimal policy π' that maximizes V in light of new observation models O' and semantic dictionary Ω' synthesized from human input during run time.

III. Planning with Ad Hoc Semantic Sensing

A. Ad Hoc Semantic Sensor Modeling

To address the problem of ad hoc observation model synthesis, we assume that human sensors use a sketch interface to draw (via pointer or touchscreen) directly on their geo-referenced map. This allows the human to quickly and intuitively specify the spatial extent of an area or object of interest in real-time. A sketch consists of a label L and set of points $\{P\}$, from which vertices can be extracted to define the convex boundaries that are needed to implement the procedure from [9]. This procedure uses the normal vectors between vertex points to synthesize a softmax likelihood function from a convex polytope defining the log-odds boundaries for different semantic class labels. Softmax functions define a likelihood function for a discrete set of observation classes $o \in O$ over the continuous state $s \in S$ using a vector of weights and a bias defined for each class,

$$p(o = j|s) = \frac{\exp(w_j^T s + b_j)}{\sum_{c=1}^{N_o} \exp(w_c^T s + b_c)} \tag{1}$$

These functions have several extremely useful properties which make them convenient to encode the model $O = p(o|s)$ for a finite set Ω defined over continuous S. They are self-normalizing, such that $\sum_o p(o|s) = 1$ for any given O at a particular s. They also model how a particular semantic class label may probabilistically dominate a region of the state space, with relatively little likelihood in other regions. The drop-off in likelihood for a given o when moving out of its dominant region is also a tunable parameter often referred to as steepness. The steepness of the boundary between classes can reflect how likely different observations are to result from similar states. Using [9], a number of points M define a model with $M + 1$ classes, with M exterior classes surrounding an interior class, which is the M-sided convex polygon defined by a convex hull on $\{P\}$. However, a sketch from a drawing interface tends to have far more points needed to define the intended polygon, and the number of vertices must be reduced in practice, e.g. using Algorithm 1.

From the points in the initial sketch, the ordered vertices of a convex hull $v_i \in \{V\}$ where $\{V\} \subseteq \{P\}$ in the 2D plane is obtained using the Quickhull algorithm. $\{V\}$ is progressively reduced until it reaches a predefined size by repeatedly removing the point contributing the least deflection angle to the line between its neighbors, calculated via the Law of Cosines for the vertex pair vectors $\overrightarrow{v_{i-1}v_i}$,

$$\Theta(v_i) = \arccos\left[\frac{\overrightarrow{v_{i-1}v_i} \cdot \overrightarrow{v_i v_{i+1}}}{\|\overrightarrow{v_{i-1}v_i}\|\|\overrightarrow{v_i v_{i+1}}\|}\right] \tag{2}$$

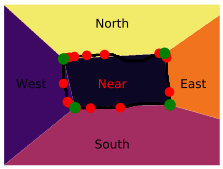

Fig. 2: Left: Convex hull verticies (red), reduced to 4 points with sequential hull reduction (green). Right: Softmax function and labels resulting from reduced points.

in a procedure inspired by the Ramer-Douglas-Peucker algorithm. This heuristic was chosen as a proxy for maximizing the area maintained by the reduced hull, but in practice other heuristics could also be used. An example sketch is shown in Fig. 2, where the initial input consists of 661 points, shown in black, making a roughly rectangular shape. The Quickhull algorithm is applied to find 21 points, shown in red, defining a convex hull on the set of points. These points are then used as input to Algorithm 1, which further reduces the number of points to the 4 points shown in green. From these 4 points, a softmax model consisting of 5 classes $\mathcal{C}$, a "Near" class and 4 cardinal directions, is then synthesized. Semantic observations can now be constructed using the label L given by the human when they made the sketch in the form, $o =$ "The Target is $\mathcal{C}$ of L." In the case of Fig. 2, the human labeled the sketch $L =$ "Lake", so a new option of semantic observation would be "The Target is North of Lake". In previous work [14], softmax models were manually created and labeled prior to the problem execution, and could not be altered online. With the ability to construct and label models from sketches, the issue of dynamically modifying observation models via human input can now be addressed.

