

# Efficient Planning under Uncertainty for a Target-Tracking Micro-Aerial Vehicle

Ruijie He, Abraham Bachrach and Nicholas Roy  
 Massachusetts Institute of Technology, Cambridge, MA, USA

**Abstract**—A helicopter agent has to plan trajectories to track multiple ground targets from the air. The agent has partial information of each target’s pose, and must reason about its uncertainty of the targets’ poses when planning subsequent actions.

We present an online, forward-search algorithm for planning under uncertainty by representing the agent’s belief of each target’s pose as a multi-modal Gaussian belief. We exploit this parametric belief representation to directly compute the distribution of posterior beliefs after actions are taken. This analytic computation not only enables us to plan in problems with continuous observation spaces, but also allows the agent to search deeper by considering policies composed of multi-step action sequences; deeper searches better enable the agent to keep the targets well-localized. We present experimental results in simulation, as well as demonstrate the algorithm on an actual quadrotor helicopter tracking multiple vehicles on a road network constructed indoors.

## I. INTRODUCTION

MAVs are increasingly used in military and civilian domains, ranging from intelligence, surveillance and reconnaissance operations, border patrol missions, as well as weather observation and disaster relief coordination efforts. In this paper, we present a target-tracking planning algorithm for a helicopter maintaining surveillance over multiple targets along road networks. Applications include an autonomous police helicopter tasked with monitoring the activity of several suspicious cars in urban environments.

Target-tracking is a sequential decision-making task that combines target-search – finding targets that are not initially visible, and target-following – maintaining visibility of the discovered targets. As the agent does not have perfect information of the targets’ poses and their subsequent actions, it has to reason about its belief of their poses when planning to keep them well-localized. Traditionally, although target-search algorithms [1], [2], [3] necessarily involve planning under uncertainty, target-following algorithms [4], [5], [6] typically focus on performing accurate belief updating and data association, rather than tackling the decision-making challenges faced by the agent. Especially when multiple targets have to be tracked by a single agent, the agent has the additional challenge of reasoning about which target to concentrate on at every timestep.

In this paper, we present the Multi-modal Posterior Belief Distribution (MMPBD) algorithm, an online, forward-search, planning-under-uncertainty algorithm for the road-constrained target-tracking problem. We have shown previously [7] that for uni-modal Gaussian representations, the sufficient statistics of how the agent’s belief are expected to evolve as actions are taken can be directly computed without

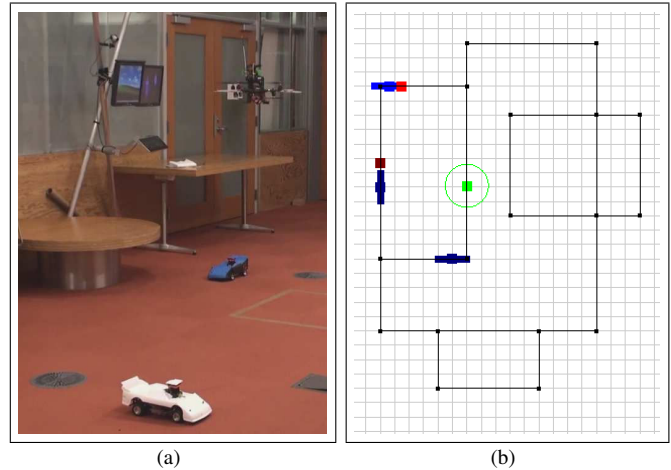


Fig. 1. (a) Our quadrotor helicopter tracking multiple ground vehicles. (b) Road-network target-tracking problem. Both the agent (green) and targets (red) are constrained to travel along the road network. The agent maintains a belief (blue/dark blue) of the targets’ poses .

having to enumerate the possible observations. This property allows the planner to search deeper by considering plans composed of multi-step action sequences, which we refer to as macro-actions.

We extend our previous algorithm by representing the agent’s belief of each target’s pose as a multi-modal Gaussian belief. For this belief representation, we show that we can similarly approximate the distribution of posterior beliefs at the end of a macro-action in an efficient manner, thereby enabling the planner to search deeper. Simulation results compare our algorithm to both the greedy strategy and a forward search strategy that does not incorporate macro-actions, and we also demonstrate our algorithm on an actual quadrotor helicopter (Figure 1a) tracking two autonomous ground vehicles from the air.

## II. PROBLEM FORMULATION

We first formulate the road-network target-tracking problem. A helicopter is tasked with having to search and maintain track of  $n$  targets ( $n \geq 2$ ) that are moving independently around a road network.

A map of the urban environment is known *a priori*, and we assume that the road network can be reduced to a graph with edges and nodes, representing roads and junctions respectively (Figure 1b). Both the agent (green square) and the targets (red square) are constrained to move along the graph, and can move in either direction along an edge (black lines). At a road junction (node), the agent and targets can

Email addresses: {ruijie, abachrac, nickroy}@mit.edu

move along any of the roads that meet at the junction. The road network makes it critical that the agent’s belief of each target is representable as a multi-modal belief (blue bars).

The agent is able to accurately localize itself in the environment, but does not have perfect information about the targets’ poses. Apart from the sensors used for self-localization, the agent has a limited range, downward-pointing camera sensor that obtains observations of a target’s pose if it is within view (green circle in Figure 1b). Given the height that the helicopter is flying at and the intrinsic parameters of the camera sensor, we can recover the effective range  $r_a$  of the camera sensor. The agent obtains a noisy observation of a target if it is less than  $r_a$  distance from the agent, and a *null* observation otherwise. The observation space is therefore continuous when the agent receives an observation, but also has a bi-modal characteristic due to the *null* observation.

We assume that the targets’ poses and dynamics are independent of the agent’s, i.e., the target is indifferent or unaware that it is being pursued. In addition, we assume that each target  $i$  moves with non-zero speed  $u_{t,i}$  around the environment, perturbed by Gaussian noise  $\epsilon_t \sim \mathcal{N}(0, R_t)$ . The agent has knowledge of the targets’ mean speed and start poses, and can travel faster than the targets. Even though the agent is aware of the targets’ start poses, this target-tracking problem still involves a target-search component because it can subsequently lose track of a target, especially when the modes split at a junction.

The agent does not know the true pose of each of the targets in the environment. Instead, it only has access to an estimate of target  $i$ ’s pose at time  $t$ , known as a belief  $b_{t,i}$ . The agent’s goal is to minimize both the uncertainty of the targets’ poses and the distance it needs to travel.

