Multi-target Tracking and Data Association on Road Networks Using Unmanned Aerial Vehicles

Brett E. Barkley, Derek A. Paley
Department of Aerospace Engineering and Institute for Systems Research
University of Maryland
College Park, Maryland 20742

Abstract— **A cooperative search and track algorithm for surveilling multiple road vehicles is presented for fixed-wing Unmanned Air Vehicles (UAVs) with a finite field of view. The road network is formed into a graph with nodes that indicate the target likelihood ratio (before detection) and position probability (after detection). Target measurement data is associated to either the likelihood ratio tracker or a Bayesian target tracker. Data association uses a similarity score generated by finding the earth mover's distance between the measurement and track probabilities. Two strategies for motion-planning of UAVs balance searching for new targets and tracking known targets. The first strategy is to loiter over the peak track probability to maximize information about a known target. The second strategy is to continue searching for new targets, returning to known targets only when the peak track probability becomes low. Results from numerical simulations are included to illustrate the performance of the algorithm and to quantify algorithm performance under the influence of added uncertainty in the detection and measurement of targets.**

TABLE OF CONTENTS

1. INTRODUCTION

Having accurate and up-to-date data from intelligence, surveillance, and reconnaissance missions has become an essential part of how the modern tactician develops strategy. As a result, the US government has released the Unmanned Systems Roadmap 2007–2032 [1] citing the specific need for target identification and designation in the realm of UAV reconnaissance. This paper's goal is to extend work on target detection on road networks using a Bayesian likelihood tracker [2], [3] to the dual problem of searching for and tracking targets after detection. Target location on a road network is represented as a probability density. Measurement data is associated using an earth mover's distance [4] similarity metric to determine whether measurements correspond to previously detected targets or new undetected targets. The UAVs search the road network using one of two control algorithms that balance tracker accuracy and target detection.

Holding such an important role in modern surveillance operations, the problem of multi-target tracking is a deep field. If no constraints are placed on the targets, tracking algorithms occupy the realm of interacting multiple model (IMM) filters described in [5], [6], [7] with Kalman filters, extended Kalman filters, and even particle filters used for linear and nonlinear target dynamics models. By applying the constraint that targets remain on a road network, simplified and less computationally costly IMM estimators can be applied to predict target motion, such as the Variable Structure IMM (V-S IMM) [8], [9], which keeps modes in use only as needed. IMM estimators based on particle filters have had success in estimating target dynamics, as in [10], [11], [12], but they run into the issue of sharp mode transitions, leading to varying levels of tracking failure [13]. By fixing the number of particles per mode, as in [6], [14] some of the errors induced by sharp mode transitions can be avoided and quicker model adaptation can be achieved. However, the robustness and tradeoff in errors and adaptation still need to be investigated further [13].

For the problem of data association among multiple measurements and trackers, a number of solutions have been developed with varying levels of success, including particle filtering [15], dynamic programming [16], [17], and maximum likelihood [18], [19] methods. However, due to the process by which they determine associations, these methods admit a high proportion of false alarms in their measurements, typically are computationally expensive, and do not pair tracking with path planning for UAVs [20].

This paper focuses on a Bayesian filtering framework to enable cooperative search and track of detected mobile targets on a road network using UAVs with a finite field of view. UAV sensor platforms cooperatively search along the road network and update the likelihood surface that represents likely target locations based on a recursive Bayesian likelihood ratio tracker [3]. Once the likelihood on the network surpasses a critical threshold, a target detection is called and a tracker is initialized. Measurement updates are provided by data association; the existing trackers and new measurements are compared using the earth mover's distance similarity metric [4].

Simulated targets are constrained to remain on the road network at all times and stop and start randomly to mimic courier behavior. The UAV sensors are characterized by probability of detection and probability of false alarm, and by the standard deviation of target location measurements. Each UAV is modeled as a Dubins car with constrained turning rate and speed. The standard deviation and probability of detection and false alarm of the onboard sensors are linked to the ground sampling distance (GSD) of the UAVs [21]. As altitude increases, GSD increases and measurement resolution decreases, leading to decreased probability of detection,

increased false alarms, and higher standard deviation in the measurement uncertainty.

For each target detected, a new Bayesian target tracker is instantiated on the road network. The Bayesian filtering methodology recursively updates the tracker with prediction and update steps. The prediction step updates the probability surface using random walk motion at the nominal speed of the targets on the road network. The update step uses planar measurements of the target to update the probability distribution. New measurements are associated by comparison to the prior distribution in the tracker using the earth mover's distance. When the distance is sufficiently low, the measurement and distribution are paired and the distribution is updated; otherwise the measurement is used to update the likelihood network.

Each UAV is guided to network nodes of higher likelihood using a set of artificial potentials. These artificial potentials cause each UAV to ascend the likelihood gradient and avoid collisions with other UAVs. Reflecting the constraint on target motion, the gradient ascension force is parallel to the edge of maximum likelihood change. A second force, known as Pauli repulsion, is applied between UAVs to prevent collisions and redundant searching. The third force is a artificial spring connecting each UAV to the edge of maximum likelihood change in field of view, which prevents excessive drift off of the network as the UAV ascends the gradient.

Two motion-planning algorithms guide UAVs to balance their search and track functions. The first strategy is to loiter on the peak probability in the nearest distribution and keep the target location estimate as accurate as possible. The second strategy is to have each UAV search for new targets and only revisit a tracker probability distribution when its peak drops below a threshold. Both strategies were tested in Monte Carlo simulations. Comparing the two algorithms, the loiter algorithm is better at tracking targets on the road network, but the difference in tracking performance between the two algorithms decreases as the search radius of UAVs increases.

The first contribution of this paper is using the structure of the road network to encode tracker probability and perform data association. By utilizing graph theory and the graph Laplacian, motion updates are modeled as a random walk on the road network. Secondly, Djikstra's algorithm is used to provide pairwise distance costs between all of the nodes that comprise the road network. This concept combined with a similarity threshold and the usage of probability networks on the road for both the target measurements and target trackers provides a simple way to determine how closely the two distributions align and whether the measurement is an appropriate match for the tracker. The third contribution is the cooperative search and track algorithms that balance finding new targets and keeping trackers active and accurate.

The paper is organized into the following sections. Section II summarizes the fundamentals of graph theory and the track-before-detect algorithm. Section III explains how the target tracking probability networks are constructed, as well as how measurement data is associated using a similarity metric. Section IV presents motion-planning strategies for balancing search and track, as well as simulations of both algorithms to compare performance. Section V summarizes the paper and provides an overview of ongoing work.

