Cooperative Bayesian Target Detection on a Real Road Network Using Aerial Vehicles

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

Abstract—**A cooperative track-before-detect algorithm for multiple ground targets is presented for fixed-wing Unmanned Air Vehicles (UAVs) with a finite field of view. The road network forms a graph whose nodes indicate the target likelihood ratio. Target observations are assimilated by a Bayesian likelihood ratio tracker in which likelihood diffuses according to the graph Laplacian of the road network. Using the likelihood ratio in a composite potential along with attractive and repulsive terms, the algorithm directs UAVs modeled as Dubins cars towards nodes with high likelihood. Results from numerical simulations are included to illustrate the algorithm.**

I. Introduction

Unmanned Aerial Vehicles (UAVs) have become a substantial topic of research and development for many national governments. Especially prevalent is the need for UAVs in the realm of intelligence, surveillance, and reconnaissance missions and for target identification and designation as described in the Unmanned Systems Roadmap 2007–2032 [1]. With low-cost aerial vehicles and powerful visual sensors widely available, the goal is to improve ground-target tracking strategies and coordination between UAVs to maximize information acquisition and accumulation. This paper describes a physics-inspired path-planning strategy based on a Bayesian likelihood ratio tracker that assimilates measurements of potential targets on a road network. The planning strategy determines UAV motion using target detections, according to the evolution of the likelihood ratio over the network. As a result, the strategy is a manifestation of the Dynamic Data-Driven Application Systems (DDDAS) paradigm [2], which uses sensor measurements to guide subsequent data collection.

Due to the importance of UAVs in a variety of missions, multi-target tracking is a popular topic in the research literature. One approach is to use Bayesian inference and either multi-hypothesis or maximum-likelihood filters to track moving targets [3]–[6]. These approaches are adequate for combining sensor measurements with tracking and detecting targets, but do not always effectively move the UAVs to find targets. The methods described in [7]–[9] offer solutions to this problem by operating in the Bayesian inference regime and using this information to move UAVs, however the techniques are focused on a single UAV.

Approaches that utilize the road-network structure lie almost entirely in the realm of particle and other nonlinear filters. This approach is a natural choice for propagating targets based on their directional dynamics by allocating probabilities and thus proportions of particles for each motion mode [10]–[12]; however, it can lead to poor estimation quality if sharp mode transitions occur. The methods described in [13]–[16] are examples of the interacting multiple model particle filter, which has been introduced as a solution to this issue by fixing the number of particles per mode regardless of mode probability. Unfortunately, while the proposed filter modification can provide lower errors and quick adaptation when targets change motion modes, there is an inherent trade-off between these two attributes, and questionable robustness to motion model violations [17].

This paper focuses on the implementation of a potential-based motion-planning strategy that enables multiple fixed-wing UAV sensor platforms to collaborate in searching for mobile ground targets on a road network. Each fixed-wing UAV is modeled using a Dubins car model with constraints on turning rate and speed. The onboard sensors have a finite field of view, probability of detection, and probability of false alarms. Sensors output an increased measurement signal if targets are in view, culminating in a target detection if a threshold is met. Targets are programmed to operate only on the road network, obey the rules of the road (e.g., one-way streets), and stop randomly to simulate courier behavior. Sensor measurements are incorporated into the likelihood network using a Bayesian likelihood ratio tracker (LRT) that makes use of a recursive model to produce a probability density function along possible target locations on the road network. In this formulation, when one or more targets are in range, likelihood increases inside the sensor range of the UAV; if a likelihood threshold is reached, then the target is detected. Subsequent measurements of a detected target would be assimilated by a target tracking filter, currently under development. UAVs have global knowledge of likelihood on the road network and the location of all other UAVs, by assumption.

The LRT track-before-detect algorithm is coupled with potential-based motion planning. As the LRT recursively modifies the likelihood surface on the road network based on proximity to targets, a physics-inspired artificial potential guides the motion of each sensor platform to nodes with higher likelihood. The sensor platform avoids local minima by continually changing the local likelihood ratio; consequently, the UAV is compelled to move away from its current location unless a target is in range. The algorithm is illustrated via simulations on real road networks extracted from OpenStreetMap databases. UAVs are initialized from the corners of the target area and targets randomly traverse the network while stopping and starting at random nodes.

Artificial forces act on the UAV to ascend the likelihood gradient, stay close to the road network, and avoid other UAVs. The first force is parallel to the edge of the graph that represents the maximum change in likelihood. This modification of the traditional gradient-climbing behavior reflects the fact that the likelihood (and targets) are constrained to the road network. The second force is a repulsive force between UAVs known as Pauli repulsion, which is typically used to model the interaction between molecules when their electron orbitals overlap. The third force is a virtual spring between the local edge of maximum likelihood and the UAV, which keeps the UAV within sensor range of the road network as it ascends the likelihood gradient.

Potential-based algorithms similar to this paper, such as [18], [19], suffer from a number of issues. Chief among them is getting stuck in local minima of the potential. Our algorithm avoids this issue by combining the motion planning strategy with the evolution of the likelihood network via the LRT. In addition, since the dynamics of the fixed-wing UAVs have been implemented using a Dubins car model, the UAVs are naturally inclined to follow their inertia along a gradient rather than get stuck in deadlock. Another important advantage of the LRT approach is the savings in computation relative to explicit calculation of mutual information among UAVs on each iteration [6].

