Decentralized Information-Rich Planning and Hybrid Sensor Fusion for Uncertainty Reduction in Human-Robot Missions

Sameera Ponda[*], Nisar Ahmed[†], Brandon Luders[*], Eric Sample[†],
Tauhira Hoossainy[†], Danelle Shah[†], Mark Campbell[†], Jonathan P. How[*]

This paper introduces a novel planning and estimation framework for maximizing information collection in missions involving cooperative teams of multiple autonomous vehicles and human agents, such as those used for multi-target search and tracking. The main contribution of this work is the scalable unification of effective algorithms for distributed high-level task planning, decentralized information-based trajectory planning, and hybrid Bayesian information fusion through a common Gaussian mixture uncertainty representation, which can accommodate multiple mission objectives and constraints as well as heterogeneous human/robot information sources. The proposed framework is validated with promising results on real hardware through a set of experiments involving a human-robot team performing a multi-target search mission.

I. Introduction

Modern day mission operations often involve large teams of networked agents, with heterogeneous capabilities, interacting together to perform the requisite mission tasks. Such missions typically involve executing several different types of task at once, such as intelligence, surveillance, and reconnaissance (ISR), target classification, rescue operations, scientific exploration, and security monitoring.[1,2] Furthermore, within the heterogeneous team, some specialized agents are better suited to handle certain types of tasks than others. For example, autonomous air and ground vehicles equipped with video can be used to perform target search and track, human operators can be used for target classification tasks, and ground teams can be deployed to perform rescue operations or engage targets.

Ensuring proper coordination and collaboration between agents in the team is crucial to efficient and successful mission execution. As a result, there has been increasing interest in exploring efficient methods to plan for mixed human-robot teams for various types of missions. Furthermore, the advancement of communication systems, sensors, and embedded technology has significantly increased the value of those solutions that are scalable to larger teams, from dozens to hundreds or even thousands of agents.[1,2] In such complex systems, care must be taken to balance the resources allocated to primary mission tasks (e.g. search and tracking) and related secondary tasks (e.g. maintenance, monitoring, safety, retrieval, etc).

There are many technical challenges associated with developing algorithms that can effectively coordinate the behavior of such teams. For example, consider a scenario where a team of human operators and autonomous robots is tasked with searching for, tracking, and classifying unknown targets in an obstacle-filled environment. A key research question is how to efficiently allocate limited agent resources with the objective of minimizing target state uncertainty as quickly as possible, while simultaneously executing required secondary tasks (e.g. vehicle status monitoring, etc). Furthermore, this task assignment process must take into account the challenges associated with the underlying autonomous motion planning and navigation that the agents must perform to successfully accomplish their tasks. For example, the vehicles must be able to autonomously plan trajectories in obstacle-filled and potentially uncertain search environments, minimizing target state uncertainty while also ensuring safety. An additional consideration for this problem is that, given that many disjoint and heterogeneous agents are collaborating to search the environment, it is important to employ efficient information fusion methods, which can be used to effectively combine sensor data acquired

[*] S. Ponda, B. Luders, and J. P. How are with the Dept. of Aeronautics and Astronautics, MIT, Cambridge, MA, {sponda, luders, jhow}@mit.edu

[†] N. Ahmed, E. Sample, T. Hoossainy, D. Shah and M. Campbell are with the Dept. of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY {nra6, ems346, th432, dcs45, mc288}@cornell.edu

by different mobile agents with information from human operators. Since most planning strategies rely on underlying agent models, developing accurate and efficient representations for agents in the team, including human operators, is crucial. In particular, modeling human agents for use in autonomous task allocation and information fusion algorithms remains a challenging problem.[3] Finally, any approach considered should be able to scale with the problem size, characterized by the number of agents and targets, without straining available computational or communication resources.

This work presents an algorithmic approach to tackle task allocation, trajectory planning and information fusion within a *unified framework*, with the objective of reducing uncertainty in the target search and tracking process, while considering the complex constraints associated with realistic human-robot missions. In this novel approach, the goal of maximizing information is a primary objective for each of the algorithms at every step, producing a cohesive framework that strategically addresses the main mission objectives. Both task planning and vehicle path planning are information based, enabling intelligent and efficient cooperative search and track strategies that are balanced alongside other mission objectives. The task allocation and trajectory planning algorithms employed are distributed, making the system scalable to large teams of operators and autonomous agents with diverse potential task sets. Furthermore, the information fusion algorithms presented in this work provide strategies to directly include "soft" inputs from human agents, that can be combined with conventional autonomous sensor information via robust particle filtering algorithms, enabling convenient recursive Bayesian updates for efficient replanning. The unified task allocation, trajectory planning and information fusion framework is validated in a real-time human-robot multi-target search experiment, demonstrating the viability of the approach.

This paper is organized as follows. Section II defines the problem statement considered by this work. Section III presents the distributed planning and information fusion framework developed to address this problem, including the overall system architecture (Section III-A), the information-rich planning algorithms (Section III-B), and the Bayesian hybrid data fusion algorithms (Section III-C). Indoor target search and track experiments for human-robot teams using the proposed framework are presented and analyzed in Section IV, followed by concluding remarks in Section V. Note that related work is provided throughout the paper, in the corresponding sections.

II. Problem Formulation and Background

This work considers the problem of planning for a team of autonomous robotic mobile agents[a] and human operators, tasked with searching for, tracking, and classifying unknown targets in an obstacle-filled dynamic environment. The robotic agents consist of heterogeneous vehicles equipped with onboard computers and a variety of sensors, such as laser range-finders, cameras and visual detection software. The human operators are static and can interact with the robotic agents directly through a computer console. The team's mission is to locate and identify a known number of targets as quickly and accurately as possible in a real-time environment. The details of this search and track problem are described below.

Assume that search region $\mathcal{S} \subseteq \mathbb{R}^3$ contains N static targets with fixed labels $i \in \{1, \ldots, N\}$ and unknown positions $\mathbf{x}_i = [x_i, y_i, z_i]^T$ with respect to some fixed origin (N is known *a priori*). The uncertainty in $\mathbf{x}_i$ is initially modeled by the probability density function (PDF) $p(\mathbf{x}_i)$. This PDF represents any prior beliefs about $\mathbf{x}_i$ (e.g. as obtained from intelligence information, previous experience, or physical considerations). Using the initial target PDFs, $\{p(\mathbf{x}_1), \ldots, p(\mathbf{x}_N)\}$, and a set of observations, $\mathcal{Z}$, acquired by the human-robot team throughout the mission, the primary objective is to detect, identify and localize all N targets in $\mathcal{S}$ as quickly and efficiently as possible. The exact specification of this objective function might include a maximum time limit, a maximum uncertainty covariance for each target, a weighted sum of these factors, or several other considerations (such as specific vehicle constraints).

It is assumed here that each target distribution $p(\mathbf{x}_i)$ is a known M_i-term Gaussian mixture (GM),

$$p(\mathbf{x}_i) = \sum_{m=1}^{M_i} w_{i,m} \, \mathcal{N}(\mu_{i,m}, \Sigma_{i,m}), \tag{1}$$

where the parameters $w_{i,m}$, $\mu_{i,m}$, and $\Sigma_{i,m}$ are respectively the weight, mean, and covariance matrix for component m of target i, with $\sum_{m=1}^{M_i} w_{i,m} = 1$. It is well-known that GMs can approximate arbitrarily com-

[a] The framework considered in this paper can be extended to incorporate human-operated mobile agents, though this is not discussed further.

plex PDFs for suitably chosen M_i and mixing components,[4] and are thus quite useful in general estimation problems with significant non-Gaussian uncertainties.[5] At any given time, the aggregated estimate of each target is given by the mean of the distribution which can be computed from the individual modes as

$$\hat{\mathbf{x}}_i = \sum_{m=1}^{M_i} w_{i,m}\,\mu_{i,m}, \tag{2}$$

with target covariance given by

$$P_i = \sum_{m=1}^{M_i} w_{i,m}\left[\Sigma_{i,m} + (\mu_{i,m} - \hat{\mathbf{x}}_i)(\mu_{i,m} - \hat{\mathbf{x}}_i)^T\right]. \tag{3}$$

The target locations are further assumed to be marginally independent, so that the joint target PDF is given by

$$p(\bar{\mathbf{x}}) = p(\mathbf{x}_1, ..., \mathbf{x}_N) = \prod_{i=1}^{N} p(\mathbf{x}_i). \tag{4}$$

If the human-robot team acquires a set of shared target observations $\mathcal{Z}_k$ up to time step k, then the distribution for $\mathbf{x}_i$ can be updated via Bayes' rule as

$$\begin{aligned} p(\mathbf{x}_i|\mathcal{Z}_k) &= \frac{1}{K} p(\mathbf{x}_i) p(\mathcal{Z}_k|\mathbf{x}_i), \\ K &= \int p(\mathbf{x}_i) p(\mathcal{Z}_k|\mathbf{x}_i) d\mathbf{x}_i, \end{aligned} \tag{5}$$

where $p(\mathcal{Z}_k|\mathbf{x}_i)$ is the likelihood function for the observations $\mathcal{Z}_k$, and K is a normalizing constant.

In the context of mixed human-robot search teams, the likelihood function, $p(\mathcal{Z}_k|\mathbf{x}_i)$, is composed of several independent models describing how measurements from various sensing platforms are stochastically generated as a function of the underlying target states. For robotic agents, the likelihood function characterizes measurements arising from typical robot sensing platforms, such as cameras and LIDAR. In human-robot search teams, human operators also contribute important target information, particularly with respect to target identification and high-level target behaviors,[6] but this information typically has limited usefulness in reducing uncertainty in $\mathbf{x}_i$, since it is either not very related (e.g. target classification), or cannot be properly modeled in $p(\mathcal{Z}_k|\mathbf{x}_i)$ unless the target has been extensively characterized through an *a priori* behavioral model. However, human operator insight is often valuable in guiding search missions, and, in many cases, it is desirable to include these "low-level" observations from operators as "soft inputs" in $\mathcal{Z}_k$ in Equation (5), thus allowing human insight to be treated as a sensor that returns continuous or categorical observations of continuous states, such as the target locations.[7,8]

An alternative characterization of the search and track problem described above involves modeling the search mission as an optimal control problem, where the objective is to place the sensing agents on trajectories that maximize the probability of finding the targets over a given time horizon. One strategy to accomplish this is to minimize the uncertainty in the posterior (Equation (5)), for example, by using a receding horizon planning strategy that accounts for sensor platform dynamics.[9] For heterogeneous multi-agent search teams, a centralized planning approach with a shared information set could be used in the optimization, but such methods usually scale poorly with the size of the search area, target population, and the number of agents. Recent work[10] considers how to perform decentralized target search in two dimensions, via a discretized representation; however, this approach also scales poorly in three dimensions and with increasing problem sizes, as well as with other realistic constraints such as target dynamics and communication constraints.

In this work, an information-based approach is employed to address the search and track problem at both the task assignment and trajectory planning levels. The solution methodologies do *not* require the discretization of the search space, although the environment is assumed to be bounded and non-convex. The task assignment process determines which agents are best suited to track which targets given their sensor configurations, current pose, and the prior target estimates provided by the GMs (Section III-B). Once the targets are assigned to the respective vehicles, the motion planning algorithm designs information-rich kinodynamically feasible trajectories which traverse this continuous environment while satisfying all state and input constraints[11] (Section III-B). The vehicles are assumed to have known dynamics and sensor/detection models (though they may be nonlinear), such that predicted trajectories can be generated deterministically. Reliable pose estimates and environmental obstacle maps are assumed to be available to each agent for convenience, although extensions to uncertain pose and maps are also possible and will be studied in future work. Furthermore, all trajectory planning is decentralized and performed by each vehicle independently; the paths of other agents are assumed unknown, although this information could be shared among the agents. While more efficient sensor fusion can be achieved in such extended search problems using GM representations,[12] there has been little prior work on how to effectively embed GMs into the planning framework. The algorithms proposed by this paper incorporate the GM target representations at each level of planning, including task allocation, trajectory planning, and human operator interface. By using computationally efficient algorithms in each of these phases, it is possible for large teams to develop real-time plans which explicitly account for the nature of the target uncertainty at every level.