### III. BELIEF UPDATING FOR TARGET-TRACKING

Formally, the transition and observation dynamics of the agent  $x_t$  tracking a single target  $s_{t,i}$  along a road edge can be written as

$$s_{t,i} = A_{t,i}s_{t-1,i} + B_{t,i}u_t + \epsilon_t \quad \epsilon_t \sim \mathcal{N}(0, R_t) \quad (1)$$

$$z_{t,i} = \begin{cases} C_t s_{t,i} + \delta_t & \text{if } |x_t - s_{t,i}| < r_a \\ \text{null} & \text{otherwise} \end{cases} \quad (2)$$

where  $\delta_t \sim \mathcal{N}(0, Q_t)$  is the Gaussian noise associated with the observation when a target is within the field-of-view of the agent’s sensor. When the target gets to a node, it chooses a new edge with dynamics  $A'_{t,i}$ , where the probability of choosing each edge is  $\frac{1}{k-1}$  for a node of degree  $k$ .

At every timestep, the agent takes an action  $a_t$  and obtains a set of observations  $z_t$ . It can then update its belief according to

$$b_{t+1}(s') = \eta p(z_t|a_t, s') \int_{s \in \mathcal{S}} p(s'|s, a_t) b_t(s) ds \quad (3)$$

where  $\eta$  is a normalization constant.

In this paper, we represent the agent’s belief over each target’s pose as a multi-modal Gaussian distribution.

$$b_{t,i} \sim \sum_j w_{t,i,j} \mathcal{N}(\mu_{t,i,j}, \Sigma_{t,i,j}) \quad (4)$$

where  $\mu_{t,i,j}$  and  $\Sigma_{t,i,j}$  are the means and covariances of mode  $j$  of target  $i$  at time  $t$ , and  $w_{t,i,j}$  is the weight of each

---

### Algorithm 1 BELIEFUPDATE() for a single target

---

**Require:** Belief state  $b_{t-1,i}$ , action  $u_{t,i}$ , observation  $z_{t,i}$

- 1: Collapse modes that are within 1 std dev.
  - 2: (*Transition update*)
  - 3: **for** each mode  $j$  **do**
  - 4:    $\bar{\mu}_{t,i,j} = A_{t,i,j}\mu_{t-1,i,j} + B_{t,i,j}u_{t,i,j}$
  - 5:    $\bar{\Sigma}_{t,i,j} = A_{t,i,j}\Sigma_{t-1,i,j}A_{t,i,j}^T + R_t$
  - 6:   Split mode if  $\bar{\mu}_{t,i,j}$  reaches a fork node.
  - 7: **end for**
  - 8: (*Observation update*)
  - 9: **if**  $z_{t,i} \neq \text{NULL}$  **then**
  - 10:   Associate  $z_{t,i}$  to edge  $e_k$
  - 11:   **for** each mode  $j$  **do**
  - 12:     **if** mode along  $e_k$  **then**
  - 13:        $K_{t,i,j} = \bar{\Sigma}_{t,i,j}C_{t,i,j}^T(C_{t,i,j}\bar{\Sigma}_{t,i,j}C_{t,i,j}^T + Q_t)^{-1}$
  - 14:        $\mu_{t,i,j} = \bar{\mu}_{t,i,j} + K_{t,i,j}(z_{t,i} - C_{t,i,j}\bar{\mu}_{t,i,j})$
  - 15:        $\Sigma_{t,i,j} = (C_{t,i,j}^TQ_t^{-1}C_{t,i,j} + \bar{\Sigma}_{t,i,j}^{-1})^{-1}$
  - 16:     **else**
  - 17:        $w_{t,i,j} = f_P w_{t-1,i,j}$ .
  - 18:     **end if**
  - 19:   **end for**
  - 20: **else** {Null observation}
  - 21:   **for** each mode  $j$  **do**
  - 22:     Truncate and refit  $(\bar{\mu}_{t,i,j}, \bar{\Sigma}_{t,i,j})$
  - 23:     Reduce weight  $w_{t,i,j}$  based on  $(1 - f_N) \times$  fraction of Gaussian truncated
  - 24:   **end for**
  - 25: **end if**
  - 26: Re-normalize weights  $w_{t,i,j} \forall j$
- 

mode. As the focus of this paper is on the multi-modal nature of the agent’s beliefs, we represent each of the modes as a 1D Gaussian, though the algorithm is easily extendable to multi-variate Gaussian representations.

A multi-modal Gaussian distribution is essentially a weighted sum of uni-modal Gaussian beliefs. This similarity enables us to leverage the popular Kalman filter algorithm for performing our belief update. Kalman filters provide a closed form, efficient means of performing a belief update to obtain the agent’s posterior belief, especially when the transition and observation models are linear with Gaussian noise (Equations 1 - 2). Unfortunately, the original Kalman filter only tracks the overall mean and variance of the agent’s belief, which is insufficient for a multi-modal distribution.

Algorithm 1 presents our modified version of the Kalman filter for performing multi-modal Gaussian belief updates of each target. This algorithm is similar to the sum of Gaussians Kalman filter from the literature [8], but has been adapted to the target-tracking problem. Following [8], at each iteration, modes within one standard deviation of each other are first collapsed into a single mode, so as to keep the belief representation as compact as possible. The Gaussian parameters of the new mode is computed by re-fitting a Gaussian to the original modes, while the new weight is found by summing the weights of the original modes.

Each of the resulting modes are then propagated forward in time based on the target’s transition dynamics, according to the equations for the Kalman filter transition update. If any of the modes reaches a node where there are multiple edges that the target can travel along, additional modes are created with the same Gaussian parameters  $\mu_{t,j}, \Sigma_{t,j}$ , but with the original weight  $w_{t,j}$  uniformly distributed amongst the resultant modes.

The observation update depends on whether an observation of the target’s pose was received. If an observation is received, we first perform a data association step linking the observation to the nearest edge  $e_k$  along the graph. The Kalman filter observation update equations are then used to update all of the modes along that edge, while the weights of all other modes for the target are set to  $f_P$ , the probability that the observation was a false positive. The weights of all the target’s modes are subsequently re-normalized.

If instead a *null* observation is obtained, the agent must still incorporate the negative information associated with the *null* observation. Modes that have support within the agent’s field-of-view are truncated according to the sensor field-of-view, and a new Gaussian is refit around the rest of the mode. In addition, we calculate the relative percentage of the belief that was truncated, and redistribute that fraction of the weight across the remaining modes of the multi-modal Gaussian belief. In this paper, we place strong emphasis on negative information when updating the agent’s belief. Modes with low weights are pruned from the agent’s belief via a variation of ratio pruning [9] to keep the belief representation tractable.