2. BAYESIAN TARGET DETECTION ON A GRAPH

Graph Theory and the Graph Laplacian

A graph is a structure in mathematics that is used to describe the relation between pairs of objects. A road graph is composed of two elements [22]: a set V of N vertices and a set E of M edges. Let $\psi : V \to \mathbb{R}^2$ return the planar coordinates of a vertex. A directed graph is described by the adjacency, $A \in \mathbb{R}^{N\times N}$, and degree, $D \in \mathbb{R}^{N\times N}$, matrices. The ij entry of the adjacency matrix represents the connectivity of nodes i and j, i.e.,

$$a_{ij} = \begin{cases} 0, & \text{if } j = i \\ 1, & \text{if there is a directed edge from j to i} \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

For an undirected graph, the adjacency matrix is symmetric about the diagonal [22]. The diagonal entries of the degree matrix D give the number of incoming connections to the corresponding node, whereas the off-diagonal entries are zero:

$$d_{ij} = \begin{cases} \sum_{j=1}^{N} a_{ij}, & \text{if } i = j \\ 0, & \text{if } i \neq j. \end{cases} \quad (2)$$

Another convenient construct in graph theory is the incidence matrix, $B \in \mathbb{R}^{N\times M}$, whose row indices represent nodes and column indices represent edges [22]. For each column, there are precisely two non-zero entries that sum to zero, indicating that exactly two nodes are connected by a single edge. (Assume no self loops.) Finally, the Laplacian matrix, $L \in \mathbb{R}^{N\times N}$, of a graph is

$$L = D - A. \quad (3)$$

The graph Laplacian matrix of a connected, undirected, graph is positive semi-definite [23].

The graph Laplacian is used throughout as an operator that describes a random walk (diffusion) between neighboring nodes. The rate of diffusion is determined by the target speed to provide a realistic spreading of target likelihood (or probability) along the network in time. Since targets may travel only along defined edges, as governed by the connectivity of the adjacency matrix, the spatial rate of change of likelihood for a vertex in the likelihood network is produced using partial derivatives along each connected edge [24]. Let $\phi \in \mathbb{R}^N$ be the likelihood over all N nodes. One way to model this exchange is with the heat equation. Assuming that the likelihood exchange rate is a constant α, the time rate of change of likelihood can be modeled in matrix vector notation as

$$\frac{d\phi}{dt} + \alpha L\phi = 0, \quad (4)$$

which is the heat equation with spatial discretization [23]. Diffusion of likelihood throughout the road network is found for discrete time steps $k = 1, 2, 3, ...$ of size Δt by solving the first-order matrix differential equation in (4) to obtain

$$\phi_k = e^{-\alpha L\Delta t}\phi_{k-1}. \quad (5)$$

Bayesian Filtering for Target Detection and Tracking

A Bayes filter is a discrete probabilistic process that recursively takes noisy measurements of a target's true state and converts them into a probability density function using a mathematical model of the target dynamics [25]. These results are used to predict and update the state space of the target. Let $\xi_k = (x_k, y_k)$ denote the target state at time step k and ζ_k denote an observation of the target at k. The predict step involves computing the conditional probability [25]

$$p(\xi_k|\zeta_{k-1}) = \int_\Omega p(\xi_k|\xi_{k-1})p(\xi_{k-1}|\zeta_{k-1})d\xi_{k-1}. \quad (6)$$

The measurement update is proportional to the product of the measurement likelihood $p(\zeta_k|\xi_k)$ and the prior probability (6), i.e.,

$$p(\xi_k|\zeta_k) = \frac{p(\zeta_k|\xi_k)p(\xi_k|\zeta_{k-1})}{p(\zeta_k|\zeta_{k-1})}. \quad (7)$$

In a Bayesian framework, simultaneous observations from multiple sensors are assimilated by executing consecutive measurement updates [25].

Similar to the standard Bayes filter, a log-likelihood ratio is used for detecting targets. This strategy is often called track-before-detect, because sensor data is acquired for possible targets before they are detected. In previous work [3], a log-likelihood ratio tracker was used to record instantaneous positions of undetected targets and to spawn target-tracking filters to track targets after detection. In a likelihood ratio tracker, the measurement probability in equation (7) is replaced with the measurement likelihood ratio [2]. The log-likelihood ratio is

$$\log \mathcal{L}(\zeta_k|\xi_k) = \log \frac{p(\zeta_k|\xi_k^+)}{p(\zeta_k|\xi_k^-)}. \quad (8)$$

Let $\mathrm{p} = \log(p)$. The update step in the log-likelihood ratio tracker becomes

$$\mathrm{p}(\xi_k|\zeta_k) = \log \frac{\mathcal{L}(\zeta_k|\xi_k)p(\xi_k|\zeta_{k-1})}{p(\zeta_k|\zeta_{k-1})} = \mathrm{p}(\zeta_k|\xi_k^+) - \mathrm{p}(\zeta_k|\xi_k^-) + \mathrm{p}(\xi_k|\zeta_{k-1}) - \mathrm{p}(\zeta_k|\zeta_{k-1}). \quad (9)$$

The first and second terms in (9) represent the newly measured positive and negative information that a target is present, respectively. The third term represents the prior information about the target, and the fourth term is a normalization constant that may be ignored if unknown.

The predict step involves updating the target probability density (or likelihood ratio) in the absence of measurement information. The graph representing the road network allows us to impose restrictions on target motion. A random-walk model describes the target motion according to the diffusivity $\alpha = \frac{\Delta t}{\Delta x} V_{max}$, where the time step of the simulation is Δt, the average node spacing is Δx, and maximum speed is V_{max}.

UAV Dynamics

The UAV dynamics are modeled using a Dubins car framework [26]. Let S_k^j be the (constant) speed at which UAV j is moving, θ_k^j be its heading, and u_k^j be the control input to the turn rate at time k. Also $\Theta_k^j = [f_k^j, g_k^j]$, $j = 1, ..., O$ represents the planar location of UAV j at time k. The unconstrained kinematics of UAV j are

$$\begin{aligned} \dot{f}_k^j &= S_k^j \cos\theta_k^j \\ \dot{g}_k^j &= S_k^j \sin\theta_k^j \\ \dot{\theta}_k^j &= u_k^j. \end{aligned} \quad (10)$$

The constraints on turn rate and speed are enforced using the saturation function. By taking derivatives of the $\dot{f}_k^j$ and $\dot{g}_k^j$ terms, using Euler's method [27], and applying saturation, the dynamics of the UAVs are determined by the forces, X_k^j and Y_k^j, along the f and g planar directions as follows:

$$\begin{aligned} \theta_k^j &= \theta_{k-1}^j + \mathrm{sat}\left(\frac{Y_k^j \cos\theta_k^j - X_k^j \sin\theta_k^j}{S_k^j}, \dot{\theta}_{max}\right)\Delta t \\ S_k^j &= \mathrm{sat}\left(S_{k-1}^j + (X_k^j \cos\theta_k^j + Y_k^j \sin\theta_k^j)\Delta t, S_{max}\right), \end{aligned} \quad (11)$$

where $\dot{\theta}_{max}$ and S_{max} are the maximum turn rate and speed of the UAV, respectively.

LRT Gradient-Search Algorithm

Each UAV's motion plan is prescribed by a combination of three artificial potentials to guide it up the gradient in likelihood ratio while simultaneously avoiding collisions. Assume the likelihood surface and geometry of the road network itself are known to all UAVs, as well as the location of every other UAV. (Distributed data fusion is outside the scope of this paper.)