III. Decentralized Planning and Fusion Framework

This section outlines the proposed framework for managing a team of human operators and autonomous vehicles engaged in a generalized target search, tracking, and identification mission. The presented approach consists of three primary algorithmic components: task allocation, trajectory planning, and information fusion. The key contribution of this work is the development of a unified framework which integrates these algorithms, allowing for the explicit consideration of target uncertainty reduction, complex constraints, and secondary objectives (e.g. safety, refueling, etc.) at every level of planning. Section III-A presents the overall system architecture. Section III-B reviews the task planning and vehicle path planning algorithms, describing how information gains are directly accounted for in the planning process, enabling the algorithms to balance information collection with other mission objectives. Finally, Section III-C presents the hybrid Bayesian fusion strategy, which combines traditional sensor models with low-level categorical human observations of target states.

A. Proposed Information-based Control Architecture

This section presents the overall system architecture for the types of planning and fusion problems considered in this work, describing the relationship between the individual components. A diagram of the generalized framework is presented in Figure 1. The main components, as shown in the figure, consist of task allocation, path planning, vehicle and sensor configurations, and state estimation and sensor fusion. The task allocation algorithm receives the latest state estimates of both the vehicles and targets, and uses this information, along with accurate models of the agents and sensors, to determine the assignment of targets to vehicles. These task assignments are then communicated to the individual vehicle path planners. The path planning algorithms design trajectories for the vehicles that minimize the target state uncertainty while considering resource consumption and obstacle avoidance. The vehicles then implement these trajectories, update their pose estimates, and collect observations via their sensors. The individual agent state and sensor data is sent to a state estimation and sensor fusion module that combines all this information to obtain the latest estimates of the agent and target states, along with measures of the estimation uncertainty.

Figure 2 shows a diagram of the proposed information-rich planning and fusion framework presented in this paper. The task allocation algorithm in the proposed approach consist of the decentralized Consensus-Based-Bundle Algorithm (CBBA)[13] augmented with information metrics, the path planning uses the Information-rich Rapidly-exploring Random Tree (IRRT)[11] algorithm, and the state estimation is performed by a recursive hybrid Bayesian fusion strategy. The hardware platform used to obtain experimental results consisted of a Pioneer rover equipped with cameras (Section IV). The key principle behind this framework is that task allocation, trajectory planning, and sensor fusion all consider acquiring information and reducing target uncertainty as the primary objectives, creating a unified framework for target tracking that addresses the main mission goals at every level. A secondary advantage is that both the task assignment and trajectory planning are decentralized, as illustrated in Figure 2, providing a scalable solution methodology which remains computationally tractable as the number of agents and targets increases. An additional contribution illustrated in this framework is the explicit use of human operators in the control and estimation loop, via a human-robot interface (HRI). In this formulation, human operators provide “soft inputs” to the sensor fusion, validating the identity of all potential target detections in addition to other target state information

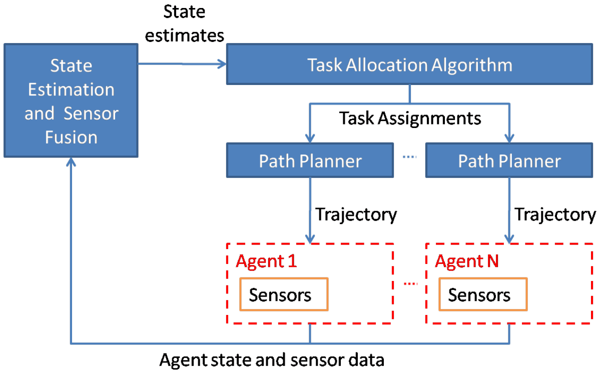


Figure 1. General system block diagram for proposed planning and fusion framework.

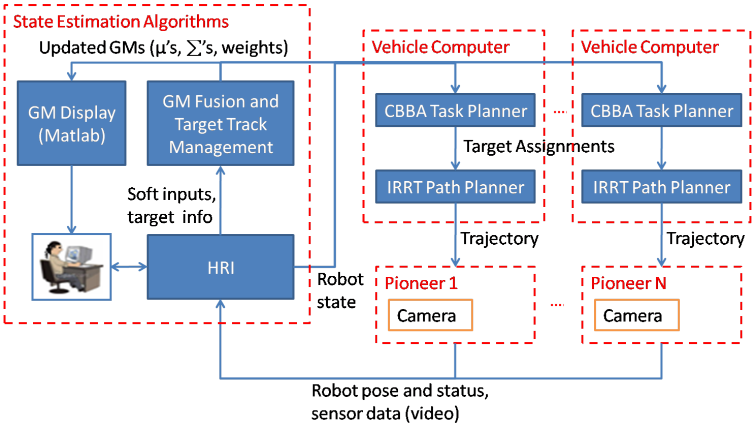
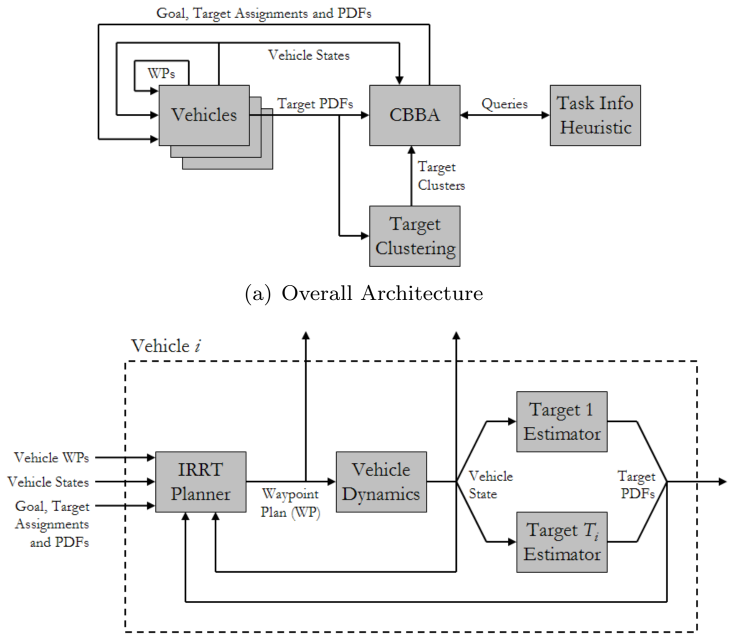


Figure 2. System block diagram for indoor human-robot target search and track experiment

which assists the robots in their search (e.g. these can include fuzzy descriptions of perceived target locations such as 'nearby landmark A' or perceived target behaviors such as 'moving quickly through the side door'). Operators may also be used to handle some secondary tasks, such as monitoring refueling operations or responding to automation failures. The following sections provide further details on these algorithmic system components.

B. Decentralized Information-Rich Planning

The performance of dynamic search and track missions is typically measured in terms of the efficiency with which the agents involved reduce target estimation uncertainty. However, trajectories that achieve this uncertainty reduction are subject to a complex set of internal and external constraints, including dynamic constraints, environmental restrictions, and sensor limitations. By using the recently-proposed Information-rich Rapidly-exploring Random Tree (IRRT) algorithm,[14] a team of agents can quickly identify feasible, uncertainty-reducing paths that explicitly embed the latest target probability distributions, whilst satisfying these constraints. While IRRT is capable of handling multiple vehicles and targets,[11] algorithmic efficiency is lost when considering realistic large-scale ISR missions. Trajectories identified for such scenarios must embed both the vehicle routing problem (in selecting which distant targets to visit) and the constrained sensor problem (in finding a vantage point to view nearby targets), and become computationally intractable as the number of agents and targets increases. By pursuing a distributed approach that partitions the target environment into disjoint tasks and allocates these tasks amongst the agents, the computational burden on the motion planners is reduced. In this work we use a decentralized task allocation algorithm called the Consensus-Based Bundle Algorithm (CBBA)[13] to distribute the targets to the individual agents. The score functions used within the CBBA task allocation framework explicitly account for the information that agents



(a) Overall Architecture

(b) Single Vehicle Architecture (leftmost block of Figure 3(a))

Figure 3. Block diagrams illustrating the overall CBBA+IRRT integrated architecture.

are able to obtain about their assigned targets.

The combination of IRRT+CBBA results in a novel multi-level algorithm which embeds information-rich trajectory planning within a task allocation framework, efficiently assigning targets and planning paths for teams of agents at the mission planning level. This real-time algorithm can leverage networks of mobile sensor agents to perform dynamic task reallocation as target estimates are updated, resulting in improved coordination and collaboration between agents while executing the mission. Figure 3 shows the proposed IRRT+CBBA architecture, where each vehicle runs an instance of the decentralized CBBA task allocation algorithm as well as its own IRRT planner. The next sections provide further detail on these two components of the decentralized planning process.

1. *Decentralized Information-Based Task Allocation*

The problem of task allocation has been extensively studied and many different methods have been considered for enabling agents to distribute tasks amongst themselves from a known mission task list (see [13, 15] and the references contained therein for more details). Centralized planners, which rely on agents communicating their state to a central server that generates a plan for the entire fleet, are commonly used in the literature. However, most of these planning architectures require high communication bandwidth, computational resources, and are typically slower to react to changes in local information. Decentralized planning algorithms, where agents make their own plans and communicate amongst themselves, have gained recent popularity, and offer several advantages over centralized planning methods.[16,17] Many of these decentralized algorithms have to be augmented with consensus algorithms for agents to converge on consistent situational awareness prior to planning,[18,19] a process that can take a significant amount of time and often requires transmitting large amounts of data.[20] A unique decentralized auction algorithm called the Consensus-Based Bundle Algorithm (CBBA)[13,15] uses a consensus protocol that acts upon the task space only, guaranteeing conflict-free solutions despite possible inconsistencies in situational awareness. CBBA is guaranteed to achieve at least 50% optimality,[13] although empirically its performance is shown to be within 93% of the optimal solution.[21] The task selection process of CBBA runs in polynomial time, demonstrating good scalability with increasing numbers of agents and tasks, making it well suited to real-time dynamic environments.

This work uses CBBA to allocate targets to the best suited agents. Figure 3(a) shows the overall target allocation architecture which is described in this section. Prior to the task allocation process, the targets are grouped into sets using K-means clustering on the target means obtained from the latest target Gaussian mixture estimates. These target sets or “tasks” can then be allocated to the individual agents using CBBA. A key advancement of the CBBA algorithm is a novel information-based scoring framework called the Task Information Heuristic (TIH), which embeds an approximation of the information gain in the assessed value of a target cluster to an agent or team. The TIH consists of selecting a starting location to enter the target cluster[b], followed by a one-step optimization process to find the best information-rich path within the cluster, providing an estimate of the locally optimal information-gathering trajectory. The path optimization involves minimizing the average A-optimality of the individual Fisher Information Matrices for each target,[22] and the algorithm continues to extend the path until this average A-optimality is below some uncertainty threshold (or some timeout is reached). The individual target Fisher Information Matrices are initialized using the inverses of the target covariance matrices obtained from the latest target PDFs, thus accounting for the actual acquired information thus far. Finally, the estimated score for the task is computed as the expected acquired information for all targets, minus the fuel resources consumed by following the optimized path. Likewise, the arrival time and task duration are approximated using the agent’s arrival time at the selected start point, and the time required to traverse the optimized path, respectively. Using the estimated scores, task durations, and arrival times, CBBA is able to allocate the tasks to the individual agents producing target lists and expected schedules for each vehicle.

2. Information-Rich Path Planning

Given the target lists produced by the task allocation process, each agent must plan a trajectory that enables the vehicle to search and track the targets assigned to it as efficiently as possible. Due to its explicit consideration of target uncertainty reduction, this work employs the Information-rich Rapidly-exploring Random Tree (IRRT) algorithm.[11,14] IRRT uses a closed-loop state prediction in conjunction with sensor models and target prior distributions to simulate a tree of candidate trajectories. Using Fisher information,[23] the value of successful measurement poses along each path can be quantified, allowing trajectories to be selected via a trade-off between uncertainty reduction and path duration. As an extension of RRT, the IRRT algorithm is amenable to the general, complex constraint characterizations often encountered in real-world planning problems. This section reviews the IRRT formulation and describes, in particular, how information collection is quantified.