Algorithm 1 Sequential Convex Hull Reduction

Function: $REDUCE$
Input: Convex Hull $\{V\}$, Target Number N
if $\{V\} == N$ **then**
  return hull
end if
for $v_i \in \{V\}$ **do**
  $\Theta(v_i) \leftarrow angle(v_{i-1}, v_i, v_{i+1})$
end for
$\{V\} \leftarrow \{V\} \setminus argmin_v \Theta(v)$
return REDUCE(hull,N)

Our implementations limit consideration to 4-point/5-class models, where the semantic labels for each class are easily mapped to simple cardinal directions by noting the North-eastern point in the reduced hull. However, it is generally possible construct multi-modal softmax (MMS) models in which a single observation maps to a sum of multiple classes

$$p(o = j|s) = \frac{\sum_{r \in \sigma(j)} \exp(w_r^T s + b_r)}{\sum_{c=1}^{N_o} \exp(w_c^T s + b_c)} \tag{3}$$

This allows sketches to consist of a larger number of reduced points, better capturing the area indicated by the sketch. Assigning labels to ad-hoc MMS models obtained through sketch remains an interesting problem for future research.

A limitation of sequential hull reduction is the need to predefine a set number of points at which to stop the progressive reduction. Ideally, the "natural" number of points needed to maximize some criteria could be chosen on a sketch by sketch basis in a geometric analogue to the Bayesian Information Criterion (BIC) score for model selection. Note that this approach for convex hull reduction differs from that used in Geometric SVM classifiers in that reduces the vertices comprising the boundaries of the hull itself, rather than reducing the hull area in a feature hyperplane.

B. Online Policy Approximation: POMCP

One state-of-the-art online POMDP solver which has seen success in continuous state spaces and complex sensing problems is Partially Observable Monte Carlo Planning (POMCP) (Algorithm 2). The core of the POMCP algorithm consists of building and exploring a tree of potential future histories for a problem in real-time. Samples $s \sim b(s)$ are simulated forward in time using a problem-specific generative model. The model is typically represented as $(s', o, r) \sim G(s, a)$; for domains with explicit models this can be factored into a set of separate functions,

$$s' \sim p(s'|s, a), \;\; o \sim p(o|s', a), \;\; r \sim R(s, a) \tag{4}$$

With either representation, a tree is built in which each node is a sequence of actions and observations following the initial belief. These sequences noted as a history h, where hao is the result of an additional action a and observation o. Each node also carries an account of states visiting it $N(h)$ and a value estimate $V(ha)$ for each action collected from the discounted rewards of its child nodes. During the exploration phase, actions are chosen via $a \leftarrow argmax_{a'} \left[V(ha') + c\sqrt{\frac{log(N(h))}{N(ha')}}\right]$, where c is an tunable exploration constant. Once exploration finishes, a is selected according to the child node of the current belief with the highest value. POMCP is an "anytime" algorithm, i.e. the best a found in a finite time will be returned.

In previous applications, this approach was constrained to operate with static observation models. However, for a problem with dynamic observation models, the beliefs and observation likelihoods represented at future observation nodes in the tree will become inconsistent with an altered observation model. This negatively impacts information gathering problems, as newer observations that can have a critical impact on decision making will not be explored as thoroughly as they would be had they been known prior to the tree's construction.

To address this, we note that the POMCP algorithm advantageously allows a tree of histories to be built quickly enough for a typical 2D target search problem, such that a decision at the first time step can be made before traversing and expanding that tree in future execution and planning

steps. Here we propose an alteration to the POMCP algorithm, in which a planning/execution step which modifies the observation model not only prunes branches of the tree of histories inconsistent with the latest action and observation, but also all future branches stemming from the current history node h which are inconsistent with the new observation model. This results in pruning *all* future nodes, effectively restarting the tree building process from scratch and ensuring that any nodes added going forward are explored according to the proper model. This approach prevents the tree of histories from containing nodes inconsistent with the current observation model, but pays both an information and effort cost. Future nodes represent imperfect information about the problem, and pruning them both discards this information and requires duplicating the effort put into exploring them. This effort includes simulating transition and reward functions in addition to the observation model which was altered. However, the speed of the POMCP algorithm still allows it to plan effectively following a sudden model change.