The prediction of likelihood using the LRT in the absence of measurements is also novel. For the typical 2-D likelihood surface, likelihood is propagated numerically using the dynamics of the target e.g., by using a random walk described by the diffusion partial differential equation on the likelihood grid. In the case of a road network modeled by a graph, the graph Laplacian is utilized to predict the possible locations of targets using a random walk model constrained to edges on the graph. The diffusivity constant represents the mobility of the targets.

The primary contribution of this paper is a physics-inspired motion-planning system for cooperative track-before-detection of multiple targets on a road network graph. Target position likelihood is assessed using a Bayesian likelihood ratio tracker and is spatially diffused along the graph using the graph Laplacian as time advances. A secondary contribution is the introduction of artificial potentials to guide UAV searchers along the gradient of maximal likelihood change on the graph while avoiding collisions. UAVs are directed along local graph edges of maximal likelihood change, thereby attracting them to positions of maximum likelihood on the graph, while simultaneously avoiding other UAVs via a Pauli repulsion force. As a result, this target detection strategy is an example of a DDDAS paradigm since UAV motion is directly influenced by measurements of target likelihood. In addition, a novel measurement of intersection density is introduced to quantify algorithm performance for road networks of varying complexity.

The paper is organized into the following sections. Section II summarizes the fundamentals of graph theory and the construction of a likelihood graph from a road network. Section III describes the likelihood-ratio detection and prediction using the graph Laplacian. Section IV presents the dynamics of the UAVs, the artificial potentials that make up the motion planning strategy, and simulations of the algorithm. Section V summarizes the paper and ongoing work.

II. Road Network Graph

This section describes our method for modeling road networks extracted from the OpenStreetMap [20] database (© OpenStreetMap contributors) and converting them into undirected graphs. Nodes from OpenStreetMap become graph vertices, which describe the geometric layout of the network, and edges, which describe the connectivity between nodes. Target likelihood is assigned to each node and is exchanged between vertices along edges by a finite-difference calculation of the Laplacian operator represented by the Laplacian matrix of the graph.

A. Graph Theory

A graph is a structure in mathematics that models the relation between pairs of objects. More specifically, a connected graph is a structure where any point on the graph can be reached from any other point on the graph [21]. A road graph is composed of three elements [22]: a set V of N vertices, a set E of M edges, and $\psi(1,...,N)$, which returns the planar coordinates of the vertices. An example of a uniformly weighted and undirected graph is shown in Figure 1, represented by

$$G = (V, E) \tag{1}$$

where $V = (1,2,3,4) \in \mathbb{R}^4$ and $E = \big\{(1,2),(1,4),(2,3),(3,4)\big\} \in \mathbb{R}^2 \times ... \times \mathbb{R}^2$.

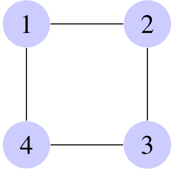


Fig. 1: An undirected graph with N=4 nodes.

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.,

$$a_{ij} = \begin{cases} 0, & \text{if } j = i \\ 1, & \text{if there is a directed edge from j to i} \\ 0, & \text{if there is no directed edge from j to i.} \end{cases} \tag{2}$$

For an undirected graph, the adjacency matrix is symmetric about the diagonal indicating bidirectional travel along that edge. The ii entries of the degree matrix D give the number of incoming connections to node i, whereas the off-diagonal entries are zero:

$$d_{ij} = \begin{cases} \sum_{j=1}^{N} a_{ij}, & \text{if } i = j \\ 0, & \text{if } i \neq j. \end{cases} \tag{3}$$

Another convenient construct in graph theory is the incidence matrix $B \in \mathbb{R}^{N \times M}$, which relates edges and nodes, with row indices representing the node indices and column indices representing the edge indices [22]. For undirected graphs, the edge direction is assigned arbitrarily by setting one entry along each column equal to one and another equal to negative one. For the graph in Figure 1, the incidence matrix is

$$B = \begin{bmatrix} 1 & 0 & 0 & 1 \\ -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & -1 \end{bmatrix}.$$

Note that for each column there are precisely two non-zero entries, since exactly two nodes are connected by a single edge. (There are no self loops.)

The Laplacian matrix $L \in \mathbb{R}^{N \times N}$ of graph G is

$$L = D - A. \tag{4}$$

The graph Laplacian is used below as an operator that models a random walk between adjacent nodes. To spread target likelihood in the network, the rate of diffusion is determined by the target speed. Diffusion of likely target positions spreads evenly in all possible directions along the road network.

B. Laplacian Operator

Since targets may only travel between connected nodes, the spatial rate of change of likelihood for a vertex in a network is modeled by partial derivatives along each connected edge. Let ϕ be the likelihood at each node, one way to model this interaction is with the heat equation

$$\frac{d}{dt}(\phi) + \alpha \nabla^2 \phi = 0, \tag{5}$$

where ∇^2 is the Laplace operator, which takes partial derivatives along each connected edge. The Laplace operator acting on each node is approximated by using finite difference methods [23].