#### IV. PLANNING UNDER UNCERTAINTY WITH MACRO-ACTIONS FOR UNI-MODAL GAUSSIAN BELIEFS

The target-tracking problem requires the agent to plan its next action at every timestep in order to minimize the cost incurred. We measure the cost of each target’s uncertainty using the weighted sum of the covariances associated with each of the target’s Gaussian modes. At each timestep, the agent incurs a cost (or negative reward) according to its belief  $b_t$  and the distance traveled in the last timestep  $d$

$$Rew(b_t, d) = - \sum_i \sum_j w_{t,i,j} tr(\Sigma_{t,i,j}) - \beta d \quad (5)$$

where  $\beta$  controls the relative importance of minimizing the distance traveled. The total cost over a fixed number of timesteps  $T$  is therefore

$$Rew_{Total} = - \sum_{t=1}^T \gamma^t \sum_i \sum_j w_{t,i,j} tr(\Sigma_{t,i,j}) - \beta d \quad (6)$$

where  $\gamma$  is a discount factor.

An approach for tackling this planning under uncertainty problem is to perform forward search, alternating between a planning and execution phase and planning online only for the belief at the current timestep. During the planning phase, a forward search algorithm creates an AND-OR tree of reachable belief states from the current belief state. The tree is expanded using action-observation pairs that are admissible from the current belief, and the beliefs at the leaf nodes are found using Equation 3. By using a value heuristic [10] that estimates the value at the fringe nodes, the expected value of executing a policy from the current belief can be propagated up the tree to the root node.

If a planner can search deep enough, it will find the optimal action for the current belief [11], [12]. Unfortunately, the number of belief states reachable within depth  $D$  is  $(|A||Z|)^D$ , where  $|A|$  and  $|Z|$  are the sizes of the action and observation sets. Not only does the search quickly become intractable as  $D$  increases, but online techniques generally

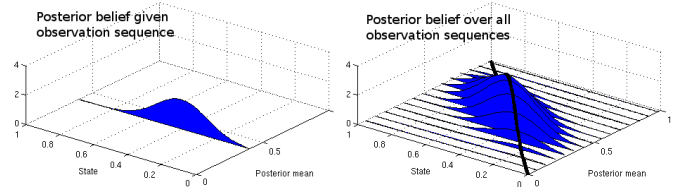


Fig. 2. Distribution of posterior beliefs for uni-modal Gaussian beliefs. a) A Gaussian posterior belief results after incorporating an observation sequence. b) Over all possible observation sequences, the distribution of posterior means is a Gaussian (black line), and for each posterior mean, a Gaussian (blue curve) describes the agent’s posterior belief.

have to meet real-time constraints, limiting the planning time available for each iteration.

One option for searching to a greater depth is to restrict our action space to a set of multi-step action sequences, also known as macro-actions, and compute the expected reward of each macro-action by sampling observation sequences of corresponding length [13]. Unfortunately, the size of the observation space and sampling complexity still grows exponentially with the length of the action sequence, making it challenging to plan to long horizons.

However, the exponential computational dependence on the action sequence length can be broken if we can analytically compute the distribution over beliefs that arises from a macro-action. For a particular macro-action, the probability of the agent obtaining an observation sequence is equivalent to the probability of obtaining the posterior belief associated with that observation sequence. Seen from another angle, every macro-action generates a distribution over beliefs, or a distribution of distributions (Figure 2). If we are able to calculate the distribution over posterior beliefs for every action sequence, and branch at the end of the action sequence by sampling posterior beliefs within this distribution, the sampling complexity is then independent of the macro-action length. Furthermore, the expected reward of an action sequence can then be computed by finding the expected rewards with respect to that distribution, rather than by sampling the possible observations.

We have previously developed the Posterior Belief Distribution (PBD) algorithm [7], such that when the agent’s belief is representable as a uni-modal Gaussian belief, we can directly compute the distribution of posterior beliefs without having to enumerate the possible observations. Specifically, given a prior Gaussian belief  $(\mu_t, \Sigma_t)$ , the resultant distribution of posterior beliefs can be characterized by a distribution of posterior means and a fixed covariance. The distribution of means is itself a normal distribution according to

$$\mu_{t+T} \sim \mathcal{N}(f(\mu_{t-1}, \{A, B, u\}_{t:t+T}), \sum_{\tau=t}^{t+T} \bar{\Sigma}_\tau C_\tau K_\tau^T) \quad (7)$$

where  $f(\mu_{t-1}, \{A, B, u\}_{t:t+T})$  is the deterministic transformation of the means according to  $\mu_{t+k} = A_{t+k}\mu_{t+k-1} + B_{t+k}u_{t+k}$ .

For the covariance update, we can also collapse multi-step covariance updates into a single step. Let  $\Sigma_{t-1} = B_{t-1}C_{t-1}^{-1}$ . At every step, the complete covariance update can be written as:

$$\Psi_t = \begin{bmatrix} B \\ C \end{bmatrix}_t = \begin{bmatrix} 0 & I \\ I & CQ_tC \end{bmatrix}_t \begin{bmatrix} 0 & A^{-T} \\ A & RA^{-T} \end{bmatrix}_t \begin{bmatrix} B \\ C \end{bmatrix}_{t-1}, \quad (8)$$

The covariance update is only dependent on the model parameters of the problem, and is independent of the observations that may be obtained. This property enables us to collapse multi-step covariance updates into a single step  $\Psi_{t+1:T} = \prod_{i=1}^T \Psi_i$ , recovering the posterior covariance  $\Sigma_{t+T}$  from  $\Sigma_{t+T} = B_{t+T} C_{t+T}^{-1}$ . For a uni-modal Gaussian belief, we therefore have a closed-form method for calculating the distribution of posterior beliefs after the agent executes a macro-action, and can then sample Gaussian parameters from this distribution of distributions to instantiate posterior beliefs for deeper forward search.

## V. APPROXIMATIONS FOR MACRO-ACTION, TARGET-TRACKING PLANNING

For the target-tracking problem, the set of macro-actions available to the agent includes the sequence of actions necessary to travel down an edge to a node, as well as hover at a position for a fixed number of timesteps. Each of these macro-actions can be thought of as open-loop policies of varying lengths, independent of the observations that the agent receives while executing the macro-action.

