Probabilistic Path Planning for Cooperative Target Tracking Using Aerial and Ground Vehicles

Huili Yu* Randal W. Beard**, *Senior Member, IEEE* Kevin Meier* Matthew Argyle*

***Abstract*—In this paper, we present a probabilistic path planning algorithm for tracking a moving ground target in urban environments using UAVs in cooperation with UGVs. The algorithm takes into account vision occlusions due to obstacles in the environments. The target state is modeled using a dynamic occupancy grid and the probability of the target location is updated using a Bayesian filter. Based on the probability of the target's current and predicted locations, the path planning algorithm is first designed to generate paths for a single UAV or UGV maximizing the sum of probability of detection over a finite look-ahead. For target tracking using multiple vehicle collaboration, a decentralized planning algorithm using an auction scheme generates paths maximizing the sum of the joint probability of detection over the finite look-ahead horizon. The decentralized planning algorithm results in linear computational growth in the number of surveillance vehicles.**

***Index Terms*—target tracking, cooperative control, path planning, miniature air vehicles.**

I. Introduction

Small unmanned air vehicles (UAVs) have recently found applications in the task of tracking moving targets on the ground. Many approaches to this topic have been presented in the last few years [1]–[4]. The main advantages of target tracking using UAVs are that they have a wide field of view and can cover large areas quickly. However, sensors mounted on UAVs are unable to localize the target on the ground accurately due to the limitations on altitude and airspeed. On the other hand, unmanned ground vehicles (UGVs) are slower with limited field of view, but they are capable of getting closer to targets and resolving their relative locations with greater accuracy [5]. In addition, in a pursuit-evasion scenario, a ground vehicle has the ability to "capture" a target, whereas an aerial vehicle can only observe and inform. Accordingly, the complimentary strength of air and ground based sensors motivates the cooperative use of both UAVs and UGVs for target tracking.

Some approaches to the target tracking problem using both UAVs and UGVs have been proposed. Reference [6] describes an information based approach to UAV/UGV cooperative tracking. This approach works well when the targets are static, and when the environment is relatively free of occlusions, allowing the efficient use of log-likelihood filters, but it is ill-suited to tracking evasive targets in complicated urban environments. Air and ground vehicle cooperation in a probabilistic pursuit-evasion framework is considered in reference [7]. But this approach does not consider sensor data fusion, complex terrain, or planning for occluded vision. Reference [8] presents a control scheme that guides a team of UGVs into a formation to effectively "corral" targets into a specific region, while a team of UAVs fly over the formation to detect targets. The approach assumes large teams of air and ground robots, and does not consider the effect of occlusions, non-navigable terrain, and data fusion.

This paper presents a probabilistic path planning algorithm for tracking a moving target in urban environments using both UAVs and UGVs. Urban terrain complicates the tracking problems because the large number of buildings and other obstacles occlude the line of sight between the sensors and the target. The main contribution of the proposed planning algorithm is to take into account the occlusions due to obstacles. We model the target state using a dynamic occupancy grid and use a second-order Markov chain model to represent the target motion. The probability distribution on the target location is updated using a Bayesian filter. For designing the planning algorithm, we define the probability of detection given the locations of the sensor and the target using a Gaussian function of the distance between the sensor and the target. To include the effect of occlusions, the probability of detection for the configurations where occlusions exist is assigned as zero. Based on the probability of detection of the target's current and predicted future locations, we design the path planning algorithm for independent target tracking by a single vehicle (UAV or UGV). The algorithm generates optimal paths maximizing the sum of the probability of detection over a finite look-ahead horizon.

For cooperative target tracking using multiple UAVs and UGVs, we define the joint probability of detection and design a decentralized suboptimal approach relying on an auction scheme to generate optimal paths maximizing the sum of joint probability of detection over the finite look-ahead horizon. The advantage of this approach is that it results in linear computational growth as the number of vehicles increases.

This paper is organized as follows. Section II describes the target state modeling and estimation using a dynamic occupancy grid. In Section III, the path planning algorithm for target tracking using a single UAV or UGV is introduced. Section IV introduces the decentralized path planning algorithm for cooperative target tracking using multiple UAVs and UGVs. Numerical results are described in Section V.

*Research assistant in Department of Electrical and Computer Engineering, Brigham Young University, Provo, USA, huiliyu.yhl@gmail.com

**Professor in Department of Electrical and Computer Engineering, Brigham Young University, Provo, USA, beard@byu.edu

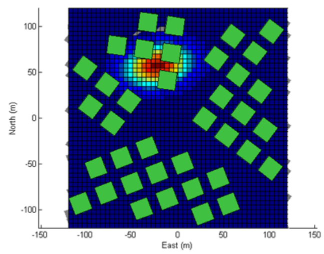
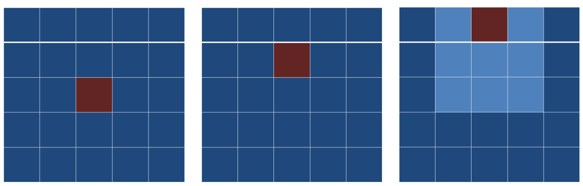


Fig. 1. The dynamic target occupancy grid.



(a) t-1 (b) t (c) t+1

Fig. 2. The target motion model. The target moves forward from time step $t-1$ to t. Accordingly, the target will be assumed to move forward with a high probability P_c at time step $t+1$. The probability $1-P_c$ will be equally divided between the neighboring cells.

II. Target state modeling and estimation

In order to plan paths for a UAV/UGV to track the target, we must estimate the target state at each time step. In this section, we describe the method for target state estimation using a dynamic occupancy grid. A discrete probabilistic model of the target motion is constructed to predict the target location using a second-order Markov chain, and is then combined with Bayes-filtered sensor measurements to update the target location.

Using a dynamic occupancy grid is a classical approach for addressing the problem of generating consistent maps from noisy and uncertain measurement data [9]. The basic idea is to represent the map as a spatial grid, where each occupancy cell has a random variable s associated with it. The random variable s has two states, *occupied* and *empty*, which correspond to the occupancy of that cell. We use a dynamic occupancy grid to represent changing belief about the target location. Figure 1 shows the dynamic occupancy grid of the target location. To calculate the probability that the target will be in a given cell at time $t+1$ we use data from the current and previous time steps, which is a second-order Markov model, and assume that the target will most likely proceed along its direction, as shown in Fig. 2. In the figure, the target moves forward from time step $t-1$ to t. Accordingly, the target will be assumed to move forward with a high probability P_c at time $t+1$. The probability $1-P_c$ will be equally divided between the neighboring cells, as shown in Fig. 2 (c). One of the advantages of using a probabilistic model of the target motion is that several potential target paths can be captured simultaneously.