From the perspective of information collection, path quality is a function of the path measurement sequence. And while CL-RRT also enjoys the benefits of smoother path planning on a stabilized vehicle model, it is the added disturbance robustness over open-loop RRT[24] and the associated accurate state prediction that are particularly useful for measurement pose prediction and, therefore, for information-based planning. Because the vehicle’s state trajectory is usually simulated with high fidelity, and the result of its prediction is notably accurate, a list of predicted measurement poses $\mathcal{M} = \langle \overline{\mu}_1, \overline{\mu}_2, ..., \overline{\mu}_l \rangle$ can be interpolated for each of many (possibly independent) sensors on the platform. These sensors need not have the same characteristics. Each sensor’s list of predicted measurement poses is generated once per node, and thereafter has no need to be updated. Given the most recent modal state estimates $\hat{\mathbf{x}}_{i,m}$ of target i with modes $m \in \{1, \ldots, M_i\}$, each measurement pose $\overline{\mu}_k, k \in \{1, \ldots, l\}$ can be checked against the sensor and environment models to assess visibility. The information for measurements deemed visible is quantified, as described below, and stored in the resulting node n_{new}. Visibility and information quantification of the $\mathcal{M}$ elements may be recomputed as target estimation data is updated.

A myriad of information-theoretic metrics exist to quantify the value of a set of measurements; we use the Fisher Information Matrix (FIM) $J_{\mathbf{Z}}(\mathbf{x})$, which describes the information contained in a set of measurements $\mathbf{z}$ about an estimation process for the vector $\mathbf{x}$. The inverse $J_{\mathbf{Z}}(\mathbf{x})^{-1}$ of the Fisher Information Matrix is exactly the Cramér-Rao Lower Bound (CRLB), a lower bound on the achievable estimation error covariance and thus a quantity to be minimized.[25] A discrete system with linear state transitions and measurements, subject to additive Gaussian white noise, can be modeled as

$$\begin{aligned} \mathbf{x}_{k+1} &= \Phi_{k+1|k}\mathbf{x}_k + \mathbf{w}_k, \\ \mathbf{z}_k &= H_k\mathbf{x}_k + \mathbf{v}_k, \end{aligned} \tag{6}$$

[b] The task start location for each vehicle is determined by computing the closest point on the outer edge of a sphere around the cluster’s centroid, whose radius is given by the average cluster spread, with an additional margin to avoid starting inside any target’s no-fly zone.

where $\Phi_{k+1|k}$ is the state transition matrix, H_k is the linear measurement matrix, $\mathbf{w}_k$ is the process noise, and $\mathbf{v}_k$ is the sensing noise. The process and sensing noises are assumed to be Gaussian, zero-mean and uncorrelated, with covariances given by Q_k and R_k respectively. For such systems, the recursive update equation for the FIM is given by[26]

$$J_{k+1} = (Q_k + \Phi_{k+1|k} J_k^{-1} \Phi_{k+1|k}^T)^{-1} + H_{k+1}^T R_{k+1}^{-1} H_{k+1}. \tag{7}$$

For stationary targets, $Q_k = 0$ and $\Phi_{k+1|k} = I$ for all k, and the recursion becomes

$$J_{k+1} = J_k + H_{k+1}^T R_{k+1}^{-1} H_{k+1}, \tag{8}$$

a particularly convenient form since the FIM in this case is additive, and the information content of a path is just the sum of the FIMs along the path edges. Using this form provides considerable computational savings over planning methods that propagate the covariance, since it does not require the computation of matrix inverses.

The linearity assumption on the observation system can be relaxed by utilizing the linearized FIM as an approximation of the CRLB inverse. Consider systems with discrete measurements $\mathbf{z}$ that are nonlinear in both the target state $\mathbf{x}_i$ and measurement pose μ, and are thus of the form

$$\mathbf{z_k} = \mathbf{h}(\mu_k, \mathbf{x}_i) + \mathbf{v_k}, \tag{9}$$

where $\mathbf{v}_k$ is a vector of zero-mean, white Gaussian sequences. The approximate FIM can be formulated by defining H_k to be the Jacobian of the nonlinear measurement function, i.e.,

$$H_k(\overline{\mu}_k, \hat{\mathbf{x}}_i) \triangleq \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right|_{\mu_k = \overline{\mu}_k, \quad \mathbf{x}_i = \hat{\mathbf{x}}_i(t)}. \tag{10}$$

Note that the assumption of Gaussian noise is retained.

The expected measurement poses $\langle \overline{\mu}_1, \ldots, \overline{\mu}_l \rangle$ can be used in the framework of the FIM to quantify the information content of a particular node in the tree. The approach is to compute FIMs for each target mode separately, approximate a lower bound on the target mixture covariance, and combine the information error from all N targets. The modal FIMs are stored in that node and are used as the initial information conditions for its children. Assuming a Gaussian mixture prior, the recursion is initiated at the root node n_{root} with $J_{root}(\hat{\mathbf{x}}_{i,m}) = P_{i,m}(t)^{-1}$, where $P_{i,m}(t) = \mathbb{E}\left[(\mathbf{x}_{i,m} - \hat{\mathbf{x}}_{i,m}(t))(\mathbf{x}_{i,m} - \hat{\mathbf{x}}_{i,m}(t))^T\right]$ is the error covariance matrix for mode m of target i at time t. For each target i, for each mode m, the FIM $J_b(\hat{\mathbf{x}}_{i,m})$ of a child node n_b is formed by a recursive update from its parent node n_a,

$$J_b(\hat{\mathbf{x}}_{i,m}) = J_a(\hat{\mathbf{x}}_{i,m}) + \sum_{k=1}^{l} \nu(\overline{\mu}_k, \hat{\mathbf{x}}_{i,m}, \hat{\mathcal{E}}) H_k^T(\overline{\mu}_k, \hat{\mathbf{x}}_{i,m}) R_k^{-1} H_k(\overline{\mu}_k, \hat{\mathbf{x}}_{i,m}), \tag{11}$$

where l is the number of measurements along the path segment, $\hat{\mathcal{E}}$ is the environment representation, and ν is a binary-valued function capturing the success/occlusion of a measurement. In this way, the tree FIMs are populated and can be recomputed, for example, after the target distributions have been updated.

In the presented approach, the cost associated with information for target i at node n_a is specified as the A-optimality criterion on a lower bound of the mixture covariance, specifically,

$$\mathcal{I}_a(\hat{\mathbf{x}}_i) = \mathtt{trace}\left(\sum_{m=1}^{M_i} w_{i,m} J_a^{-1}(\hat{\mathbf{x}}_{i,m})\right). \tag{12}$$

The A-optimality criterion has been shown to be better suited than other FIM optimality conditions for the 3D target tracking case.[22] In the multi-target case, convex combinations of the A-optimality costs can be found by summing over the targets,

$$\mathcal{I}_a = \sum_{i=1}^{N} q_i \, \mathcal{I}_a(\hat{\mathbf{x}}_i), \qquad \sum_{i=1}^{N} q_i = 1 \tag{13}$$

Algorithm 1 IRRT, Tree Expansion

```
 1: Take a sample $x_{samp}$ from the environment
 2: Identify the nearest node $n_{near}$ using mixture of exploration, optimization, and information heuristics
 3: $\overline{x}(t+k) \leftarrow$ final state of $n_{near}$
 4: while $\overline{x}(t+k) \in \mathcal{X}_{free}$ and $\overline{x}(t+k)$ has not reached $x_{samp}$ do
 5:     Use reference law to generate $\overline{r}(t+k)$
 6:     Use control law to generate $\overline{u}(t+k)$
 7:     Use prediction model to simulate $\overline{x}(t+k+1)$
 8:     $k \leftarrow k+1$
 9: end while
10: for each feasible node $n$ generated do
11:     Update cost estimates for $n$
12:     Compute simulated measurement poses $n.\mathcal{M}$
13:     Compute FIMs using (11)
14:     Add $n$ to $\mathcal{T}$
15: end for
```

Algorithm 2 IRRT, Execution Loop

```
 1: $t \leftarrow 0$
 2: Initialize tree $\mathcal{T}$ with node at $x(0)$
 3: while $x(t) \neq \mathbf{x}_{goal}$ do
 4:     Update the current state $x(t)$ and target estimates $\hat{x}_i \; \forall i$
 5:     Propagate the state $x(t)$ by $\Delta t \rightarrow \overline{x}(t+\Delta t)$
 6:     while time remaining for this timestep do
 7:         Expand tree by adding nodes
 8:     end while
 9:     Update FIMs throughout $\mathcal{T}$ using (11)
10:     Use information-based cost metric to identify best feasible path, $\{n_{root}, \ldots, n_{select}\}$
11:     Apply best feasible path, if one exists
12:     $t \leftarrow t + \Delta t$
13: end while
```

where the relative weights q_i can be used to bias information collection towards some targets (e.g. mission-critical targets). Summation of the A-optimality costs is consistent with the nature of the multi-objective problem.[c]

The ability to simulate expected measurement poses is used in two ways to extend the CL-RRT algorithm for information gathering. First, these expected measurements are used to bias tree growth toward regions of high information-gain[11] (Algorithm 1)[d]. Second, the vehicle selects paths from the tree that minimize a cost function which explicitly considers information, in addition to path cost and remaining cost-to-go.

Whenever new feasible nodes n_{new} are generated for the tree, the predicted measurement poses $\mathcal{M}$ are stored within the node (line 12). These measurement poses are used to compute the FIMs based on the current target estimates $\hat{\mathbf{x}}_{i,m}$ for all i and m, both when the node is created (line 13) and whenever the best path is selected, as discussed next.

The IRRT execution loop is presented in Algorithm 2. In the IRRT algorithm, a single, multi-objective cost metric is used (Algorithm 2, line 10), which considers both progress toward the goal and the value of information collection. This cost function here takes the form

$$C(n_a) = \alpha_\tau \tau \left(n_a | n_{root}\right) + \tau^* \left(n_a\right) + \alpha_{\mathcal{I}} \mathcal{I}_a, \tag{14}$$

where $\tau(n_a|n_{root})$ is the simulated time to travel from the root node n_{root} to node n_a, $\tau^*(n_a)$ is the lower-bound cost-to-go (e.g. Euclidean or Dubins length divided by average speed) from n_a to the goal, and $\mathcal{I}_a$ is the information-related cost component. The weights α_τ and $\alpha_{\mathcal{I}}$ can be adjusted to reflect the relative importance of information gathering and of following minimal-time paths to the goal. To ensure all recent measurements are taken into account, the latest target estimates are measured at the beginning of each execution loop (line 4), which are then used to update the FIM of each node in the tree (line 9). Though this FIM update is performed on the entire tree on each pass, this is a computationally efficient operation compared to other aspects of the algorithm, such as constraint evaluation.

Of particular note with this cost function is that it can be shown to result in "smooth" mission-level behaviors, in the sense that negligible churning between information collection and goal directedness exists. Rather, the planner is always conscious of the inherent tradeoff and will generate behaviors that, for example, conclude missions by maneuvering to collect information while remaining relatively close to the goal. It should also be noted as a limitation of IRRT, and RRTs in general, that mission-critical requirements like maximum allowable duration and/or minimum required information collection are not well handled; it is difficult enough to find, let alone guarantee that one could find, a feasible solution to such requirements in finite time. Despite this, IRRT has been shown through simulations to perform well empirically under a number of previously prohibitive general constraints. Furthermore, recent flight results have demonstrated the viability of the IRRT approach, incorporating multiple vehicles, complex uncertainty models and sensors in the loop.[11]

[c] It should be noted that simply summing the FIMs (and not the associated A-optimality costs) over all targets at a given measurement pose is imprudent; for example, two targets with singular FIMs could in their sum form a nonsingular FIM, thereby masking the momentary unobservability of each target's estimation process.