The first force is derived from the maximum gradient of the log-likelihood graph in the UAV's field of view. Let μ_k^j represent the index of the edge with maximum likelihood change in search range of UAV j and $\Delta\phi_k$ describe the edge-wise likelihood differences at time k, i.e., [3]

$$\Delta\phi_k = B^T \phi_k. \quad (12)$$

Let the maximum difference in likelihood along an edge in search range of UAV j be $\Delta\phi_k(\mu_k^j)$. If the maximum likelihood change is contained in multiple edges, a single edge is chosen randomly. Since μ_k^j represents only the edge index of the largest difference in likelihood, the direction of the gradient along the edge is also needed. Recall that ψ returns the position of a node in the network; let n_1 and n_2 be the head and tail, respectively, of the edge in row μ_k^j of B^T. The likelihood gradient ∇R_k^j is thus

$$\nabla R_k^j = \Delta\phi_k(\mu_k^j)\frac{\psi(n_1) - \psi(n_2)}{\left\|\psi(n_1) - \psi(n_2)\right\|}. \quad (13)$$

The second artificial force is the gradient of the repulsive portion of the Lennard-Jones potential [28], known as Pauli repulsion. Let $\epsilon = 1$ be the depth of the well and $\sigma = 2\rho$ the distance at which the potential between two UAVs is zero. The gradient of the Lennard-Jones potential is

$$\nabla P_k^j = -48 \sum_{i \neq j}^{O} \left(\sigma^{12} ||\Theta_k^j - \Theta_k^i||^{-13} \right) \frac{\Theta_k^j - \Theta_k^i}{||\Theta_k^j - \Theta_k^i||}, \quad (14)$$

where repulsion occurs only when UAVs have overlapping search radii.

The third force is a spring force connecting the UAV to the node of highest likelihood along the edge of maximum gradient, i.e., $n_{max} \in \{n_1, n_2\}$ such that $\phi_k(n_{max})$ is greatest. If the node of interest is out of sensor range, the spring force brings the UAV closer to the nodes of interest. The spring force is

$$\nabla Q_k^j = -K \left(||\Theta_k^j - \psi(n_{max})|| - \rho \right) \frac{\Theta_k^j - \psi(n_{max})}{||\Theta_k^j - \psi(n_{max})||}, \quad (15)$$

where K is the spring constant and ρ is the rest length.

The total control force applied to UAV j is

$$F_k^j = [X_k^j, Y_k^j] = \nabla R_k^j + \nabla P_k^j + \nabla Q_k^j, \quad (16)$$

where the components are scaled as above.

3. Target Data Association Governed by Earth Mover's Distance

Track-after-detect filters are initialized for each detected target when the likelihood network reaches a critical threshold ϕ_{max}. The new tracker is initially populated with a univariate normal distribution based on measurements of the target position. Having trackers and the LRT functioning at the same time necessitates a procedure to determine whether subsequent measurements from the UAVs should be used to update the LRT or a particular target's tracker. This procedure is called data association. This section explores the procedure used to generate probability measurements from target detections and how to properly associate new measurements using the earth mover's distance metric.

Target Detection and Tracker Instantiation

Consider a measurement data model based on an imperfect sensor with a finite range of view. Let targets within the sensor range ρ be detected with probability P_d and false-alarm rate of P_f per time step [29]. Combining the two probabilities, the sensitivity m of each sensor is [29]

$$m = z(P_d) - z(P_f), \quad (17)$$

where $z(\cdot)$ represents the z-transformation into standard deviation units given by the quantile function [29]

$$z(p) = \sqrt{2}\text{erf}^{-1}(2p - 1).$$

Let w_k represent unit-normal measurement noise in standard deviation units at time step k. Without a target, the measurement is $\zeta_k = w_k$, whereas with a target the measurement is $\zeta_k = m + w_k$. Assuming a zero-mean Gaussian sensor model [30] transforms the log-likelihood ratio (8) into

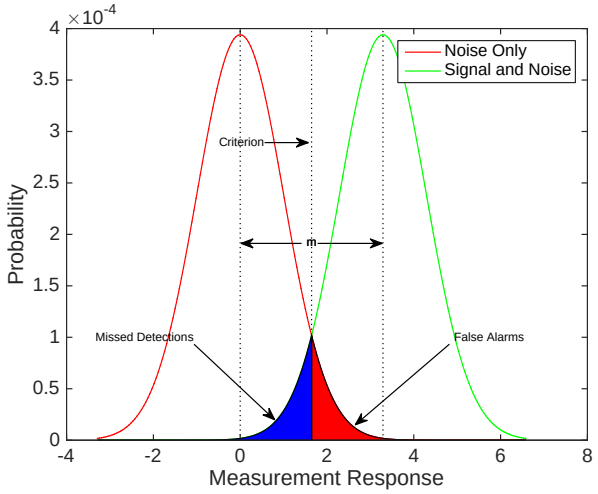


Figure 1: Probability of Occurrence plot with $P_d = 0.95$ and $P_f = 0.05$

$$\log \mathcal{L}(\zeta_k | \xi_k) = -\frac{(\zeta_k - m)^2}{2} + \frac{\zeta_k^2}{2} = m \left(\zeta_k - \frac{m}{2} \right).$$

Note, the log-likelihood ratio is only applied to the prior likelihood inside a disc of radius ρ centered on the UAV location. Targets in sensor range are declared detected and measurements of the target locations are generated when the target likelihood on the LRT surface reaches ϕ_{max}. The likelihood inside of ρ is subsequently suppressed to zero.

Target measurements include location. Recalling that $\xi_k = (x_k, y_k)$ represents the actual location of a target, the measurement of the target location by UAV j at time step k is

$$\tilde{\xi}_k^j = \xi_k + \nu(0, s),$$

where ν is Gaussian measurement noise with zero mean and standard deviation, s. (The location of a measurement in the absence of a target (i.e., a false alarm), is generated randomly from a uniform distribution centered on the UAV.)

To prevent superfluous target measurements from entering the data association process, we introduce a criterion of $c = m/2$, where m is given by equation (17), as shown on the probability of occurrence graph in Figure 1. This choice represents the intersection of the probability of occurrence curves for noisy measurements with and without signals.

After a detection, the initial track probability is formed using the measurement of the target location $\tilde{\xi}_k^j$ and the locations ψ of the nodes that compose the road network. Recall that s is the standard deviation of the position measurement noise. The measurement probability density gathered by UAV j at time k is thus

$$U_k^j = \frac{1}{s\sqrt{2\pi}} e^{-\frac{(\tilde{\xi}_k^j - \psi(V))^2}{2s^2}}.$$

Measurement Data Association and Measurement Update

Earth mover's distance (EMD) is a solution to the transportation problem introduced by Rubner, Tomasi, and Guibas [4]. Comparing two piles (signatures) can be effectively performed by finding how much dirt (probability) must be moved from one pile to the other until they are of identical height. Specifically, the EMD represents the minimum cost required to transform one signature into another, where a unit cost is moving one unit of probability by one unit of distance. In this case the signatures are the probability densities of the instantiated trackers, P_k^l, and the measurements produced by UAV j at time k, $\hat{U}_k^j$.