Let $x_T(t)$ represent the target state at time t and let $P(x_T(t))$ represent the posterior probability that the target is at $x_T(t)$, which is used as the prior probability of the target location at time $t+1$. The dynamic occupancy grid utilizes a Bayesian filter to implement approximate posterior estimation for each grid cell. The Bayesian filter consists of two phases: prediction and update. The prediction phase uses the target motion model given by $P(x_T(t+1)|x_T(t),x_T(t-1))$, which represents the probability that the target is at $x_T(t+1)$ at time $t+1$ given its location $x_T(t)$ at time t and its location $x_T(t-1)$ at time $t-1$. As mentioned above, we represent the target motion model using a second-order Markov chain. The predicted target probability at time $t+1$ before the new measurements are taken into account are then given by

$$\begin{aligned}&\bar{P}(x_T(t+1)) = \\ &\int\int P\left(x_T(t+1)|x_T(t),x_T(t-1)\right)\cdot \\ &P(x_T(t-1))\cdot P(x_T(t))dx_T(t-1)dx_T(t).\end{aligned} \tag{1}$$

When the position of the target is observed by a member of the UAV/UGV team, the occupancy grid is updated to reflect the new information. This update is the measurement phase in the Bayesian filter. Let N represent the number of vehicles on the UAV/UGV team. The measurement model is represented by $P(z_i|x_T)$ which is the probability of receiving the measurement z_i from the i^{th} observation platform (UAV/UGV) given that the target is located at x_T. We assume that all N vehicles can communicate with each other and share their measurements with each other as global information. Under the assumption that the measurements made by each vehicle are independent, the posterior probability that the target is at $x_T(T+1)$ at time step $t+1$ is given by

$$P(x_T(t+1)) = \eta \sum_{i=1}^{N} P(z_i|x_T)\bar{P}(x_T(t+1)), \tag{2}$$

where η is a normalization factor. Equations (1) and (2) constitute the Bayesian filter for updating the posterior probability of the target location. We should note that if a measurement is not received at every time step, then the probability of the target location is updated using Eq. (1). At the beginning of an observation mission, when the target has not been observed by any platforms, the probability of the target location can be initialized as a uniform distribution. However, if priori information is known, then the probability map can be initialized using this information.

III. Path planning for a single vehicle based on predicted target behavior

Given the probability of the target's current and probable future locations computed using the procedure described in the previous section, we design a path planning algorithm for tracking the target by a single vehicle (UAV or UGV). The objective of the algorithm is to generate a parameterized path over a finite look-ahead T_L. Future paths can be parameterized in a number of different ways including a set of roll angles or a set of waypoints. To be general, let Θ_i represent the path parameterization over the time horizon $[t,t+T_L]$. We use

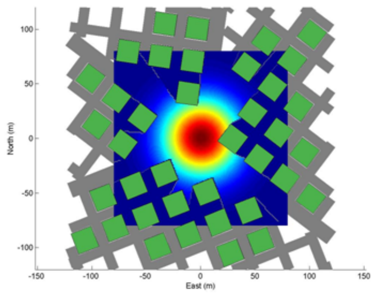


Fig. 3. The probability of detection when buildings occlude the line of sight vector. The location of the target is at (0,0) and the location of the sensor is varied.

the notation $x_i(t,\sigma,\Theta_i)$ to represent the predicted location of the i^{th} sensor platform (UAV/UGV) at time $t+\sigma$, given its current location at time t, and the path parameterization Θ_i, where $\sigma \in [0,T_L]$.

Let D_i represent the event that the target is detected by the i^{th} vehicle, and let $P(D_i|x_i,x_T)$ represent the probability that the target is detected by the i^{th} vehicle when it is at x_i and the target is at x_T. We represent $P(D_i|x_i,x_T)$ as a Gaussian function of the distance between the target and the sensor:

$$P(D_i|x_i,x_T) = \eta \exp(-\frac{1}{2}(x_i - x_T)^\top \Sigma^{-1}(x_i - x_T)), \quad (3)$$

where Σ is the covariance and η is a normalization factor. Please note that other non-Gaussian distribution functions can be used to represent $P(D_i|x_i,x_T)$ as well. The planning algorithm does not require the Gaussian distribution assumption and we use the Gaussian distribution only as an example. In this paper, we pay particular attention to the problem of tracking a moving target in urban and cluttered environments where occlusions due to buildings and other terrain are prevalent. We assume that an elevation map of the environment is available to the vehicles before the mission begins. To include the effect of occlusions, we also include the map as a dependence in the probability of detection where $P(D_i|x_i,x_T,m)$ is the probability that the target is detected by the i^{th} vehicle given that the i^{th} vehicle is at x_i, the target is at x_T, and the map is given by m. We then compute

$$P(D_i|x_i,x_T,m) = \eta \begin{cases} P(D_i|x_i,x_T) & \text{LOS is not occluded,} \\ 0 & \text{otherwise,} \end{cases} \quad (4)$$

where again, η is a normalization factor. Figure 3 shows an example probability of detection in an urban environment. In this example, the target location x_T is held constant at (0,0), while the vehicle location x_i is varied in one meter increments on both axes. As shown, when a building occludes the line of sight, the probability of detection is zero. Occlusions are detected by testing each line of sight vector for intersections with each building polygon in the map. Since we assume that the map is known, $P(D_i|x_i,x_T,m)$ can be precomputed for each pair of x_i and x_T and stored in memory.

Using the law of total probability, the probability that the target is detected when the i^{th} vehicle is at x_i is

$$P(D_i|x_i) = \int P(D_i|x_i,x_T)P(x_T)dx_T. \quad (5)$$

We are interested in maximizing the probability of detection over the future time horizon $[t, t+T_L]$. Using the path parameterization discussed above, we have that the probability of detection at future time $\sigma \in [0,T_L]$ is given by

$$P(D_i|x_i(t,\sigma,\Theta_i)) = \quad (6)$$
$$\int P(D_i|x_i(t,\sigma,\Theta_i),x_T(t+\sigma))P(x_T(t+\sigma))dx_T(t+\sigma).$$

For optimizing the path parameterization Θ_i, we propose a return function, which measures the sum of probability of detection over the look-ahead window and is given by

$$J_i(\Theta_i) = \int_0^{T_L} P(D_i|x_i(t,\sigma,\Theta_i))d\sigma. \quad (7)$$

The myopic planning problem for a single vehicle is then to find Θ_i to maximize $J_i(\Theta_i)$. The optimum path parameters are then passed to the low level autopilot and followed for a time less than or equal to T_L.