C. Online Policy Revision: DOMCP

Algorithm 3 details an innovation which leverages an alternative representation of particles in POMCP to adapt when a new observation model $p(o'|s)$ is introduced between planning steps. This algorithm, Dynamically Observable Monte-Carlo Planning (DOMCP), stores a complete descriptor consisting of a current state, action, reward, resulting state, and observation for every particle in node h. This differs from prior work which represented each particle only as the state and accumulated the reward as a sum total alongside a collection of states. Tracking this information requires only a minor change to the POMCP algorithm:

$$h \leftarrow h \cup s \Longrightarrow h \leftarrow h \cup [s, a, r, s', o]$$

A small additional memory usage cost is required, upper bounded by a constant multiple of the cost of storing the states themselves. This allows state trajectories and their associated value to be redirected between nodes as observations are resampled from $p(o'|s)$. Each time a state is reassigned to a new node, its value contribution to that node and its children is reassigned with it, subtracted from its previous node. In this way, a state history which had previously been threaded through the entire tree can be redirected to a newly created observation node only when receiving the new observation, preserving the information contained in nodes retained from the original tree. This results in a tree with observation likelihoods matching those of the current problem model, as in Fig. 3. It is interesting to note that although this work addresses the addition of new observations to the observation model, in principle DOMCP addresses any substantive change to the observation model in use. For instance, if observations are subtracted, the observation sampling step simply doesn't allocate any states to subtracted nodes.

Algorithm 2 POMCP

Function: $SEARCH$
Input: History h, Belief B
repeat
  $s \sim B(s)$
  $SIMULATE(s, h, 0)$
until TIMEOUT( )
return $argmax_a V(ha)$

Function: $SIMULATE$
Input: State s, History h, Depth d
if $d > d_{max}$ **then**
  return 0
end if
if $h \notin T$ **then**
  for $a \in A$ **do**
    $T(ha) \leftarrow (N_{init}(ha), V_{init}(ha), \emptyset)$
  end for
  return γ ESTIMATE_VALUE(s,h,d)
end if
$a \leftarrow argmax_{a'} \left[V(ha') + c\sqrt{\frac{log(N(h))}{N(ha')}} \right]$
$s' \sim p(s'|s, a), \;\; o \sim p(o|s', a), \;\; r \sim R(s, a)$
$R \leftarrow r + \gamma SIMULATE(s', hao, d + 1)$
$h \leftarrow h \cup s$
$N(h) \leftarrow N(h) + 1$
$V(ha) \leftarrow V(ha) + \frac{R - V(ha)}{N(ha)}$
return R

IV. Simulated Applications

A. DOMCP Demonstration

In order to demonstrate the viability of the DOMCP algorithm, we first present a one dimensional toy problem, building on a heavily modified version of [26]. In this problem, a single robotic agent attempts to localize it's position in a continuous open space, and reach a designated goal state. The robot can move left and right across the continuous state

Algorithm 3 DOMCP Redistribution Step

Function: $REDISTRIBUTE$
Input: History h, Likelihood Model $p(o|s)$
for $[s, a, r, s', o] \in h$ **do**
  $o' \sim p(o'|s')$
  if $o \neq o'$ **then**
    $h \leftarrow h \setminus [s, a, r, s', o]$
    $h \leftarrow h \cup [s, a, r, s', o']$
    $PURGE(hao, s')$
  end if
end for
for $hao \in h.children$ **do**
  $REDISTRIBUTE(hao, p(o|s))$
end for

Function: $PURGE$
Input: History h, State s
if $[s, a, r, s', o] \in h, \forall a, r, s', o$ **then**
  $h \leftarrow h \setminus [s, a, r, s', o]$
  $PURGE(hao, s')$
end if