In order to perform efficient planning under uncertainty, we seek to perform belief updates over a macro-action without having to consider the possible observations that could be obtained. We want to compute the joint distribution over the weights and mode statistics, which we will then sample from to obtain posterior beliefs for deeper forward search. For the target-tracking problem, we make explicit two assumptions that are necessary to be able to update the agent's beliefs efficiently: 1) a distance-varying covariance function can model the bimodal characteristic of the observation model, and 2) the observations are accurate enough that over the span of a macro-action, the agent will observe the target at least once if it is within the agent sensor's field-of-view at some point along the macro-action.

### A. Distance-varying observation noise covariance function

A single mode  $j$  in a multi-modal Gaussian belief of target  $i$  is represented as a Gaussian distribution,  $\mathcal{N}(\mu_{t,i,j}, \Sigma_{t,i,j})$ . As per our problem formulation (Equation 2), the agent has a sensor with a limited field-of-view,  $r_a$ . If the target is within the sensor's field-of-view, the agent will receive a noisy position observation  $z_{t,i}$  of the target, whereas if the target is outside of the agent's field-of-view, the agent receives a *null* observation. The possibility of having both noisy position observations *and* a negative observation makes it appear difficult to consider the observation space without branching on the observations.

We propose using a modified noise covariance function to unify the two observation modes. Recall that when the target is within the agent's field-of-view, it receives an observation that is perturbed by Gaussian noise  $\delta_t \sim \mathcal{N}(0, Q_t)$ . The covariance of the Gaussian noise,  $Q_t$ , is then used to calculate the Kalman gain  $K_{t,i,j}$  and update the Gaussian parameters  $(\mu_{t,i,j}, \Sigma_{t,i,j})$  according to

$$K_{t,i,j} = \bar{\Sigma}_{t,i,j} C_t^T (C_t \bar{\Sigma}_{t,i,j} C_t^T + Q_t)^{-1} \quad (9)$$

$$\mu_{t,i,j} = \bar{\mu}_{t,i,j} + K_{t,i,j} (z_{t,i} - C_t \bar{\mu}_{t,i,j}) \quad (10)$$

$$\Sigma_{t,i,j} = (C_t^T Q_t^{-1} C_t + \bar{\Sigma}_{t,i,j}^{-1})^{-1} \quad (11)$$

When the agent's belief of the target's location has little support within the agent's field-of-view, we observe that the belief update according to Algorithm 1 will result in a posterior belief that remains essentially unchanged from the prior belief before the observation update, since little of the belief will be truncated. We also note that for a traditional Kalman filter update, if the noise covariance is very large, then the Kalman filter equations (Equation 9 - 11) result in a posterior belief that also remains essentially unchanged.

When a particular mode of the agent's belief has strong support within the agent's field-of-view, a *null* observation will result in the weight of the mode converging towards zero. Under such circumstances, the update of the individual mode with negligible weights is irrelevant. We delay the discussion of weights updating to Section V-B.

Therefore, we can unify the two observation modes (*null* observation or not) by using an observation noise covariance that varies with different distances between the agent and the target. Specifically, we represent the set of observations using the following covariance function:

$$Q_t(x_t, s_{t,i}) = C_1 - C_2 \mathcal{N}(s_{t,i} | x_t, C_3 r_a) \quad (12)$$

where  $C_1$  is the noise covariance value where getting an observation would have little effect on the belief update, and  $C_1 - C_2 \geq 0$  determines the covariance of the observation noise when the agent is directly above the target.  $C_3$  is a scaling constant that makes the drop off of the covariance function a direct function of the sensor field-of-view.

Despite the possibility of obtaining a *null* observation, the Gaussian parameters of each mode  $j$  of the agent's belief of target  $i$  can therefore be calculated by using a single covariance function (Equation 12). Since the observation noise covariance,  $Q_t$ , is now a function of the agent and the target's poses, and the target poses are unknown, we must integrate over the agent's belief of the target to compute and use the expected noise covariance  $\Sigma_{t,z,i} = E[Q_t]$  for each mode

$$\Sigma_{t,z,i,j} = \int_{s_{t,i}} \mathcal{N}(s_{t,i} | \bar{\mu}_{t,i,j}, \bar{\Sigma}_{t,i,j}) \quad (13)$$

$$\begin{aligned} & \times (C_1 - C_2 \mathcal{N}(s_{t,i} | x_t, C_3 r_a)) ds_{t,i} \\ & = C_1 - C_2 \int_{s_{t,i}} \mathcal{N}(s_{t,i} | \bar{\mu}_{t,i,j}, \bar{\Sigma}_{t,i,j}) \quad (14) \\ & \times \mathcal{N}(s_{t,i} | x_t, C_3 r_a) ds_{t,i} \end{aligned}$$

We note that in a Normal distribution, for every value of  $s_{t,i}, x_t$ ,  $p(s_{t,i} | x_t, C_3 r_a) = p(x_t | s_{t,i}, C_3 r_a)$ , which implies that the above equation can be written as:

$$\Sigma_{t,z,i,j} = C_1 - C_2 \int_{s_{t,i}} \mathcal{N}(s_{t,i} | \bar{\mu}_{t,i,j}, \bar{\Sigma}_{t,i,j}) \quad (15)$$

$$\begin{aligned} & \times \mathcal{N}(x_t | s_{t,i}, C_3 r_a) ds_{t,i} \\ & = C_1 - C_2 \mathcal{N}(s_{t,i} | \bar{\mu}_{t,i,j}, \bar{\Sigma}_{t,i,j} + C_3 r_a) \quad (16) \end{aligned}$$

This calculation of  $\Sigma_{t,z,i,j}$  assumes that the belief is known, i.e., there is a deterministic value of the mean and covariance. However, when performing a belief update after a macro-action, we obtain a distribution over the means (Equation 7). Assuming  $\bar{\mu}_{t,i,j} \sim \mathcal{N}(m_{t,i,j}, S_{t,i,j})$ , the noise

term can then be calculated according to:

$$\Sigma_{t,z,i,j} = C_1 - C_2 \int_{s_{t,i}} \int_{\bar{\mu}_{t,i,j}} \mathcal{N}(x_t|s_{t,i}, C_3 r_a) \quad (17)$$

$$\times \mathcal{N}(s_{t,i}|\bar{\mu}_{t,i,j}, \bar{\Sigma}_{t,i,j}) \mathcal{N}(\bar{\mu}_{t,i,j}|m_{t,i,j}, S_{t,i,j}) d\bar{\mu}_{t,i,j} ds_{t,i}$$

$$= C_1 - C_2 \int_{s_{t,i}} \mathcal{N}(x_t|s_{t,i}, C_3 r_a) \quad (18)$$

$$\times \mathcal{N}(s_{t,i}|m_{t,i,j}, \bar{\Sigma}_{t,i,j} + S_{t,i,j}) ds_{t,i}$$

$$= C_1 - C_2 \mathcal{N}(x_t|m_{t,i,j}, C_3 r_a + \bar{\Sigma}_{t,i,j} + S_{t,i,j}) \quad (19)$$