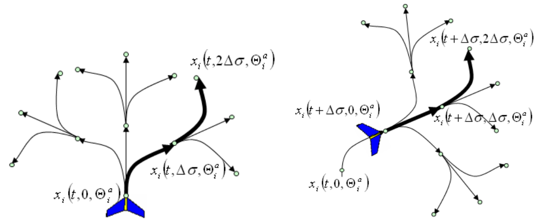
For independent target tracking using a UAV, we parameterize the paths by roll angles since different roll angles generate different paths. Let $\Phi_i = [-\phi_{max}, \phi_{max}]$ represent the set of roll angles, where $\pm\phi_{max}$ is the positive/negative maximal roll angle. An optimization method can be used to find the optimal roll angle over the set Φ_i that maximizes $J_i(\Theta_i)$.

To solve the optimization problem, in this paper we discretize Φ_i as a finite set of roll angles represented by $\Phi_{id} = \{\phi_1, \phi_2, \cdots, \phi_m\}$, where $\phi_1 = -\phi_{max}$ and $\phi_m = \phi_{max}$ and we also discretize the look-ahead window $[0,T_L]$ as $T_d = \{0, \Delta\sigma, \cdots, n\Delta\sigma\}$, $\Delta\sigma = T_L/n$, which is the n-step look-ahead horizon. Let $\Theta_i^a \in \Phi_{id} \times \cdots \times \Phi_{id}$ represent the path parameterization over the n-step look-ahead horizon. For the n-step look-ahead planning horizon, the cost function given by Eq. (7) becomes

$$J_i(\Theta_i^a) = \sum_{j=0}^{n} P(D_i|x_i(t, j\Delta\sigma, \Theta_i^a)). \quad (8)$$

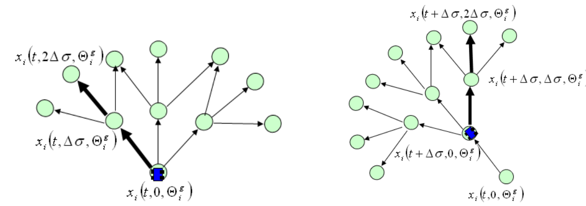
To maximize the return function given by Eq. (8), we recursively search a tree representing a set of potential paths over the n-step look-ahead horizon. Each node in the tree represents the UAV configuration at a certain stage and it has multiple children, each of which represents the resulting configuration at the next stage corresponding to a certain roll angle.

The path planning algorithm for target tracking using a single UAV can be described as follows. When the UAV is at the configuration $x_i(t,0,\Theta_i^a)$ at time t, the algorithm has already determined an optimal path $\tau_i(t) = \{x_i(t,0,\Theta_i^a), x_i(t,\Delta\sigma,\Theta_i^a), \cdots, x_i(t,n\Delta\sigma,\Theta_i^a)\}$. The UAV is maneuvered towards $x_i(t,\Delta\sigma,\Theta_i^a)$. During that period, the algorithm first takes $x_i(t,\Delta\sigma,\Theta_i^a)$ as the tree root and the tree



(a) t (b) $t+\Delta\sigma$

Fig. 4. A two-step look-ahead path planning tree for the UAV, where $T_d = \{0, \Delta\sigma, 2\Delta\sigma\}$ and $\Phi_{id} = \{\phi_1, \phi_2, \phi_3\}$.



(a) t (b) $t+\Delta\sigma$

Fig. 5. A two-step look-ahead path planning tree for the UGV.

is pruned by only maintaining the branches with the root at $x_i(t, \Delta\sigma, \Theta_i^a)$. The tree is then extended by one stage and the new tree is searched to find a new path $\tau_i(t+\Delta\sigma)$. Once the UAV reaches $x_i(t, \Delta\sigma, \Theta_i^a)$, the new path $\tau_i(t+\Delta\sigma)$ has been generated. We repeat this process recursively so that the UAV is always maneuvered to configurations where the probability of detection is high.

Figure 4 shows a two-step look-ahead planning horizon tree, where $T_d = \{0, \Delta\sigma, 2\Delta\sigma\}$ and $\Phi_{id} = \{\phi_1, \phi_2, \phi_3\}$. When the UAV is at the configuration $x_i(t, 0, \Theta_i^a)$ at time t and the path $\tau_i(t) = \{x_i(t, 0, \Theta_i^a), x_i(t, \Delta\sigma, \Theta_i^a), x_i(t, 2\Delta\sigma, \Theta_i^a)\}$ has been found, as shown in Fig. 4(a). In Fig. 4(b), the UAV is maneuvered to $x_i(t, \Delta\sigma, \Theta_i^a)$ and the branches whose root is not at $x_i(t, \Delta\sigma, \Theta_i^a)$ are removed. The tree is then extended by one step horizon and the new tree is searched to find a new path $\tau_i(t+\Delta\sigma) = \{x_i(t+\Delta\sigma, 0, \Theta_i^a), x_i(t+\Delta\sigma, \Delta\sigma, \Theta_i^a), x_i(t+\Delta\sigma, 2\Delta\sigma, \Theta_i^a)\}$. Once the UAV reaches $x_i(t, \Delta\sigma, \Theta_i^a)$, the new path $\tau_i(t+\Delta\sigma)$ has been found. Given a tree, searching the tree and finding a path can be solved efficiently using dynamic programming [10].