A linear programming problem can formalized for the case of moving probability on a road network as follows. Let P be the first signature with q elements indexed by i and Q be the second signature with r elements indexed by g [4]. The ground distance metric between the elements of P and Q is represented by the matrix $\mathbb{D} = [\mathbb{d}_{ig}]$ and is obtained by applying Dijkstra's algorithm in a pairwise fashion between the two signature's elements along the road network graph. The overall cost of work is [4]

$$WORK(P, Q, F) = \sum_{i=1}^{q} \sum_{g=1}^{r} \mathbb{d}_{ig} f_{ig}. \tag{18}$$

The flow $F = [f_{ig}]$ that minimizes (18) can be found when subject to the following constraints for probability distributions with equal total probabilities [31]

$$f_{ig} \geq 0 \qquad 1 \leq i \leq q, 1 \leq i \leq r \tag{19}$$

$$\sum_{g=1}^{r} f_{ig} = P(i) \qquad 1 \leq i \leq q \tag{20}$$

$$\sum_{i=1}^{q} f_{ig} = Q(g) \qquad 1 \leq g \leq r \tag{21}$$

$$\sum_{i=1}^{q} \sum_{g=1}^{r} f_{ig} = \sum_{i=1}^{q} P(i) = \sum_{g=1}^{r} Q(g) = 1. \tag{22}$$

Constraint (19) requires supplies transferred from P to $Q(g)$ to be nonnegative. Constraint (20) ensures that the probability matched to Q is equal to the probability in $P(i)$. Similarly, constraint (21) ensures that the probability matched to P is equal to the probability in $Q(g)$; and constraint (24) requires that the signature with the most probability be moved, which is known as the total flow [4]. In this case, both signatures are normalized, so the total flow is one. With an F that minimizes the overall cost of the signature transformation, the earth mover's distance is [4], [32]

$$EMD\,(P, Q) = \frac{\sum_{i=1}^{q} \sum_{g=1}^{r} \mathbb{d}_{ig} f_{ig}}{\sum_{i=1}^{q} \sum_{g=1}^{r} f_{ig}} = \sum_{i=1}^{q} \sum_{g=1}^{r} \mathbb{d}_{ig} f_{ig}.$$

Let $\mathcal{I}_k$ denote the total number of trackers running at time k and $\mathcal{T}_k^j$ denote the total number of measurements at time step k produced by UAV j. Given the EMD between the measurement and tracker distributions, an $\mathbb{R}^{\mathcal{I}_k \times \mathcal{T}_k^j}$ data association matrix DA can be formed to succinctly compile all of the EMD costs (see Algorithm 1).

The approach for associating measurements to trackers requires iteratively finding the row (tracker) and column (UAV measurement) in matrix DA corresponding to the minimum EMD cost. Let $\mathcal{E}$ represent the maximum EMD outcome that would be considered an association between a measurement and tracker. If the minimum EMD cost in DA is below $\mathcal{E}$, then the measurement and tracker associated with that cost are associated with one another. The posterior of the tracker $\mathbb{P}_k^l$ is updated as described in Algorithm 1 (see line 11).

Next, other EMDs generated using the newly associated tracker and measurement are removed from DA and a new search for the minimum EMD cost is started. If no minimum EMD can be found, then any additional unassociated measurements are used to update the LRT surface according to the procedure described in Section 2. This procedure is repeated on each timestep for each UAV.

Algorithm 1 EMD Data Association

Require: $\mathcal{I}_k, \mathcal{T}_k^j, \hat{U}_k^j, P_k^l, O, \mathcal{E}$

1: ▷ Where $\mathcal{I}_k$ is the total number of instantiated trackers, $\mathcal{T}_k^j$ the total number of measurements, $\hat{U}_k^j$ the measurement probability, P_k^l the tracker probability, O the number of UAVs, and $\mathcal{E}$ the EMD threshold
2: Repeat for each timestep k
3: **for** $j = 1 : O$ **do**
4: **for** $l = 1 : \mathcal{I}_k$ **do**
5: **for** $q = 1 : \mathcal{T}_k^j$ **do**
6: $DA(l, q) = EMD\left(P_k^l, \hat{U}_k^j(:, q)\right)$
7: **end for**
8: **end for**
9: Note the index l, q of min(EMD) in DA
10: **while** $DA(\min(EMD)) < \mathcal{E}$ **do**
11: Generate posterior: $\mathbb{P}_k^l = P_k^l \times \hat{U}_k^j(:, q)$
12: Set the elements $(:, q)$ and $(l, :)$ in DA to $\mathcal{E} + 1$
13: Note the new index l, q of min(EMD) in DA
14: **end while**
15: Note the number of unused P_k^l
16: **if** number of unused $P_k^l > 0$ **then**
17: Update LRT with measurement
18: **end if**
19: **end for**

The posteriors achieved after all of the measurements have been associated appropriately are diffused as in (5) and formed into the prior for the next time step i.e.,

$$P_{k+1}^l = e^{-\alpha L \Delta t} \mathbb{P}_k^l. \tag{19}$$

Let P_{min} be the minimum probability threshold before a tracker instantiation is dissolved. Prior to the beginning of the next time step $k+1$, the probability contained in each prior P^l_{k+1} is evaluated to determine if the maximum probability has dropped below P_{min}. If this situation occurs, tracker l is dissolved and the track probability becomes zero.

4. COORDINATED CONTROL OF TARGET DETECTION AND TRACKING

Tracking Preliminaries

Both tracking algorithms in this section require that UAV measurements are assigned to a tracker. In each tracker, the location of maximum probability indicates the most likely position of the target. As a result, without perfect knowledge of target location, the UAV assigned to provide measurement updates to the tracker has the highest chance of providing updated data if the location of highest probability is in sensor range. The assignment is performed by the following protocol for linking UAVs to trackers.

At any given time, a varying number $\mathcal{I}_k$ of trackers can be running. Also remember that P^l_k denotes the probability in tracker l at time step k for tracker $l = 1, ..., \mathcal{I}_k$ and recall Θ^j_k represents the planar position at time step k of UAV $j = 1, ..., O$. Recalling that ψ contains the positions of each node in the network, and utilizing the max function to extract the index of the maximum probability in P^l_k, the distance between the peak of that distribution and each UAV can be represented by the Euclidean distance between Θ^j_k and $\psi(\max(P^l_k))$.

These pairwise Euclidean distances are used to form an $\mathbb{R}^{O\times\mathcal{I}_k}$ distance matrix between each UAV and instantiated tracker. A minimum travel distance metric is used to choose pairings of UAVs to trackers. Since each UAV can only be assigned to one tracker, certain combinations of sums along the pairwise distance matrix are inadmissible options. These possibilities are pruned and the combination that yields the sum of assignments with minimum distance traveled from UAV to tracker maximum location is chosen.

Let the assigned peak probability and index of that probability be represented by $\mathcal{P}^{lj}_k$, indicating that the peak probability in tracker l is linked to UAV j at time k. This assignment information is then fed through to the subsequent search and track algorithms.

Strategy 1: Loiter Tracking

The first strategy for tracking targets after detection is loiter tracking. With no trackers running, this algorithm reduces to the LRT gradient-search algorithm described in Section 2. When a tracker is instantiated, the minimum travel distance metric is used at each time step to assign each UAV to a peak probability. That UAV will feed new measurements of the target into the tracker through the data association process. The attraction between a UAV and the associated tracker peak probability is performed using an artificial spring potential as described in Section 2. In this case, the spring is attached to the node of peak probability in the assigned tracker. The associated spring force between the paired UAV j and peak probability $\mathcal{P}^{lj}_k$ is

$$\nabla L^{lj}_k = -K\left(||\Theta^j_k - \psi(\mathcal{P}^{lj}_k)||\right)\frac{\Theta^j_k - \psi(\mathcal{P}^{lj}_k)}{||\Theta^j_k - \psi(\mathcal{P}^{lj}_k)||}, \quad (20)$$

where K is the spring constant and the rest length is zero.