The expected noise from an observation therefore depends not only on the agent’s pose and expectation of the target’s pose, but also on the agent’s uncertainty of the target’s pose. The larger the uncertainty in the target’s pose, the more uniformly noisy the observation is expected to be with changes in the agent’s pose, which corresponds to a lower expectation of being able to reduce the uncertainty associated with that particular mode.

More importantly, individual modes can now be updated in a consistent, closed-form manner. After executing a macro-action, each mode of the agent’s belief can be represented by a distribution of means (Equation 7), as well as a fixed covariance (Equation 8), with  $Q_t$  replaced by  $\Sigma_{t,z,i,j}$ .

### B. “Macro-observations”

Having discussed how the individual modes of the agent’s multi-modal Gaussian belief can be updated independently of the possible observations, we now turn to updating the weights of the agent’s posterior belief. Unfortunately, it is not reasonable for us to update the weights without incorporating the observations, since by the very nature of the belief updating process, the weights will vary drastically depending on whether the agent has made a positive observation during the macro-action.

In order to reduce the branching factor due to the number of possible observations, we introduce the notion of a “macro-observation” when updating the weights, corresponding to whether *null* observations are received when the multi-step action sequence is executed. For each multi-modal belief of a single target  $i$ , we maintain two different sets of weights,  $W_{Y,i}$  and  $W_{N,i}$ , corresponding to the two separate scenarios where the agent does and does not receive at least a single positive observation while executing a macro-action.

As discussed in Algorithm 1, the agent’s belief is updated after receiving a positive observation by first associating the observation to an edge. When predicting the possible observations for planning, one option is to sample possible data associations and recover the corresponding weights of the modes. Alternatively, we can leverage the distance-varying covariance function that was developed in Section V-A to update the posterior weights of all modes directly. The distance-varying covariance function exhibits the property that the noise covariance of the observation model grows with increasing distance between the agent and the mean of a Gaussian mode. In addition, the Fisher information associated with the observation model measures the certainty of a state estimate due to measurement data alone [14]. In the context of a multi-modal Gaussian belief, the Fisher information associated with the observation model for each

mode therefore corresponds to the certainty that an observation is associated with that mode. Applying Bayes rule  $p(s|z) \propto p(z|s)p(s)$ , we can therefore update the weights of each mode according to

$$w_{t,i,j} \propto C_t \Sigma_{t,z,i,j}^{-1} C_t^T w_{t-1,i,j} \quad (20)$$

where  $C_t \Sigma_{t,z,i,j}^{-1} C_t^T$  is the Fisher information associated with the observation model for the Gaussian mode  $j$  of target  $i$  and  $C_t$  maps the state space to observation space. We then re-normalize the weights.

On the other hand, when a *null* observation is obtained, modes that have support within the agent’s field-of-view should be truncated and a new Gaussian refit around the rest of the mode. Unfortunately, no closed-form solution exists for performing such a truncation. As a result, we take advantage of the intuition that if *null* observations are obtained throughout the course of a macro-action, a mode that has strong support within the agent’s field-of-view will have a posterior weight that tends towards zero. We therefore compute the set of weights associated with *null* observations,  $W_{N,i}$ , by setting the weights of modes that have support in the agent’s field-of-view to zero, and renormalize the rest of the weights.

Separately, we calculate the probability  $\alpha$  of the agent obtaining a positive observation while executing the macro-action. This probability can be obtained by computing the average percentage of the agent’s belief that has support within the field-of-view of the agent’s sensor over the course of the macro-action, assuming that the macro-action had been executed in an open-loop fashion.  $\alpha$  therefore describes the probability that the agent’s multi-modal weights will be distributed according to  $W_{Y,i}$ , as opposed to  $W_{N,i}$ , assuming that the agent does not incorporate any of the observations obtained while executing the macro-action. We then sample from these two sets of weights according to  $\alpha$  to instantiate posterior beliefs for deeper forward search.

Given that the series of belief updates associated with a macro-action have been approximated with a macro-observation, it would appear that an alternative approach would be to reduce the graph map structure into a series of discrete cells, with each cell depicting a connected edge and/or node. However, such a simplification of the problem would result in actions with varying time durations, and performing a belief update using an asynchronous discrete-event model is non-trivial, as evidenced by the computationally complex Partially Observable Semi-Markov Decision Processes (POSMDP) [15] framework.

## VI. MULTI-MODAL POSTERIOR BELIEF DISTRIBUTION (MMPBD) ALGORITHM

Algorithm 2 summarizes the Multi-Modal Posterior Belief Distribution (MMPBD) algorithm for efficient macro-action planning in the target-tracking problem. At every timestep, this forward search algorithm generates the list of macro-actions based on the agent’s current state. It then evaluates each of the macro-actions, and executes the first action from the macro-action with the minimum expected cost. The agent then updates its belief based on the action taken and observation received via Algorithm 1, and repeats the planning process at the next timestep. Hence, even though the planning process only considers macro-actions, it replans

---

**Algorithm 2** Multi-Modal Posterior Belief Distribution (MMPBD) algorithm
 

---

**Require:** Agent’s initial belief  $b_0$ , Macro-action search depth  $H_f$ , No. belief samples per macro-action  $N_z$

- 1:  $t \leftarrow 0$
- 2: **while not** EXECUTIONEND() **do**
- 3:   Generate Macro-action list  $A_{seq}$
- 4:   **for**  $a_{seq,i} \in A_{seq}$  **do**
- 5:      $Q(a_{seq,i}, b_t) = \text{ROLLOUTMACRO}(a_{seq,i}, b_t, H_f, N_z)$
- 6:   **end for**
- 7:   Execute first action  $\hat{a}_t$  of  $\hat{a}_{seq} = \text{argmax} Q(b_t, a_{seq})$
- 8:   Obtain new observation  $z_t$  and cost  $C_t$
- 9:    $b_{t+1} = \text{BELIEFUPDATE}(b_t, a_t, z_t)$
- 10:  $t \leftarrow t + 1$
- 11: **end while**