[d] See [24] for more information on existing components.

C. Recursive Bayesian Hybrid Data Fusion

This section describes the components of the "State Estimation" block in Figure 2, which combines observations made by human and robot agents to update the Gaussian Mixture target PDFs used by the CBBA task allocation algorithm and IRRT path planning algorithm for each agent. The proposed sensor fusion process is centralized and leads to recursive Bayesian GM updates. Decentralized recursive Bayesian fusion with GMs remains a challenging problem[27] and will be addressed in future work.

1. *Overview of Gaussian Mixture Fusion Updates and Measurement Models*

Let $\mathcal{Z}_k = \left\{\mathbf{Z}_1^{total}, \ldots, \mathbf{Z}_k^{total}\right\}$ be the set of all available observations up to time k from human and robot agents, where $\mathbf{Z}_k^{total} = \left\{\mathbf{Z}_k^r, \mathbf{Z}_k^h\right\}$ is the set of observations $\mathbf{Z}_k^r$ from robot agents at time k and the set of observations $\mathbf{Z}_k^h$ from human agents at time k. For N_r robot agents and N_h human agents, $\mathbf{Z}_k^r$ contains $N_r^k \leq N_r$ measurements $\mathbf{z}_k^{r,j} \in \mathbb{R}^{n_r \times 1}$ for $j \in \{1, \ldots, N_r\}$ and $\mathbf{Z}_k^h$ contains $N_h^k \leq N_h$ measurements $\mathbf{z}_k^{h,j} \in \mathbb{R}^{n_h \times 1}$ for $j \in \{1, \ldots, N_h\}$, where n_r and n_h are the fixed sizes of the robot and human measurement vectors, respectively. The observations $\mathbf{z}_k^{r,j}$ and $\mathbf{z}_k^{h,j}$ are generally non-linear and described stochastically by non-Gaussian likelihood functions $p(\mathbf{z}_k^{r,j}|\mathbf{x}_i)$ and $p(\mathbf{z}_k^{h,j}|\mathbf{x}_i)$, where it is assumed that reliable agent state estimates and environment maps are available so that only $\mathbf{x}_i$ is uncertain. A more general treatment of the fusion problem that includes uncertain target dynamics, agent states and environment maps is also possible, but is omitted here for brevity.

If $\mathbf{Z}_k^r$ and $\mathbf{Z}_k^h$ are conditionally independent given $\mathbf{x}_i$, the Bayesian posterior (Equation (5)) can be recursively computed as

$$p(\mathbf{x}_i|\mathcal{Z}_k) = \frac{1}{K} p(\mathbf{x}_i|\mathcal{Z}_{k-1}) p(\mathbf{Z}_k^{total}|\mathbf{x}_i) = \frac{1}{K} p(\mathbf{x}_i|\mathcal{Z}_{k-1}) p(\mathbf{Z}_k^r|\mathbf{x}_i) p(\mathbf{Z}_k^h|\mathbf{x}_i), \tag{15}$$

where

$$K = \int p(\mathbf{x}_i|\mathcal{Z}_{k-1}) p(\mathbf{Z}_k^r|\mathbf{x}_i) p(\mathbf{Z}_k^h|\mathbf{x}_i) d\mathbf{x}_i \tag{16}$$

is a normalization constant, and

$$\begin{aligned} p(\mathbf{Z}_k^r|\mathbf{x}_i) &= \prod_{\mathbf{z}_k^{r,j} \in \mathbf{Z}_k^r} p(\mathbf{z}_k^{r,j}|\mathbf{x}_i), \\ p(\mathbf{Z}_k^h|\mathbf{x}_i) &= \prod_{\mathbf{z}_k^{h,j} \in \mathbf{Z}_k^h} p(\mathbf{z}_k^{h,j}|\mathbf{x}_i), \\ p(\mathbf{x}_i|\mathcal{Z}_0) &= p(\mathbf{x}_i). \end{aligned} \tag{17}$$

Since $p(\mathbf{x}_i|\mathcal{Z}_k) = p(\mathbf{x}_i|\mathcal{Z}_{k-1}, \mathbf{Z}_k^r, \mathbf{Z}_k^h)$, Equation (15) is factored into sequential Bayesian updates for $\mathbf{Z}_k^r$ and $\mathbf{Z}_k^h$,

$$p(\mathbf{x}_i|\mathcal{Z}_{k-1}, \mathbf{Z}_k^r) = \frac{1}{K_r} p(\mathbf{x}_i|\mathcal{Z}_{k-1}) p(\mathbf{Z}_k^r|\mathbf{x}_i) \tag{18}$$

$$p(\mathbf{x}_i|\mathcal{Z}_{k-1}, \mathbf{Z}_k^r, \mathbf{Z}_k^h) = \frac{1}{K_h} p(\mathbf{x}_i|\mathcal{Z}_{k-1}, \mathbf{Z}_k^r) p(\mathbf{Z}_k^h|\mathbf{x}_i), \tag{19}$$

where

$$\begin{aligned} K_r &= \int p(\mathbf{x}_i|\mathcal{Z}_{k-1}) p(\mathbf{Z}_k^r|\mathbf{x}_i) d\mathbf{x}_i, \\ K_h &= \int p(\mathbf{x}_i|\mathcal{Z}_{k-1}, \mathbf{Z}_k^r) p(\mathbf{Z}_k^h|\mathbf{x}_i) d\mathbf{x}_i. \end{aligned}$$

Finally, (18) and (19) are evaluated using N_r^k and N_h^k recursive Bayesian updates, respectively,

$$p(\mathbf{x}_i|\mathcal{Z}_{k-1}, ..., \mathbf{z}_k^{r,j}) = \frac{1}{K_k^{r,j}} p(\mathbf{x}_i|\mathcal{Z}_{k-1}, ..., \mathbf{z}_k^{r,j-1}) p(\mathbf{z}_k^{r,j}|\mathbf{x}_i), \tag{20}$$

$$p(\mathbf{x}_i|\mathcal{Z}_{k-1}, \mathbf{Z}_k^r, ..., \mathbf{z}_k^{h,j}) = \frac{1}{K_k^{h,j}} p(\mathbf{x}_i|\mathcal{Z}_{k-1}, \mathbf{Z}_k^r, ..., \mathbf{z}_k^{h,j-1}) p(\mathbf{z}_k^{h,j}|\mathbf{x}_i), \tag{21}$$

where the indices (r,j) and (h,j) respectively denote the agents with measurements in $\mathbf{Z}_k^r$ and $\mathbf{Z}_k^h$ for $j \in \{1,...,N_k^r \text{ or } N_k^h\}$, and the constants $K_k^{r,j}$ and $K_k^{h,j}$ are the required normalizing integrals. After (18) and (19) are evaluated at each time step k for each target i, the updated GM PDFs are fed back to the agents so that they can replan and execute tasks more efficiently via CBBA and IRRT in light of newly acquired information.

$\mathbf{Z}_k^r$ and $\mathbf{Z}_k^h$ are both assumed here to contain discrete multi-category observations with respect to $\mathbf{x}_i$. Prior to actual target acquisition, the discrete target detector outputs for each robot vehicle can be used to update the continuous target PDFs via Equation (18), as the probability of detection/no detection function can be treated as the likelihood $p(\mathbf{z}_k^{r,j}|\mathbf{x}_i)$ for the binary observation $\mathbf{z}_k^{r,j} \in \{$“no detection”, “detection”$\}$.[9,10] Since $\mathbf{z}_k^{r,j}$ depends on the vehicle pose at time k and on $\mathbf{x}_i$, the fusion of “no detection” observations squashes $p(\mathbf{x}_i|\mathcal{Z}_{k-1})$ with vehicle j’s “no detection” likelihood at time k, thus shifting probability mass to regions of $\mathcal{S}$ where the target is more likely to be detected.

Similar “GM squashing” updates can also be induced in Equation (19) by “soft” human observations $\mathbf{z}_k^{h,j}$ that take the form of ambiguous categorical location descriptions. For instance, a human agent might observe “something is around landmark A”, “something is behind wall 2”, or “nothing is in front of the robot”, where the prepositional phrases “around landmark A”, “behind wall 2”, and “in front of the robot” can be treated as fuzzy categorical labels for coarse range and bearing measurements relative to known locations on a common map. As with probability of detection models, such terms can be modeled probabilistically via likelihood functions that squash the target PDFs towards/away from specified map reference points via Bayes’ rule.

The likelihood functions $p(\mathbf{z}_k^{r,j}|\mathbf{x}_i)$ and $p(\mathbf{z}_k^{h,j}|\mathbf{x}_i)$ are modeled here via multimodal softmax (MMS) models, which enable simple piecewise linear representations of “continuous-to-discrete” probability surface mappings.[8] The top left of Figure 4(b) shows an example 2D MMS model of a triangular probability of detection region for a camera-based target detector mounted to a robot agent facing east. This particular MMS likelihood model has the form

$$p(\mathbf{z}_k^{r,j} = c|\mathbf{x}_i) = \frac{\sum_{s\in\sigma(c)} \exp(w_s^T \xi)}{\sum_{r\in\{\sigma(D)\cup\sigma(ND)\}} \exp\left(w_r^T \xi\right)}, \tag{22}$$

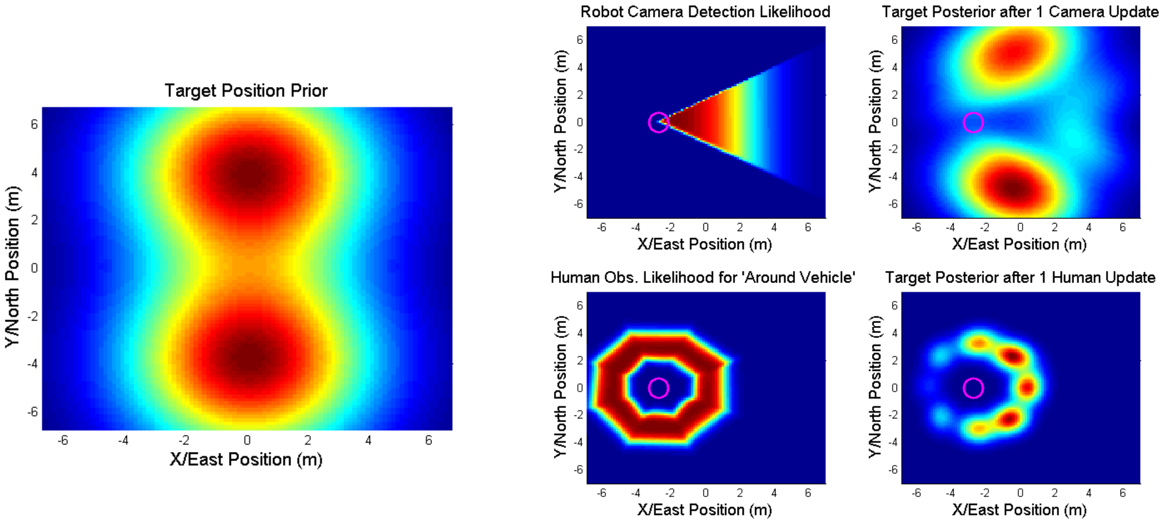
where $\xi = [\mathbf{x}_i, 1]^T$, $c \in \{$“no detection” (ND), “detection” (D)$\}$ is the observed category of $\mathbf{z}_k^{r,j}$. The weights w_s in (22) are parameters for the two mutually exclusive subcategories, $\sigma(D)$ and $\sigma(ND)$, that geometrically define the possible observation categories of $\mathbf{z}_k^{r,j}$ as a function of ξ, where $s \in \sigma(ND)$ or $s \in \sigma(D)$. The camera detection model in Figure 4(b) uses 3 subcategories in $\sigma(ND)$ to describe $\mathbf{z}_k^{r,j}$ = “no detection” as being most probable outside of the triangle, and 1 subcategory in $\sigma(D)$ to describe $\mathbf{z}_k^{r,j}$ = “detection” as being most probable inside the triangle. The bottom left of Figure 4(b) shows an example 2D MMS likelihood model corresponding to a coarse human range observation relative to a robot vehicle, with 3 possible categorical values $\mathbf{z}_k^{h,j} \in \{$“next to”, “around”, “far from”$\}$. The form of the likelihood function $p(\mathbf{z}_k^{h,j} = c|\mathbf{x}_i)$ for this ternary model is similar to that of the binary camera model in (22); here $\sigma($“next to”$)$ contains 1 subcategory (defining the hole of the ring), $\sigma($“around”$)$ contains 8 subcategories (each defining a segment of the octagon ring), and $\sigma($“far from”$)$ contains 8 subcategories (each defining a convex region extending from an outer face of the ring), for a total of 17 weights[e].