Using the loiter tracking strategy, an assigned UAV no longer follows the likelihood gradient; it does avoid collisions using a modified avoidance force. The assigned UAV is no longer actively repulsed from unassigned UAVs, only from assigned UAVs. Let $\mathcal{J}$ be the set of all UAVs assigned to trackers. The Pauli repulsion force for assigned agents becomes

$$\nabla P^j_k = -48\sum_{\substack{i\neq j,\\ i\in\mathcal{J}}}^{O}\left(\sigma^{12}||\Theta^j_k - \Theta^i_k||^{-13}\right)\frac{\Theta^j_k - \Theta^i_k}{||\Theta^j_k - \Theta^i_k||}, \quad (21)$$

where $\epsilon = 1$ and $\sigma = 2\rho$.

The total control forces applied from the loiter strategy to UAV j are represented as

$$F^j_k = \begin{cases} \nabla L^{lj}_k + \nabla P^j_k, & j \in \mathcal{J} \\ \nabla R^j_k + \nabla P^j_k + \nabla Q^j_k, & j \notin \mathcal{J}, \end{cases} \quad (22)$$

where $\mathcal{J}$ is the set of all UAVs assigned to trackers.

Strategy 2: Search-and-Loiter Tracking

The second strategy is the Search-and-Loiter algorithm. When a tracker l is assigned to an agent j, the peak probability in that tracker is encoded in $\mathcal{P}^{lj}_k$. By continually monitoring the peak probability, the accuracy of the tracker can be determined at each time step. As the peak drops, so too does the probabilistic knowledge of the true target location. As a result, to maintain the health of the tracker, new measurements are required to keep the tracker accurate.

The loiter strategy improves tracker accuracy by continually providing the tracker with new measurements, thereby keeping the peak probability high. The second strategy relies on the tracker to maintain its target estimate even in the absence of measurements. A loiter threshold ϱ tells the assigned UAV when the estimated target location is no longer adequate. So long as $\mathcal{P}^{lj}_k$ remains above ϱ (as monitored by the assigned UAV), the UAV is free to continue searching the likelihood network for other targets. When $\mathcal{P}^{lj}_k$ drops below ϱ, the UAV returns to the target and provides new measurements of the target to bring $\mathcal{P}^{lj}_k$ back above ϱ.

The total control forces for Search-and-Loiter tracking applied to UAV j are represented as

$$F^j_k = \begin{cases} \nabla L^{lj}_k + \nabla P^j_k, & j \in \mathcal{J}\ \&\ \mathcal{P}^{lj}_k \leq \varrho \\ \nabla R^j_k + \nabla P^j_k + \nabla Q^j_k, & j \in \mathcal{J}\ \&\ \mathcal{P}^{lj}_k > \varrho \\ \nabla R^j_k + \nabla P^j_k + \nabla Q^j_k, & j \notin \mathcal{J}. \end{cases} \quad (23)$$

(Note, the same Pauli repulsion rules apply as in the loiter tracking strategy. The term ∇P^j_k changes based on whether

the UAV is assigned or not according to Equations (14) and (21).)

Simulation Parameters and Metrics

Four parameters determine simulation performance: intersection density (ID), coverage rate, guidance algorithm, and UAV-to-target ratio $\widetilde{O}$. ID is the number of intersections per square mile of a road-network snapshot [33].

To describe the ability of UAVs to find targets, the relative coverage rate is found using the procedure in [34]. Let V_{max} be the maximum speed of targets on the road network. Let $\tilde{\theta}$ represent the angle between the velocity of a target and a UAV, n_T represent the target density over the confines of the map, $S = 2\rho + \pi\rho^2/(V_{max}\Delta t)$ represent the cross section of coverage between the target and mobile sensors, and $\sqrt{\xi^2 + \Theta^2 - 2\Theta\xi\cos\tilde{\theta}}$ represent the relative velocity between UAVs and targets. Relative coverage rate estimates the sensor coverage of targets per unit time for a UAV and is defined as [21]

$$\mathcal{C} = n_T \mathcal{S}\sqrt{\xi^2 + \Theta^2 - 2\Theta\xi\cos\tilde{\theta}}, \tag{24}$$

which is similar to the mean free path theory from the kinetic theory of gas molecules found in physics.

Under the assumption of the random mobility model, the relative velocity becomes the average relative speed between the targets and the UAVs [34]

$$\frac{1}{2\pi}\int_0^{2\pi}\sqrt{\xi^2 + \Theta^2 - 2\Theta\xi\cos\tilde{\theta}}d\tilde{\theta}. \tag{25}$$

The target density within the confines of the map is

$$n_T = \frac{T}{\mathcal{A}}, \tag{26}$$

where $\mathcal{A}$ is the convex hull of the road network.

For simulations, a single road network in downtown Baltimore, Maryland with ID 194 ($\mathcal{A}$ = 4.59×10^6 m^2) was used and the altitude of the UAVs was varied linearly from $h = 457$ m to 1829 m. This range was chosen to model variations in the standard operating altitude of the ScanEagle [35], which normally performs surveillance at 457 m. The UAV-to-target ratio $\widetilde{O}$ was varied between 0.25, 0.5, 1.0, and 1.5.

Changes in altitude are accompanied by changes in the fidelity of the sensors onboard for detecting and gathering measurements of target location. The change in accuracy can be extended to changes in the simulation model by varying P_d, P_f, ρ, and s with altitude. P_d, P_f, and s were assumed to vary linearly with altitude according to

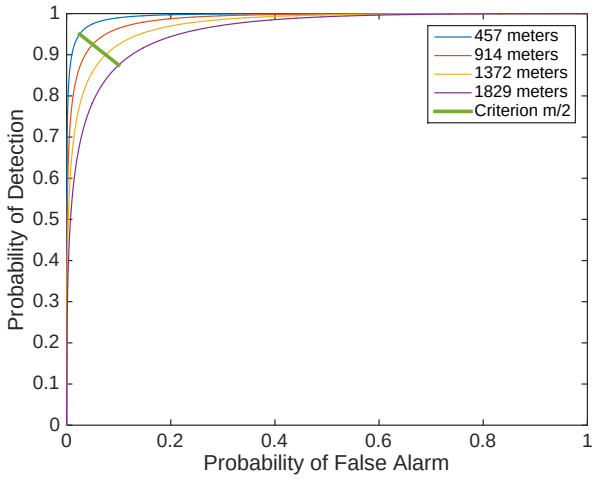


Figure 2: ROCs for linearly increasing altitude

Table 1: Relation between altitude, P_d, P_f, s, and ρ

Altitude (m)	$\mathbf{P_d}$	$\mathbf{P_f}$	**s (m)**	$\boldsymbol{\rho}$ **(m)**
457.2	0.95	0.025	5	93.5
914.4	0.925	0.05	10	187.0
1372.0	0.90	0.075	15	280.0
1828.8	0.875	0.10	20	374.0

$$P_d = 0.975 - 0.025\frac{h}{457} \tag{27}$$

$$P_f = 0.025\frac{h}{457} \tag{28}$$

$$s = \frac{5}{457}h. \tag{29}$$

Variations in P_d and P_f produce the series of receiver operating characteristic curves (ROC) [29] in Figure 2.