---



---

**Algorithm 3** ROLLOUTMACRO()
 

---

**Require:** Action sequence  $a_{seq}$ , belief state  $b_t$ , remaining search depth  $h$ , no. belief samples per macro-action  $N_z$

- 1: **if**  $h = 0$  **then**
- 2:   Perform Kalman filter update with null observation out to predefined time depth.
- 3:   return  $\sum_t \text{Rew}(b_t, \Sigma_{t+j})$
- 4: **else** {Expand Macro-Action}
- 5:   **for** each target  $i$  **do**
- 6:     Compute prob.  $\alpha_i$  of obtaining observation along  $a_{seq}$
- 7:     **for** each step in  $a_k \in a_{seq}, k = \{1, \dots, |a_{seq}|\}$  **do**
- 8:       Collapse modes that are within 1 std dev. apart
- 9:       **for** each mode  $j$  in belief  $b_{t,i}$  **do**
- 10:         Compute params  $(m_{t+k,i,j}, S_{t+k,i,j}, \Sigma_{t+k,i,j})$
- 11:         Split modes if end of edge reached
- 12:         Update weights  $w_{y,t+k,i,j}, w_{n,t+k,i,j}$
- 13:       **end for**
- 14:        $V = V + \alpha \text{Rew}(W_{Y,t+k,i}, \Sigma_{t+k,i})$
- 15:        $V = V + (1 - \alpha) \text{Rew}(W_{N,t+k,i}, \Sigma_{t+k,i})$
- 16:     **end for**
- 17:      $V = V + \beta d$
- 18:   **end for**
- 19:   **for**  $p = 1$  to  $N_z$  **do**
- 20:     Gen. samples with weights  $\{W_Y, W_N\}$  accord. to  $\alpha$
- 21:     Sample belief modes from  $(m_{t+a_{seq}}, S_{t+a_{seq}}, \Sigma_{t+a_{seq}})$
- 22:     Obtain action sequence list  $A_{seq}^{next}$
- 23:     **for**  $a_{seq,p}^{next} \in A_{seq}^{next}$  **do**
- 24:        $Q(b_p, a_{seq,p}^{next}) =$   
        $\text{ROLLOUTMACRO}(a_{seq,p}^{next}, b_p, h - 1, N_z)$
- 25:     **end for**
- 26:      $V = V + \gamma \frac{1}{N_z} \text{argmax}_{a_{seq,p}^{next}} Q(b_p, a_{seq,p}^{next})$
- 27:   **end for**
- 28:   return  $V$
- 29: **end if**

---

after every timestep, taking into account the latest observation and information available.

Algorithm 3 summarizes the procedure for calculating the expected cost of each macro-action. After computing the probability  $\alpha$  of obtaining a positive observation during the macro-action, the distribution of posterior beliefs for both the individual modes and the weights are calculated according to Section V-A and V-B respectively.

After calculating the expected immediate cost associated with the macro-action, the agent then samples from the posterior belief distribution to instantiate beliefs for deeper forward search. Given that the covariance of each mode is constant, we perform importance sampling only on the posterior mean distribution and the weights, and associate them with the posterior covariances to obtain samples of

posterior beliefs. These beliefs are then used to perform an additional layer of depth-first search, and this process repeats for a pre-determined search depth  $H_f$ .

At the leaf nodes, a value heuristic is used to provide an estimate of being at the belief node [10]. Because we perform a forward search out to a fixed macro-action depth, rather than a fixed primitive-action depth, the belief nodes may have expanded out to different timesteps. Our value heuristic therefore accumulates the cost of the agent’s belief out to a predefined timestep, and for these additional timesteps assumes that the beliefs are updated without obtaining any meaningful observations.

## VII. SIMULATION EXPERIMENTS

We implemented our target-tracking algorithm in several simulated environments, such as those shown in Figure 1b. The targets circle the environment in a predominantly clockwise fashion, but at a map junction, the targets may choose to travel along one of multiple paths. In addition, different paths can subsequently recombine onto the same edge. For this set of experiments, the agent only had to track two targets, although the algorithm can be directly extended to more targets.

Figure 3 provides snapshots of a simulation run of the MMPBD algorithm. The problem begins with the agent heading in the general direction of the targets (3a, b). Because one of the targets will be arriving at a branching point, the agent hovers at the branching point to observe which outgoing edge is chosen by the target (3b). The agent then moves over to localize the other target (3c). This behavior shows that the agent can anticipate when modes will split at a junction, and search deep enough to recognize that if it does not localize the target at the junction, it will be harder to localize the targets subsequently. In addition, the agent typically focuses on localizing a single target until the uncertainty of the other targets is large enough that it offsets the cost of traveling to those uncertain targets (3d), or if the modes of the other targets are about to split (3e). The cost of motion also biases the agent to choose shorter paths (3f).

We compared our algorithm to two other strategies – a greedy strategy and a forward search strategy without macro-actions. For the greedy strategy, the agent always travels towards the mode that results in the smallest negative rewards, which roughly translates to the mode with the largest weighted covariance and a small cost-to-travel. This strategy causes the agent to oscillate between the different targets, but cannot anticipate when modes may split at road junctions.

We also implemented a forward search algorithm that did not have access to macro-actions. Primitive actions are evaluated by performing belief updates according to Algorithm 1. However, the continuous observation space of the target-tracking problem would have made the branching factor too large to be computationally feasible. Instead, for every action that was evaluated, we only branched on whether an observation was obtained or not, thereby giving the naive forward search algorithm access to the macro-observations that were discussed in Section V-B.