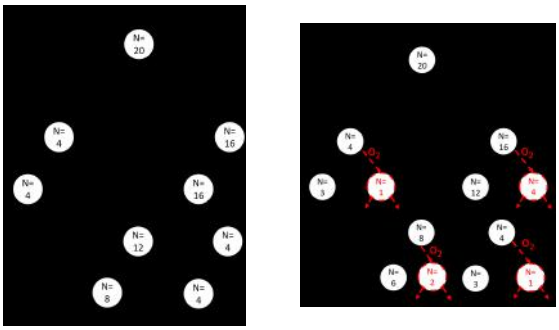

Fig. 3: POMCP tree before vs. after DOMCP redistribution.

space with actions subject to Gaussian dynamics ϕ with a mean shifted by the expected outcome:

$$p(s'|a = left, s) = \phi(s'|\mu = s - 0.5, \Sigma = 1) \quad (5)$$

$$p(s'|a = right, s) = \phi(s'|\mu = s + 0.5, \Sigma = 1) \quad (6)$$

Initially, as shown in Fig. 4(a), the robot has access to two possible observations, representing "West" and "East" halves of the state space. These observations are defined by a two class softmax function $p(o = \text{west}|s)$ and $p(o = \text{east}|s)$. Two parameters, a weight and a bias, are required for each class. After several steps, the observation model is changed as shown in Fig. 4(b), to include an additional observation "Far West" with $p(o = \text{far west}|s)$. This introduces a new class to the softmax model, modifying the denominators of the existing classes accordingly. At this point, having reached the simulated observation distribution shown in Fig. 4(a), which consists of 5578 individual particles $[s, a, r, s', o]$, both the replanting and redistribution approaches are separately applied, with the resulting planning trees shown in Figs. 4(b) and 4(c). As shown, each of the new trees distributes its observations according to the new observation model $p(o'|s)$, but the DOMCP algorithm retains 4096 of the original particles without having to resimulate their states, while POMCP must execute an additional planning step involving both transition and observation simulations.

B. Interface Application

We applied the DOMCP algorithm to the dynamic target search scenario described in Section IIa. Both the pursuer 'p' and the target 't' operate in 2D space $S_{robot} = \mathbb{R} \times \mathbb{R}$, with a combined states $s_{p,t} \in [S_p, S_t]^T$. The target states are unknown as it executes a Gaussian random walk with a standard deviation of 8 meters,

$$s'_r \sim \mathcal{N}(s_r, 8^2 \cdot \mathcal{I}) \quad (7)$$

while the pursuer's state is known. The pursuer can shift it's state by 10 meters with actions in the 4 cardinal directions, North, South, East, and West, in addition to actions querying the human in "Robot Pull" events. The pursuer is rewarded,

$$R(s, \forall a \in A) = \begin{cases} 100 & \text{if: } dist(s_p, s_r) < 25 \\ -1 & \text{if: } dist(s_p, s_r) > 25 \end{cases} \quad (8)$$

and any state in which the distance between the target and the pursuer is less than 25 meters is treated as a terminal state. The pursuer also carries an onboard visual sensor which allows it to detect if the target is within 50 meters in any direction but gives no indication of bearing. The human collaborator is allowed to push information to the robot through the use of drop-down menus encoding the possible semantic observations, as well as introduce semantic observations through sketching models on the map as in Fig. 5. Each of these models is constructed as a softmax function with the methods introduced in Section IIIa. In this work, sketches were reduced to 4 points before softmax model synthesis. Thus the observation space consists of a "North of", "South of", "East of", "West of", and "Near" in relation to each sketch as well as in relation to the pursuer.

The pursuer and human start out with an outdated map of an environment. The updated map contains significant additional features which can be given semantic labels, and also lack several features present in the outdated map. As the pursuer transverses the environment, the updated map is revealed to the human, along with the target if it is present in the newly explored area. The pursuer acts to sweep out belief in an efficient manner while exploiting information from the human. As more of the updated map is revealed, the human is able to label additional landmarks and give a broader variety of semantic observations as in Fig. 5.

