ABSTRACT

Title of Thesis: Multi-target Detection, Tracking,
and Data Association on Road Networks
Using Unmanned Aerial Vehicles

Brett E. Barkley, Master of Science, 2017

Thesis directed by: Professor Derek A. Paley
Department of Aerospace Engineering

A cooperative detection and tracking algorithm for multiple targets constrained to a road network is presented for fixed-wing Unmanned Air Vehicles (UAVs) with a finite field of view. Road networks of interest are formed into graphs with nodes that indicate the target likelihood ratio (before detection) and position probability (after detection). A Bayesian likelihood ratio tracker recursively assimilates target observations until the cumulative observations at a particular location pass a detection criterion. At this point, a target is considered detected and a position probability is generated for the target on the graph. Data association is subsequently used to route future measurements to update the likelihood ratio tracker (for undetected target) or to update a position probability (a previously detected target). Three strategies for motion planning of UAVs are proposed to balance searching for new targets with tracking known targets for a variety of scenarios. Performance was tested in Monte Carlo simulations for a variety of mission parameters, including tracking on road networks with varying complexity and using UAVs at various altitudes.

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

by

Brett E. Barkley

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Master of Science
2017

Advisory Committee:
Professor Derek A. Paley, Chair/Advisor
Professor Robert M. Sanner
Professor Huan Xu

Acknowledgments

This thesis would not have been possible without the guidance and support of a large group of people. I will inadvertently forget to mention everyone, so for that I apologize. First of all, I would like to give thanks to my advisor, Dr. Derek Paley, for providing me with the opportunity to conduct this work and for his direction and support throughout the process. My technical writing skills and verbal communication of concepts was not great when I first started and Dr. Paley provided guidance and patience while I was learning. I could not have imagined that I would learn so much in two years, so for that I am grateful.

Thank you to my committee, Dr. Robert Sanner and Dr. Mumu Xu, for providing their time to provide me with feedback and discussion about my research. Additionally a big thank you to Dr. Sanner for his no-nonsense approach to teaching control theory. I decided to focus on control systems in large part due to your undergraduate controls class and I have been very happy with that decision as I have learned more from you, Dr. Paley, and Dr. Imraan Faruque.

I couldn't have even made it to this point without the help of all those in the Aerospace department and office, especially Tom Hurst, Aileen Hentz, LaVita Williams, Otto Fandino, Becky Sarni, and all those working behind the scenes. I am especially grateful to Aileen, who has been an invaluable resource and exceptionally kind from when I first transfered to the University of Maryland, to now as I prepare to complete graduate school.

Thanks to Greg Malling, Eddie Tolliver, and all my past coaches who taught

me what hard work can do for me and how to take ownership of success and failure. I would not have made it to this point academically, or more importantly as a human being without you. To all those who have coached me and rolled with me at Crazy 88 Mixed Martial Arts, thanks for giving me an outlet that I could always count on to clear my head and refocus myself.

To all my fellow CDCL members, especially Will Craig, Brian Free, Debdipta Goswami, Frank Lagor, Jinseong Lee, Helene Nguewou, Will Scott , and Daigo Shishika thank you for all the discussions and information you have given me over the past two years. Whenever I had a frustrating problem I really appreciated being able to talk it out with you all.

To all my friends, thank you for being there to balance out the rigors of graduate school. Being able to relax at the end of the week was in large part possible because I had great people to spend time with. To Andrew, James, Jonah, Kevin, Mark, Matt, Rashawn, Ryan, Scott, Victor, and the rest of my football family, thank you for all the support and weird times throughout the years. Whether it be talking in school parking lots in the middle of the night, or digging holes for no apparent reason, it was always a pleasure.

To my family, thank you for your willingness to listen to my digressions into whatever the problem of the day was. To my mother and father, Pam and Brian, thank you for providing me with such an excellent upbringing. Whether it was indulging my fleeting new hobbies or just taking me to the arcade to play Tekken, you were always loving and supportive to me and for that I am immensely grateful. To my siblings, Leigh and Christine (Kibbie), thank you for disagreeing with me so often when we were kids and teens. I learned so much from you in our arguments and I really appreciate you both. To my aunt and uncle, Kathy and Chuck, thank you for the conversation and hospitality whenever I visited you all in Arizona. I will never forget all of the adventures (and food) we had together. To my cousin and his wife, Adam and Kelly, thank you for all of the support and friendship you have shown me throughout the years, as well as allowing me to live with you for a summer. It has been such a pleasure visiting with you all and discussing anything from dogs to NWA albums.

Finally, this research would not have been possible without the support of the L-3 Graduate Scholarship and the efforts of Brian Funk, Dallas Hopper, and Jerry Peterson. I really appreciate all the discussions and guidance you provided me with as I have progressed through this work.

Table of Contents

List of Figures

Chapter 1: Introduction

1.1 Background and Motivation

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 [2] citing the specific need for target identification and designation in the realm of UAV reconnaissance. 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 thesis' goals are twofold. First, to implement 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 [3], which uses sensor measurements to guide subsequent data collection. The second goal is to extend this methodology for target detection on road networks using a Bayesian likelihood tracker [4,5] to the dual problem of cooperatively searching for and tracking targets after detection. Expected locations for detected targets on the road network are represented as a probability density and new observations of each target update the expected location after being appropriately associated to a particular target using an earth mover's distance [6] similarity metric. The UAVs search the road network using one of three 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 with many proposed methodologies. If no constraints are placed on the targets, tracking algorithms occupy the realm of interacting multiple model (IMM) filters described in [7–9] with Kalman filters, extended Kalman filters, and even particle filters used for linear and nonlinear target dynamics models. Other approaches use Bayesian inference and either multi-hypothesis or maximum-likelihood filters to track moving targets [10–13]. 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 [14–16] 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.

By constraining targets to 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) [17,18], which keeps modes in use only as needed. IMM estimators based on particle filters have had success in estimating target dynamics, as in [19–21], but they run into the issue of sharp mode transitions, leading to varying levels of tracking failure [22]. The methods described in [8,23–25] 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 tradeoff between these two attributes, and questionable robustness to motion model violations [22].

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 [26], dynamic programming [27, 28], and maximum likelihood [29, 30] 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 [10].

1.2 Problem Formulation

This thesis focuses on methodologies for 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 by updating the shared likelihood surface that represents likely target locations based on a recursive Bayesian likelihood ratio tracker (LRT) [5]. 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 [6, 31].

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 [32]. 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.

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 on the road network. 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 and attempt to detect targets.

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 distributions are sufficiently similar, the measurement and distribution are paired and the distribution is updated; otherwise the measurement is used to update the likelihood network.

Three motion-planning algorithms guide UAVs to balance their search and track functions. The first strategy, loiter, has each UAV loiter on the peak probability in the nearest distribution to keep the target location estimate as accurate as possible. The second strategy, search-and-loiter, has each UAV search for new targets and only revisit a tracker probability distribution when its peak probability drops below a threshold. The third strategy, search-and-reacquire, formulates the problem of reacquiring the target as an optimal search problem and seeks to maximize the probability of relocalizing the target in finite time. The first two strategies were tested in Monte Carlo simulations, but the third strategy was not due to time constraints on this thesis. Suggestions for improvements to all three strategies and methods for testing the third strategy in simulation are offered as future directions of this work.

1.3 Contributions

1. The formulation of a physics-inspired motion-planning system for cooperative search and track 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. Target detections and subsequent measurements are then assimilated using a Bayesian filter and diffused using the graph Laplacian once again. Both the target position likelihood and detected target's probable positions are used to facilitate cooperation between the motion planning of the UAVs. Potential-based algorithms similar to this paper, such as [33, 34], suffer from a number of issues. Chief among them is getting stuck in local minima of the potential. Our algorithm is novel in that it avoids this issue by combining the motion planning strategy with the evolution of the likelihood network via the LRT. 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, 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 [13].

2. The prediction of likelihood and probability in the absence of measurements

is a novel contribution to target tracking constrained to a road network. 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.

3. Usage of Djikstra's algorithm paired with Earth mover's distance to compare distributions on a road network graph when performing data association. 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 update for the tracker.

4. Created three potential based algorithms for revisiting detected targets on a road network. The loiter algorithm loiters over the peak of an assigned tracker, the search-and-loiter algorithm searches for targets and then returns to loiter over the peak of the tracker, and the search-and-reacquire formulates the reacquisition of targets as a optimal search problem.

5. Usage of the intersection density metric to quantify algorithm performance for road networks of varying complexity.

1.4 Outline of Thesis

The thesis is organized into the following sections. Chapter 2 summarizes the fundamentals of graph theory and some structures that are relevant to tracking on a road network graph. Chapter 3 introduces the likelihood-ratio tracker and discusses implementing it on a road network for gathering information about multiple undetected targets. Chapter 4 explains the usage of a Bayesian filter for tracking detected targets on a road network and data association using Earth mover's distance. Chapter 5 explains the model for the UAVs, the artificial potentials that guide UAV search and track behavior, and how the potentials are combined to implement three search and track strategies. Chapter 6 provides results for Monte Carlo simulations that validate the performance of the proposed search and track algorithms for a variety of mission profiles. Chapter 7 summarizes the paper and provides an overview of future research directions.

Chapter 2: Road Network Graph

2.1 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 [35]. A road graph is composed of three elements [36]: 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 2.1, represented by

$$G = (V, E) \tag{2.1}$$

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

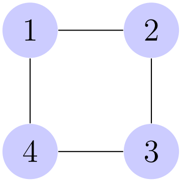


Figure 2.1: An undirected graph with N=4 nodes.

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.,

$$a_{ij} = \begin{cases} 0, & \text{if } j = i \\ 1, & \text{if there is a directed edge from j to i} \\ 0, & \text{if there is no directed edge from j to i.} \end{cases} \tag{2.2}$$

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:

$$d_{ij} = \begin{cases} \sum_{j=1}^{N} a_{ij}, & \text{if } i = j \\ 0, & \text{if } i \neq j. \end{cases} \tag{2.3}$$

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 [36]. 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 2.1, 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.)