Figure 4 reports the performance of the different strategies for a typical run. Initially, all three strategies obtain similar performance, oscillating between the targets to keep them

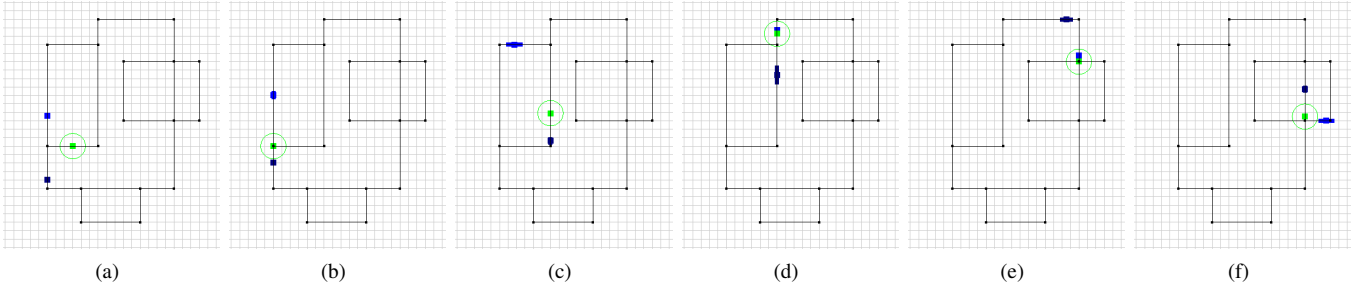


Fig. 3. Snapshots of a simulation run with the MMPBD algorithm. The agent (green) is able to follow targets (d), anticipate mode splitting (b),(e), and intercept targets (f).

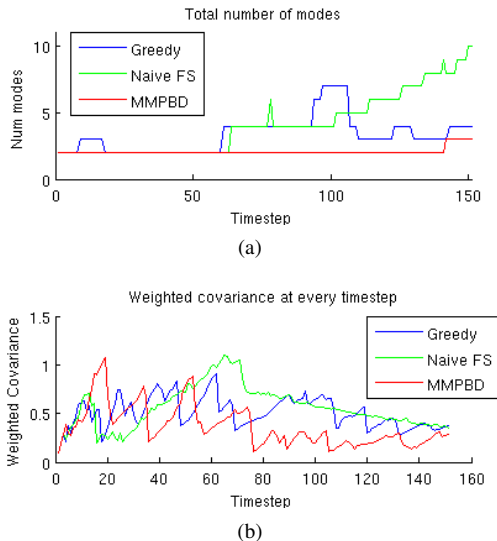


Fig. 4. Comparison of different strategies over a single run. The targets are better localized if the agent’s belief has a smaller weighted covariance.

well-localized. However, as the modes approach the road junctions, the greedy strategy does not anticipate when the modes split. Similarly, the naive forward search strategy cannot search deep enough to realize that it would be harder for the agent to localize the target after the modes have split. The increase in the number of target modes (Figure 4a) for these benchmark algorithms make it harder for the agent to reduce the uncertainty at subsequent timesteps.

Table I summarizes the performance of the three strategies over 10 runs. Although the MMPBD algorithm results in the longest distance traveled, the agent is better able to localize the targets, ensuring that its belief of each target is unimodal most of the time. In contrast, the naive forward search algorithm causes the agent to travel the shortest distance because it focuses on trying to localize a single target.

	Dist. Traveled	Ave. Modes	Total Cost
MMPBD	138.76	1.0810	-51.7698
Greedy	133.52	1.5240	-61.2492
Naive FS	112.20	1.7747	-85.1101

TABLE I  
PERFORMANCE OF DIFFERENT PLANNERS OVER 10 RUNS.

### VIII. REAL-WORLD EXPERIMENTS

Finally, as a proof of concept, we demonstrated the MMPBD algorithm on an indoor quadrotor helicopter (Figure 1a) tasked with tracking two ground vehicles. We set up a mock-up of a road network indoors (Figure 5a), with two

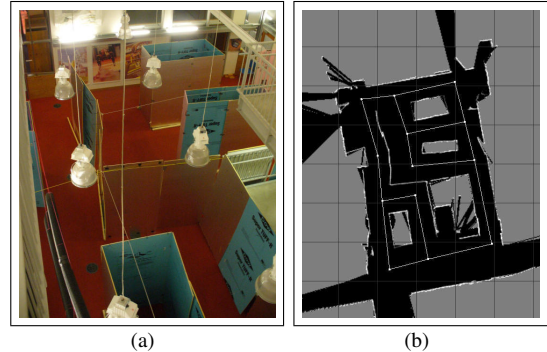


Fig. 5. (a) Mock-up of road network constructed indoors. (b) Graph structure extracted from SLAM map.

autonomous cars driven at approximately constant speeds in the environment. In previous work [16], [17], we have developed a quadrotor helicopter that is capable of autonomous flight in unstructured and unknown indoor environments.

We made use of a publicly available GMapping algorithm [18] to build a map of the environment (Figure 5b), before performing a skeletonization of the map to construct the road-network. Since neither target detection, nor data association, is the focus of this paper, we made the targets easy to detect by using distinct colors for each of the targets, and used a simple blob detection algorithm to detect them.

Figure 6 shows snapshots of the helicopter’s path during a target-tracking run, as well as the images captured from the onboard camera. The helicopter exhibited behaviors similar to those discussed in Section VII, oscillating between the different targets to keep them well-localized.

### IX. RELATED WORK

Variants of the target-tracking problem have been well-researched in the past, and it is beyond the scope of this paper to review that body of work here. However, most of the existing literature has only focused on either the target-searching or target-following problems, as opposed to the unified approach that is presented here.

A sub-class of problems known as road-constrained target-tracking is often used to describe problems where the agent’s belief of the targets’ pose could have multiple modes. In cases of multiple-hypothesis tracking of a single target, the belief updating is traditionally done with particle filters ([4],[5],[6]). An exception is [19], which explores both the Gaussian sum approximation and particle filter approaches. However, these algorithms typically focus on the problem of performing accurate belief updating and data association, rather than on the decision-making for the agent.

In the planning domain, modern approaches to planning with incomplete state information are typically based on

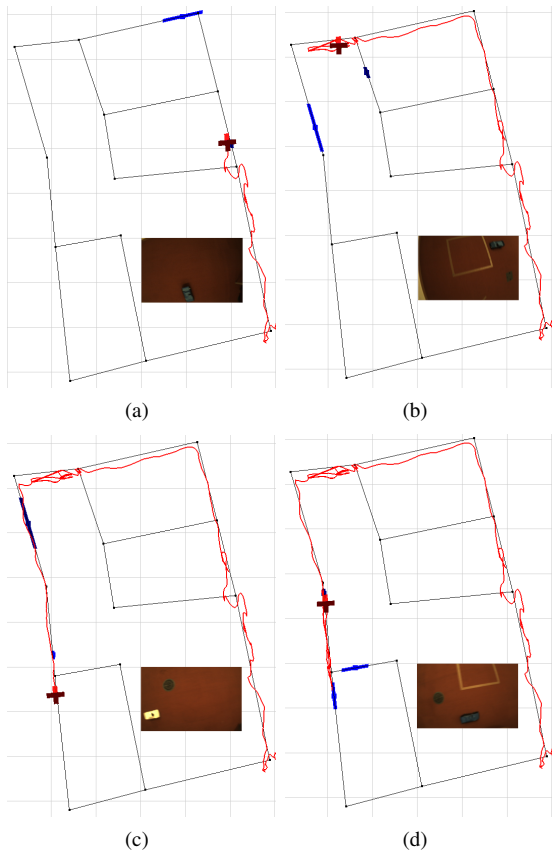


Fig. 6. Real-world target-tracking demonstration. The path traveled by the helicopter (red/brown) is shown in red. The helicopter oscillates between both target beliefs (blue/dark blue) to localize the targets. Inset: Image captured from downward-pointing camera.