The DOMCP algorithm was tested alongside POMCP and a maximum aposteriori (MAP) control algorithm. The MAP approach used a greedy heuristic which directed the robot towards the mode of the pursuer's belief of the target's location. This is presented as a baseline implementation of a reasonable low cost planner which is still capable of achieving the goal of capturing the target, albeit less optimally than POMCP and DOMCP. The POMCP algorithm is implemented using the reboot method discussed in Section III.B, in which future planning steps are pruned from the decision tree after each sketch is made.

Five independent test scenarios were constructed from human input and tested on each of the approaches. For each scenario, the pursuer's starting location, initial belief, all target positions, sketches, and human inputs relating to sketches were held constant across algorithms, while inputs related to the position of the pursuer, such as "The Target is East of You", were given in roughly equivalent measure between the three in accordance with the position of the pursuer for that simulation. This ensures fair comparison between the algorithms, testing their ability to maximize the use of human information to quickly reach their goal. As shown in Table I, all three approaches are capable of capturing the target in a reasonable amount of time. As expected, the MAP heuristic showed the worst performance in all cases while POMCP and DOMCP preformed similarly on the whole. While DOMCP achieve a slightly faster time to catch on average, the low number of simulations prevents any significant statistical difference from being identified. Indeed, it would be unexpected to see any major discrepancy between POMCP and DOMCP in this application with respect to decision making as there are relatively few model alterations over the course of each run. However, in problems where more frequent sketches occur, or computational savings are

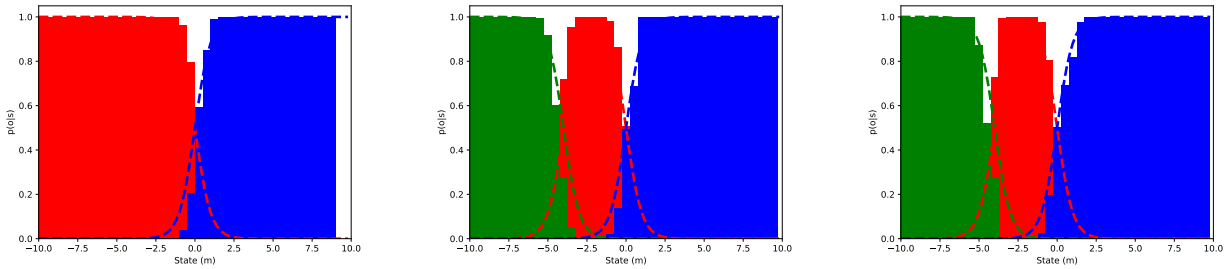

Fig. 4: 1D likelihood (a) before update, (b) after POMCP search, (c) after DOMCP redistribution without extra search

Fig. 5: Example search run using DOMCP with sketching interface: (left to right) robot starts with broad uncertainty in target position and an outdated map, and few options for human input; robot receives sketch "Parking" sketch, then inquires if the target is north of it; human alerts robot that target east of "Building" sketch, finally leading to the updated belief.

Target Search Interface Simulation Results

Method	Mean Time to Capture	Std Dev
MAP	126.2 s	±37
POMCP	94.6 s	±24
DOMCP	88.2 s	±19

TABLE I: Average Time to Capture with Standard deviations

more essential, we expect DOMCP to provide an advantage.

When executing searches, human information played a significant role in the ability of the pursuer to accurately localize the target. While the actions suggested by a particular algorithm dictate the regions searched and therefore the speed with which the target is caught, the human can shift belief much more rapidly. Humans, however, are not perfect, and mistaken observations or wrongly answered questions can lead to inconsistent beliefs and extended searches.

Fortunately, humans also have the ability to reason about previous measurements with new information, allowing them to recognize mistakes and work to correct them. An example of this behavior is showcased in Fig. 6, where the human mistakenly indicates the exact opposite direction to the pursuer. After several steps the human is able to use their drone scouting options to re-localize the target, and gives multiple rapid observations to inform the pursuer. The target's capture was delayed, but could have been more so if the human had not actively corrected a previous error. While not addressed here, existing work [27] could provide robustness to updates involving repeated or erroneous observations.