As altitude increases, the field of view of the camera footprint increases. Let F_L be the focal length of the sensor and S_r the sensor radius. The radius of the circular sensor footprint for each height is

$$\rho = \frac{S_r}{F_L}h. \tag{30}$$

A summary of simulation parameters are provided for a camera with $F_L = 90$ mm and $S_r = 18.4$ mm in Table 1. Additional parameters used for the simulation are described in Table 2.

The performance of our simulations is compared using a number of metrics [36], including the number of valid tracks (NVT), the number of spurious tracks (NST), the number of valid associations (NVA), and the number of false associations (NFA). Each of these measures is recorded for every time step and Monte Carlo trial, and are averaged over the

Table 2: Simulation parameters

Parameter	Value (units)	Definition
Δt	0.3 (sec)	time step
ϕ_{max}	10	target detection threshold
$\dot{\theta}_{max}$	70 (°/sec)	UAV max turn rate
S_{max}	80 (mph)	UAV speed
V_{max}	50 (mph)	target speed
α	0.4470 (m^2/s)	target diffusivity
c	m/2	criterion
$\mathcal{E}$	300	EMD threshold
P_{min}	0.07	track dissolve threshold
C	50	LRT decay term
ϱ	0.3	loiter threshold
$\mathcal{A}$	4.59×10^6 m^2	convex hull of network

entire data set for a particular scenario described by the four parameters: intersection density (ID), coverage rate, guidance algorithm, and UAV-to-target ratio $\tilde{O}$. These metrics may also be combined into additional useful metrics, including the measure of completeness (MOC), which is the ratio of valid trackers to total number of targets [36]. Each altitude was simulated for both of the tracking algorithms over 50 trials in a custom Matlab simulation environment.

Simulation Results

Figure 3 shows the change in MOC vs. time for four different $\tilde{O}$'s and a variety of different coverage rates. For all four UAV-to-target ratios, both algorithms are characterized by two sequences: the (mostly) positive linear aggregation of information about the targets and road network, and the plateau achieved when some percentage of the total targets are found. As coverage rate increases the slope of the linear aggregation portion becomes sharper, indicating quicker data collection. This trend results in the plateau being higher, and thus leads to MOC becoming close to unity for the maximum coverage rate used in simulation. Having the highest coverage rate consistently yields the quickest and most complete tracking of targets for all four UAV-to-target ratios.

Comparing the performance of the two algorithms, the loiter algorithm perform substantially better in quickly gathering information and achieving a higher steady-state MOC for lower coverage rates. However, as the coverage rate increases, this gap in steady-state MOC between each algorithm becomes smaller and the two algorithms both perform.

The algorithms perform relatively well for different UAV-to-target ratios. For $\tilde{O} = 0.25$ there is 1 UAV and 4 targets. As a result, this UAV-to-target ratio achieves at best a steady-state average of 25% because both algorithms are compelling the UAV to focus on keeping one tracker accurate rather than tracking all possible targets. On the other side of the spectrum is the case of $\tilde{O} = 1.5$, which involves 3 UAVs and 2 targets. As coverage rate increases, so too does the steady-state MOC, reaching close to 100% relatively early in the simulation.

Let the percentage of valid trackers PVT be

$$PVT = \frac{NVT}{NVT + NST}. \tag{31}$$

For all four UAV-to-target ratios the difference in PVT for each algorithm is negligible and both instantiate valid trackers 97% of the time for a wide range of altitudes. However, by highlighting a couple trials from the case of 4 targets and 1 UAV, some of the characteristics of Search-and-Loiter relative to the loiter algorithm become visible. For case study 1, Search-and-Loiter detects a third target that the loiter algorithm never sees and keeps occasional tracks on a second target that loiter loses. In case study 2, Search-and-Loiter loses its track of all targets after matching the performance of the loiter algorithm. However, after regaining a track on one target, Search-and-Loiter performs substantially better than loiter and has instantiated trackers on all four targets by the end of the simulation. In case study 3, Search-and-Loiter performs better than pure loiter, tracking as many as three targets. Search-and-loiter does experience a quick tracker loss, indicative of the UAV returning to perform its loiter assignment in Search-and-Loiter, but returning too late. This problem can be solved by further tuning of the return threshold in this strategy, as well as a superior target motion model.

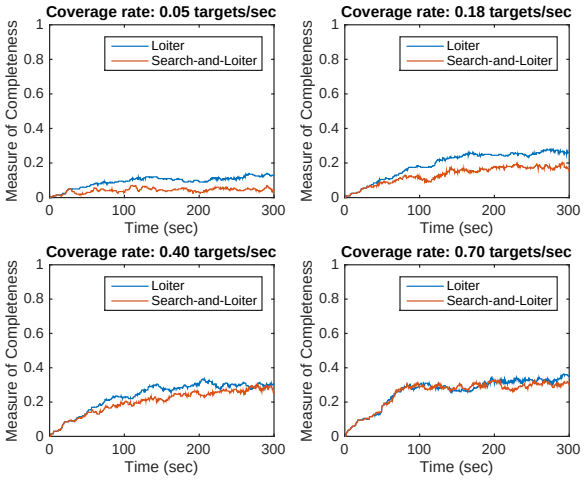
Let the percentage of valid associations PVA be

$$PVA = \frac{NVA}{NVA + NFA}. \tag{32}$$

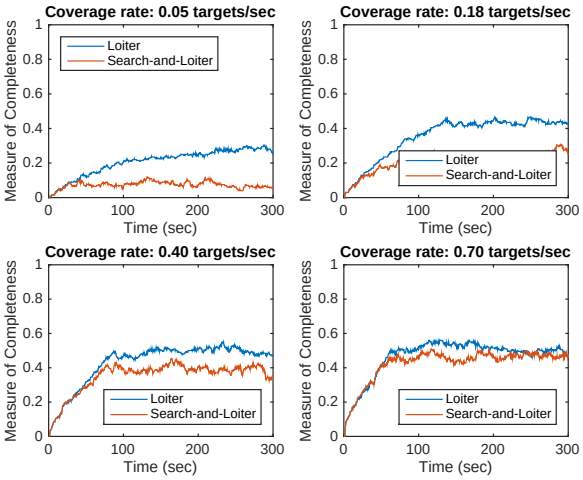
Figure 5 shows the relationship between coverage rate $\mathcal{C}$ and PVA for both algorithms and all four UAV-to-target ratios. In these cases, differences between algorithms are slightly more defined than for the case of PVT. The loiter algorithm performs slightly better in providing trackers with accurate measurement updates than the Search-and-Loiter algorithm. The performance difference can be explained by recalling that UAVs employing the loiter strategy are continually providing updates to the tracker and their search radius always has the target they are tracking fully in view. In addition, for all cases except $\tilde{O} = 0.25$, the PVA seems to plateau and then drop indicating that the quality of the measurements being provided is dropping and more incorrect associations are occurring. This negative slope in PVA decreases as $\tilde{O}$ increases. The decrease in PVA with increasing coverage rate was an expected outcome based on a measurement model uncertainty that increased with altitude, but the effect of increasing $\tilde{O}$ was not anticipated.