Despite their flexibility, the highly nonlinear/non-Gaussian nature of MMS likelihood models means that the exact posteriors on the left-hand sides of Equations (20) and (21) are unfortunately no longer closed-form, since the required integrals for the normalizing constants $K_k^{r,j}$ and $K_k^{h,j}$ cannot be determined analytically. In fact, the resulting hybrid Bayesian updates in (20) and (21) can only be performed via approximation methods such as discretization or Monte Carlo sampling.[28]

2. Hybrid Bayesian Gaussian Mixture Updates via Monte Carlo Importance Sampling

As discussed in [8], fast Monte Carlo importance sampling techniques can be used to obtain accurate GM approximations for the required posteriors on the left-hand sides of Equations (20) and (21) when the right-hand sides contain GM priors and MMS likelihoods. This leads to a recursive approximate Bayesian fusion strategy in which the priors and posteriors in (20) and (21) are always represented as GMs, thus ensuring that human and robot agents can incorporate new information from each other in a consistent and compact form.

[e] See [8] for further details on MMS models.



(a) (b)

Figure 4. (a) Bimodal GM prior with $\mu_1 = [0, 4]^T$, $\mu_2 = [0, -4]^T$, $\Sigma_1 = \Sigma_2 = 5 \cdot \mathbf{I}_2$, and $w_1 = w_2 = 0.5$, (b) *left (top and bottom)*: MMS likelihood models for camera detection probability and human range observation, where probabilities are close to 1 for red and close to 0 for blue; *right (top and bottom)*: corresponding Gaussian mixture posterior approximations for GM prior in (a) (robot vehicle position indicated by magenta circle).

The basic importance sampling algorithm used here to evaluate Equations (20) and (21) for non-dynamic GM priors and MMS likelihoods is given in Algorithm 3. Importance sampling approximates expectations under an intractable posterior distribution by using weighted samples drawn from a known "importance" distribution $q(\mathbf{x}_i)$, which ideally has a similar shape to the posterior; this idea underlies the well-known particle filter, which represents the priors and posteriors for non-Gaussian Bayesian filtering via weighted samples at each time step.[29] Algorithm 3 extends this representation by treating the input prior as a full GM and compressing the weighted particles from importance sampling measurement updates into a new GM posterior model. This not only provides a consistent and compact uncertainty representation for task and path planning for all time steps, but also helps avert sample degeneracy problems that can lead to inaccurate/unreliable Bayesian fusion.[29]

Since the importance distribution q should be as close as possible to the true posterior for efficient sampling,[29] Algorithm 3 also tailors q to the measurement updates in (20) and (21). For updates via Equation (20), q is set to the m^{th} component of the input GM, since its components are typically very close to the true posterior modes when "detection/no detection" observations arrive from robot vehicles at a sufficiently high rate. For updates via Equation (21), a variational Bayes algorithm is used to determine a q close to the true posterior, which can shift far away from the modes of the input GM when human agents make "surprising" observations (in such cases, severe sample degeneracy could result if q were instead set to the m^{th} component of the input GM). A more detailed description of Algorithm 3, including further details on the selection of q and evaluation of the importance weights ω^n in step 2, can be found in [8].

Figure 4 illustrates some typical fusion results using Algorithm 3 . Figure 4(a) shows the 2 component GM prior used in this demonstration with the MMS observation likelihoods shown in Figure 4(b). The top right of Figure 4(b) shows the 6 component GM posterior following an update with a robot vehicle "no detection" observation, which pushes probability mass away from the triangular detection region of the robot's camera. The bottom right of Figure 4(b) shows an 18 component GM posterior following the human observation "target around robot", where almost all of the probability mass from the GM in Figure 4(a) is forced into the ring-shaped region of the corresponding MMS likelihood model.

3. Practicalities

Mixture management: If the input GM of Algorithm 3 has M_i components and $p(\mathbf{z}_k^{t,j}|\mathbf{x}_i)$ has S_z relevant subcategories corresponding to $\mathbf{z}_k^{t,j}$, then the output GM will have $M_i \cdot S_z$ components. Thus, GM

Algorithm 3 Importance Sampling Measurement Update Algorithm

```
Inputs: Agent type $t \in \{r, h\}$, agent index $j \in \{1, ..., N_k^t\}$, input GM $p(\mathbf{x}_i|\mathcal{Z}_{k-1}, ..., \mathbf{z}_k^{t,j-1}) = \sum_{m=1}^{M_i} w_{i,m}\,\mathcal{N}(\mu_{i,m}, \Sigma_{i,m})$,
MMS observation likelihood $p(\mathbf{z}_k^{t,j}|\mathbf{x}_i)$, observation $\mathbf{z}_k^{t,j}$ with $S_z$ subcategories in $p(\mathbf{z}_k^{t,j}|\mathbf{x}_i)$
Output: Posterior GM $p(\mathbf{x}_i|\mathcal{Z}_{k-1}, ..., \mathbf{z}_k^{t,j}) = \sum_{h=1}^{S_z \cdot M_i} w_{h,i}\,\mathcal{N}(\mu_{h,i}, \Sigma_{h,i})$
Initialize h ← 0
for Input GM component $m = 1$ to $M_i$ do
  for Each subcategory $s \in \sigma(z_k^{t,j})$ do
    Set h ← h + 1
    if $t = r$ for updating via (20) then
      Set $q(\mathbf{x}_i) = \mathcal{N}(\mu_{i,m}, \Sigma_{i,m})$
    else if $t = h$ for updating via (21) then
      Set $q(\mathbf{x}_i)$ using variational Bayes method (see [8])
    end if
    1. Draw $N$ samples $\{\mathbf{x}_i^1, ..., \mathbf{x}_i^N\}$ from $q(\mathbf{x}_i)$
    2. Evaluate importance weights $\{\omega^1, ..., \omega^N\}$ (see [8])
    3. Estimate the conditional measurement likelihood $l(m, s) = \frac{1}{N}\sum_{n=1}^{N} \omega^n$
    4. Re-normalize importance weights so that $\sum_{n=1}^{N} \omega^n = 1$
    5. Estimate posterior component $h$ mixing weight $w_{h,i}$, mean $\mu_{h,i}$ and covariance $\Sigma_{h,i}$,

       $$w_{h,i} = w_{i,m} \cdot l(m, s), \quad \mu_{h,i} = \sum_{n=1}^{N} \omega^n \mathbf{x}_i^n, \quad \Sigma_{h,i} = \sum_{n=1}^{N} \omega^n (\mathbf{x}_i^n \mathbf{x}_i^{nT}) - \mu_{h,i}\mu_{h,i}^T \tag{23}$$

  end for
end for
Re-normalize mixing weights so that $\sum_{h=1}^{S_z \cdot M_i} w_{h,i} = 1$
```

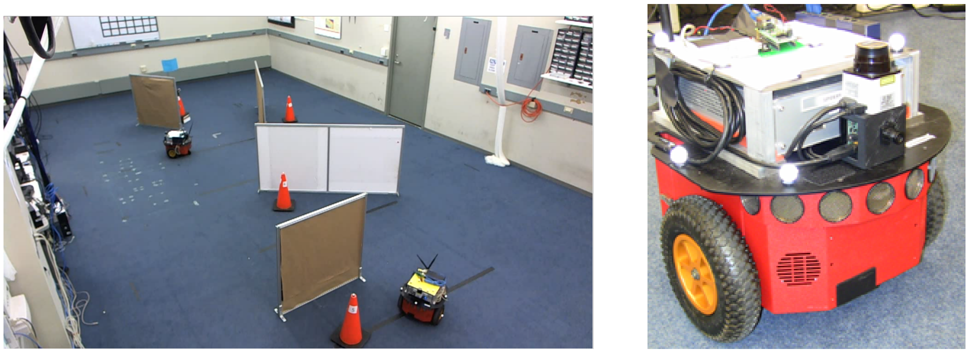
compression techniques should be used after each measurement update to prevent the number of GM terms in Equations (20) and (21) from becoming prohibitively large after each time step. As discussed in [8], many GM compression algorithms can be used, although all incur some information loss with respect to the original output GM. Salmond's merging algorithm[30] is used here after each application of Algorithm 3, so that the compressed output GM contains no more than M_{max} mixands that preserve the overall mean and covariance of the uncompressed output GM. While the best value of M_{max} is strongly problem/implementation-dependent, M_{max} must in general be tuned to balance between (i) minimizing the time costs of evaluating and compressing (20) and (21) for each target (so that agents can replan in a timely manner) and (ii) maintaining the most accurate possible GM approximation of (5) for each target (so that agents can use as much information as possible for replanning). To this end, it should also be noted that the fusion updates for multiple independent target GMs can be run in parallel, while the nested **for** loops in Algorithm 3 can be split into $M_i \cdot S_z$ parallel importance sampling updates.

False alarms and data association for ambiguous human observations: It is assumed here that human agents perfectly filter out false target detections from $\mathbf{Z}_k^r$, and that the soft location information in Z_k^h is completely reliable. While it is theoretically possible to extend the proposed fusion process to accommodate false alarms in $\mathbf{Z}_k^r$ and human errors/uncertainties in $\mathbf{Z}_k^h$, these extensions are omitted here for brevity.

Data association issues arise in Equation (21) when human observations $\mathbf{z}_k^{h,j}$ are not target specific (e.g. if a human reports the location of "something" or "nothing" in the absence of target ID information). For example, the "positive" observation $\mathbf{z}_k^{h,j}$ ="Something is nearby the robot" could apply to any remaining target, but only truly corresponds to one target. However, the "negative" observation $\mathbf{z}_k^{h,j}$ ="Nothing is nearby the tree" corresponds to all remaining targets. The fusion of such ambiguous measurements can be handled by probabilistic data association techniques.[31] The naive data association method of [8] is used here for the typical case where human observations only describe either "something" or "nothing" without target ID information.

IV. Indoor Target Search and Track Experiments

This section describes the experiments conducted at Cornell's Autonomous Systems Lab to validate the proposed planning and fusion methods for an actual cooperative human-robot target search and identification mission. Five static targets (orange traffic cones with numbered labels) were hidden in an obstacle-filled en-



(a) Search area used for experiment

(b) Pioneer 3DX rover equipped with computer, LIDAR, and camera

Figure 5. Real-time search and track experiments for human-robot teams performed at Cornell's Autonomous Systems Lab

vironment and assigned random state priors. A human-robot team consisting of two completely autonomous ground rovers and a human analyst were tasked with detecting, identifying, and localizing all $N = 5$ targets in under 10 minutes. The 5m x 10.5m indoor search area used for the experiment is shown in Figure 5(a).

Due to the relatively small scale of the problem, in these experiments, only the robotic agents were involved in the "Task Allocation" and "Path Planning" process described in Figure 1; the human operator was instead considered a stationary sensor whose sole task was to provide information (i.e. target verification and possibly soft location information) through a desktop computer console to the "State Estimation and Sensor Fusion" block. The targets $i \in \{1, \ldots, 5\}$ were placed at fixed locations $\mathbf{x}_i^{true}$ throughout a field featuring four movable wall obstacles measuring between 1m and 1.5m, which obstructed the human operator's direct line-of-sight to some of the targets when seated at the console. The operator had access to the live video streams from the vehicle cameras, displayed at the console, to assist in the classification process. The operator could also send information to the robots via the Human-Robot Interface (HRI) to improve their autonomous search.