V. Conclusion

We developed and tested a novel online planning solution for collaborative human-robot search tasks in dynamic and unknown environments. The approach uses an extension of the POMCP algorithm called Dynamically Observable Monte-Carlo Planning (DOMCP) to adapt prior planning for the purpose of optimally exploiting semantic natural language observations by a human sensor. The innovations included a method for construction of ad-hoc semantic observation models from human sketches and the integration of modified observation models into tree-based planners. The approach was validated on a small scale problem, where a large proportion of relevant information in the planning tree could be retained after altering the model. It was further demonstrated on a simulated collaborative human-robot dynamic target search problem. The DOMCP algorithm provided sensible action choices and retained planning information as the human modified the problem with ad-hoc semantic sensor models through a sketching interface.

In ongoing work, this approach will be extended by allowing human model alterations to both the transitions and rewards as well. Whereas observations can be modified without corresponding environmental components, transitions and rewards must be grounded in real elements of the system. In these cases, human sketches could be leveraged to update the robot's internal model. We also plan to integrate DOMCP with more realistic robotic sensor models and pursue a hardware implementation of a robotic search task.

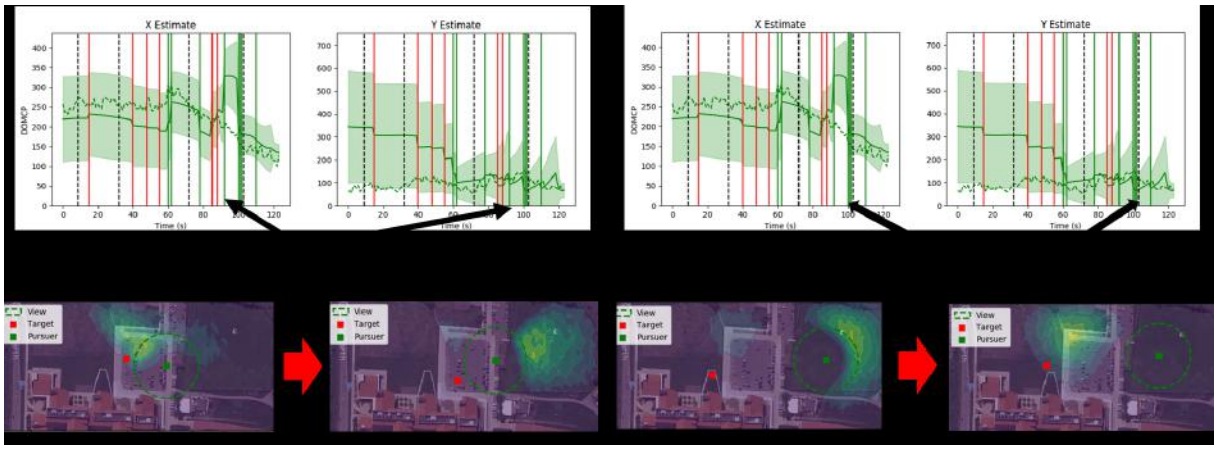

Fig. 6: Shaded estimates of the robber's location, with green and red vertical lines representing positive and negative human inputs respectively. (Left) The human gives an erroneous observation, shifting the belief away from the target's state. (Right) The human realizes their mistake and attempts to counteract their previous error.

REFERENCES

[1] T. Kaupp, B. Douillard, F. Ramos, A. Makarenko, and B. Upcroft, "Shared environment representation for a human-robot team performing information fusion," *Journal of Field Robotics*, vol. 24, no. 11-12, pp. 911–942, 2007.

[2] F. Bourgault, A. Chokshi, J. Wang, D. Shah, J. Schoenberg, R. Iyer, F. Cedano, and M. Campbell, "Scalable bayesian human-robot cooperation in mobile sensor networks," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2008, pp. 2342–2349.

[3] J. Frost, A. Harrison, S. Pulman, and 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.

[4] B. Khaleghi, A. Khamis, and 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.

[5] A. Dani, M. McCourt, J. W. Curtis, and S. 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.

[6] D. Yi, S. Choudhury, and S. Srinivasa, "Incorporating qualitative information into quantitative estimation via sequentially constrained hamiltonian monte carlo sampling," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 4648–4655.