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 target likelihood at time k for node n is $\phi_k(n) = p(\xi_k | \zeta_k)\big|_n$, where $n = 1, ..., N$. For vertex 1 in Figure 1, the spatial rate of change of likelihood would be represented as

$$\nabla^2(\phi_k(1)) = \frac{\phi_k(2) + \phi_k(3) + \phi_k(4) - 3\phi_k(1)}{h^2}, \tag{6}$$

where h is the node spacing (assumed to be identical for all edges).

Let $\mathcal{N}(n)$ represent the neighbor set of all vertices connected to node n. Assuming that the exchange rate is a constant α, the time rate of change of likelihood is modeled by

$$\frac{d}{dt}(\phi_k(n)) = -\alpha \sum_{j \in \mathcal{N}(n)} a_{nj}(\phi_k(n) - \phi_k(j)). \tag{7}$$

In matrix-vector notation, equation (7) becomes

$$\frac{d}{dt}(\phi_k) + \alpha L \phi_k = 0, \tag{8}$$

which is the heat equation with spatial discretization. The Laplacian matrix L implements the finite difference calculation of the Laplacian operator ∇^2 [24]. Thus the diffusion of likelihood throughout the road network represented by a graph is found by solving the first-order matrix differential equation in (8).

The graph Laplacian matrix of a connected undirected graph is positive semi-definite [24]. As a result, other than zero, the graph Laplacian has all positive real eigenvalues, which indicates that the information on the graph will be conserved; (8) reaches an equilibrium that is the average of the initial likelihood. The use of the Laplacian matrix as a method for information diffusion and target prediction is further described in Section IIIB.

C. OpenStreetMap Data Structure and Importing

OpenStreetMap data is exported for a particular map snapshot using the Overpass Turbo web-based data mining tool [25]. Turbo allows users to write and implement scripts that limit the number and type of results returned for a particular map export. To obtain roadways accessible only by cars, we restrict the types of paths that we export as shown in Figure 2. The resulting data file includes the bounds of the export data (in longitude and latitude), all of the nodes in the road network, and the ways (lists of adjacent nodes) that define the individual roads in the network. Each node contains a unique node id, longitude and latitude coordinates, and a tag that describes what kind of road element it is (e.g., a highway). Each way has a unique id and includes all of the node ids that compose it. Road data is parsed into a Matlab struct data structure that represents the road network.

III. Likelihood Ratio Detection and Tracking

A. Likelihood Ratio Tracker

The instantaneous position of likelihood targets is found using a log-likelihood ratio tracker [6]. A log-likelihood ratio tracker is effective for detecting possibly multiple targets based on recursive Bayesian estimation. This methodology is often called track-before-detect because it accumulates sensor data about possible targets before they are detected. The particular methodology used in this framework is based on previous work done in physics-inspired motion planning [26].

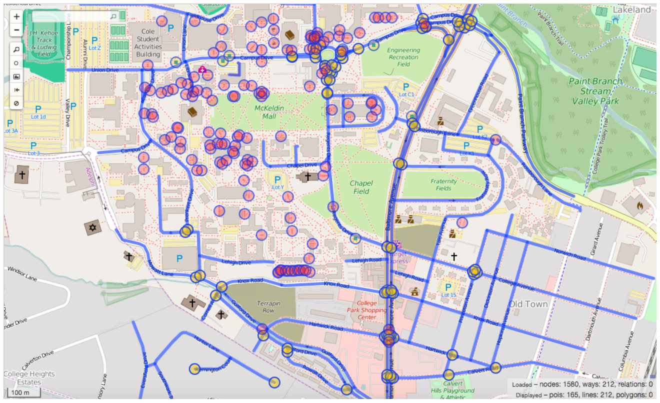

Lakeland
Cole Student Activities Building
McKeldin Mall
Engineering Recreation Field
Paint Branch Stream Valley Park
Chapel Field
Fraternity Fields
Terrapin Row
College Park Shopping Center
Old Town
College Avenue
Knox Road
Calvert Road
Dartmouth Avenue
Columbia Avenue
College Heights Estates
100 m
Loaded – nodes: 1580, ways: 212, relations: 0
Displayed – pois: 165, lines: 212, polygons: 0

Fig. 2: OpenStreetMap snapshot

A Bayes filter is a probabilistic methodology for recursively converting noisy measurements of a target's state space into a probability density function using a mathematical model of the target dynamics. The filter is applied in discrete time steps to predict and update the two-dimensional position of a target. Recall $\xi_k = (x_k, y_k)$ denotes the target state at time step k and ζ_k denotes an observation of the target at k. The predict step involves computing the conditional probability [6]

$$p(\xi_k|\zeta_{k-1}) = \int_{\Omega} p(\xi_k|\xi_{k-1}) p(\xi_{k-1}|\zeta_{k-1}) d\xi_{k-1}. \tag{9}$$

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

$$p(\xi_k|\zeta_k) = \frac{p(\zeta_k|\xi_k) p(\xi_k|\zeta_{k-1})}{p(\zeta_k|\zeta_{k-1})}, \tag{10}$$

where

$$p(\zeta_k|\zeta_{k-1}) = \int_{\Omega} p(\zeta_k|\xi_k) p(\xi_k|\zeta_{k-1}) d\xi_k$$

is the integral of the numerator. In this framework, simultaneous observations from multiple sensors are assimilated by executing consecutive measurement updates.