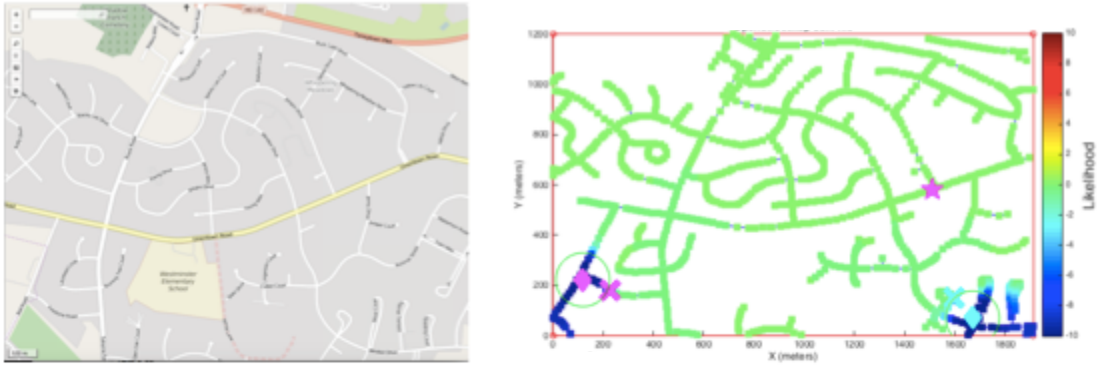


Figure 2.2: Real road network converted to a likelihood network.

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

$$L = D - A. \tag{2.4}$$

The graph Laplacian matrix is used throughout as an operator that describes the diffusion of information (described by a random walk) between neighboring nodes as described by the graph structure. If the information at each node is assumed to be the likelihood that a target is present, the rate of diffusion is determined by the target speed to provide a realistic spreading of target likelihood along the network in time. An example of a real road network in College Park, MD converted into a likelihood network is provided in Figure 2.2. Targets are colored stars, UAVs (with finite field of view denoted by a green circle) are colored diamonds, and the current UAV waypoints are denoted as colored x's.

2.2 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 $\phi \in \mathbb{R}^N$ 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 as

$$\frac{d}{dt}(\phi) + \alpha \nabla^2 \phi = 0, \tag{2.5}$$

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 [37].

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 2.1, the spatial rate of change of likelihood would be represented as

$$\nabla^2(\phi_k(1)) = \frac{\phi_k(2) + \phi_k(3) + \phi_k(4) - 3\phi_k(1)}{h^2}, \tag{2.6}$$

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

$$\frac{d}{dt}(\phi_k(n)) = -\alpha \sum_{j \in \mathcal{N}(n)} a_{nj}(\phi_k(n) - \phi_k(j)). \tag{2.7}$$

In matrix-vector notation, equation (2.7) becomes

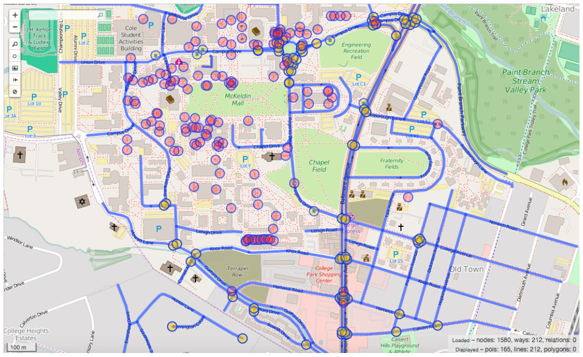
$$\frac{d}{dt}(\phi_k) + \alpha L \phi_k = 0, \tag{2.8}$$

which is the heat equation with spatial discretization. The Laplacian matrix L implements the finite difference calculation of the Laplacian operator ∇^2 [38]. Thus the diffusion of likelihood throughout the road network represented by a graph is found by solving the first-order matrix differential equation in (2.8).

The graph Laplacian matrix of a connected undirected graph is positive semi-definite [38]. 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; equation (2.8) reaches an equilibrium that is the average of the initial likelihood. The use of the Laplacian matrix as a method for target position motion modeling in both the likelihood-ratio tracker (for undetected targets) and probabilistic trackers (for detected targets) is further described in Sections 3.2 and 4.2.

2.3 OpenStreetMap Data Structure and Importing

OpenStreetMap data is exported for a particular map snapshot using the Overpass Turbo web-based data mining tool [39]. 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, the types of paths that are exported are restricted as shown in Figure 2.3. 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



Loaded – nodes: 1580, ways: 212, relations: 0
Displayed – pois: 165, lines: 212, polygons: 0

Figure 2.3: OpenStreetMap snapshot

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 [40].

Chapter 3: Likelihood Ratio Target Detection on a Graph

The instantaneous position of likely target locations prior to detections is found using a log-likelihood ratio tracker (LRT) [13]. 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 [4] and [5].

3.1 Likelihood Ratio Tracker

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 [13]

$$p(\xi_k|\zeta_{k-1}) = \int_{\Omega} p(\xi_k|\xi_{k-1})p(\xi_{k-1}|\zeta_{k-1})d\xi_{k-1}. \tag{3.1}$$

The measurement update is proportional to the product of the measurement likelihood $p(\zeta_k|\xi_k)$ and the predicted state [13], i.e.,

$$p(\xi_k|\zeta_k) = \frac{p(\zeta_k|\xi_k)p(\xi_k|\zeta_{k-1})}{p(\zeta_k|\zeta_{k-1})}, \tag{3.2}$$

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, the measurement likelihood is replaced 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

$$\log \mathcal{L}(\zeta_k|\xi_k) = \log \frac{p(\zeta_k|\xi_k^+)}{p(\zeta_k|\xi_k^-)} = \log(p(\zeta_k|\xi_k^+)) - \log(p(\zeta_k|\xi_k^-)). \tag{3.3}$$

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

$$\begin{gathered} \mathrm{p}(\xi_k|\zeta_k) = \log \frac{\mathcal{L}(\zeta_k|\xi_k)p(\xi_k|\zeta_{k-1})}{p(\zeta_k|\zeta_{k-1})} = \\ \mathrm{p}(\zeta_k|\xi_k^+) - \mathrm{p}(\zeta_k|\xi_k^-) + \mathrm{p}(\xi_k|\zeta_{k-1}) + \mathrm{p}(\zeta_k|\zeta_{k-1}). \end{gathered} \tag{3.4}$$

The first term in (3.4) 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.

3.2 Likelihood-Ratio Tracker Prediction Step

The predict step (3.2) involves updating the target probability density function in the absence of measurement information based on a model of the target motion. The graph representing the road network allows the specification of requirements that constrain target motion to remain on the graph. For disparate target types in a track-before-detect LRT framework, a motion model that distributes likely target positions in a broad way was considered to be a conservative, but safe assumption. Although the model may not localize targets with high accuracy in the absence of measurements, possible target locations will at least be contained within the likelihood distribution. The motion model used to fulfill this requirement was a random walk on a graph.

A random walk model is described by the diffusion in (2.8). The diffusivity coefficient of targets on the network can be found by releasing an ensemble of particles constrained to random walk on a line with the speed properties of the target. A gaussian fit of the particle distribution on the line at a particular time t can found. The mean square displacement of the fit is related to the diffusion of the targets according to the Einstein-Smoluchowsky equation for a one-dimensional gaussian [41]

$$MSD = 2\alpha t. \tag{3.5}$$

Solving the Einstein-Smoluchowsky equation for α provides a measure of diffusivity for a target on a line that can be extended to the road network graph.

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

$$\frac{d}{dt}(\phi_k) + \alpha L \phi_k + \frac{\alpha}{C}\phi_k = 0. \tag{3.6}$$

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:

$$\frac{d}{dt}(\phi_k) = \left(-\alpha L - \frac{\alpha}{C} I\right)\phi_k, \tag{3.7}$$

which implies

$$\phi_k = e^{-\alpha(L + C^{-1}I)\Delta t}\phi_{k-1}. \tag{3.8}$$

Eq. (3.8) 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.

3.3 Likelihood-Ratio Tracker Measurement Update Step

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 [42]. Combining these two probabilities, the sensitivity m of each sensor is

$$m = z(P_d) - z(P_f), \tag{3.9}$$

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

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

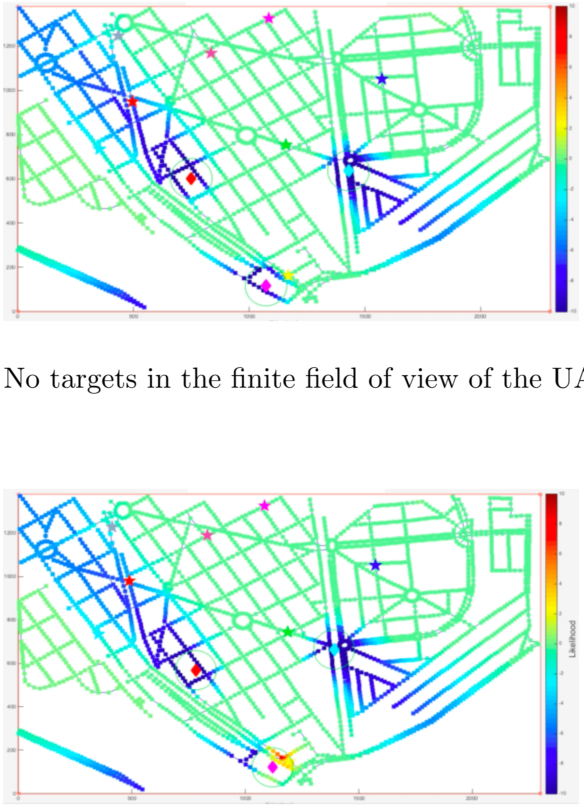
For example, $P_d = 0.95$ and $P_f = 0.1$ yields $m = 2.92$. Let w_k represent unit-normal measurement noise in standard deviation units at time step k. When the target is absent the measurement data is $\zeta_k = w_k$, whereas when the target is present, the measurement data is $\zeta_k = m + w_k$. Assuming a zero-mean Gaussian sensor model yields [43]

$$p(\zeta_k|\xi_k^-) = \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{\zeta_k^2}{2}\right) \tag{3.10}$$

$$p(\zeta_k|\xi_k^+) = \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{(\zeta_k - m)^2}{2}\right). \tag{3.11}$$

The log-likelihood ratio (3.3) becomes

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



(a) No targets in the finite field of view of the UAVs

(b) One target in the finite field of view of the magenta UAV

Figure 3.1: Updates being provided by the LRT sensor measurement model.

where m is a function of the sensor P_d and P_f given by (3.9). Note, the log-likelihood ratio is applied to the prior located inside a disc of radius ρ centered on the UAV location. The application of measurements to update the likelihood network is shown in Figure 3.1. Recall that targets are colored stars and UAVs (with finite field of view denoted by a green circle) are colored diamonds. The red and cyan UAVs observe no targets between the two time steps and lower likelihood in their field of view according to the log-likelihood ratio. The magenta UAV observes a target between the two images and raises likelihood in its field of view.