[7] N. R. Ahmed, E. M. Sample, and M. Campbell, "Bayesian multicategorical soft data fusion for human–robot collaboration," *IEEE Transactions on Robotics*, vol. 29, no. 1, pp. 189–206, 2013.

[8] N. Ahmed, M. Campbell, D. Casbeer, Y. Cao, and D. Kingston, "Fully bayesian learning and spatial reasoning with flexible human sensor networks," in *Proceedings of the ACM/IEEE Sixth International Conference on Cyber-Physical Systems*. ACM, 2015, pp. 80–89.

[9] N. Sweet and 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.

[10] L. Burks and N. Ahmed, "Optimal continuous state POMDP planning with semantic observations," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC 2017)*. IEEE, 2017, pp. 1509–1516.

[11] D. Shah, J. Schneider, and M. Campbell, "A sketch interface for robust and natural robot control," *Proceedings of the IEEE*, vol. 100, no. 3, pp. 604–622, 2012.

[12] F. Boniardi, B. Behzadian, W. Burgard, and G. D. Tipaldi, "Robot navigation in hand-drawn sketched maps," in *2015 European conference on mobile robots (ECMR)*. IEEE, 2015, pp. 1–6.

[13] K. G. Lore, N. Sweet, K. Kumar, N. Ahmed, and 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.

[14] L. Burks, I. Loefgren, L. Barbier, J. Muesing, J. McGinley, S. Vunnam, and N. Ahmed, "Closed-loop Bayesian semantic data fusion for collaborative human-autonomy target search," in *2018 21st International Conference on Information Fusion*. IEEE, 2018, pp. 2262–2269.

[15] G. Shani, R. Brafman, and S. Shimony, "Adaptation for changing stochastic environments through online pomdp policy learning," in *Proc. Eur. Conf. on Machine Learning*. Citeseer, 2005, pp. 61–70.

[16] H. Kurniawati and V. Yadav, "An online pomdp solver for uncertainty planning in dynamic environment," in *Robotics Research*. Springer, 2016, pp. 611–629.

[17] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial Intelligence*, vol. 101, no. 1-2, pp. 99–134, 1998.

[18] E. Brunskill, L. P. Kaelbling, T. Lozano-Pérez, and N. Roy, "Planning in partially-observable switching-mode continuous domains," *Annals of Mathematics and Artificial Intelligence*, vol. 58, no. 3-4, pp. 185–216, 2010.

[19] J. M. Porta, N. Vlassis, M. T. Spaan, and P. Poupart, "Point-based value iteration for continuous POMDPs," *Journal of Machine Learning Research*, vol. 7, no. Nov, pp. 2329–2367, 2006.

[20] S. Ross, B. Chaib-draa, and J. Pineau, "Bayes-adaptive pomdps," in *Advances in neural information processing systems*, 2008, pp. 1225–1232.

[21] 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*, vol. 3, 2003, pp. 1025–1032.

[22] M. Hauskrecht, "Value-function approximations for partially observable markov decision processes," *Journal of artificial intelligence research*, vol. 13, pp. 33–94, 2000.

[23] D. Silver and J. Veness, "Monte-Carlo planning in large POMDPs," in *Advances in neural information processing systems*, 2010, pp. 2164–2172.

[24] Z. N. Sunberg and M. J. Kochenderfer, "Online algorithms for pomdps with continuous state, action, and observation spaces," in *Twenty-Eighth International Conference on Automated Planning and Scheduling*, 2018.

[25] A. Goldhoorn, A. Garrell, R. Alquézar, and A. Sanfeliu, "Continuous real time POMCP to find-and-follow people by a humanoid service robot," in *2014 IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2014, pp. 741–747.

[26] M. Egorov, Z. N. Sunberg, E. Balaban, T. A. Wheeler, J. K. Gupta, and M. J. Kochenderfer, "POMDPs.jl: A framework for sequential decision making under uncertainty," *Journal of Machine Learning Research*, vol. 18, no. 26, pp. 1–5, 2017. [Online]. Available: http://jmlr.org/papers/v18/16-300.html

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