5. CONCLUSION

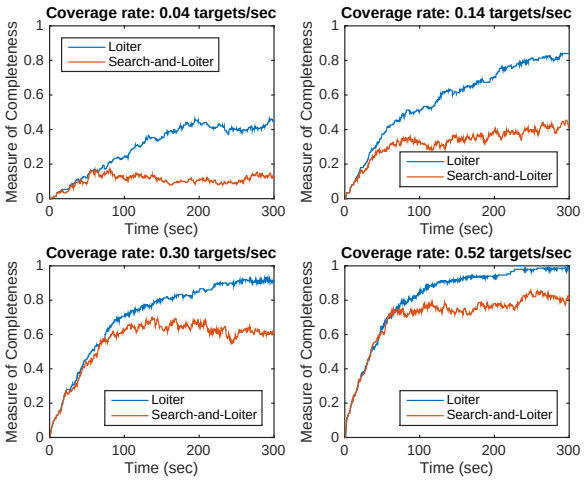
The variety of mission requirements for surveillance operations make the search and track problem difficult and computationally intensive. We present a Bayesian filtering framework for cooperative search-and-track of mobile targets on a road network using UAVs with finite field of view. UAVs generate measurements of targets on a likelihood network and call detections once the local likelihood passes a critical threshold. Measurements from detected targets are used to create a measurement probability distribution that indicates locations of targets. A data-association framework takes prior tracker probability distributions and new measurement probability distributions and compares them using the earth



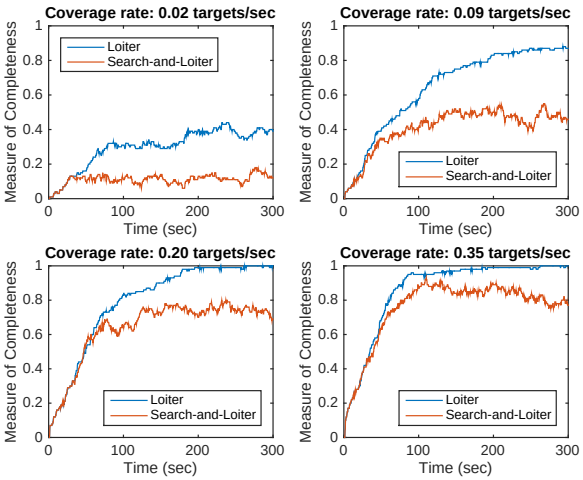
(a) For UAV-to-target ratio 0.25



(b) For UAV-to-target ratio 0.5

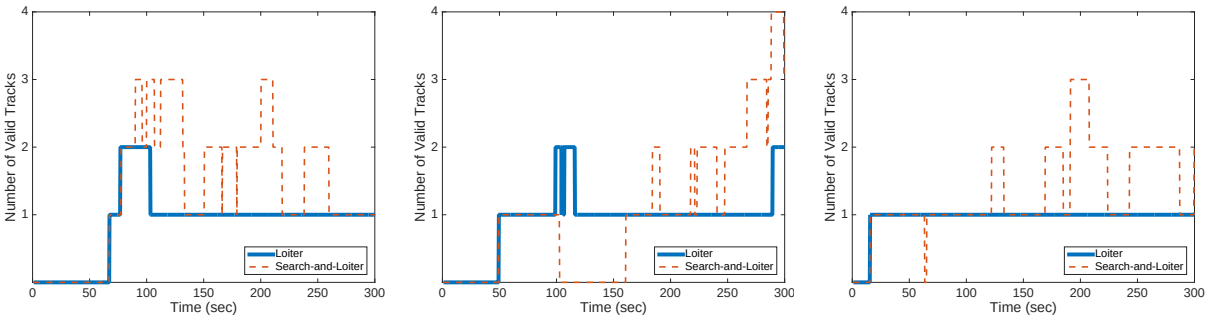


(c) For UAV-to-target ratio 1.0



(d) For UAV-to-target ratio 1.5

Figure 3: MOC vs. time for increasing coverage rate. The loiter algorithm consistently performs better in tracking all of the targets than the Search-and-Loiter algorithm. As coverage rate increases, the time required to reach the plateau in performance decreases and the steady-state performance increases.



(a) Case Study 1 **(b)** Case study 2 **(c)** Case Study 3

Figure 4: Case study of valid tracks vs. time for 4 targets and 1 UAV at an altitude of 1828.8 m. The performance of Search-and-Loiter exceeds that of pure loiter in both maximum number of targets tracked and cumulative time tracking each target.

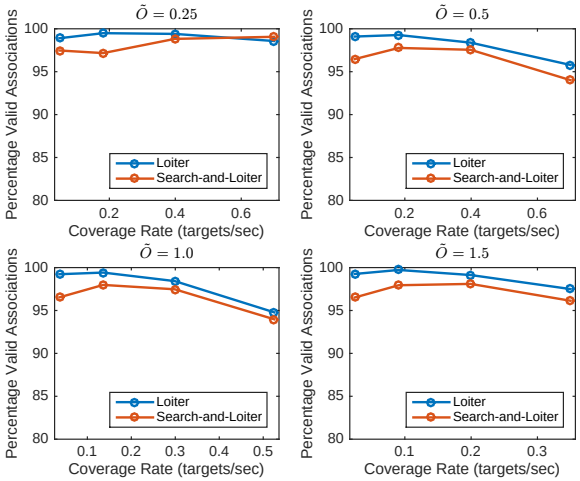


Figure 5: The percentage of associations that are valid vs. coverage rate. The percentage of valid associations decreases with increased coverage rate for low UAV-to-target ratios, but becomes constant at high UAV-to-target ratios.

mover's distance. Utilizing artificial potentials, two motion-planning strategies were created to balance finding undetected targets with keeping trackers accurate.

On average, the loiter algorithm performed better than the Search-and-Loiter algorithm in achieving high levels of MOC and in accuracy of trackers and data associations. However, in the cases where there are many fewer UAVs than targets, the Search-and-Loiter algorithm can provide temporary tracks on many more targets than pure loiter and give operators a better understanding of the target distribution on the road network. The average outcome being in favor of loiter is not surprising since the Search-and-Loiter algorithm is highly dependent on both the motion model (a random walk) and the threshold for returning to the tracker. Both of these were determined heuristically and they could both be greatly improved through optimization based on the target model. As a result, important ongoing contributions to tracking algorithm development in this framework would include a more accurate motion model for targets, more accurate variation in measurement uncertainty for higher altitude UAVs, and an optimal Search-and-Loiter algorithm rather than a heuristic-based algorithm.

ACKNOWLEDGMENTS

The authors would like to acknowledge Jerry Petersen and Brian Funk of L-3 Unmanned Systems for their suggestions and discussion related to the design of this simulation and algorithm. Brett Barkley is supported by the University of Maryland L-3 Graduate Scholarship.

REFERENCES

[1] Department of Defense, "Unmanned Systems Roadmap: 2007-2032," Tech. Rep., 2007.

[2] N. Sydney, D. A. Paley, and D. Sofge, "Physics-inspired motion planning for information-theoretic target detection using multiple aerial robots," *Autonomous Robots*, pp. 1–11, 2015.

[3] B. Barkley and D. Paley, "Cooperative bayesian target detection on a real road network using aerial vehicles," in *Proc. of International Conference on Unmanned Aircraft Systems*, June 2016.