For independent target tracking using a single UGV, we decompose the roads into cells and construct a graph using those cells since the UGV can only move along the roads. Similarly, we discretize the look-ahead window $[0, T_L]$ as the n-step look-ahead horizon $T_d = \{0, \Delta\sigma, \cdots, n\Delta\sigma\}$, where $\Delta\sigma = T_L/n$. For each stage, the paths to the next stage are parameterized by the waypoints, denoted by Θ_i^g, which are the centers of the neighboring cells. The cost function to be maximized for target tracking using a single UGV is given by

$$J_i(\Theta_i^g) = \sum_{j=0}^{n} P(D_i | x_i(t, j\Delta\sigma, \Theta_i^g)). \tag{9}$$

Similarly, the n-step look-ahead planning horizon tree is constructed. The connectivity of the graph determines the extension of the tree. Figure 5 shows a two-step look-ahead planning horizon tree, where the circles represent the nodes and the tree is extended based on the connectivity of the graph. At time t, the UGV is at the configuration $x_i(t, 0, \Theta_i^g)$ and the path $\tau_i(t) = \{x_i(t, 0, \Theta_i^g), x_i(t, \Delta\sigma, \Theta_i^g), x_i(t, 2\Delta\sigma, \Theta_i^g)\}$ has been found as shown in Fig. 5(a). The UGV is maneuvered to $x_i(t, \Delta\sigma, \Theta_i^g)$ and the branches whose root is not at $x_i(t, \Delta\sigma, \Theta_i^g)$ are removed, as shown in Fig. 5(b). The tree is then extended by one stage and the new tree is searched to find a new path $\tau_i(t+\Delta\sigma)$ using dynamic programming. Once the UGV reaches $x_i(t, \Delta\sigma, \Theta_i^g)$, the new path $\tau_i(t+\Delta\sigma) = \{x_i(t+\Delta\sigma, 0, \Theta_i^g), x_i(t+\Delta\sigma, \Delta\sigma, \Theta_i^g), x_i(t+\Delta\sigma, 2\Delta\sigma, \Theta_i^g)\}$ has been found.

IV. Path planning for multiple vehicle collaboration

The approach described in the previous section can easily be extended to multiple vehicles. Let I be an index set of vehicles and let $\mathbf{x}_I$ be the combined state of all vehicles whose index is in I. Let D_I represent the event that at least one vehicle in I can detect the target. The probability that at least one of vehicles detects the target given $\mathbf{x}_I$ and the target location x_T is denoted by $P(D_I | \mathbf{x}_I, x_T)$. It can be shown using standard probabilistic reasoning, that if the measurements made by each vehicle are independent, then

$$P(D_I | \mathbf{x}_I, x_T) = 1 - \prod_{i \in I} \left(1 - P(D_i | x_i, x_T)\right). \tag{10}$$

This formula is significant, because it shows that the joint probability of detection can be computed by combining the probability of detection for each vehicle. The probability that at least one of vehicles detects the target given $\mathbf{x}_I$ is denoted by

$$P(D_I | x_I) = \int P(D_I | x_I, x_T) P(x_T) dx_T. \tag{11}$$

Let Θ_I be the combined path parameters for all vehicles in the index set I. We can define the optimization criteria similar to Eq. (7) that are over the index set I, where

$$J(\Theta_I) = \int_0^{T_L} P(D_I | \mathbf{x}_I(t, \sigma, \Theta_I)) d\sigma. \tag{12}$$

The joint team optimization problem is to let I include all UAVs and UGVs on the team, and to maximize the return function $J(\Theta_I)$ at each planning instant. Unfortunately, this problem is NP-complete and so the computational time will grow exponentially in the number of UAVs and UGVs. In addition, solving the full joint optimization problem requires a centralized implementation. To mitigate these problems, we propose using a decentralized suboptimal approach that relies on an auction scheme. To best describe our approach, we need some additional notation. Let I and K represent two index sets where $I \bigcap K = \emptyset$, and let $J(\Theta_I | \Theta_K)$ represent the return function defined by Eq. (12) but where the path parameters for the vehicles in I are free variables and the path parameters for the vehicles in K are fixed.

Consider that there are N vehicles in the groups. The decentralized algorithm that we use consists of N steps. Let Θ_i^j represent the path parameterization for the i^{th} vehicle at the j^{th} step of the algorithm. The first step of the algorithm is for each vehicle to maximize $J(\Theta_i^1)$, $i=1,\cdots,N$, and to send the optimal myopic return to the other vehicles in the network. If k_1 is the index of the vehicle such that $k_1 = \underset{i}{\operatorname{argmax}}(J(\Theta_i^1))$, then the path of the $(k_1)^{th}$ vehicle over the look-ahead window $[0,T_L]$ is parameterized by $\Theta_{k_1}^1$, and each vehicle assigns $K=k_1$. At the second step, the remaining vehicles maximize $J(\Theta_i^2|\Theta_K)$, $i=1,\cdots,k_1-1,k_1+1,\cdots,N$ and send the resulting optimal value to the group. If k_2 is the index of the vehicle such that $k_2 = \underset{i}{\operatorname{argmax}}(J(\Theta_i^2|\Theta_K))$, then the path of the $(k_2)^{th}$ vehicle is parameterized by $\Theta_{k_2}^2$. The $(k_2)^{th}$ vehicle is added to K such that $K=\{k_1,k_2\}$. The process repeats until all vehicles $\{k_1,k_2,\cdots,k_N\}$ have been assigned path parameters $\{\Theta_{k_1}^1,\Theta_{k_2}^2,\cdots,\Theta_{k_N}^N\}$. For the N vehicles, $N-1$ auctions will be required. Let $|\Theta_i|$ represent the cardinality of the set Θ_i. The advantage of this approach is that rather than optimizing over $|\Theta_i|^N$ parameters, the process requires N optimizations over $|\Theta_i|$ parameters, resulting in linear computational growth. Algorithm 1 shows the decentralized path planning algorithm.

Algorithm 1: The decentralized path planning algorithm

```
Initialize I = {1,2,···,N}, K = ∅;
for j ← 1 to N do
    Each UAV optimizes its own myopic return
    J(Θ_i^j | Θ_K), ∀i ∈ I;
    Find the vehicle with the maximum myopic return;
    Parameterize the path for that vehicle using the path
    parameters causing the maximum myopic return;
    Remove the vehicle index from I and add the
    vehicle index to K.
end
```

We compare the suboptimal solution obtained by the decentralized algorithm and the optimal solution obtained by maximizing the joint return function (12) over all vehicles. We first present two properties of the return function (12), as shown in Lemmas 1 and 2.

Lemma 1: The cost function (12) satisfies $J(\Theta_{I\cup L}) - J(\Theta_L) \geq J(\Theta_{I\cup K}) - J(\Theta_K)$, $\forall L \subseteq K$.