Chapter 4: Bayesian Tracking on a Graph

Bayesian track-after-detect filters are initialized on the road network for each detected target when the likelihood network reaches a critical threshold ϕ_{max}. The new tracker is instantiated with a normalized distribution based on the previous LRT measurements of the likely 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 chapter explores the procedure used to generate and update probability distributions from target detections and how to properly associate new measurements using the earth mover's distance metric.

4.1 Target Detection and Bayesian Tracker Instantiation

The Bayesian track-after-detect filters in this thesis are probability distributions that are restricted to the nodes of the road network graph. Each target detection leads to an additional track-after-detect filter being instantiated on the road network wherein each node has an associated probability of the newly detected target being present. As a result, future measurements of a detected must be routed

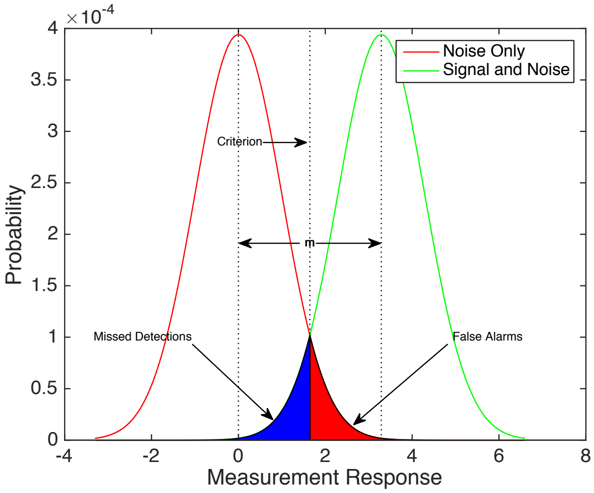
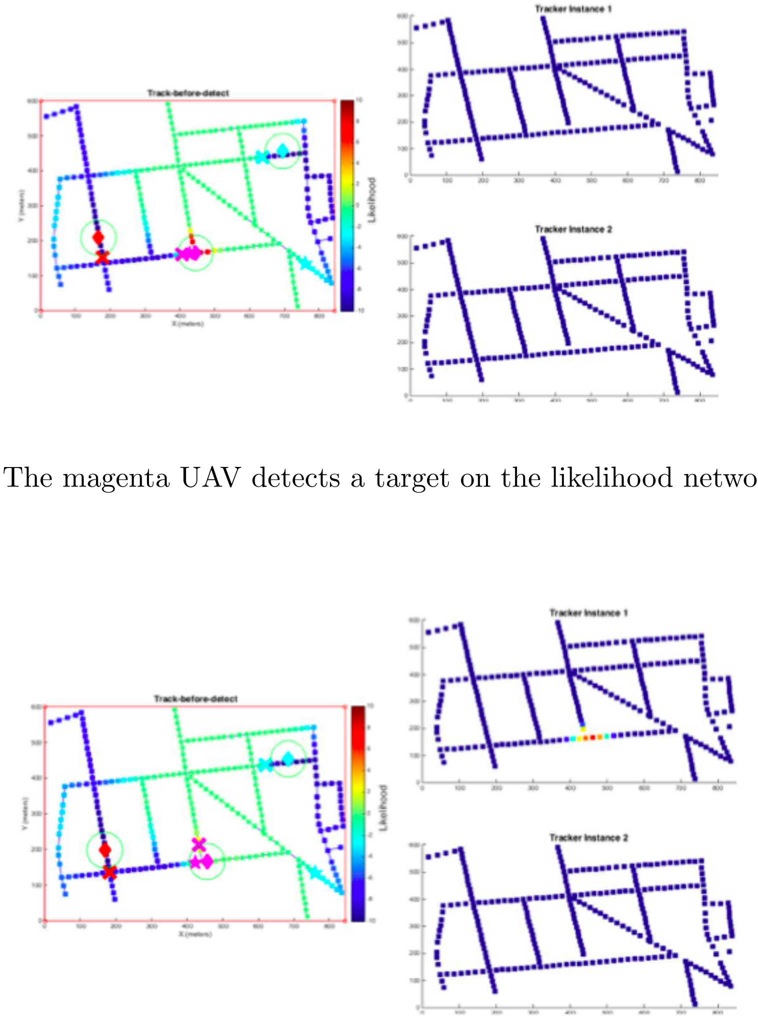


Figure 4.1: Probability of Occurrence plot with $P_d = 0.95$ and $P_f = 0.05$.

to one of the instantiated trackers to keep the probability distribution accurate in localizing the target location.

To prevent superfluous target measurements from entering the data association process, a criterion of $c = m/2$ is introduced, where m is given by equation (3.9), as shown on the probability of occurrence graph in Figure 4.1. This choice represents the intersection of the probability of occurrence curves for noisy measurements with and without signals and determines how often false alarms and missed detection occur.

After a detection, the initial track probability for a target is formed using the likelihood obtained from all connected nodes that are above ϕ_{max}. These connected nodes' positions and likelihood distribution (normalized to one) form the basis for a new tracker on the road network. The connected nodes from the likelihood surface are then suppressed to zero likelihood as shown in Figure 4.2.



(a) The magenta UAV detects a target on the likelihood network.

(b) A Bayesian target tracker is instantiated.

Figure 4.2: Tracker instantiation upon detection of a target on the likelihood network.

4.2 Target Tracker Prediction Step

Once a target is detected, the problem of determining a motion model that will predict the motion of the target in the absence of direct measurements is reintroduced. In the case where agile UAVs or other sensor platforms are available to provide continuous measurement updates of the target location, an accurate motion model in the Bayesian tracker is less important. However, in the case where there are fewer UAVs than targets and no additional tracking resources available, UAVs need to leave detected targets and search for undetected ground vehicles. As a result, a mismatch in the motion model and the actual target dynamics can be disastrous when attempting to reacquiring targets.

A number of potential solutions to achieving balance between searching and revisiting targets is covered further in Chapter 5, but all require that the motion model match the target dynamics relatively well, or at least encompass the worst cases of the target motion. To achieve this goal, the tracker prediction step utilizes the random walk on a graph described by the graph Laplacian to provide a conservative estimate of the target location. Unlike the random walk motion model in the LRT, the model for the tracker does not require any consensus at an equilibrium value indicating neither low nor high probability after a long period of time. Instead, if the maximum probability in the tracker has dropped below some threshold, P_{min}, the tracker is dissolved and the track probability becomes zero. The diffusion of probability utilizes the matrix vector form in Equation 2.8 and can be solved with a matrix exponential as follows:

$$\frac{d}{dt}(\phi_k) = -\alpha L \phi_k, \tag{4.1}$$

which implies

$$\phi_k = e^{-\alpha L \Delta t} \phi_{k-1}. \tag{4.2}$$

4.3 Target Tracker Measurement Update and Data Association

After a detection, the initial track probability is formed and updated using the motion update described in 4.2. The next step is to update the distribution with new measurements of the tracked target. Target measurements all include a localization of the target position in addition to the measurement provided by the log-likelihood ratio tracker. 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.)