A. Experimental Setup

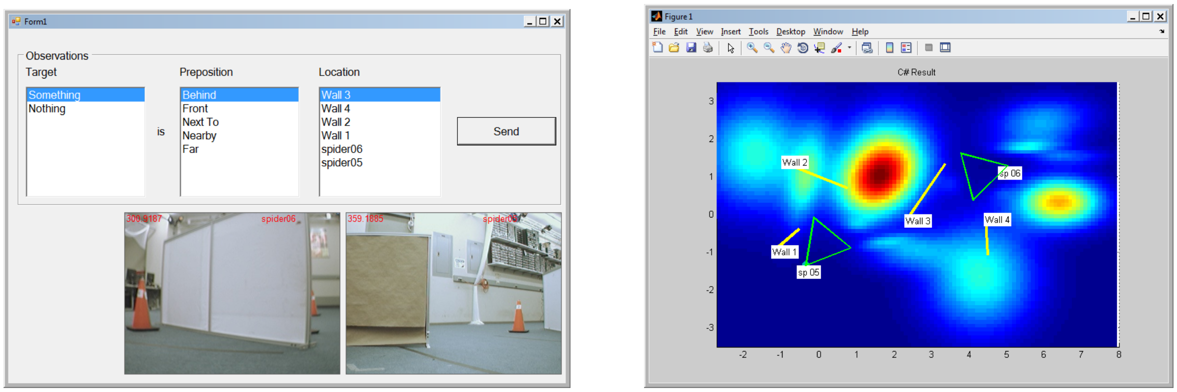
1. *Hardware*

The robot platform used for these experiments was the Pioneer 3DX with the following mounted devices: Mini ATX based computer with a 2.00 GHz Intel$^{\circledR}$ Core^{TM} 2 processor, 2 GB of RAM and WiFi networking to control the robot, a Unibrain Fire-I OEM Board camera, and a Hokuyo URG-04 LX LIDAR sensor (Figure 5(b)). A computer with a 2.66 GHz Intel$^{\circledR}$ Core^{TM} 2 Duo processor and 2 GB of RAM performed the State Estimation and Human-Robot Interface (HRI) tasks. An additional computer with similar hardware executed task allocation in a simulated decentralized environment. A computer with two 2.66 GHz Intel$^{\circledR}$ Xeon$^{\circledR}$ processors and 4 GB of RAM implemented high-level path planning for the experimental trials. A Vicon motion-tracking system performed robot localization and pose calculations.

2. *Messaging, Local Controller, and Target Detection Algorithms*

The Pioneers sent three types of messages: images, detected targets, and robot data. Each robot data message was sent at 20 Hz and contained the robot's name, ID number, pose, local timestamp, most recent LIDAR scan, current path, current trajectories, and path completion status. This data was processed for high-level task and path planning, as well as low-level local robot control and target detection.

Local Pioneer robot controllers enabled the vehicles to autonomously follow the waypoint trajectories created by the path planners while avoiding dynamic collisions. The Pioneer controllers were provided with a map of the search environment in order to plan around known obstacles as well as state data for any dynamic obstacles. Local path following used a D* planning algorithm to find an optimal path and a pure pursuit controller to generate velocity commands. The D* planner was required to avoid newly detected



(a) Main operator user interface window, with menus for soft inputs and windows for camera feeds

(b) Overhead field map with the combined target GM PDF, walls, and robot locations

Figure 6. Screenshots from the HRI console available to the human operator

obstacles and other robots in the field, as well as to provide a safety measure against possible inconsistencies between Vicon localization and actual field positions. Objects detected by the Pioneer LIDAR units were considered collision hazards if they were within a 0.4m threshold, with a bearing between 0 and π radians (in front of the vehicle). When a collision was imminent, the robots searched their latest LIDAR scan returns for possible escape directions that avoided these potential collisions. In some cases, a robot could get 'stuck' inside obstacles if they strayed too close to them, which necessitated human intervention to 'rescue' the robot.

The positions of other robots were unknown to the local path planners in this experimental setup, and the local robot controllers were instead responsible for avoiding robot-robot collisions[f]. The state data for each robot was broadcast at 2 Hz so that each local D* planner could model the other robot as a dynamic obstacle. A simple fixed precedence algorithm was used to resolve potential robot-robot collisions if the robots came within a fixed distance of each other. The trajectories sent from the vehicle computer to the Pioneer controller were varied between rates of 4 sec and 10 sec; the effects of these two different rates are examined below.

To detect and localize potential targets, the Pioneer computer used OpenCV "blob" detection algorithms on areas of the camera images that were within thresholds corresponding to the color of a standard traffic cone under the lab's lighting conditions. A cone detection algorithm developed in OpenCV was then employed to provide the HRI with a bounding box of possible cone-shaped blobs in a "potential target detection" message. To simulate realistic mission conditions given the small size of the field, the HRI restricted detection ability to a forward range of 1m by ignoring potential targets with bounding boxes of insufficient height. The HRI associated potentially valid targets with the generated image, and with the LIDAR scan with the closest timestamp to that image. The location of the possible target was then roughly estimated using the LIDAR points with the best match in bearing to the bounding box. If the estimated location was not for an already identified target or a false alarm, this information was prompted to the user, who either identified the specified target or classified it as a false alarm.

3. Human-Robot Interface

In addition to assisting in target detection, the HRI, shown in Figure 6, allowed the human operators to send soft sensor information to update the Gaussian mixture model for each target, as described in Section III.C. The observations from the human agent involved positive/negative information using the previously described "something/nothing" format and the preposition models shown in Section III.C to describe target location information relative to field landmarks or the robots themselves via a scroll menu. Using the HRI, the operator also had available the camera views of each Pioneer and a top-down map view of the field, which included all known obstacles, robot locations, identified target/false alarm locations, and the latest

[f] Note that IRRT has the capability to perform dynamic obstacle/agent avoidance if augmented with the appropriate state data for those dynamic obstacles/agents. This capability was not leveraged in these experiments, however, in the next set of trials this feature will be enabled.

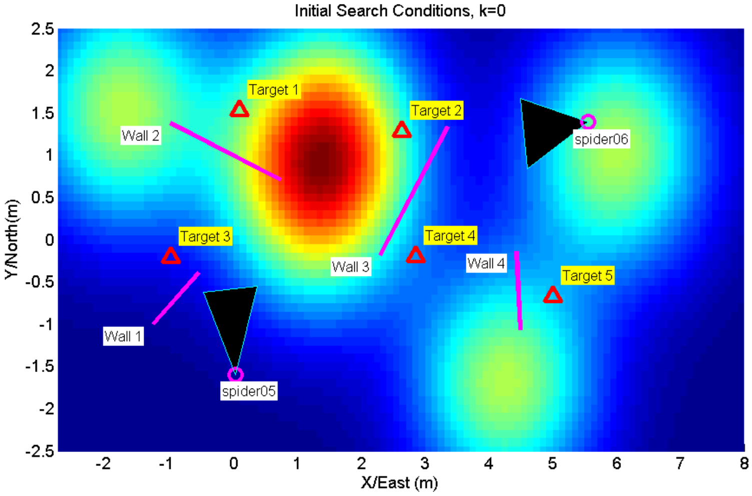


Figure 7. Field map showing walls (magenta lines), true target locations (red triangles), initial target prior for combined target GM, and initial vehicle locations (circles) with camera detection field of view (black triangles).

combined target PDF. Note that the vehicles were fully autonomous and that the HRI could not directly send any control signals to the robots (except for an emergency stop command). Any operator inputs to the robots were via "soft" observations only, as described in Section III.C.

B. Search Performance Metrics and Results

Each target $i \in \{1, \dots, 5\}$ was assigned a Gaussian prior $p(\mathbf{x}_i) = \mathcal{N}(\mu_i, \Sigma_i)$ with covariance $\Sigma_i = I$ and mean μ_i. The values for μ_i and $\mathbf{x}_i^{true}$ that were used for the experiments were

$$\begin{aligned}
\mu_1 &= [1.50, 1.30]^T, & \mathbf{x}_1^{true} &= [0.103, 1.526]^T\\
\mu_2 &= [4.25, -1.70]^T, & \mathbf{x}_2^{true} &= [2.648, 1.28]^T\\
\mu_3 &= [1.25, 0.55]^T, & \mathbf{x}_3^{true} &= [-0.973, -0.214]^T\\
\mu_4 &= [-1.75, 1.55]^T, & \mathbf{x}_4^{true} &= [2.867, -0.201]^T\\
\mu_5 &= [6.00, 1.05]^T, & \mathbf{x}_5^{true} &= [5.012, -0.679]^T.
\end{aligned}$$

The PDF surface for the combined target prior is shown in Figure 7, along with the 2D field map, initial search vehicle locations and orientations, and the true target locations and labels. Note that in some cases, $\mathbf{x}_i^{true}$ is sometimes quite far from the specified μ_i prior, which creates a fairly challenging search problem.

Multiple search trials were conducted to compare team search performance using several metrics under different planning and information fusion modalities. The experiments included trials with and without human operator soft inputs and varied task allocation replan rates. Planner performance was evaluated via the following metrics: (1) number of targets detected/identified, (2) individual and combined vehicle distances traveled (not including rotations in place), (3) time required to find the targets, and (4) information acquired throughout the mission. The latter metrics reflect the efficiency of the combined decentralized planning algorithms in exploring the search space[g]. The information gain at each time-step, following updates from the GM target state estimator, was computed via the Kullback-Leibler divergence (KLD) between the updated combined (undetected) target GM posterior in Equation (24) at time-step $k+1$ and the combined

[g] Note that mission duration could be less than the allotted 10 minutes either because: all targets were found and identified successfully, or because the trajectory planner or robot controller became "stuck" in an irrecoverable state (e.g. inside an obstacle buffer), in which case the mission was terminated since the robots could not continue autonomously without significant human intervention. Hence, the mission termination condition also served as a loose qualitative measure of planner performance, although it should be emphasized that this is sensitive to the particular tuning and implementations of the local robot controllers and the experimental environment.

target GM posterior from time-step k, given by

$$\mathrm{KL}\left[p(\bar{\mathbf{x}}|\mathcal{Z}_{k+1})||p(\bar{\mathbf{x}}|\mathcal{Z}_k)\right] = \int_{-\infty}^{\infty} p(\bar{\mathbf{x}}|\mathcal{Z}_{k+1}) \ln\left[\frac{p(\bar{\mathbf{x}}|\mathcal{Z}_{k+1})}{p(\bar{\mathbf{x}}|\mathcal{Z}_k)}\right] d\bar{\mathbf{x}}. \tag{24}$$

As discussed in [9], the KLD can be thought of as a distance measure between two probability distributions, which quantifies the amount of information acquired about the combined target estimate $\bar{\mathbf{x}}$ from time k to $k+1$. KLDs are nonnegative and are zero only if both PDFs in the integrand are identical, which implies that no new information is acquired. The sum of the KLD over the mission duration reflects the overall average changes in the uncertainty of the undetected target locations during the search mission. The cumulative sum of Equation (24) from mission start to end was computed for each trial offline via discretization techniques.

1. *Effect of Human Operator Soft Inputs*

To quantify the benefit of the human operator soft inputs to the Bayesian fusion performance, trials were conducted to compared the proposed GM target state estimator which fused soft human information to a baseline GM target state estimator that ignored soft human information. A basic greedy Markov Decision Process (GMDP) path planner served to generate default trajectories. For each robot, the GMDP discretized the combined target PDF for the robot's assigned targets, and then selected the cell with the largest PDF share as the goal location. A path to that location was then planned using value iteration,[32] where the robot was allowed to move into adjacent cells by moving either left, right, up, down, or diagonally. For the trials where the soft human observations were ignored, only the probability of no detection information from the robot camera and pose information were used to update the target GMs. In such cases, the human agent was instructed not to send any observations to the robots, other than responding to target confirmations. For the trials where the human's observations were fused, the human could send soft information observations to the robots at will, as long as the robots were not awaiting target confirmation responses. In all cases, the maximum number of GM components per target was set to 15. The trials were repeated using a 4 sec replan rate for the CBBA task allocation component and a 10 sec replan rate. Table 1 shows the detected targets along with the time it took the team to find them in each of the trials.