Proof: Based on Eq. (10), we can see that, $\forall L \subseteq K$

$$
\begin{aligned}
&P(D_{I\cup L}|\mathbf{x}_{I\cup L},x_T) - P(D_L|\mathbf{x}_L,x_T)\\
&-P(D_{I\cup K}|\mathbf{x}_{I\cup K},x_T) + P(D_K|\mathbf{x}_K,x_T) =\\
&-\prod_{i\in I\cup L}(1-P(D_i|x_i,x_T)) + \prod_{i\in L}(1-P(D_i|x_i,x_T))\\
&+\prod_{i\in I\cup K}(1-P(D_i|x_i,x_T)) - \prod_{i\in K}(1-P(D_i|x_i,x_T)) =\\
&\left(1-\prod_{i\in I}(1-P(D_i|x_i,x_T))\right)\cdot\\
&\left(\prod_{i\in L}(1-P(D_i|x_i,x_T)) - \prod_{i\in K}(1-P(D_i|x_i,x_T))\right) \geq 0.
\end{aligned}
$$

Substituting the above inequality into Eqs. (11) and (12) leads to $J(\Theta_{I\cup L}) - J(\Theta_L) - J(\Theta_{I\cup K}) + J(\Theta_K) \geq 0$, $\forall L \subseteq K$, which completes the proof. ∎

Lemma 2: The cost function (12) satisfies $J(\Theta_I) \leq J(\Theta_K)$, $\forall I \subseteq K$.

Proof: Based on Eq. (10), we can see that, $\forall I \subseteq K$,

$$
\begin{aligned}
&P(D_I|\mathbf{x}_I,x_T) - P(D_K|\mathbf{x}_K,x_T) =\\
&-\prod_{i\in I}(1-P(D_i|x_i,x_T)) + \prod_{i\in K}(1-P(D_i|x_i,x_T)) =\\
&\prod_{i\in I}(1-P(D_i|x_i,x_T))\left(\prod_{i\in K\setminus I}(1-P(D_i|x_i,x_T)) - 1\right) \leq 0.
\end{aligned}
$$

Substituting the above inequality into Eqs. (11) and (12) leads to $J(\Theta_I) - J(\Theta_K) \leq 0$, $\forall I \subseteq K$, which completes the proof. ∎

Lemmas 1 and 2 show that the return function (12) satisfies the submodular and nondecreasing properties presented by [11]. The submodular property of an objective function states that the more measurements are added to the objective function, the less valuable an individual measurement becomes. The nondecreasing property of an objective function indicates that the addition of a measurements to the objective function increases the value of the function [11]. Let $\{\bar{\Theta}_1,\cdots,\bar{\Theta}_N\}$ and $\{\hat{\Theta}_1,\cdots,\hat{\Theta}_N\}$ represent the optimal path and the suboptimal path generated by Algorithm 1, respectively, for a team of N vehicles. Theorem 1 shows the comparison of the optimal and suboptimal solutions.

Theorem 1: For a team of N vehicles, the decentralized path planning algorithm given by Algorithm 1 generates a suboptimal path $\{\hat{\Theta}_1,\cdots,\hat{\Theta}_N\}$, such that $\frac{1}{2}J(\{\bar{\Theta}_1,\cdots,\bar{\Theta}_N\}) \leq J(\{\hat{\Theta}_1,\cdots,\hat{\Theta}_N\}) \leq J(\{\bar{\Theta}_1,\cdots,\bar{\Theta}_N\})$, where $\{\bar{\Theta}_1,\cdots,\bar{\Theta}_N\}$ is the optimal path and $J(\cdot)$ is the return function given by Eq. (12).

Proof: Since $\{\bar{\Theta}_1,\cdots,\bar{\Theta}_N\}$ is the optimal solution to the return function (12), it is apparent that $J(\{\hat{\Theta}_1,\cdots,\hat{\Theta}_N\}) \leq J(\{\bar{\Theta}_1,\cdots,\bar{\Theta}_N\})$. We only need to show $\frac{1}{2}J(\{\bar{\Theta}_1,\cdots,\bar{\Theta}_N\}) \leq J(\{\hat{\Theta}_1,\cdots,\hat{\Theta}_N\})$.

Based on Lemma 2, we have that

$$\begin{aligned}
&J(\{\bar\Theta_1,\cdots,\bar\Theta_N\}) \le J(\{\hat\Theta_1,\cdots,\hat\Theta_N,\bar\Theta_1,\cdots,\bar\Theta_N\}) = \\
&J(\{\hat\Theta_1,\cdots,\hat\Theta_N,\bar\Theta_1,\cdots,\bar\Theta_N\}) - \\
&J(\{\hat\Theta_1,\cdots,\hat\Theta_N,\bar\Theta_1,\cdots,\bar\Theta_{N-1}\}) + \\
&J(\{\hat\Theta_1,\cdots,\hat\Theta_N,\bar\Theta_1,\cdots,\bar\Theta_{N-1}\}) - \\
&J(\{\hat\Theta_1,\cdots,\hat\Theta_N,\bar\Theta_1,\cdots,\bar\Theta_{N-2}\}) + \cdots + \\
&J(\{\hat\Theta_1,\cdots,\hat\Theta_N,\bar\Theta_1,\bar\Theta_2\}) - J(\{\hat\Theta_1,\cdots,\hat\Theta_N,\bar\Theta_1\}) + \\
&J(\{\hat\Theta_1,\cdots,\hat\Theta_N,\bar\Theta_1\}) - J(\{\hat\Theta_1,\cdots,\hat\Theta_N\}) + \\
&J(\{\hat\Theta_1,\cdots,\hat\Theta_N\}) - J(\{\hat\Theta_1,\cdots,\hat\Theta_{N-1}\}) + \cdots + \\
&J(\{\hat\Theta_1,\hat\Theta_2\}) - J(\hat\Theta_1) + J(\hat\Theta_1).
\end{aligned}$$