the Partially Observable Markov Decision Process (POMDP) framework. [20] presents a POMDP formulation of the target-tracking problem, and applies SARSOP to generate an offline policy. However, this formulation focuses on tracking a single target. Recently, [21] formulated the target-tracking problem for an MAV over multiple targets as a POMDP by assuming uni-modal Gaussian beliefs of the targets' poses, but avoids having to represent the beliefs as multi-modal Gaussian distributions by assuming that obstacles obstruct the agent's view of the targets but not the targets' movement. [22] similarly employs Bayesian techniques for performing search task allocation for multiple vehicles under uncertainty.

Our approach is closely related to the body of online, forward POMDP techniques that have recently been developed. [10] provides a survey of these methods, but these have thus far been limited to discrete belief representations, branching on the individual actions and observations to determine the set of reachable posterior beliefs. In contrast, our approach uses parametric belief representations and branches only at the end of macro-actions. Macro-actions have previously been explored in the context of POMDPs [13], but have not been used as part of a forward-search algorithm. More generally, the Partially Observable Semi-Markov Decision Process (POSMDP) [15] framework allows for actions that can persist for extended time periods, but effective POSMDP algorithms do not currently exist.

## X. CONCLUSION

We have demonstrated the value of probabilistic planning for tracking targets in an urban environment. By using a multi-modal Gaussian representation of the agent's beliefs, we can plan efficiently by considering multi-step action sequences, and we have demonstrated the performance of our algorithm in both simulation and on actual hardware. Future work includes analyzing the errors induced by the approximations introduced, as well as comparing the algorithms on larger environments with more targets.

## XI. ACKNOWLEDGEMENTS

Ruijie He, Abraham Bachrach and Nicholas Roy were supported by NSF IIS under grant # 0546467, the ARO MAST CTA and ONR under MURI N00014-07-1-0749. We would like to thank Emma Brunskill, Finale Doshi, Josh Joseph, and Javier Velez for valuable discussions and feedback. Daniel Gurdan, Jan Stumpf and Markus Achtelik provided the quadrotor helicopter and the support of Ascending Technologies. Adam Bry, Anton De Winter, Garrett Hemann, Albert Huang, and Samuel Prentice assisted with the development of the software and hardware for the helicopter demonstration.

## REFERENCES

- [1] I. Suzuki and M. Yamashita, "Searching for a mobile intruder in a polygonal region," *SIAM Journal on Computing*, vol. 21, p. 863, 1992.
- [2] B. Gerkey, S. Thrun, and G. Gordon, "Visibility-based pursuit-evasion with limited field of view," *IJRR*, vol. 25, no. 4, p. 299, 2006.
- [3] C. Geyer, "Active target search from UAVs in urban environments," in *Proc. ICRA*, 2008.
- [4] C. Hue, J. Le Cadre, P. Perez, and R. Irisa, "Sequential Monte Carlo methods for multiple target tracking and data fusion," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, p. 309, 2002.
- [5] C. Agate and K. Sullivan, "Road-constrained target tracking and identification using a particle filter," in *Proc. SPIE*, 2003.
- [6] D. Streller, "Road map assisted ground target tracking," in *Proc. ICIF*, 2008.
- [7] R. He and N. Roy, "Efficient POMDP Forward Search by Predicting the Posterior Belief Distribution," MIT, Tech. Rep., 2009.
- [8] H. Sorenson and D. Alspach, "Recursive Bayesian Estimation Using Gaussian Sums," *Automatica*, vol. 7, p. 465, 1971.
- [9] T. Schmitt, M. Beetz, R. Hanek, S. Buck, *et al.*, "Watch their moves: Applying probabilistic multiple object tracking to autonomous robot soccer," in *Proc. NCAI*, 2002.
- [10] S. Ross, J. Pineau, S. Paquet, and B. Chaib-draa, "Online planning algorithms for POMDPs," *JAIR*, vol. 32, p. 663, 2008.
- [11] M. Hauskrecht, "Value-function approximations for partially observable Markov decision processes," *JAIR*, vol. 13, p. 33, 2000.
- [12] M. Puterman, *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, Inc., 1994.
- [13] G. Theodorou and L. Kaelbling, "Approximate planning in POMDPs with macro-actions," in *Proc. NIPS*, 2003.
- [14] P. Maybeck, *Stochastic models, estimation, and control, Volume 1*. Academic Press, 1982.
- [15] S. Mahadevan and N. Khaleeli, "Robust mobile robot navigation using partially-observable semi-Markov decision processes," Tech. Rep., 1999.
- [16] R. He, S. Prentice, and N. Roy, "Planning in information Space for a quadrotor helicopter in a GPS-denied environments," in *Proc. ICRA*, 2008.
- [17] A. Bachrach, R. He, and N. Roy, "Autonomous flight in unstructured and unknown indoor environments," in *Proc. EMAV*, 2009.
- [18] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE Transactions on Robotics*, vol. 23, no. 1, p. 34, 2007.
- [19] M. Ulmke, W. Koch, and W. FGAN-FKIE, "Road-map assisted ground moving target tracking," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 42, no. 4, p. 1264, 2006.
- [20] D. Hsu, W. Lee, and N. Rong, "A point-based POMDP planner for target tracking," in *Proc. ICRA*, 2008.
- [21] A. Scott, Z. Harris, and E. Chong, "A POMDP framework for coordinated guidance of autonomous UAVs for multitarget tracking," *EURASIP Journal on Advances in Signal Processing*, 2009.
- [22] L. Bertuccelli and J. How, "Bayesian forecasting in multi-vehicle search operations," in *Proc. AIAA GNC*, 2006.