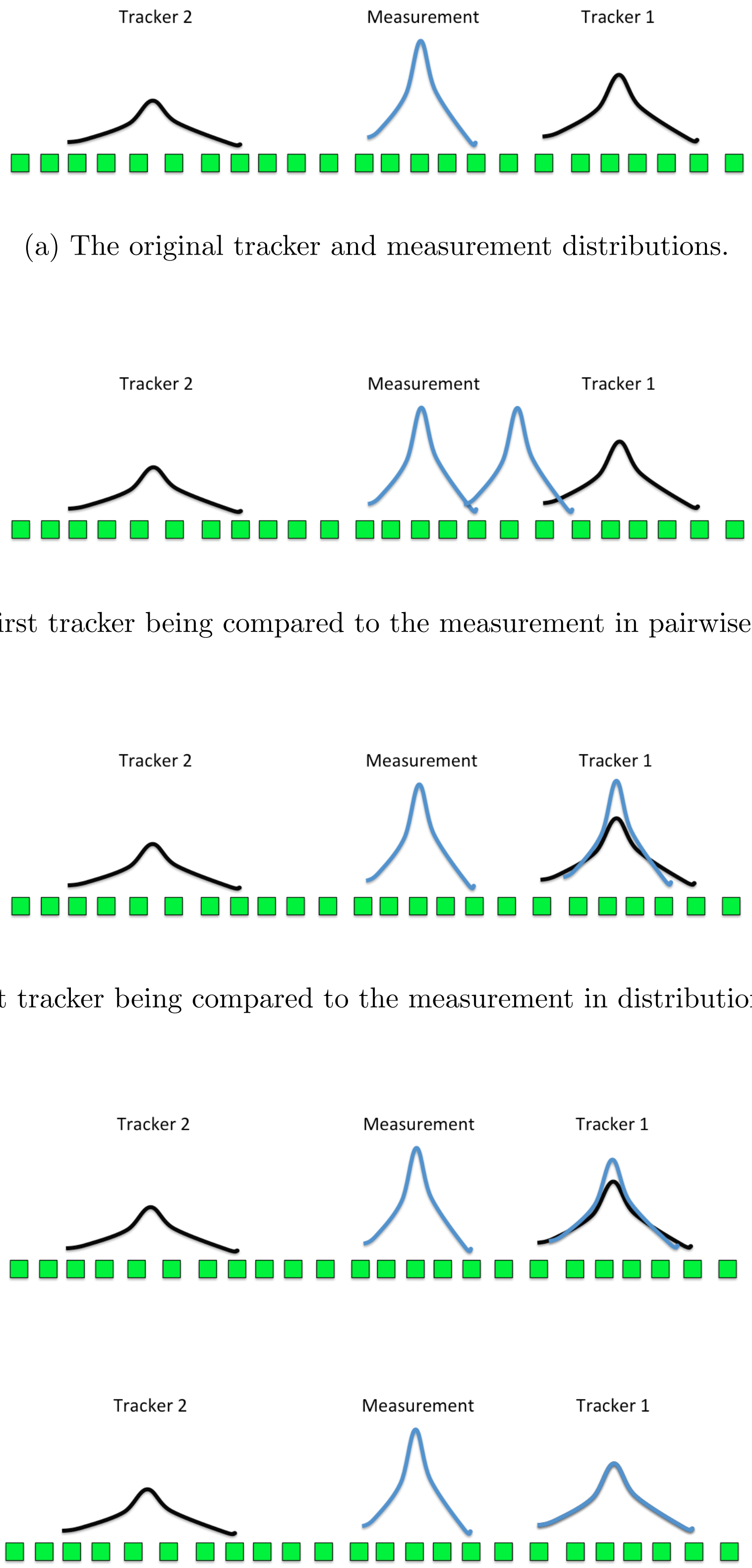
Recall that ψ provides locations of the nodes that compose the road network and s is the standard deviation of the position measurement noise. The measurement probability density gathered by UAV j at time k is thus

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

As new measurements are generated, each U_k^j needs to update a single tracker, or the log-likelihood ratio must be used to update the likelihood network. Determining the correct update is known as the data association problem [13] and can be formalized into a transportation problem when the measurements and distributions are constrained to a road network.

Earth mover's distance (EMD) is a solution to the transportation problem introduced by Rubner, Tomasi, and Guibas [6]. 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. For target tracking on a road network the signatures are the probability densities of the instantiated trackers, P_k^l, and the measurements produced by UAV j at time k, $\hat{U}_k^j$. The idea is that if any measurement and tracker instantiations are similar enough to one another (based on a comparison threshold) then the most similar tracker and measurement pair are associated and the measurement is used to update the tracker. A case of two trackers being compared to a a measurement distribution using a EMD transportation model is depicted in Figure 4.3. In the case depicted, neither tracker instantiation would receive the measurement update because the comparison threshold was not reached. The measurement would instead be used to update the likelihood network.

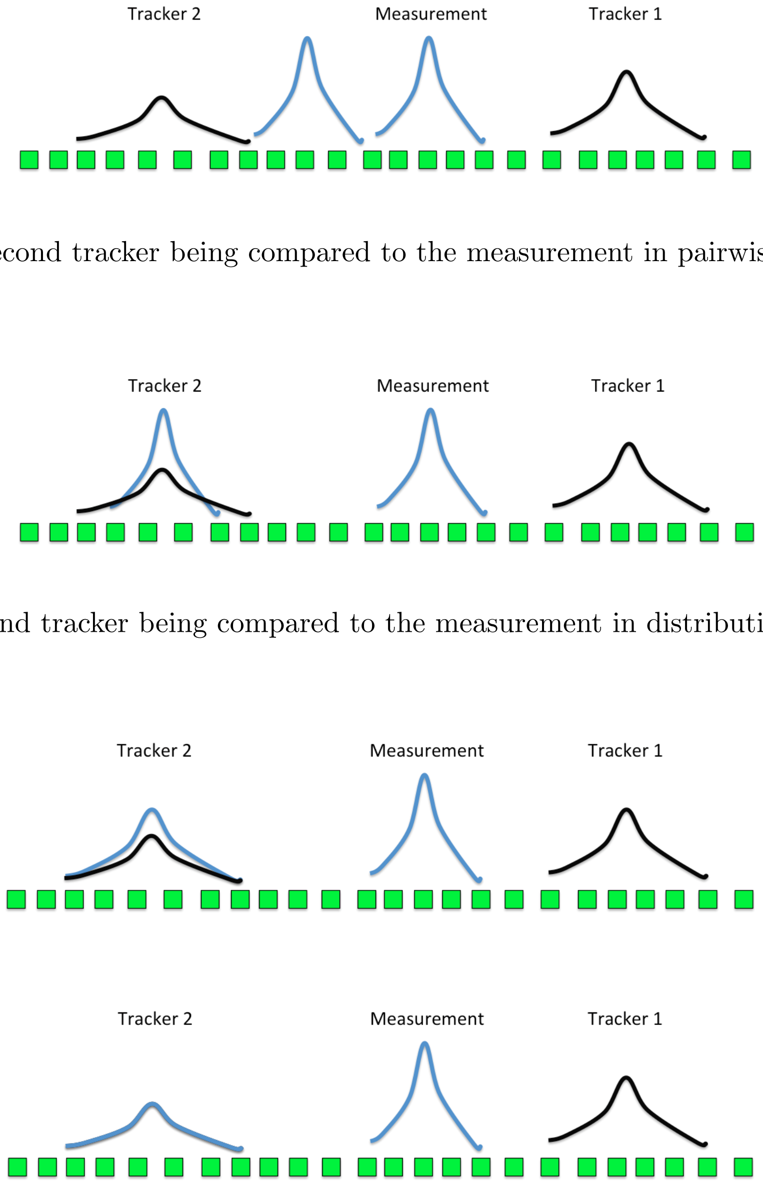
A linear programming problem can be formalized for the case of moving probability on a road network as follows. Let P be the first signature with q elements



(a) The original tracker and measurement distributions.

(b) The first tracker being compared to the measurement in pairwise distance.

(c) The first tracker being compared to the measurement in distribution similarity.



(d) The second tracker being compared to the measurement in pairwise distance.

(e) The second tracker being compared to the measurement in distribution similarity.

Figure 4.3: Depiction of the EMD process.

indexed by i and Q be the second signature with r elements indexed by g [6]. 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 [6]

$$WORK(P, Q, F) = \sum_{i=1}^{q} \sum_{g=1}^{r} \mathbb{d}_{ig} f_{ig}. \tag{4.3}$$

The flow $F = [f_{ig}]$ that minimizes (4.3) can be found when subject to the following constraints for probability distributions with equal total probabilities [44]

$$f_{ig} \geq 0 \qquad 1 \leq i \leq q, 1 \leq i \leq r \tag{4.4}$$

$$\sum_{g=1}^{r} f_{ig} = P(i) \qquad 1 \leq i \leq q \tag{4.5}$$

$$\sum_{i=1}^{q} f_{ig} = Q(g) \qquad 1 \leq g \leq r \tag{4.6}$$

$$\sum_{i=1}^{q} \sum_{g=1}^{r} f_{ig} = \sum_{i=1}^{q} P(i) = \sum_{g=1}^{r} Q(g) = 1. \tag{4.7}$$

Constraint (4.4) requires supplies transferred from P to $Q(g)$ to be nonnegative. Constraint (4.5) ensures that the probability matched to Q is equal to the probability in $P(i)$. Similarly, constraint (4.6) ensures that the probability matched to P is equal to the probability in $Q(g)$; and constraint (4.7) requires that the signature with the most probability be moved, which is known as the total flow [6]. 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 [6, 45]

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

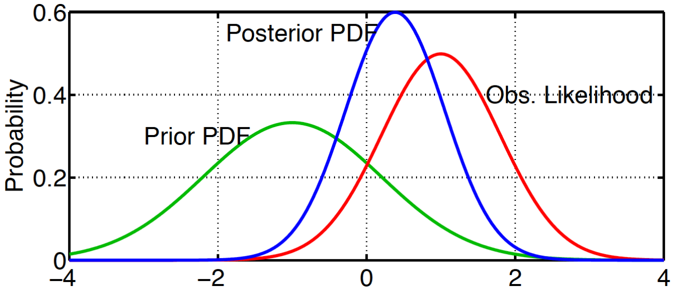


Figure 4.4: Observation (measurement) update being applied for a Bayesian filter.

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 generated as described in Algorithm 1 (see line 11) and is shown visually for a Bayesian filter in Figure 4.4.

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

3.3. This procedure is repeated on each timestep for each UAV.

The posteriors achieved after all of the measurements have been associated appropriately are diffused as in (3.8) and formed into the prior for the next time step i.e.,

$$P^l_{k+1} = e^{-\alpha L \Delta t} \mathbb{P}^l_k. \tag{4.8}$$

Recall that P_{min} is 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.