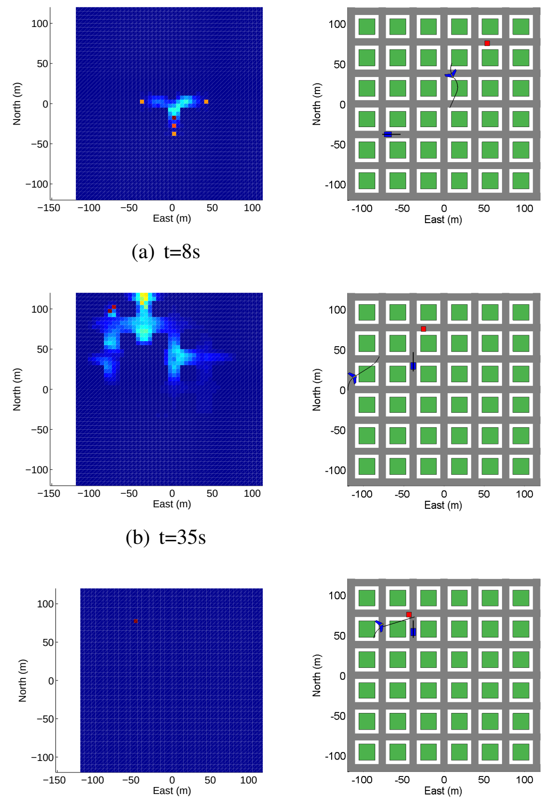
Based on Lemma 1, we further have that

$$\begin{aligned}
&J(\{\hat\Theta_1,\cdots,\hat\Theta_N,\bar\Theta_1,\cdots,\bar\Theta_N\}) \le \\
&J(\{\hat\Theta_1,\cdots,\hat\Theta_{N-1},\bar\Theta_N\}) - J(\{\hat\Theta_1,\cdots,\hat\Theta_{N-1}\}) + \\
&J(\{\hat\Theta_1,\cdots,\hat\Theta_{N-2},\bar\Theta_{N-1}\}) - J(\{\hat\Theta_1,\cdots,\hat\Theta_{N-2}\}) + \\
&\cdots + J(\{\hat\Theta_1,\hat\Theta_2,\bar\Theta_3\}) - J(\{\hat\Theta_1,\hat\Theta_2\}) + \\
&J(\{\hat\Theta_1,\bar\Theta_2\}) - J(\hat\Theta_1) + J(\bar\Theta_1) + \\
&J(\{\hat\Theta_1,\cdots,\hat\Theta_N\}) - J(\{\hat\Theta_1,\cdots,\hat\Theta_{N-1}\}) + \\
&\cdots + J(\{\hat\Theta_1,\hat\Theta_2,\hat\Theta_3\}) - J(\{\hat\Theta_1,\hat\Theta_2\}) + \\
&J(\{\hat\Theta_1,\hat\Theta_2\}) - J(\hat\Theta_1) + J(\hat\Theta_1) = \\
&J(\{\hat\Theta_1,\cdots,\hat\Theta_{N-1},\bar\Theta_N\}) + J(\{\hat\Theta_1,\cdots,\hat\Theta_N\}) \\
&J(\{\hat\Theta_1,\cdots,\hat\Theta_{N-2},\bar\Theta_{N-1}\}) - J(\{\hat\Theta_1,\cdots,\hat\Theta_{N-1}\}) \\
&\cdots + J(\{\hat\Theta_1,\bar\Theta_2\}) - J(\{\hat\Theta_1,\hat\Theta_2\}) - J(\hat\Theta_1) + J(\bar\Theta_1).
\end{aligned}$$

In addition, the suboptimal path $\{\hat\Theta_1,\hat\Theta_2,\cdots,\hat\Theta_N\}$ is obtained by the auction scheme. The order of all vehicles in the team can be arranged in the way that the i^{th} vehicle is the vehicle to which the planning algorithm assigns the path at the i^{th} step. Accordingly, we have $J(\bar\Theta_1) \le J(\hat\Theta_1)$, $J(\hat\Theta_1,\bar\Theta_2) = J(\bar\Theta_2|\hat\Theta_1) \le J(\hat\Theta_2|\hat\Theta_1) = J(\hat\Theta_1,\hat\Theta_2)$, $\cdots$, $J(\{\hat\Theta_1,\cdots,\hat\Theta_{N-2},\bar\Theta_{N-1}\}) = J(\bar\Theta_{N-1}|\hat\Theta_1,\cdots,\hat\Theta_{N-2}) \le J(\hat\Theta_{N-1}|\hat\Theta_1,\cdots,\hat\Theta_{N-2}) = J(\{\hat\Theta_1,\cdots,\hat\Theta_{N-1}\})$, and $J(\{\hat\Theta_1,\cdots,\hat\Theta_{N-1},\bar\Theta_N\}) = J(\bar\Theta_N|\hat\Theta_1,\cdots,\hat\Theta_{N-1}) \le J(\hat\Theta_N|\hat\Theta_1,\cdots,\hat\Theta_{N-1}) = J(\{\hat\Theta_1,\cdots,\hat\Theta_N\})$. Therefore, we have that $J(\{\bar\Theta_1,\cdots,\bar\Theta_N\}) \le J(\{\hat\Theta_1,\cdots,\hat\Theta_N,\bar\Theta_1,\cdots,\bar\Theta_N\}) \le 2J(\{\hat\Theta_1,\hat\Theta_2,\cdots,\hat\Theta_N\})$, which completes the proof. ■

V. Numerical results

The algorithm was tested using a simulation environment developed in MATLAB/SIMULINK, snapshots of which are shown in Fig. 6, where green blocks represent the buildings. A single UAV and UGV were used to track a target cooperatively. A 49×49 occupancy grid, where the size of each cell is 5m×5m, was used to model the target state. The simulator uses six state navigation equations for the aircraft and uses four state navigation equations for the ground vehicle. Three-step look-ahead horizon paths were planned for the UAV and the UGV with the sample interval $\sigma = 2$s. The set of roll angles for the UAV was $\{-30^\circ,-15^\circ,0^\circ,15^\circ,30^\circ\}$. The camera mounted on the UAV was assumed to be gimballed



(a) t=8s

(b) t=35s

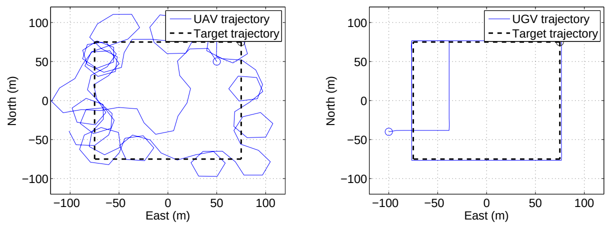
(c) t=40s

Fig. 6. The snapshots of target occupancy grid and the paths of the UAV and the UGV for the first type of target motion.

so that it was always pointed down with a field of view of 40°. An omnidirectional camera was used for the UGV and the area it can observe was a square of 30m ×30m. The covariance of the detection probability was $\Sigma = 20$. The parameter P_c was set at 0.9. In the simulation, the UAV flew at an altitude of 120m. The speed for the UAV, UGV, and the target was set to 10 m/s, 4 m/s, and 6 m/s respectively.