In a likelihood-ratio tracker, we replace the measurement likelihood with the measurement likelihood ratio. The numerator of the likelihood ratio represents the conditional probability of the measurement given that the target is present (ξ_k^+), whereas the denominator represents the conditional probability of the measurement given that the target is not present (ξ_k^-). Thus, the log-likelihood ratio is

$$\log \mathcal{L}(\zeta_k|\xi_k) = \log \frac{p(\zeta_k|\xi_k^+)}{p(\zeta_k|\xi_k^-)} = \log(p(\zeta_k|\xi_k^+)) - \log(p(\zeta_k|\xi_k^-)). \tag{11}$$

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

$$\mathrm{p}(\xi_k|\zeta_k) = \log \frac{\mathcal{L}(\zeta_k|\xi_k) p(\xi_k|\zeta_{k-1})}{p(\zeta_k|\zeta_{k-1})} = \tag{12}$$
$$\mathrm{p}(\zeta_k|\xi_k^+) - \mathrm{p}(\zeta_k|\xi_k^-) + \mathrm{p}(\xi_k|\zeta_{k-1}) + \mathrm{p}(\zeta_k|\zeta_{k-1}).$$

The first term in (12) represents the newly obtained, positive information that a target is present. Likewise, the second term represents the newly obtained, negative information that no target is present. The third term represents the prior information about the target, and the fourth term is a normalization constant that may be safely ignored if unknown.

When the target probability reaches a critical threshold at a location in the graph, the target is declared detected. If the targets do not pass the threshold then the target probabilities are maintained as hypotheses for future iterations.

B. Predict step: Integrating the diffusion equation

The predict step (10) involves updating the target probability density function in the absence of measurement information. The graph representing the road network allows us to specify requirements that are imposed on target motion. A random-walk model is described by the diffusion in (8). The diffusivity coefficient for the network is set according to parameters that describe both the road and the graph model. The relationship between the diffusivity α, the time step of the simulation Δt, the node spacing Δx, and target max speed V_{max} is

$$\alpha = \frac{\Delta t}{\Delta x} V_{max}. \tag{13}$$

The graph Laplacian is a conservative operator, so the sum of target likelihood in the network never changes and thus boundary conditions are naturally enforced for the road network. However, with no additional measurements along the network, likelihood throughout the graph should reach consensus at a equilibrium value reflecting neither high nor low likelihood, indicative of an unknown target distribution after a long period of time. This outcome is achieved by adding in an additional exponential decay term to the heat equation. The updated differential equation is

$$\frac{d}{dt}(\phi_k) + \alpha L \phi_k + \frac{\alpha}{C} \phi_k = 0. \tag{14}$$

The α/C term ensures that likelihood decays much slower than it spreads between nodes as long as $C \gg \alpha$.

Let $I \in \mathbb{R}^{N \times N}$ be the identity matrix. By rearranging the terms of the first-order matrix differential equation, it can be solved with a matrix exponential as follows:

$$\frac{d}{dt}(\phi_k) = \left(-\alpha L - \frac{\alpha}{C} I \right) \phi_k, \tag{15}$$

which implies

$$\phi_k = e^{-\alpha(L + C^{-1} I)\Delta t} \phi_{k-1}. \tag{16}$$

Eq. (16) is the solution for the evolution of likelihood over the road network, with a heat diffusivity equal to the mobility of the targets on the graph, and a non-conservative decay term to reflect the loss (positive or negative) of likelihood as time evolves.

C. Update step: The sensor measurement model

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 probability of P_f per measurement time step [27]. Combining these two probabilities, the sensitivity m of each sensor is

$$m = z(P_d) - z(P_f), \tag{17}$$

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

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

For example, $P_d = 0.95$ and $P_f = 0.1$ yields $m = 2.92$. Let w_k represent unit-normal measurement noise in standard deviation units at time step k. When the target is absent the measurement data is $\zeta_k = w_k$, whereas when the target is present, the measurement data is $\zeta_k = m + w_k$. Assuming a zero-mean Gaussian sensor model yields [28]

$$\begin{aligned} p(\zeta_k|\xi_k^-) &= \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{\zeta_k^2}{2}\right) \\ p(\zeta_k|\xi_k^+) &= \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{(\zeta_k - m)^2}{2}\right). \end{aligned}$$

The log-likelihood ratio (11) becomes

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

where m is a function of the sensor P_d and P_f given by (17). Note, the log-likelihood ratio is applied to the prior located inside a disc of radius ρ centered on the UAV location.

IV. Gradient-Search Algorithm

A. UAV Dynamics

Presuming the UAVs are fixed-wing aircraft, the UAV dynamics are modeled using a Dubins car framework [29]. 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. The constraints on turn rate and speed are enforced using the saturation function.