Algorithm 1 EMD Data Association

```
Require: $\mathcal{I}_k, \mathcal{T}_k^j, \hat{U}_k^j, P_k^l, O, \mathcal{E}$
▷ 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
Repeat for each timestep $k$
for $j = 1 : O$ do
    for $l = 1 : \mathcal{I}_k$ do
        for $q = 1 : \mathcal{T}_k^j$ do
            $DA(l, q) = EMD\left(P_k^l, \hat{U}_k^j(:, q)\right)$
        end for
    end for
    Note the index $l, q$ of min(EMD) in $DA$
    while $DA(\min(EMD)) < \mathcal{E}$ do
        Generate posterior: $\mathbb{P}_k^l = P_k^l \times \hat{U}_k^j(:, q)$
        Set the elements $(:, q)$ and $(l, :)$ in $DA$ to $\mathcal{E} + 1$
        Note the new index $l, q$ of min(EMD) in $DA$
    end while
    Note the number of unused $P_k^l$
    if number of unused $P_k^l > 0$ then
        Update LRT with measurement
    end if
end for
```

Chapter 5: Coordinated Control of UAVs for Target Detection and Tracking

The control algorithms introduced in this thesis for balancing detecting and tracking targets on a road network rely on each UAV being assigned to a probabilistic target tracker. Without much prior knowledge, the location in a tracker that would be expected to provide the highest probability of reacquiring a target is the peak of the tracker distribution. As a result, the protocol linking UAVs to target trackers utilizes each UAV's proximity to the peak of each tracker, which is given by the Euclidean distance between Θ_k^j and $\psi(\max(P_k^l))$.

Let $\mathcal{I}_k$ represent the number of trackers running at time step k. A $\mathbb{R}^{O\times\mathcal{I}_k}$ matrix represents the distance between each UAV and instantiated tracker. Pruning combinations of sums along the pairwise distance matrix that pair a UAV with more than one tracker, the pairing combination that yields the sum of assignments with minimum distance traveled from UAV to tracker peak probability is chosen. The assigned peak probability and index of that probability are $\mathcal{P}_k^{lj}$, indicating that tracker l is linked to UAV j at time k.

With a link established between a tracker and a UAV, three motion planning algorithms were developed to guide the UAVs: loiter, search-and-loiter, and search-and-reacquire. The loiter algorithm seeks to keep the UAV over the peak of the assigned tracker distribution and continuously provide new measurements of the target. This represents one extreme where UAVs exclusively keep trackers accurate and disregard searching for new targets. Search-and-loiter represents a more balanced approached wherein UAVs search for new targets until their assigned target tracker loses localization of the target. At this point, the UAV returns and loiters over the peak of the tracker in an attempt to relocalize the target location. Finally, the search-and-reacquire algorithm formalizes reacquiring detected targets after searching for new targets as an optimal search problem.

5.1 UAV Dynamics

The UAVs considered for this thesis are fixed-wing aircraft modeled using a Dubins car framework [46] as shown in Figure 5.1. 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

$$\begin{aligned} \dot{x}_k^j &= S_k^j \cos \theta_k^j \\ \dot{y}_k^j &= S_k^j \sin \theta_k^j \\ \dot{\theta}_k^j &= u_k^j. \end{aligned} \tag{5.1}$$

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 force, $F_k^j = \left[X_k^j, Y_k^j\right]$ along the x and y directions

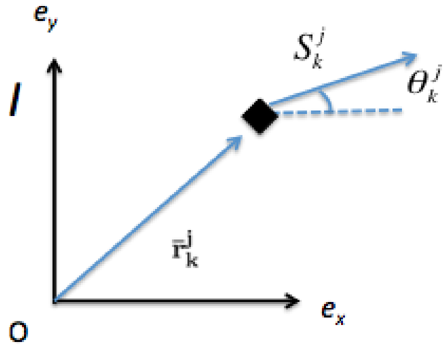


Figure 5.1: Dubins car at fixed altitude in inertial frame, $I = (O, e_x, e_y, e_z)$.

as follows:

$$\begin{aligned} \dot{S}_k^j \cos\theta_k^j - \dot{\theta}_k^j S_k^j \sin\theta_k^j &= X_k^j \\ \dot{S}_k^j \sin\theta_k^j + \dot{\theta}_k^j S_k^j \cos\theta_k^j &= Y_k^j. \end{aligned} \tag{5.2}$$

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

$$\begin{aligned} \dot{\theta}_k^j &= \frac{Y_k^j \cos\theta_k^j - X_k^j \sin\theta_k^j}{S_k^j} \\ \dot{S}_k^j &= X_k^j \cos\theta_k^j + Y_k^j \sin\theta_k^j. \end{aligned} \tag{5.3}$$

Using Euler's method [47] and applying saturation models yields

$$\begin{aligned} \theta_k^j &= \theta_{k-1}^j + \text{sat}\left(\frac{Y_k^j \cos\theta_{k-1}^j - X_k^j \sin\theta_{k-1}^j}{S_{k-1}^j}, \dot{\theta}_{max}\right)\Delta t \\ S_k^j &= \text{sat}\left(S_{k-1}^j + (X_k^j \cos\theta_{k-1}^j + Y_k^j \sin\theta_{k-1}^j)\Delta t, S_{max}\right), \end{aligned} \tag{5.4}$$

$$\text{where sat}(x, x_{\max}) = \begin{cases} x, & |x| \le x_{\max} \\ x_{\max}, & x > x_{\max} \\ -x_{\max}, & x < -x_{\max.} \end{cases} \tag{5.5}$$

5.2 Artificial Potentials and Likelihood 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 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.

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 for 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 likelihood 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

$$\Delta\phi_k = B^T\phi_k. \tag{5.6}$$

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 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

$$\nabla R_k^j = \Delta\phi_k(\mu_k^j)\frac{\psi(n_1) - \psi(n_2)}{\left\|\psi(n_1) - \psi(n_2)\right\|} \tag{5.7}$$

and is shown being applied to a UAV graphically in Figure 5.2. The UAV ascends the likelihood gradient, while feeding new measurements to the likelihood surface, ensuring that a local maximum will not be reached.

The second artificial force is the gradient of the repulsive portion of the Lennard-Jones potential [48], 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. Pauli repulsion is utilized because it is tunable for

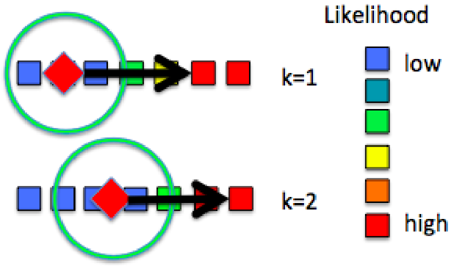


Figure 5.2: The likelihood gradient force vector directing a UAV.

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

$$P_k^j = 4\epsilon \sum_{i\neq j}^{O} (\sigma^{12} ||\xi_k^j - \xi_k^i||^{-12}), \tag{5.8}$$

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 5.3. 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 (5.8) is

$$\nabla P_k^j = -48 \sum_{i\neq j}^{O} \left(\sigma^{12} ||\xi_k^j - \xi_k^i||^{-13}\right) \frac{\xi_k^j - \xi_k^i}{||\xi_k^j - \xi_k^i||}, \tag{5.9}$$

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. This potential is depicted in Figure 5.4 where two UAV's search radii overlap one another and the Pauli repulsion force guides them away from each other.

The coupling of maximum gradient force and Pauli repulsion prevents multiple UAVs from approaching the same node. As the likelihood surface updates with

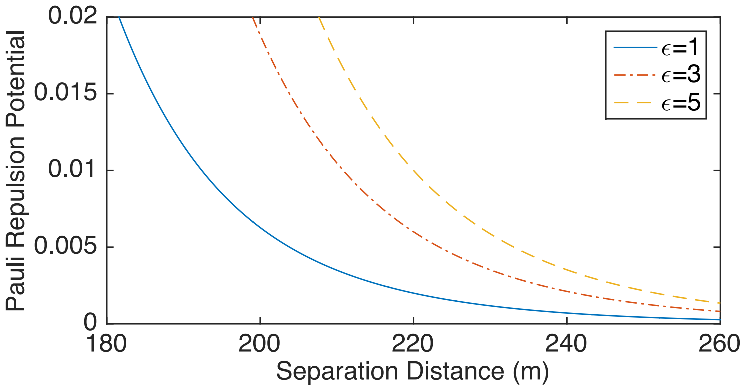


Figure 5.3: Pauli repulsion with $\sigma = 2\rho$

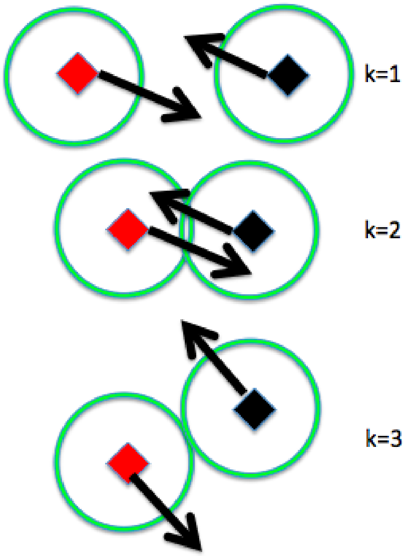


Figure 5.4: Depiction of the Pauli repulsion gradient directing two UAVs.

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 as depicted in Figure 5.5. 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

$$Q_k^j = -\frac{1}{2}K\left(||\xi_k^j - \psi(n_{max})|| - \rho\right)^2, \tag{5.10}$$

and the spring force is

$$\nabla Q_k^j = -K\left(||\xi_k^j - \psi(n_{max})|| - \rho\right)\frac{\xi_k^j - \psi(n_{max})}{||\xi_k^j - \psi(n_{max})||}, \tag{5.11}$$

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

The net artificial force applied to UAV j when ascending the likelihood gradient in search of new targets is

$$F_k^j = \nabla R_k^j + \nabla P_k^j + \nabla Q_k^j, \tag{5.12}$$

where the components are internally scaled as described above.