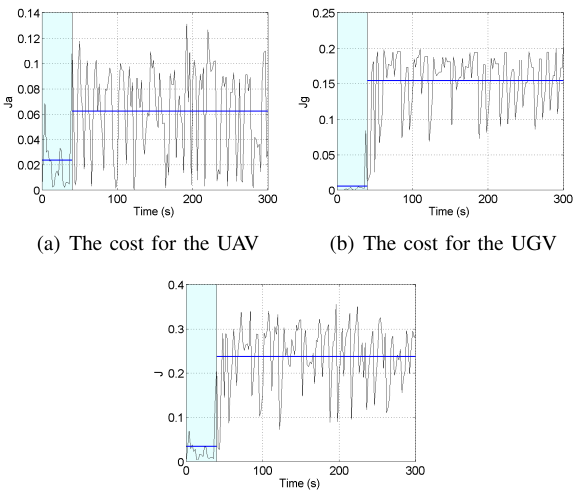
We evaluate the performance of the target tracking algorithm for two types of target motion. For the first type of target motion, the target is initially placed at North-East coordinate (75m,75m) and it will move among the waypoints (75m,75m), (75m,-75m), (-75m,-75m) and (-75m,75m) in turn. The motion model of the target is not known by the UAV and the UGV.

Figure 6 shows the snapshots of the target occupancy grid and the paths of the UAV and the UGV for cooperatively tracking the target at different time steps for a 300 second simulation run. There exist 36 buildings in the environment, each of which is 40m high. The algorithm assumes the target is initially located at the origin. It then updates the target occupancy grid using the Bayesian filter described in Section II and plans the corresponding paths such that the joint cost function is maximized, using Algorithm 1. By doing so, the UAV and the UGV can eventually detect the target at time $t = 40$s, as shown in Fig. 6(c).



(a) UAV target tracking (b) UGV target tracking

Fig. 7. The trajectories of the target, the UAV and the UGV for the first type of target motion.



(a) The cost for the UAV (b) The cost for the UGV

(c) The joint cost

Fig. 8. The cost associated with the UAV, the UGV and their cooperation.

Figure 7 shows the trajectories of the target, the UAV, and the UGV. Figure 8 shows the cost associated with the UAV, the UGV, and their cooperation. Based on the figure, the cost associated with the UAV shows more oscillations than the cost associated with the UGV. This is due to the fact that the UAV must always fly along its orbits and cannot stop. The algorithm takes around 40s to detect the target, as shown in the green area. The UAV and the UGV then start to track the target. The straight horizontal lines in the figure show the mean of the cost associated with the UAV, the UGV, and their cooperation before and after the first detection.

Figure 9 shows the average time percentage that the target was observed by at least one vehicle (in the field of view of at least one camera and unoccluded) over 100 simulation runs versus time, where the time percentage is evaluated per 20 seconds. Each simulation lasts 1000 seconds and different initial positions of the UAV and the UGV were chosen based on a uniform distribution. From Figure 9, we can see that the average time percentage increases as time progresses.

For the second type of target motion, the target starts at a random initial location on the road network and then moves along the roads. When the target reaches an intersection, it will turn left, turn right, or go straight based on a uniform distribution. Figure 10 shows the snapshots of the target

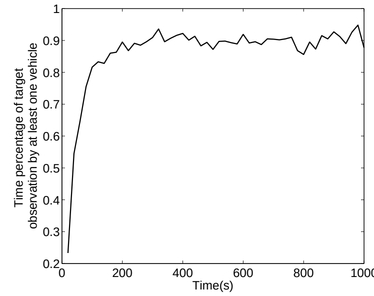
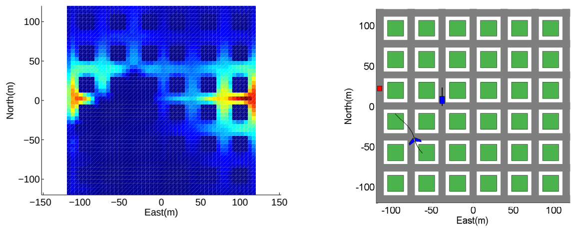


Fig. 9. The average time percentage that the target was observed by at least one vehicle over 100 simulation runs for the first type of target motion.

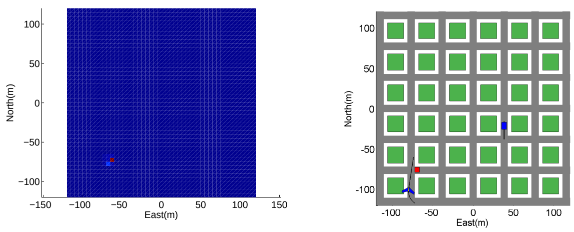
occupancy grid and the paths for the UAV and the UGV for cooperatively tracking the target at different time steps for a 500 second simulation run. Using the planning algorithm, the UAV and the UGV search the target and detect it at time $t = 80$s, as shown in Fig. 10 (b). Figure 11 shows the trajectories of the UAV, the UGV, and the target. We also evaluate the average time percentage that the target was observed by at least one vehicle over 100 simulation runs, as shown in Fig. 12. Each simulation last 1000 seconds and different initial positions of the UAV, the UGV, and the target were chosen based on a uniform distribution.

To show the performance of the algorithm in different environments, we also implemented the algorithm in the environments with varying building height and density. For each environment, we executed 100 simulation runs. Each simulation run lasted 300s and different initial positions of the UAV and the UGV were chosen based on a uniform distribution. The target was set to move based on the first type of target motion model. We evaluated two criteria: (a) the search time for the first detection denoted by T_s, and (b) the time percentage of target loss by both the UAV and the UGV (out of the field of view of both cameras or occluded by buildings) denoted by ρ.

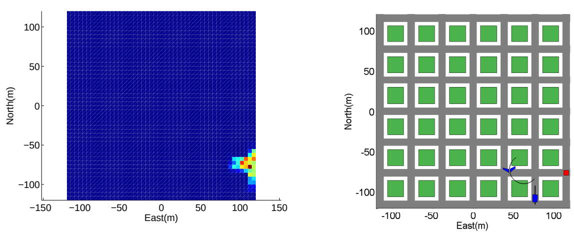
Figure 13 shows the change of average values of T_s and ρ over 100 simulations versus the height of the buildings for the environment where 36 buildings exist. Figure 14 shows the change of average values of T_s and ρ versus the density of the environment where the building height is 100m. In the simulation, we consider the environment where at most 36 buildings exist. The density of the environment is assumed to be the ratio of the number of existing buildings over 36. Based on Fig. 13 and Fig. 14, as the building height and the density of the environment increase, the planning algorithm takes longer time to search the target and the time percentage of target loss by both the UAV and the UGV increases. The results of the Monte Carlo simulations shown by Fig. 13 and Fig. 14 provide insight into the applicability of the planning algorithm for addressing the target tracking problem in an environment with a certain degree of occlusions, although the tracking performance of the planning algorithm degrades as more occlusions occur.