The unconstrained kinematics of UAV $j = 1, ..., O$ are defined by

$$\begin{aligned} \dot{x}_k^j &= S_k^j \cos\theta_k^j \\ \dot{y}_k^j &= S_k^j \sin\theta_k^j \\ \dot{\theta}_k^j &= u_k^j. \end{aligned} \tag{18}$$

By taking derivatives of the $\dot{x}_k^j$ and $\dot{y}_k^j$ terms, assuming unit mass, the dynamics of the UAVs are determined by the forces, X_k^j and Y_k^j, along the x and y directions as follows:

$$\begin{aligned} \dot{S}_k^j \cos\theta_k^j - \dot{\theta}_k^j S_k^j \sin\theta_k^j &= X_k^j \\ \dot{S}_k^j \sin\theta_k^j + \dot{\theta}_k^j S_k^j \cos\theta_k^j &= Y_k^j. \end{aligned} \tag{19}$$

Solving for $\dot{S}_k^j$ and $\dot{\theta}_k^j$ yields

$$\begin{aligned} \dot{\theta}_k^j &= \frac{Y_k^j \cos\theta_k^j - X_k^j \sin\theta_k^j}{S_k^j} \\ \dot{S}_k^j &= X_k^j \cos\theta_k^j + Y_k^j \sin\theta_k^j. \end{aligned} \tag{20}$$

Using Euler's method [30] and applying saturation models yields

$$\begin{aligned} \theta_k^j &= \theta_{k-1}^j + \mathrm{sat}\left(\frac{Y_k^j \cos\theta_k^j - X_k^j \sin\theta_k^j}{S_k^j}, \dot{\theta}_{max}\right)\Delta t \\ S_k^j &= \mathrm{sat}\left(S_{k-1}^j + (X_k^j \cos\theta_k^j + Y_k^j \sin\theta_k^j)\Delta t, S_{max}\right). \end{aligned} \tag{21}$$

$$\text{where } \mathrm{sat}(x, x_{\max}) = \begin{cases} x, & |x| \leq x_{\max} \\ x_{\max}, & x > x_{\max} \\ -x_{\max}, & x < -x_{\max}. \end{cases} \tag{22}$$

B. Likelihood gradient and Pauli repulsion forces

Each UAV's motion plan is prescribed by a combination of three artificial potentials to guide it up the gradient in likelihood ratio while preventing 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. (These assumptions may be relaxed, but that is not the objective of this work.)

The first force is derived from the maximum gradient of the log-likelihood graph in a limited field of view. Although the UAV has global knowledge of the nodes that compose the likelihood network, a finite field of view with radius ρ is adopted to allow the UAV to navigate using the local maximum gradient. Without this restriction, the maximum gradient might be extracted from anywhere in the network and the resulting gradient force might send the UAV off the road network entirely.

Let $\mathcal{N}_k^j$ be the set of indices of all vertices in sensor range of a UAV j at time k corresponding to the non-zero row entries in the corresponding columns of B. The likelihood of all nodes in range for a particular UAV is represented by $\phi(\mathcal{N}_k^j)$. To find the gradient of the graph, the distance between connected nodes is assumed to be constant. The gradient magnitude is the difference between adjacent nodes and its orientation is along the edge that connects them, which can be extracted from the incidence matrix B as follows.

The edge-wise likelihood differences $\Delta\phi_k$ at time k are

$$\Delta\phi_k = B^T \phi_k. \tag{23}$$

Let $\mathcal{M}_k^j$ be the set of row indices of B^T corresponding to the edges that connect to nodes inside the search range of the UAV and $\mu_k^j \in \mathcal{M}_k^j$ represent the index of the edge with maximum likelihood change. The maximum difference in likelihood along an edge in $\mathcal{M}_k^j$ is $\Delta\phi_k(\mu_k^j)$. If the maximum

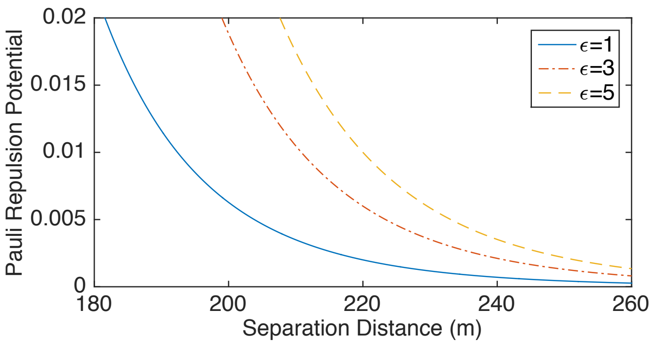


Fig. 3: Pauli repulsion with $\sigma = 2\rho$

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 contains the positions for each 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 difference operator B^T acting on ϕ_k finds the difference between connected nodes n_1 and n_2 by subtracting n_2 from n_1. If the difference is positive, the gradient points from n_2 to n_1 along edge μ_k^j and vice versa if the distance is negative. The likelihood gradient ∇R_k^j is thus

$$\nabla R_k^j = \Delta\phi_k(\mu_k^j)\frac{\psi(n_1)-\psi(n_2)}{\left\|\psi(n_1)-\psi(n_2)\right\|}. \tag{24}$$

The second artificial force is the gradient of the repulsive portion of the Lennard-Jones potential [31], known as Pauli repulsion. The Lennard-Jones potential is typically used as a computationally efficient way to model intermolecular gas dynamics, and Pauli repulsion in particular describes repulsion between molecules as their electron orbitals overlap. We utilize Pauli repulsion because it is tunable for avoiding collisions and redundant searching.