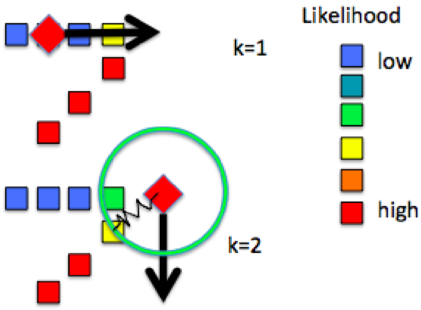


Figure 5.5: Depiction of the spring force directing a UAV back onto the road network after a sudden gradient direction change.

5.3 Loiter Tracking Strategy

The loiter algorithm presents an option for target tracking that works best in the case when the number of UAVs available is greater than or equal to the number of targets. The UAVs simply loiter over targets they detect and no longer search for new targets. In the case of more targets than UAVs, simply loitering over targets doesn't provide an adequate track and search balance, but is a conservative choice to maintain tracks on detected targets.

Having the UAV loiter over the location in the tracker with the highest target probability requires an artificial attractive force between a UAV and the associated tracker. The artificial force is achieved using a spring potential as described for drift reduction in searching for targets, however 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 $\mathcal{P}_k^{lj}$ is

$$\nabla L_k^{lj} = -K\left(||\Theta_k^j - \psi(\mathcal{P}_k^{lj})||\right)\frac{\Theta_k^j - \psi(\mathcal{P}_k^{lj})}{||\Theta_k^j - \psi(\mathcal{P}_k^{lj})||}, \tag{5.13}$$

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

Assigned UAVs no longer search for new targets using the likelihood gradient search and have a modified collision avoidance force. The assigned UAV is no longer actively repulsed from all UAVs, but only from other loitering UAVs. Let $\mathcal{J}$ be the set of all UAVs assigned to trackers. The modified Pauli repulsion force for assigned agents is

$$\nabla P_k^j = -48\sum_{\substack{i\neq j,\\ i\in\mathcal{J}}}^{O}\left(\sigma^{12}||\Theta_k^j - \Theta_k^i||^{-13}\right)\frac{\Theta_k^j - \Theta_k^i}{||\Theta_k^j - \Theta_k^i||}, \tag{5.14}$$

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

The total control force applied from the loiter strategy to UAV j is

$$F_k^j = \begin{cases} \nabla L_k^{lj} + \nabla P_k^j, & j \in \mathcal{J} \\ \nabla R_k^j + \nabla P_k^j + \nabla Q_k^j, & j \notin \mathcal{J}. \end{cases} \tag{5.15}$$

5.4 Search-and-Loiter Tracking Strategy

The search-and-loiter algorithm is a more aggressive, but naive, solution to providing more balance in detecting new targets and reacquiring detected targets. The assigned UAVs temporarily leave their loiter over detected targets when the localization of their assigned target is known with high accuracy via successive measurement updates.

In the absence of sensor measurements, the localization of the tracked target drops due to the motion model updates of the probabilistic trackers. When the loss of localization drops below a defined revisit threshold, UAVs return and loiter over the peak of their assigned tracker in an attempt to reacquire the target.

Just as for loiter, when a tracker l is assigned to an agent j, the peak probability in that tracker is encoded in $\mathcal{P}_k^{lj}$. The assigned UAV continually monitors the peak probability even as it searches for new targets, so the perceived localization accuracy of the tracker can be determined at each time step. As the motion model updates the tracker in the absence of measurements, the probabilistic knowledge of the true target location becomes less accurate. This issue makes new measurements of the target essential to maintaining the health of the tracker.

The loiter strategy solves the problem of tracker accuracy by continually providing the tracker with new measurements, thereby keeping the peak probability high. The search-and-loiter strategy relies more heavily on the accuracy of tracker motion model to maintain its target estimate even in the absence of measurements. A loiter threshold ϱ tells the assigned UAV when the estimated target location has become too inaccurate. So long as $\mathcal{P}_k^{lj}$ remains above ϱ, the UAV continues searching the likelihood network for other targets. When $\mathcal{P}_k^{lj}$ drops below ϱ, the UAV returns to the node of maximum target probability and attempts to relocalize the target and provide an updated $\mathcal{P}_k^{lj}$ that is above ϱ.

The total control force for Search-and-Loiter tracking applied to UAV j is

$$F_k^j = \begin{cases} \nabla L_k^{lj} + \nabla P_k^j, & j \in \mathcal{J} \ \& \ \mathcal{P}_k^{lj} \leq \varrho \\ \nabla R_k^j + \nabla P_k^j + \nabla Q_k^j, & j \in \mathcal{J} \ \& \ \mathcal{P}_k^{lj} > \varrho \\ \nabla R_k^j + \nabla P_k^j + \nabla Q_k^j, & j \notin \mathcal{J}. \end{cases} \tag{5.16}$$

(Note, the same Pauli repulsion rules apply as in the loiter tracking strategy. The term ∇P_k^j changes based on whether the UAV is assigned or not according to Equations (5.9) and (5.14).)

5.5 Search-and-Reacquire Tracking Strategy

Reacquisition of targets is often not as simple as revisiting the location in the tracker that has the maximum probability of a target. When there is a mismatch between the target model and the actual target dynamics, the target dynamics do not fall within a Gaussian distribution confined to the road network, or the target reaches the edges of the Gaussian distribution, revisiting the peak of the tracker distribution often results in a detected target being lost. As a result, a conservative target motion model is beneficial because, although it is not highly accurate at localizing the exact position of a target, it does provide a probability for all possible locations that the target could be. Such is the case for a motion model based on a random walk; rarely will targets follow a random walk in reality, but as long as the random walk diffuses at the speed of the targets, the actual location of the target should fall within the distribution produced in the absence of measurements for a short period of time.

One useful formulation of the reacquisition of a target after an extended search for new targets is the orienteering problem with time windows (OPTW) as first described in [49]. A depiction of one of Marius Solomon's benchmark cases for vehicle routing problems [50] is shown in Figure 5.6.

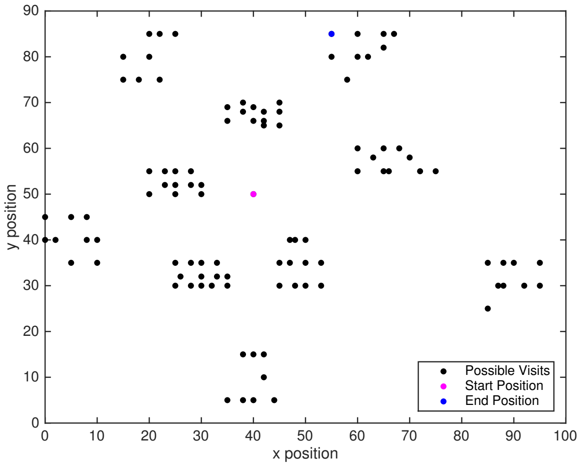


Figure 5.6: Benchmark dataset c101 [1] for 100 nodes from Solomon's test problems.

Each node of the tracker network has a probability of the target being located there (profit) and the time window when profit may be collected at each node can be tailored to provide less constraint than the typical orienteering problem. The orienteering problem solution provides an optimal route for a vehicle that maximizes the collected profit within the constraint of the service time windows of each node. A heuristic developed in [51] was modified to give a locally optimal set of waypoints for the UAV to visit that maximizes the chances of reacquiring the target.

The OPTW takes a set V of N nodes, and assigns each of them a score S_i, a time required for service T_i, and a time window $[O_i, C_i]$ within which the nodes can be serviced. A start and end node $\mathcal{S}$, E (that will no longer have a profit associated with them) are chosen, and the travel time (cost) c_{ig} between various nodes $i, g \in V$ is assumed to be the pairwise Euclidean distance between them divided by the UAV travel speed S_{max}. Due to the time windows and a limited time budget T_{max}, not all locations can be visited. The goal of the OPTW is to find a single route that maximizes collected profits within T_{max}, obeys the time windows for serviced nodes, only visits each node once at most, and starts and ends at the prescribed nodes.

Assume for the formulation of the problem that W is a large constant, s_i is the start of service at location i, and

$$x_{ig} = \begin{cases} 1, & \text{a visit to node } i \text{ is followed by a visit to } g \\ 0, & \text{otherwise,} \end{cases}$$

$$y_i = \begin{cases} 1, & \text{if node } i \text{ is visited in the prescribed route} \\ 0, & \text{otherwise.} \end{cases}$$

This problem can be formalized into a integer programming problem as in [51]

$$Max \sum_{i=2}^{N-1} S_i y_i \tag{5.17}$$

$$\sum_{g=2}^{N-1} x_{\mathcal{S}g} = \sum_{i=2}^{N-1} x_{iE} = 1 \tag{5.18}$$

$$\sum_{g=2}^{N-1} x_{g\mathcal{S}} = \sum_{i=2}^{N-1} x_{Ei} = 0 \tag{5.19}$$

$$\sum_{i=1}^{N-1} x_{ik} = \sum_{g=2}^{N} x_{kg} = y_k \quad (k = 2, ..., N-1) \tag{5.20}$$

$$s_i + T_i + c_{ig} - s_g \leq M(1 - x_{ig}) \quad (i, g \in V) \tag{5.21}$$

$$y_k \leq 1 \quad (k = 2, ..., N-1) \tag{5.22}$$

$$\sum_{i=1}^{N-1} \left(T_i y_i + \sum_{g=2}^{N} c_{ig} x_{ig} \right) \leq T_{max} \tag{5.23}$$

$$O_i \leq s_i \quad (i \in V) \tag{5.24}$$

$$s_i \leq C_i \quad (i \in V) \tag{5.25}$$

$$x_{ig}, y_i \in \{0, 1\} \quad (i, g \in V). \tag{5.26}$$

The objective function in (5.17) maximizes the collected profit from a given route $\mathcal{R}$. Constraints (5.18) and (5.19) enforces the requirement that route $\mathcal{R}$ must start and end at points $\mathcal{S}$ and E. Constraints (5.20) and (5.21) provide a requirement for the connectivity and timing of the route. Constraint (5.22) requires that every location may only be visited at most once, while constraint (5.23) keeps the time to traverse the tour smaller than the available time budget. Constraints (5.24) and (5.25) ensure that service is provided to nodes within their time windows. Constraint (5.26) requires that all nodes in the integer program are assigned as either a one or a zero.