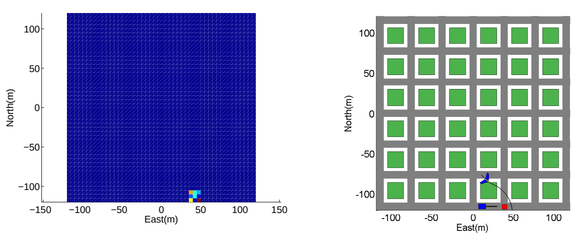
(a) t=30s



(b) t=80s

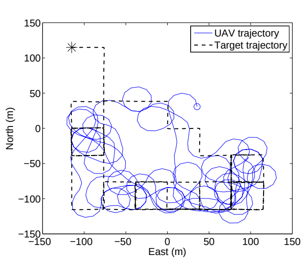


(c) t=310s

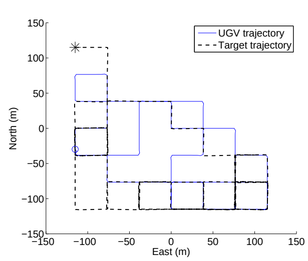


(d) t=480s

Fig. 10. The snapshots of target occupancy grid and the paths of the UAV and the UGV for the second type of target motion.



(a) Building Height vs. T_s



(b) Building Height vs. ρ

Fig. 11. The trajectories of the target, the UAV, and the UGV for the second type of target motion.

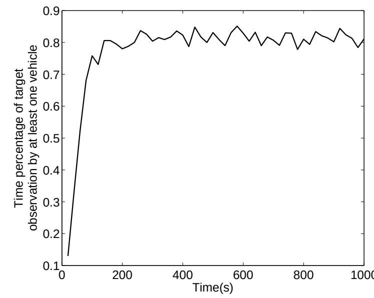
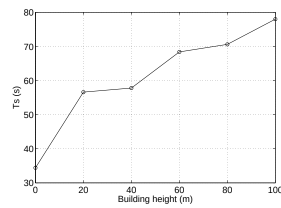
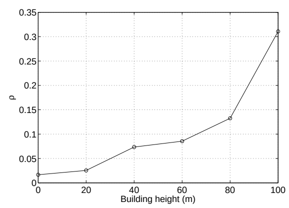


Fig. 12. The average time percentage that the target was observed by at least one vehicle over 100 simulation runs for the second type of target motion.

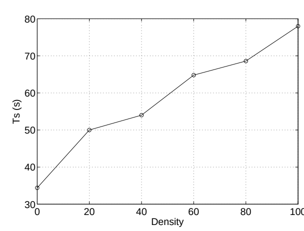


(a) Height vs. T_s

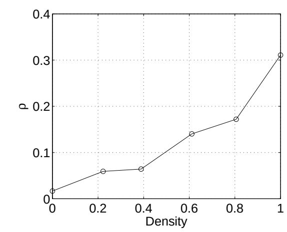


(b) Height vs. ρ

Fig. 13. The change of T_s and ρ versus the building height.



(a) Density vs. T_s



(b) Density vs. ρ

Fig. 14. The change of T_s and ρ versus the density of the environment.

VI. Conclusions

We have presented a path planning algorithm for tracking a moving target in urban environments using both UAVs and UGVs. The algorithm takes into account occlusions between the sensors and the target. We use a dynamic occupancy grid to model the target state and use a Bayesian filter to update the probability of the target location. For target tracking by a single vehicle, we design a path planning algorithm to generate paths maximizing the sum of probability of detection over a finite look-ahead horizon. For target tracking using multiple vehicles, we designed a decentralized path planning algorithm relying on an auction scheme to generate paths maximizing the sum of the joint probability of detection over the finite horizon. The numerical results demonstrate the feasibility of the path planning algorithm for addressing the target tracking problem in urban environments.

VII. Acknowledgement

This research was supported by OSD and AFRL under contract FA8650-08-C-1411 with BYU serving as a subcon-

tractor to SET Corporation.

REFERENCES

[1] J. Kim and Y. Kim, "Moving target tracking in dense obstacle areas using UAVs," in *Proc. 17th IFAC World Congress, Seoul, Korea*, 2008.

[2] Z. Tang and U. Ozguner, "Sensor fusion for target track maintenance with multiple UAVs based on Bayesian filtering method and hospitability map," in *Proc. 42nd IEEE Conference on Decision and Control*, vol. 1, 9–12 Dec. 2003, pp. 19–24.

[3] U. Zengin and A. Dogan, "Real-time target tracking for autonomous UAVs in adversarial environments: A gradient search algorithm," *IEEE Transactions on Robotics*, vol. 23, no. 2, pp. 294–307, April 2007.

[4] S. Kanchanavally, R. Ordonez, and J. Layne, "Mobile target tracking by networked uninhabited autonomous vehicles via hospitability maps," in *Proc. American Control Conference*, vol. 6, 2004, pp. 5570–5575.

[5] B. Grocholsky, R. Swaminathan, J. Keller, V. Kumar, and G. Pappas, "Information driven coordinated air-ground proactive sensing," in *Proc. IEEE International Conference on Robotics and Automation ICRA 2005*, 18–22 April 2005, pp. 2211–2216.

[6] B. Grocholsky, J. Keller, V. Kumar, and G. Pappas, "Cooperative air and ground surveillance," *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, pp. 16–25, Sept. 2006.

[7] R. Vidal, O. Shakernia, H. J. Kim, D. H. Shim, and S. Sastry, "Probabilistic pursuit-evasion games: theory, implementation, and experimental evaluation," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 662–669, Oct. 2002.

[8] H. G. Tanner, "Switched UAV-UGV cooperation scheme for target detection," in *Proc. IEEE International Conference on Robotics and Automation*, 10–14 April 2007, pp. 3457–3462.

[9] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. The M.I.T. Press: Cambridge, Massachusetts, 2005.

[10] F. L. Lewis, *Optimal Control*. New York: John Wiley & Sons, 1986.

[11] J. Tisdale, Z. Kim, and K. Hedric, "Autonomous UAV path planning and estimation," *IEEE Robotics and Automation Magazine*, vol. 16, no. 2, pp. 35–42, June 2009.