[4] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *International Journal of Computer Vision*, vol. 40, no. 2, pp. 99–121, 2000.

[5] L. Hong, N. Cui, M. Bakich, and J. R. Layne, "Multirate interacting multiple model particle filter for terrain-based ground target tracking," *Proc. IEE Control Theory and Applications*, vol. 153, no. 6, pp. 721–731, Nov 2006.

[6] M. Ekman and E. Sviestins, "Multiple model algorithm based on particle filters for ground target tracking," in *10th Conf. Proc. on Information Fusion*, July 2007, pp. 1–8.

[7] T. Yang, H. A. P. Blom, and P. G. Mehta, "Interacting multiple model-feedback particle filter for stochastic hybrid systems," in *52nd Conf. Proc. IEEE Conference on Decision and Control*, Dec 2013, pp. 7065–7070.

[8] X.-R. Li and Y. Bar-Shalom, "Multiple-model estimation with variable structure," *IEEE Transactions on Automatic Control*, vol. 41, no. 4, pp. 478–493, Apr 1996.

[9] T. Kirubarajan, Y. Bar-Shalom, K. R. Pattipati, and I. Kadar, "Ground target tracking with variable structure imm estimator," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 36, no. 1, pp. 26–46, Jan 2000.

[10] C. S. Agate and K. J. Sullivan, "Road-constrained target tracking and identification a particle filter," *Proc. SPIE*, vol. 5204, pp. 532–543, 2003.

[11] D. Salmond, M. Clark, R. Vinter, and S. Godsill, "Ground target modelling, tracking and prediction with road networks," in *10th Conf. Proc. on Information Fusion*, July 2007, pp. 1–8.

[12] C. Yang, M. Bakich, and E. Blasch, "Nonlinear constrained tracking of targets on roads," in *8th Conf. Proc. on Information Fusion*, vol. 1, July 2005, pp. 235–242.

[13] C. Kreucher, A. Hero, and K. Kastella, "Multiple model particle filtering for multitarget tracking," in *12th Annual Workshop on Adaptive Sensor Array Processing. Lexington, MA*, 2004.

[14] U. Orguner, T. Schon, and F. Gustafsson, "Improved target tracking with road network information," in *Proc. IEEE Aerospace Conf.*, March 2009, pp. 1–11.

[15] M. G. Rutten, B. Ristic, and N. J. Gordon, "A comparison of particle filters for recursive track-before-detect," in *Proc. 7th International Conference on Information Fusion*, vol. 1, July 2005.

[16] S. Buzzi, M. Lops, and L. Venturino, "Track-before-detect procedures for early detection of moving target from airborne radars," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 3, pp. 937–954, July 2005.

[17] S. M. Tonissen and R. J. Evans, "Peformance of dynamic programming techniques for track-before-detect," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 32, no. 4, pp. 1440–1451, Oct 1996.

[18] S. M. Tonissen and Y. Bar-Shalom, "Maximum likelihood track-before-detect with fluctuating target amplitude," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, no. 3, pp. 796–809, Jul 1998.

[19] C. Jauffret and Y. Bar-Shalom, "Track formation with bearing and frequency measurements in clutter," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 26, no. 6, pp. 999–1010, Nov 1990.

[20] W. Blanding, P. Willett, and Y. Bar-Shalom, "Multiple target tracking using maximum likelihood probabilistic data association," in *Conf. Proc. IEEE Aerospace Conference*, March 2007, pp. 1–12.

[21] A. Orych, "Review of methods for determining the spatial resolution of uav sensors," *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 40, no. 1, p. 391, 2015.

[22] J. A. Bondy, *Graph Theory With Applications*. Oxford, UK: Elsevier Science Ltd., 1976.

[23] B. Schoelkopf and M. Warmuth, *Learning Theory and Kernel Machines: 16th Annual Conference on Computational Learning Theory and 7th Kernel Workshop*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2003.

[24] "Discrete Laplace-Beltrami operators for shape analysis and segmentation," *Computers and Graphics*, vol. 33, no. 3, pp. 381–390, 2009.

[25] L. Stone, R. Streit, T. Corwin, and K. Bell, *Bayesian Multiple Target Tracking, Second Edition:*, ser. Radar/Remote Sensing. Artech House, 2013.

[26] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *American Journal of Mathematics*, vol. 79, no. 3, pp. 497–516, 1957.

[27] E. Hairer, S. P. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations I (2nd Revised. Ed.): Non-stiff Problems*. New York, NY, USA: Springer-Verlag New York, Inc., 1993.

[28] J. E. Jones, "On the determination of molecular fields. ii. from the equation of state of a gas," *Proc. of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 106, no. 738, pp. 463–477, 1924.

[29] N. A. Macmillan and C. D. Creelman, *Detection Theory — A user's guide*, 2nd, Ed. Mahwah, New Jersey, London: Lawrence Erlbaum Associates, 2005.

[30] M. A. Richards, *Fundamentals of radar signal processing*. New York: McGraw-Hill, 2014.

[31] E. Levina and P. Bickel, "The earth mover's distance is the Mallows distance: some insights from statistics," in *Proc. 8th IEEE International Conference on Computer Vision*, vol. 2, 2001, pp. 251–256 vol.2.

[32] M. Alipour, "EMD (earth mover's distance) mex interface," https://www.mathworks.com/matlabcentral/fileexchange/12936, 2006.

[33] J. Dill, "Measuring network connectivity for bicycling and walking," in *83rd Annual Meeting of the Transportation Research Board*, 2004, pp. 11–15.

[34] G. Y. Keung, B. Li, Q. Zhang, and H. D. Yang, "The target tracking in mobile sensor networks," in *IEEE Global Telecommunications Conference*, 2011, pp. 1–5.

[35] C. W. Lum and B. Waggoner, "A risk based paradigm and model for unmanned aerial systems in the national airspace," in *Proc. AIAA Infotech Aerospace Conference*, 2011.

[36] A. A. Gorji, R. Tharmarasa, and T. Kirubarajan, "Performance measures for multiple target tacking problems," in *Proc. of the 14th International Conference on Information Fusion*, 2011, pp. 1–8.

BIOGRAPHY

Brett E. Barkley *is a graduate student in the Department of Aerospace Engineering at the University of Maryland. Barkley received the B.S. degrees in Aerospace Engineering and Engineering Physics from the University of Maryland and Elon University, respectively, in 2015. His research investigates the use of cooperative algorithms to balance finding new targets with tracking known targets in environments with highly variable visibility.*

Derek A. Paley *is the Willis H. Young Jr. Associate Professor of Aerospace Engineering Education in the Department of Aerospace Engineering and the Institute for Systems Research at the University of Maryland. Paley received the B.S. degree in Applied Physics from Yale University in 1997 and the Ph.D. degree in Mechanical and Aerospace Engineering from Princeton University in 2007. He received the National Science Foundation CAREER award in 2010, the Presidential Early Career Award for Scientists and Engineers in 2012, the University of Maryland E. Robert Kent Teaching Award for Junior Faculty in 2014, and the AIAA National Capital Section Engineer of the Year in 2015. Paley's research interests are in the area of dynamics and control, including cooperative control of autonomous vehicles, adaptive sampling with mobile networks, and spatial modeling of biological groups.*