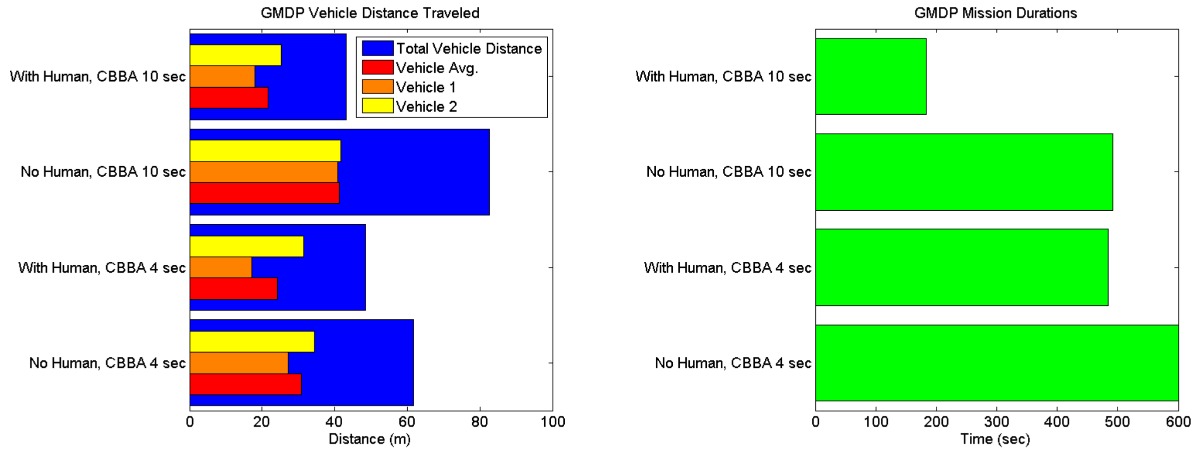
Table 1. Results for Human Operator Soft Input Experiments: Targets Detected and Time to Detect Targets (in order of acquisition)

Case	Targets, no Human	Time (s), no Human	Targets, with Human	Time (s), with Human
4 sec CBBA	2,5,4,1	98,246,496,543	4,3,2,1,5	25,199,241,286,336
10 sec CBBA	1,4,5,3	65,209,262,427	2,4,3,1,5	49,60,79,347,365

Figure 8 compares the mission performance for the different trials, showing the vehicle distances traveled (Figure 8(a)) and the mission durations (Figure 8(b)) for each case. These results highlight the benefit of the human operator soft inputs, showing that by fusing in the information provided by the operator, the robotic team is able to locate and identify the targets more quickly and efficiently than without the operator inputs.

2. *Information-Based Search and Track*

Next, the IRRT trajectory planning approach described in Section III.B was implemented and used to generate information-based paths for the robotic agents. Multiple trials were conducted for different task allocation replan rates (4 sec vs. 10 sec), and for cases with and without the human operator soft inputs described above. Table 2 presents the results for these trials, showing the targets acquired throughout the mission along with the time it took the team to located and identify them. Figure 9 shows the mission durations and vehicle distances traveled for the different trials, and Figure 10 shows the information acquired by the team throughout the mission for the different experiments. There are some interesting observations that can be made from these results. Firstly, as shown in Figure 10, the trials incorporating human soft inputs achieved a higher information content than the trials without human inputs for both the 10 sec and



(a) Individual and combined vehicle distances traveled during each trial

(b) Total mission duration for each trial

Figure 8. Results for Human Operator Soft Input Experiments: Comparison of mission durations and distances traveled with and without human operator soft inputs.

Table 2. IRRT Targets Detected and Time to Detect Targets (in order of acquisition)

Case	Targets, no Human	Time (s), no Human	Targets, with Human	Time (s), with Human
4 sec CBBA	5,3	148,287	3,4,1,5	77,176,178,282
10 sec CBBA	5,4,2,1	27,85,121,339	4,2,1,5	30,144,222,421

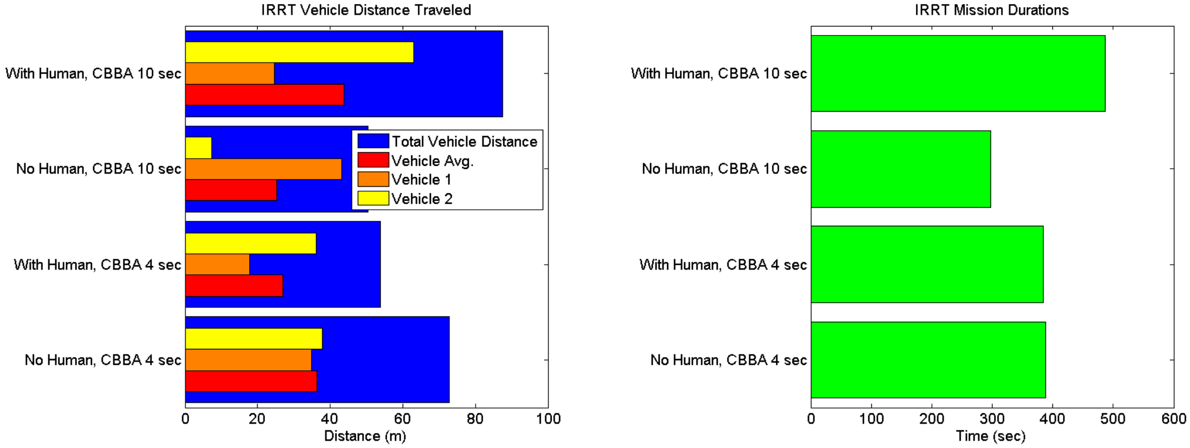
4 sec replan cases. In fact, in the 4 sec replan case the autonomous team found only 2 out of the 5 targets, but using human inputs it was able to find 4 (see Table 2). A second observation is that the information acquired using a 10 sec replan rate was consistently higher than that obtained using a 4 sec replan rate (for both with and without human inputs). This is due to a tradeoff between replanning often to include the latest target estimate information vs. allowing IRRT enough time to generate a quality plan before using it. Finally, it is worth noting that the impact of human inputs and replan rates on vehicle distances and total mission completion times was inconclusive for this set of experiments. This is partly due to the fact that vehicle distances and mission times were affected by the collision avoidance software that would sometimes stop or reroute vehicles before returning them to the trajectories planned by IRRT. The next section provides a more detailed discussion on the different elements that impacted the results, especially with regards to mission times and vehicle distances.

C. Discussion

The hardware implementation results for the multi-target search and identification mission provide several interesting and useful insights about the proposed information-rich planning and hybrid fusion framework for human-robot teams. This section describes the benefits of the proposed architecture as well as lessons learned and suggestions for future work.

1. Effects of soft human inputs on Bayesian fusion performance

The performance metrics above show that the fusion of human information generally improved target search efficiency; in particular, the rates of average information gain with respect to the undetected target PDF without human inputs were smaller than the rates of average information gain with human input. This shows that although human agents are rate limited and less precise than conventional robot sensors, proper Bayesian fusion of soft categorical human data can lead to significant improvements in team performance



(a) Individual and combined vehicle distances traveled during each trial

(b) Total mission duration for each trial

Figure 9. Results for Information-Based Search and Track Experiments: Comparison of mission durations and distances traveled

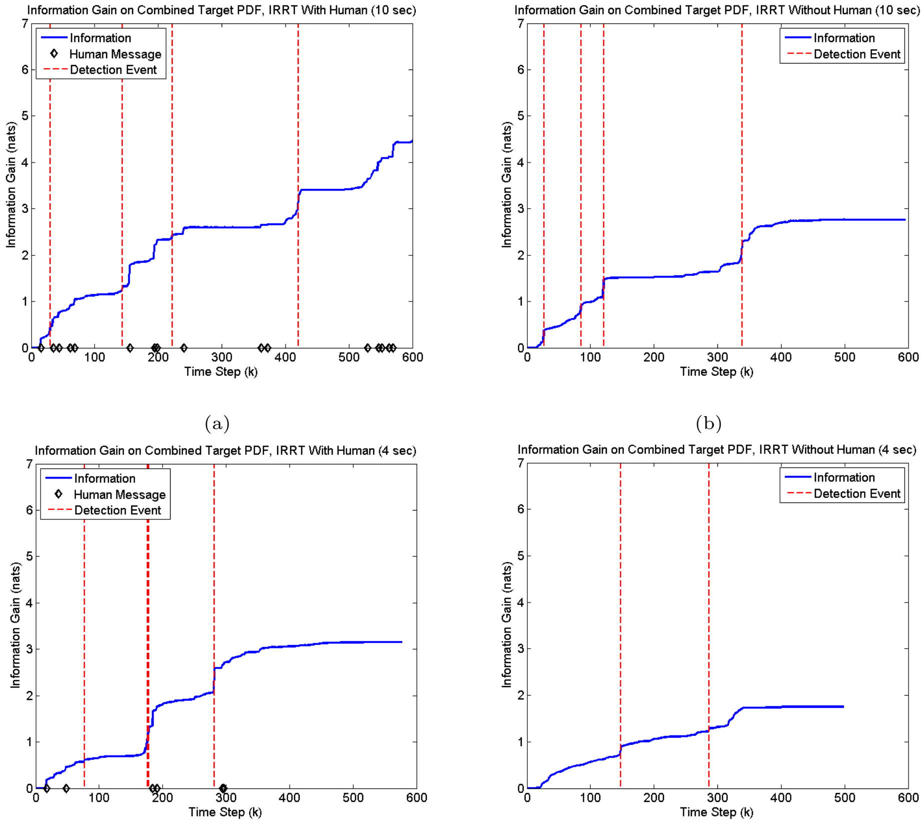
for information-based tasks. Soft categorical human inputs are especially interesting to consider for data fusion in ISR-type missions, due to their natural interpretability to human operators and their high degree of flexibility. The GM-based data fusion framework presented here for soft human inputs can also readily accommodate existing Bayesian fusion methods for conventional sensors (e.g. such as those based on Kalman filtering).

However, care must always be taken in practice to properly characterize the context of soft human inputs before fusion takes place. For example, the meaning (and hence the likelihood functions) of 'nearby' is quite different in the statements 'the car is nearby the tree' and 'Newark is nearby New York City'. In the experiments presented here, this issue is resolved through the use of a simple messaging protocol and a limited but unambiguous set of contextual cues (i.e. the 'wall' and 'robot' location reference points) that can be passed to an interpreter, which is constructed offline. More sophisticated messaging protocols or interfaces (e.g. natural language) could also be implemented with additional effort for other applications, as long as sufficient training data is available for properly translating the desired soft categories into meaningful probabilistic likelihood functions, as described in Section III.

Interestingly, while human sensor inputs are clearly advantageous from an information fusion perspective, they remain challenging to directly accommodate and account for within information-based planning algorithms, since humans are not as predictable or reliable as conventional sensors such as cameras and LIDAR. In particular, the highly intermittent and nonlinear/non-Gaussian nature of soft categorical human inputs makes it difficult to predict the expected amount of information to be contributed by human sensor agents over a finite horizon. As a result, information-based planners must be tuned carefully to the styles of different human operators in order to ensure good team performance. Future work will explore different strategies for coping with these issues, such as explicitly polling the human operator for inputs to improve predictability and calculation of information gain bounds with respect to intermittent human observations.