UAVs that need to adapt in real time require minimal computation times, so the proposed solution is based on a locally optimal iterated local search heuristic (ILS) introduced in [51], rather than an exact optimal solution. The ILS uses an insertion heuristic that iteratively takes possible visits, which don't violate the integer program formulation, and finds the locally optimal insertion position in the current route $\mathcal{R}$. For each extra visit added to the tour, the insertion heuristic presents a quick method to verify that time windows are not violated for tour nodes after the insertion position. This procedure is done by defining two variables for each visit that have already been included in the route, the $Wait_i$ and the $MaxShift_i$. $Wait$ is either the amount of difference between the arrival at the node a_i and the opening of the time window O_i, or 0 if there is no wait since $a_i \in [O_i, C_i]$

$$Wait_i = Max(0, O_i - a_i). \tag{5.27}$$

$MaxShift$ is the maximum time that the service of a member of the current route $\mathcal{R}$ can be delayed before an inserted visit becomes infeasible. The $MaxShift$ for node $i \in \mathcal{R}$ is defined to be

$$MaxShift_i = min(C_i - s_i, Wait_{i+1} + MaxShift_{i+1}). \tag{5.28}$$

The maximum shift for node i is dependent on the sum of the available time to delay all future nodes in $\mathcal{R}$ and the wait time for the next node, unless the ability to complete the service of i is the limiting factor due to the closing of its service time window.

According to [51], this methodology makes the evaluation of feasibility for proposed insertions possible in constant time, rather than linear time. The total time shift for all subsequent visits associated with an insertion of an extra visit $g \notin \mathcal{R}$ between nodes $i, n \in \mathcal{R}$ is

$$Shift_g = c_{ig} + Wait_g + T_g + c_{gn} - c_{in}. \tag{5.29}$$

To check the feasibility of inserting g, the following condition is checked

$$Shift_g \leq MaxShift_g. \tag{5.30}$$

The time shift associated with each proposed insertion position, for each $g \notin \mathcal{R}$, is found in this manner and the lowest possible shift is considered to be the best possible insertion position for that g. Determining the g to be inserted on each iteration requires finding which g maximizes the ratio

$$Ratio_g = \frac{(S_g)^2}{Shift_g} \tag{5.31}$$

out of all remaining $g \notin \mathcal{R}$.

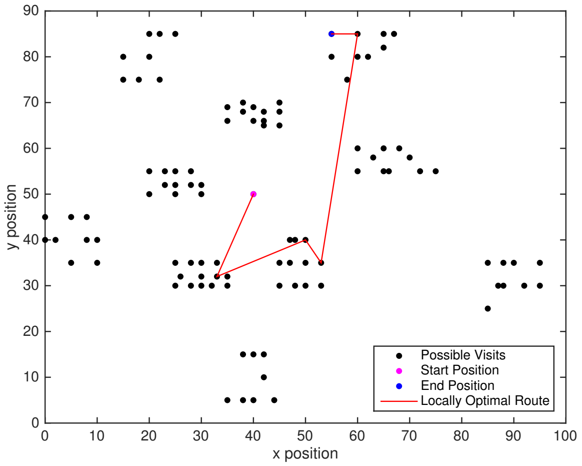


Figure 5.7: Locally optimal solution to OPTW problem in Figure 5.6.

The visit g that maximizes the ratio is inserted in the position in $\mathcal{R}$ that minimized $Shift$ and the visits after the insertion position are updated with the following formulas

$$Shift_g = c_{ig} + Wait_g + T_g + c_{gn} - c_{in}$$

$$Wait_{n^*} = Max(0, Wait_n - Shift_g$$

$$a_{n^*} = a_n + Shift_g$$

$$Shift_n = Max(0, Shift_g - Wait_n)$$

$$s_{n^*} = s_n + Shift_n$$

$$MaxShift_{n^*} = MaxShift_n - Shift_n.$$

These equations are used to update all visits after n until the $Shift$ is reduced to zero. The value of MaxShift for visits prior to the insertion position in the route are then updated as needed with Equation (5.28). A route that locally maximizes profits for the benchmark case provided in Figure 5.6 is shown in Figure 5.7.

The orienteering problem with time windows and the solution heuristic is modified for UAVs trying to optimally reacquire detected targets as follows. The profits associated with each node are the probabilities contained in an assigned P_k^l tracker, so possible insertion nodes are all nodes contained in P_k^l. Additionally, the position of the UAV when it begins trying to reacquire a target needs to be the start position of the route $\mathcal{R}$ (and as a result has zero profit associated with it). Attempts to reacquire the target occur at some global simulation time $\mathcal{T}$ when $\mathcal{P}_k^{lj} < \varrho$.

The end position requirement is removed by focusing the time windows for service to be the amount of time between when the UAV attempts to reacquire the target in tracker l at global simulation time $\mathcal{T}$ and when the tracker dissolves because $\mathcal{P}_k^{lj}$ drops below P_{min}. The time difference T_r is determined by simulating the motion update step in Equation (4.2) until $\mathcal{P}_k^{lj}$ reduces from ϱ to P_{min}, which is the time available for a revisiting UAV to attempt to reacquire the target. $\mathcal{T}$ becomes the lower time window O_i and $\mathcal{T} + T_r$ becomes the upper time window for all nodes $i \in P_k^l$, as well as for $\mathcal{S}$ located at the position of UAV j at time $\mathcal{T}$. The end node E can be chosen arbitrarily from the set of all nodes not in P_k^l. The only requirement is that $O_E = \mathcal{T}$ and $C_E > max(c_{iE}) + \mathcal{T} + T_r \; \forall \; i \in P_k^l$, which means it is reachable within the end of the time budget no matter where the second to last node in $\mathcal{R}$ is located. The limited time budget T_{max} must also be larger than $T_r + max(c_{iE})$ so that the reacquisition problem is well posed.

The traversal of the locally optimal reacquisition route prescribed terminates with either an acquisition of the target, at which point the UAV relocalizes the target and begins searching for new detections, or with a tracker being dissolved when the target cannot be found. These two situations make the application of wait times unnecessary because the reacquisition of the target is constrained within the uniform time windows for $\mathcal{R}$. Additionally, each node is considered serviced as soon as the UAV search footprint provides coverage of the node location, which makes service times instantaneous.

The dynamics of the assigned UAV are guided to waypoints in $\mathcal{R}$ by again utilizing a modified artificial spring force

$$\nabla W_k^{lj} = -K \left(||\Theta_k^j - \psi(\mathcal{R}(i))|| \right) \frac{\Theta_k^j - \psi(\mathcal{R}(i))}{||\Theta_k^j - \psi(\mathcal{R}(i))||}, \tag{5.32}$$

where K is the spring constant and the rest length is zero. The spring is connected to the next unserviced element $i \in \mathcal{R}$ and is updated for each time step to continually evolve along $\mathcal{R}$.

The total control force for Search-and-Reacquire tracking applied to UAV j is

$$F_k^j = \begin{cases} \nabla W_k^{lj} + \nabla P_k^j, & j \in \mathcal{J} \ \& \ \mathcal{P}_k^{lj} \leq \varrho \\ \nabla R_k^j + \nabla P_k^j + \nabla Q_k^j, & j \in \mathcal{J} \ \& \ \mathcal{P}_k^{lj} > \varrho \\ \nabla R_k^j + \nabla P_k^j + \nabla Q_k^j, & j \notin \mathcal{J}. \end{cases} \tag{5.33}$$

(Note again that the same Pauli repulsion rules apply as in the loiter and search-and-loiter tracking strategies. The term ∇P_k^j changes based on whether the UAV is assigned or not according to Equations (5.9) and (5.14).)

Chapter 6: Simulation Results

6.1 Performance of the Likelihood Gradient Search Algorithm

Table 6.1: 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

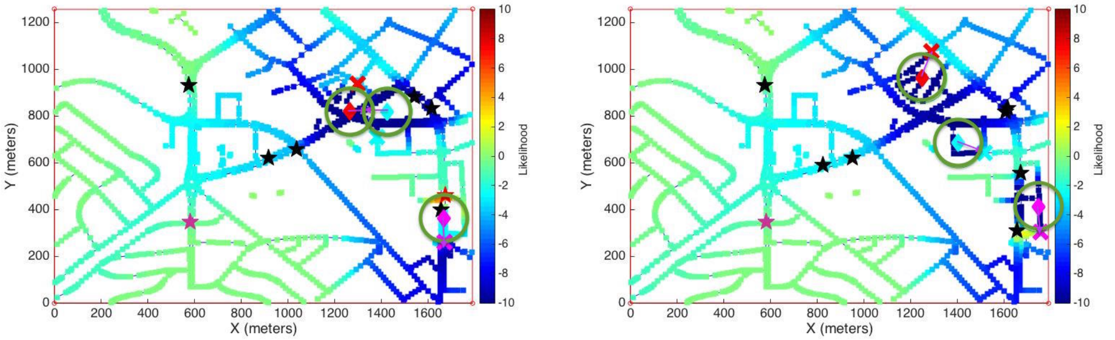
Figure 6.1 provides a snapshot of the gradient search algorithm in simulation. The likelihood along the network is represented graphically by the color of each node. High likelihood is red, neutral likelihood is green, and low likelihood is blue. Targets are distributed throughout the network as colored 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 6.2.