Recall $\xi_k^j = [x_k^j, y_k^j], j = 1, ..., O$. The Pauli repulsion potential for UAV j is

$$P_k^j = 4\epsilon\sum_{i\neq j}^{O}(\sigma^{12}||\xi_k^j - \xi_k^i||^{-12}), \tag{25}$$

where ϵ is the depth of the well and σ is the distance at which the potential between two UAVs is zero. An example of the potential between two UAVs is shown in Figure 3. Note that as the distance between UAVs becomes large, the potential becomes very flat, which implies that repulsive interactions only occur when UAVs are close. The gradient of (25) is

$$\nabla P_k^j = -48\sum_{i\neq j}^{O}\left(\sigma^{12}||\xi_k^j - \xi_k^i||^{-13}\right)\frac{\xi_k^j - \xi_k^i}{||\xi_k^j - \xi_k^i||}, \tag{26}$$

where ϵ is set to one for proportionality to the likelihood gradient force, and σ is set to twice the search radius ρ to make repulsion occur only when UAVs have overlapping search radii.

The coupling of maximum gradient force and Pauli repulsion prevents multiple UAVs from approaching the same node. As the likelihood surface updates with negative measurements in the search radius of a UAV, the gradient towards that portion of the graph decreases and any other local UAVs have less incentive to approach. Any UAVs with coincident paths are also diverted due to Pauli repulsion. As a result, only UAVs approaching the same node from separate paths will come into close proximity and will be diverted either by the local gradient updating away from the common node as measurements of the common node are collected or by Pauli repulsion.

The third force is a spring potential connecting the UAV to the node of higher likelihood along the edge of maximum gradient, i.e., $n_{max} \in \{n_1, n_2\}$ such that $\phi_k(n_{max})$ is greatest. The spring potential is used to counteract drift induced by sudden changes in the gradient direction. If the node of interest goes out of sensor range, the spring force acts on the UAV and brings the UAV closer to the nodes of interest. The rest length of the spring is set to the sensor range of the UAV to keep the edge of max likelihood change in measurement range, while still allowing the UAV to measure nearby edges, thereby maximizing information collection. The spring potential is

$$Q_k^j = -\frac{1}{2}K\left(||\xi_k^j - \psi(n_{max})|| - \rho\right)^2, \tag{27}$$

and the spring force is

$$\nabla Q_k^j = -K\left(||\xi_k^j - \psi(n_{max})|| - \rho\right)\frac{\xi_k^j - \psi(n_{max})}{||\xi_k^j - \psi(n_{max})||}, \tag{28}$$

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

The net artificial force applied to UAV j is

$$F_k^j = \nabla R_k^j + \nabla P_k^j + \nabla Q_k^j, \tag{29}$$

where the components are internally scaled as described above.

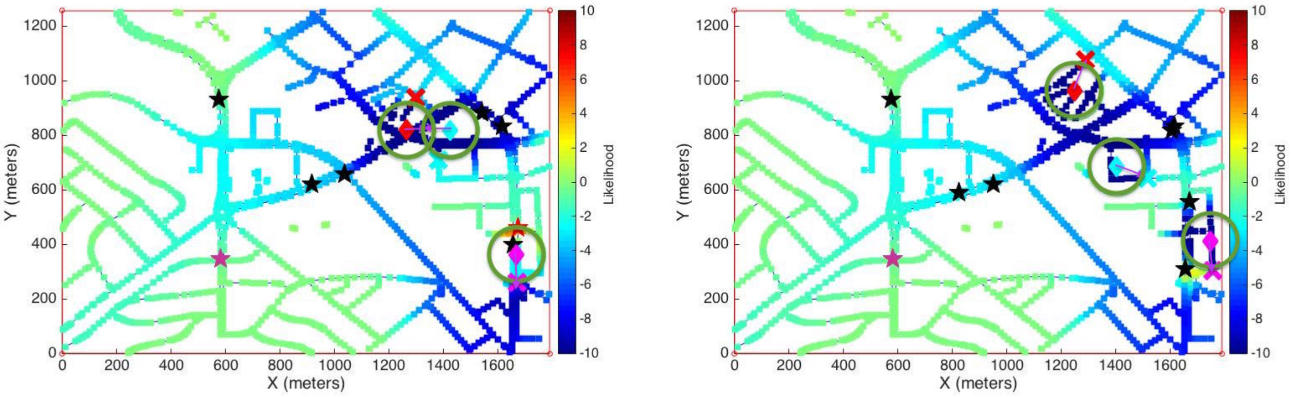
C. Simulation Results

Figure 4 provides a snapshot of the algorithm in simulation. The likelihood along the network is represented graphically by the color of each node. High likelihood is red, neutral likelihood is green, and low likelihood is blue. Targets are distributed throughout the network as colored stars, whereas UAVs and their sensor ranges are represented by colored diamonds and green circles, respectively. Additional parameters used for the simulation are described in Table II.