2. Analysis of information-based search and track

While the experiments using the proposed IRRT-CBBA fusion architecture illustrated the viability of a unified information-based planning framework, there were several interesting issues observed and lessons learned. Firstly, the experiments using a 4 sec task allocation replan rate vs. a 10 sec replan rate highlighted a tradeoff between plan relevance and plan quality. The quality of the plans generated by IRRT improves the longer the algorithm is allowed to run, thus the 10 sec replan rate created more informative trajectories which enabled the team to acquire larger amount of information throughout the mission (see Figure 10). On the other hand, a faster replan rate ensures that the current plan remains relevant in light of changing information, such as updated GM estimates and critical information provided by human operators. For this particular mission, a 10 sec task allocation replan rate proved more efficient than a 4 sec rate, but in a more



(a)

(b)

(c)

(d)

Figure 10. Results for Information-Based Search and Track Experiments: KLD information gain for the combined undetected target PDF. Plots show information acquired for (a) 10 sec CBBA replan with human inputs, (b) 10 sec CBBA replan without human inputs, (c) 4 sec CBBA replan with human inputs, and (d) 4 sec CBBA replan without human inputs. Red lines indicate target detection/ID events and black diamonds denote instances where a soft human observation message is fused.

dynamic environment it is likely that the relevance of the plan will become more important than the quality, thus favoring faster replan rates. This tradeoff is problem and implementation dependent, and should be carefully considered when implementing these system to achieve the best performance.

Secondly, the complications associated with performing actual hardware experiments impacted the performance of the system resulting in discrepancies between expected and actual performance. For example, the information-based trajectories planned by the IRRT component were not being executed exactly by the rovers during the experiments for a variety of reasons such as collision avoidance, delays, and modeling inaccuracies. The low-level dynamic collision avoidance software often changed the shape of the trajectory, creating detours so that the vehicles would not hit each other. These detours often resulted in larger vehicle distances and longer mission completion times. Although the IRRT algorithm presented in [11] accounts for dynamic collision avoidance, the distributed hardware nature of the experiment did not allow us to take advantage of this feature, however, this capability will be included in future iterations of these experiments. The other reason mentioned above involved delays between planning the trajectory and communicating it to the vehicles. Although minor, these delays impacted the trajectory following capability and the next iteration of experiments will attempt to minimize these delays as much as possible. The third reason was due to mismatches between the IRRT's model of the vehicle dynamics and sensor model, and those actually used by the rovers during execution (such as many of the D* effects). IRRT specifies exact trajectories that include position, orientation and sensor location, but the low-level path following software consisted only of waypoint following, causing discrepancies between the planned and actual sensor location/orientation required for obtaining accurate measurements (although the vehicles often got really close to taking a proper measurement, they would pass by without looking directly at the target, and therefore there would be no reward for executing the almost perfect trajectory). In addition, the actual measurement procedure was slightly different than that modeled in the IRRT framework, causing mismatches in predicted and actual performance. In particular, the vehicle sensor limitations in the vision processing software were more conservative than the sensor models used by IRRT. Thus, while the trajectories generally exhibited good behavior, the “reward” in terms of valid observations was often insufficient. Furthermore, there was significant sensitivity in the actual measurement procedure making it difficult to obtain a good measurement. For example, the target cones had to be centered at a certain height and distance for the vision processing software to accept them, and white labels used to mark the targets were interfering with the vision software’s classifiers, therefore, even when the vehicles were looking directly at the targets, a proper measurement was not received. These real-world considerations are hard to model within the IRRT framework, and future work should consider ways to incorporate robustness into both the planning and image processing components.

Lastly, it should also be noted that, out of all these trials, only the two GMDP trials with human input terminated in under 10 minutes because all targets were successfully detected and identified. All other trials ended either because the 10 minute time limit was reached (GMDP, no human, 4 sec CBBA replan rate) or because the robots determined that they were irreversibly “trapped” by obstacles (often by moving too close to previously detected targets or walls), thus requiring human intervention to break free (all other trials). To ensure some consistency for meaningful comparisons in light of these difficulties, these latter trials were only accepted after at least 5 minutes of the search had elapsed, otherwise the trial was discarded and restarted. As mentioned before, early termination due to motion infeasibility was usually caused by discrepancies in the low level controllers, Vicon state estimation, and/or environmental setup, which are issues that will be addressed in the next iteration of trials. In spite of all the real-world considerations that arise when dealing with actual hardware, note that at least 4 out of the 5 targets were successfully detected and identified in all but one of the eight total trials, illustrating the potential benefits of this information-based planning and fusion architecture for search and track missions.

V. Conclusions and Ongoing Work

Motivated by the need for scalable and flexible information-based planning algorithms and robust fusion methods to exploit heterogeneous information sources in large-scale semi-autonomous systems, this paper introduces a new planning and estimation framework for optimizing information collection in cooperative human-robot missions. To this end, a unified approach to high-level distributed task planning, information-based path planning, and hybrid Bayesian information fusion using Gaussian mixtures is presented and validated in a real hardware experiment involving multi-target search with a cooperative human robot team. The results illustrate the benefits of including information acquisition as a goal at every level of planning, as well as showing that by including human operator soft inputs into the Bayesian fusion framework the performance and efficiency of the autonomous search team can be greatly improved.

Future work will consider extending the proposed planning and fusion framework in several ways to accommodate other realistic estimation problems for cooperative human-robot search missions. These extensions include (but are not limited to): dynamic target tracking with continuous sensor information fusion and multiple model uncertainties; environmental map uncertainties to accommodate simultaneous localization and mapping (SLAM);[32] 3D target dynamics and sensor model updates (e.g. using UAV sensor platforms); decentralized fusion with Gaussian mixtures; and improved false alarm modeling and data association techniques for soft information fusion. Future work will also consider task-planning extensions for human sensor agents and for human-operated mobile agents, as well as task and path planning for mobile human agents. Since CBBA is well-suited to handling heterogeneous agents that can perform a wide variety of tasks, extensions to incorporate realistic secondary mission objectives such as refueling and automation failure handling (e.g. assignment of a human agent to tele-operate a ‘trapped’ robot to rescue it) will also be studied.

Acknowledgments

This research was supported in part by MURI FA9550-08-1-0356 and a National Science Foundation Graduate Research Fellowship.

References

[1] "FY2009-2034: Unmanned Systems Integrated Roadmap," Tech. rep., Office of the Secretary of Defense, USA, April 2009.

[2] "Technology Horizons: A Vision for Air Force Science and Technology During 2010-2030," Tech. rep., Office of the Chief Scientist of the U.S. Air Force (AF/ST), May 2010.

[3] Southern, D., *Human-Guided Management of Collaborating Unmanned Vehicles in Degraded Communication Environments*, Master's thesis, MIT Department of Electrical Engineering and Computer Science, June 2010.

[4] Bishop, C., *Pattern Recognition and Machine Learning*, Springer, New York, 2006.

[5] Kotecha, J. and Djuric, P., "Gaussian Sum Particle Filtering," *IEEE Trans. on Sig. Proc.*, Vol. 51, No. 10, 2003, pp. 2602–2612.

[6] Lewis, M., Wang, H., Velgapudi, P., Scerri, P., and Sycara, K., "Using humans as sensors in robotic search," *FUSION 2009*, pp. 1249–1256.

[7] Kaupp, T., Douillard, B., Ramos, F., Makarenko, A., and Upcroft, B., "Shared Environment Representation for a Human-Robot Team Performing Information Fusion," *Journal of Field Robotics*, Vol. 24, No. 11, 2007, pp. 911–942.

[8] Ahmed, N., Sample, E., Ho, K., Hoossainy, T., and Campbell, M., "Categorical Soft Data Fusion via Variational Bayesian Importance Sampling, with Applications to Cooperative Search," *to appear in the 2011 American Control Conference (ACC 2011)*.

[9] Bourgault, F., *Decentralized Control in a Bayesian World*, Ph.D. thesis, University of Sydney, 2005.

[10] Ferrari, S., Foderaro, G., and Tremblay, A., "A Probability Density Function Approach to Distributed Sensors' Path Planning," *Proc. of the 2010 Int'l Conf. on Robotics and Automation (ICRA 2010)*, Anchorage, Alaska, pp. 432–439.

[11] Levine, D., *Information-rich Path Planning under General Constraints using Rapidly-exploring Random Trees*, Master's thesis, Massachusetts Institute of Technology, 2010.

[12] Upcroft, B., Ong, L., Kumar, S., Ridley, M., and Bailey, T., "Rich Probabilistic Representations for Bearing-Only Decentralised Data Fusion," *FUSION 2005*, 2005.

[13] Choi, H.-L., Brunet, L., and How, J. P., "Consensus-Based Decentralized Auctions for Robust Task Allocation," *IEEE Trans. on Robotics*, Vol. 25 (4), 2009, pp. 912 – 926.

[14] Levine, D., Luders, B., and How, J. P., "Information-rich Path Planning with Arbitrary Constraint using Rapidly-exploring Random Trees," *AIAA Infotech@Aerospace Conference*, Atlanta, GA, April 2010 (AIAA-2010-3360).

[15] Brunet, L., *Consensus-Based Auctions for Decentralized Task Assignments*, Master's thesis, Massachusetts Institute of Technology, 2008.

[16] Curtis, J. and Murphey, R., "Simultaneous Area Search and Task Assignment for a Team of Cooperative Agents," *Proceedings of AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2003.

[17] McLain, T. W. and Beard, R. W., "Coordination Variables, Coordination Functions, and Cooperative-Timing Missions," *Journal of Guidance, Control, and Dynamics*, Vol. 28(1), 2005, pp. 150–161.

[18] Ren, W., Beard, R. W., and Kingston, D. B., "Multi-agent Kalman consensus with relative uncertainty," *American Control Conference (ACC)*, 8-10 June 2005, pp. 1865–1870 vol. 3.

[19] Olfati-Saber, R., Fax, J., and Murray, R., "Consensus and Cooperation in Networked Multi-Agent Systems," *Proceedings of the IEEE*, Vol. 95, No. 1, Jan. 2007, pp. 215–233.

[20] Alighanbari, M. and How, J. P., "Decentralized Task Assignment for Unmanned Aerial Vehicles," *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference*, 2005.

[21] Bertuccelli, L., Choi, H., Cho, P., and How, J., "Real-time Multi-UAV Task Assignment in Dynamic and Uncertain Environments," *AIAA Guidance, Navigation, and Control Conference*, 2009.

[22] Ponda, S. S., *Trajectory Optimization for Target Localization Using Small Unmanned Aerial Vehicles*, Master's thesis, Massachusetts Institute of Technology, 2008.

[23] Fisher, R. A., "On the Mathematical Foundations of Theoretical Statistics," *Philosophical Transactions of the Royal Society of London, Series A: Mathematical and Physical Sciences*, Vol. 222, 1922, pp. 309–368.

[24] Luders, B., Karaman, S., Frazzoli, E., and How, J. P., "Bounds on Tracking Error using Closed-Loop Rapidly-Exploring Random Trees," *American Control Conference (ACC)*, Baltimore, MD, 2010, pp. 5406–5412.

[25] Rao, C. R., "Information and the accuracy attainable in the estimation of statistical parameters," *Bulletin of the Calcutta Mathematical Society*, Vol. 37, 1945, pp. 81–89.

[26] Arulampalam, S., Gordon, N., and Ristic, B., *Beyond the Kalman Filter: Particle Filters for Tracking Applications*, Artech House, Boston, MA, 2004.

[27] Julier, S., "An Empirical Study into the Use of Chernoff Information for Robust, Distributed Fusion of Gaussian Mixture Models," *FUSION 2006*, 2006, pp. 1–8.

[28] Lerner, U., *Hybrid Bayesian Networks for Reasoning About Complex Systems*, Ph.D. thesis, Stanford University, 2002.

[29] Arulumpalam, M., Maskell, S., Gordon, N., and Clapp, T., "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking," Vol. 50, No. 2, 2002, pp. 174–188.

[30] Salmond, D., "Mixture Reduction Algorithms for Uncertain Tracking," Tech. Rep. 88004, Farnborough, UK: Royal Aerospace Est., 1988.

[31] Kirubarajan, T. and Bar-Shalom, Y., "Probabilistic Data Association Techniques for Target Tracking in Clutter," *Proc. of the IEEE*, Vol. 92, No. 3, 2004, pp. 536 – 557.

[32] Thrun, S., Burgard, W., and Fox, D., *Probabilistic Robotics*, MIT Press, Cambridge, MA, 2005.