Figure 4 shows a number of the behaviors described throughout this paper. Focusing first on the magenta UAV

TABLE I: Road network parameters

Parameter (units)	Verizon Center, DC	Silver Spring, MD	New York, NY	Westminster, MD
Area of snapshot (mi^2)	0.863	0.868	0.870	0.887
Convex hull of network (mi^2)	0.817	0.842	0.841	0.846
Number of intersections	510	372	268	138
ID (int/mi^2)	624.356	441.777	318.595	163.175



(a) Time step $k = 230$

(b) Time step $k = 235$

Fig. 4: Snapshots of LRT detection and repulsion

TABLE II: Simulation parameters

Parameter	Value (units)	Definition
σ	240 (m)	repulsive threshold
ϵ	1 (kg m^2/s^2)	Pauli repulsion depth
O	3-6	number of UAVs
T	8	number of targets
Δt	0.3 (sec)	time step
Δx	10 (m)	node spacing
P_d	95%	probability of detection
P_f	10%	probability of false alarm
ϕ_{max}	10	target detection threshold
$\dot{\theta}_{max}$	50 (°/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
ρ	120 (m)	UAV sensor range
C	50	decay term

in Figure 4(a), a red target is just entering the sensor range of the UAV. The UAV raises the likelihood for all nodes in sensor range as indicated by the orange and red nodes. In Fig. 4(b), with the threshold achieved, the red target has changed color to black, indicative of a target detection and transition from track-before-detect to track-after-detect for that target. The black target is now invisible to all UAV sensors since the likelihood surface is used only for track-before-detect. (Another algorithm will track future movements of the detected targets; it is not described here.)

Focusing now on the red and cyan UAVs, their sensor ranges have just overlapped in Figure 4(a). As a result, a repulsion force acts on both UAVs, introducing an additional force into the Dubins car dynamics. This force becomes larger as the UAVs get closer, and eventually leads the two UAVs to turn away from one another while continuing their gradient ascending behavior in Figure 4(b).

UAVs have limited turn rates, so when confronted with two equally large maximum gradients in range, each UAV reacts within its dynamic constraints; this indirectly introduces a tiebreak in the case of multiple edges with the same maximum change in likelihood. As discussed before, if multiple local edges have the same maximum likelihood change, a random edge among that maximum set will be chosen. Even if the random edge is chosen in a direction not aligned with the current heading of the UAV, the saturation introduced in the speed of the UAV, and the limited turn radius, will compel the UAV to move in approximately the same direction. On

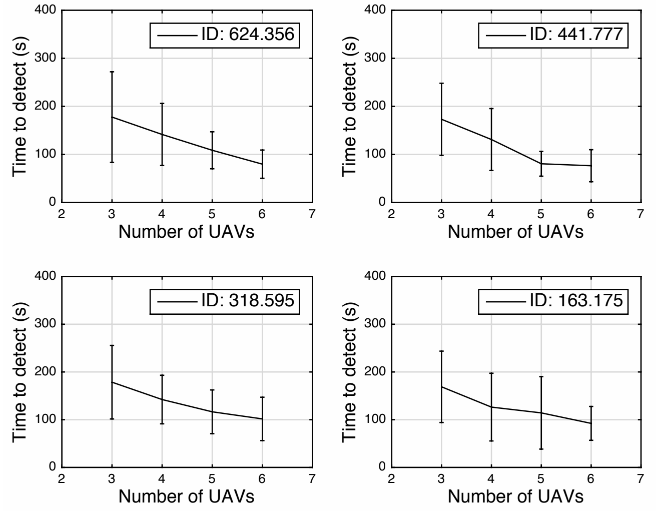


Fig. 5: Number of UAVs vs. Time to detect for constant area

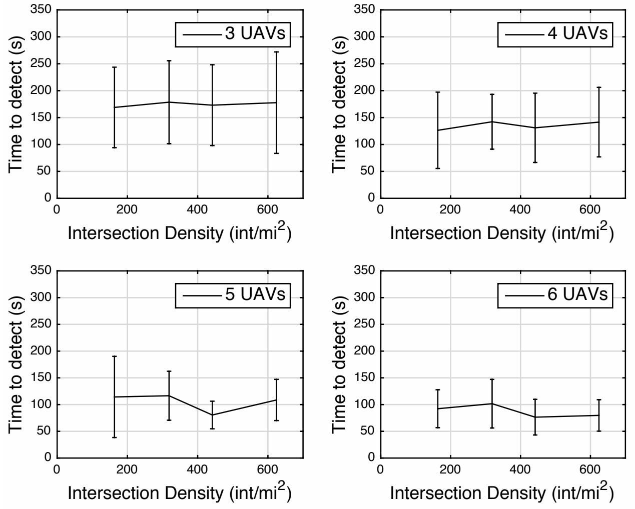


Fig. 6: Intersection Density vs. Time to detect for constant area

the subsequent time step, the UAV's sensor area will have shifted and the edges that are re-evaluated for gradients will have changed as well, causing the UAVs to prefer gradients along their current direction rather than changing direction substantially to pursue a maximum gradient.

Performance of the algorithm on a road network was determined using three metrics: intersection density (ID), time to detect, and number of UAVs. Time to detect represents the time required to detect all targets on the road network and ID is the number of intersections per square mile of a road-network snapshot [32]. The four different road networks in Table I have linearly decreasing intersection density for approximately the same size snapshot. Between 3–6 UAVs were released on each road network to search for targets. Twenty–five Monte Carlo simulations were run using the parameters described in Table II for each UAV configuration, resulting in one hundred trials for each of the four road network snapshots.

Figure 5 shows the relationship between number of UAVs and time to detect for each road network snapshot. As the number UAVs searching increased, the time to detect decreased. Figure 6 shows the relationship between intersection density and time to detect for increasing numbers of UAVs. The time to detect for a particular number of UAVs does not change substantially, indicative of a balanced algorithm that can search spaces of various road complexity without suffering a loss in production. In addition, as the number of UAVs searching increases, the variance in time to detection decreases. This result indicates that the algorithm becomes more efficient at detecting targets on similarly sized snapshots as the number of UAVs increases.

V. Conclusion

Multiple mission profiles require strong motion planning algorithms that are autonomous and cooperative. We present a physics-inspired track-before-detect algorithm for multiple UAVs and targets on a road network. UAVs take measurements of target position on the road network and generate log-likelihood ratio estimates. The likelihood surface is diffused throughout possible target locations using the graph Laplacian of the road network graph. UAVs are directed up the maximum local likelihood gradient by a coupled gradient and spring force, while avoiding collisions using Pauli repulsion. Simulations show the algorithm performance for a set of parameters. The next step in the algorithm is introducing a track-after-detection component to UAV's motion planning. In the absence of measurements, likelihood for detected target locations will be propagated throughout the network based on target dynamics. UAVs will be directed to revisit possible detected target locations when a threshold in uncertainty in target position is reached.

Acknowledgment

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 control algorithm. Brett Barkley is supported by the University of Maryland L-3 Graduate Scholarship. This research is also partially supported by Air Force Office of Scientific Research grant No. FA95501310162.

References

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

[2] F. Darema, "Dynamic data driven applications systems: A new paradigm for application simulations and measurements," in *4th Conf. Proc on Computational Science-ICCS*. Springer, 2004, pp. 662–669.

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

[4] D. Clark and J. Bell, "Bayesian multiple target tracking in forward scan sonar images using the phd filter," *IEEE Conf. Proc. on Radar, Sonar and Navigation*, vol. 152, no. 5, pp. 327–334, October 2005.

[5] C.-C. Huang and S.-J. Wang, "A Bayesian hierarchical framework for multitarget labeling and correspondence with ghost suppression over multicamera surveillance system," *IEEE Conf. Proc. on Automation Science and Engineering*, vol. 9, no. 1, pp. 16–30, Jan 2012.

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

[7] J. Kim and Y. Kim, "Moving ground target tracking in dense obstacle areas using UAVs," *Proc. of the 17th IFAC World Congress*, vol. 1, 2008.

[8] M. A. Peot, T. W. Altshuler, A. Breiholz, R. A. Bueker, K. W. Fertig, A. T. Hawkins, and S. Reddy, "Planning sensing actions for UAVs in urban domains," in *SPIE Proc.*, vol. 5986, 2005, pp. 59 860J–59 860J–10.

[9] F. Rafi, S. Khan, K. Shafiq, and M. Shah, "Autonomous target following by unmanned aerial vehicles," in *Defense and Security Symposium*. International Society for Optics and Photonics, 2006, pp. 623 010–623 010.

[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] M. Ulmke and W. Koch, "Road-map assisted ground moving target tracking," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 42, no. 4, pp. 1264–1274, October 2006.

[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. 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.

[16] S. Gattein, B. Pannetier, and P. Vannoorenberghe, "Analysis and integration of road projection methods for multiple ground target tracking," in *Proc. 8th Conf. on Information Fusion*, vol. 1, July 2005, pp. 227–234.

[17] C. Kreucher, A. Hero, and K. Kastella, "Multiple model particle filtering for multitarget tracking," in *the Twelfth annual workshop on adaptive sensor array processing. Lexington, MA*, 2004.

[18] R. Gayle, W. Moss, M. Lin, and D. Manocha, "Multi-robot coordination using generalized social potential fields," in *Proc. IEEE Conf. on Robotics and Automation*, May 2009, pp. 106–113.

[19] M. Schwager, D. Rus, and J.-J. Slotine, "Unifying geometric, probabilistic, and potential field approaches to multi-robot deployment," *The International Journal of Robotics Research*, vol. 30, no. 3, pp. 371–383, 2011.

[20] OpenStreetMap, "Openstreetmap," 2016. [Online]. Available: https://www.openstreetmap.org

[21] B. Bollobs, *Modern graph theory*, ser. Graduate texts in mathematics. Springer, 1998.

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

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

[24] 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.

[25] M. M. Haklay and P. Weber, "Openstreetmap: User-generated street maps," *IEEE Pervasive Computing*, vol. 7, no. 4, pp. 12–18, Oct. 2008.

[26] 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.

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

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

[29] 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.

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

[31] 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.

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