Figure 6.1 shows a number of the behaviors described throughout this thesis. Focusing first on the magenta UAV in Figure 6.1a, a red target is just entering the sensor range of the UAV. The UAV raises the likelihood for all nodes in sensor

Table 6.2: 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
$\mathbb{T}$	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

range as indicated by the orange and red nodes. In Fig. 6.1b, 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. For the purposes of validating the track-before-detect gradient search algorithm, the black



(a) Time step $k = 230$

(b) Time step $k = 235$

Figure 6.1: Snapshots of LRT detection and repulsion

target is now invisible to all UAV sensors and the UAVs are unconcerned about how that target is being tracked after detection.

Focusing now on the red and cyan UAVs, their sensor ranges have just overlapped in Figure 6.1a. 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 6.1b.

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 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 gradient search algorithm on road networks 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 [52]. The four different road networks in Table 6.1 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 6.2 for each UAV configuration, resulting in one hundred trials for each of the four road network snapshots.

Figure 6.2 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 6.3 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 gradient search algorithm becomes more efficient at detecting targets on similarly sized snapshots as the number of UAVs increases.

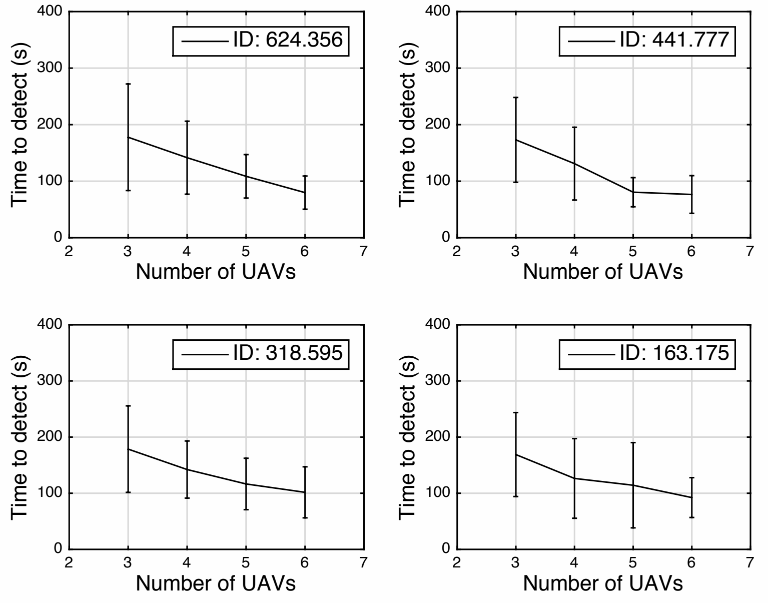


Figure 6.2: Number of UAVs vs. Time to detect for constant area

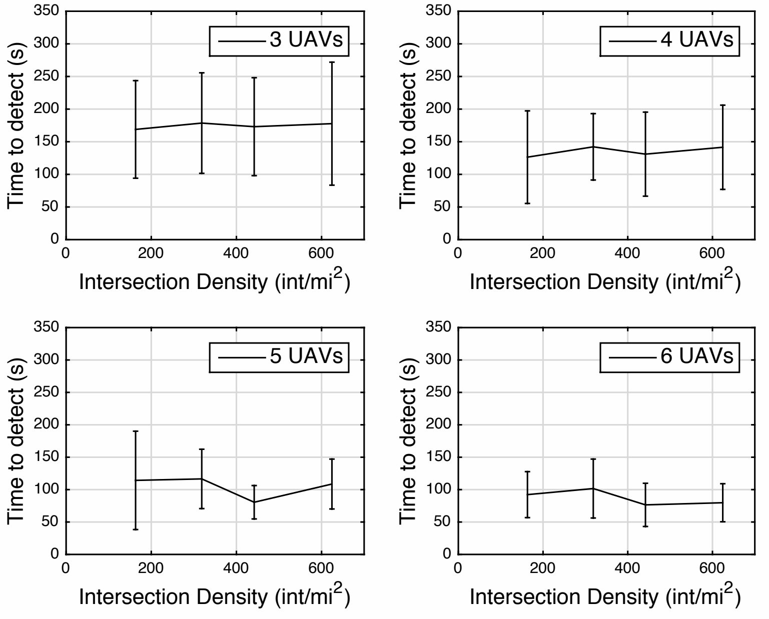


Figure 6.3: Intersection Density vs. Time to detect for constant area

6.2 Performance of Search and Track Algorithms

Four parameters help determine simulation performance for a search and track problem on a road network: 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 [52].

To describe the ability of UAVs to find targets, the relative coverage rate is found using the procedure in [53]. 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, $n_{\mathbb{T}}$ 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 [32]

$$\mathcal{C} = n_{\mathbb{T}}\mathcal{S}\sqrt{\xi^2 + \Theta^2 - 2\Theta\xi\cos\tilde{\theta}}, \tag{6.1}$$

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 [53]

$$\frac{1}{2\pi}\int_0^{2\pi}\sqrt{\xi^2 + \Theta^2 - 2\Theta\xi\cos\tilde{\theta}}d\tilde{\theta}. \tag{6.2}$$

The target density within the confines of the map is

$$n_{\mathbb{T}} = \frac{\mathbb{T}}{\mathcal{A}}, \tag{6.3}$$

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

For simulations, a single road network in downtown Baltimore, Maryland with ID 194 ($\mathcal{A} = 4.59 \times 10^6$ 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 [54], 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

$$P_d = 0.975 - 0.025\frac{h}{457} \tag{6.4}$$

$$P_f = 0.025\frac{h}{457} \tag{6.5}$$

$$s = \frac{5}{457}h. \tag{6.6}$$

Variations in P_d and P_f produce the series of receiver operating characteristic curves (ROC) [42] in Figure 6.4.

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

$$\rho = \frac{S_r}{F_L}h. \tag{6.7}$$

A summary of simulation parameters are provided for a camera with $F_L = 90$

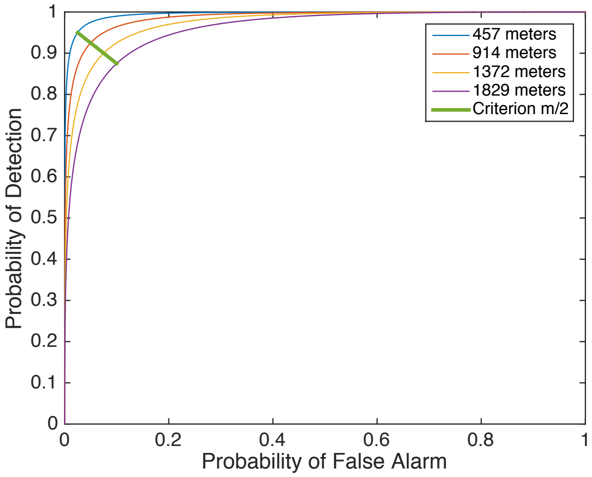


Figure 6.4: ROCs for linearly increasing altitude

Table 6.3: 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

mm and $S_r = 18.4$ mm in Table 6.3. Additional parameters used for the simulation are described in Table 6.4.

The performance of simulations is compared using a number of metrics [55], 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).

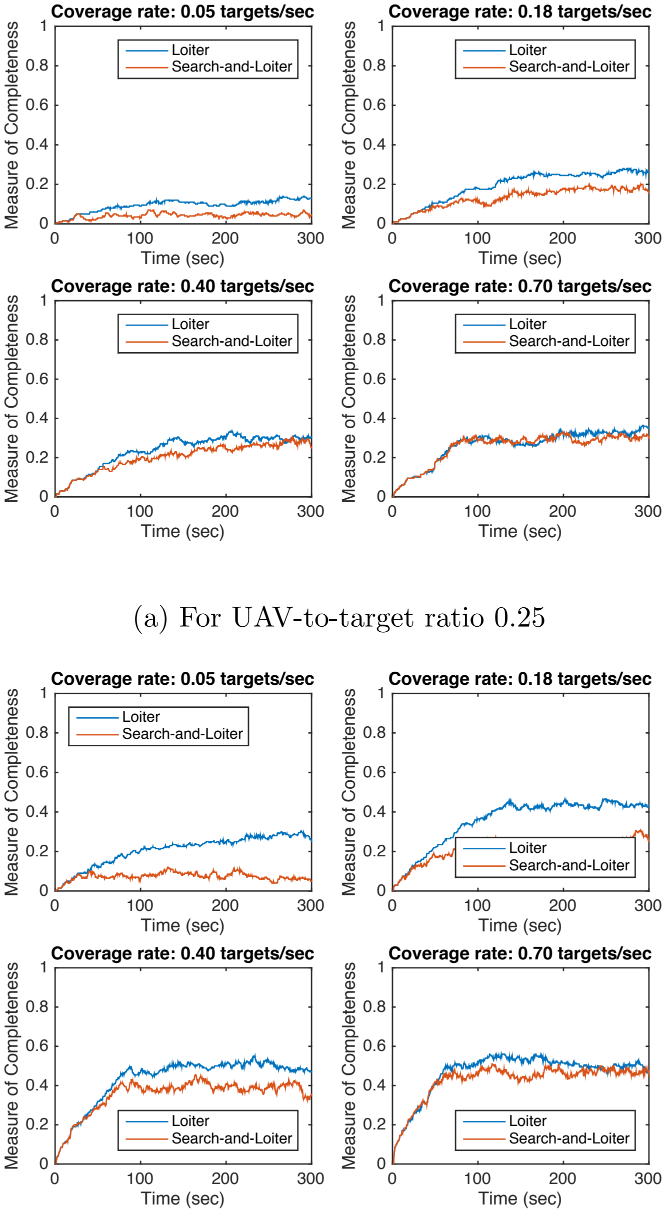
Table 6.4: Simulation parameters

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

Each of these measures is recorded for every time step and Monte Carlo trial, and are averaged over the entire data set for a particular scenario described by the four parameters: intersection density (ID), coverage rate, guidance algorithm, and UAV-to-target ratio $\widetilde{O}$. 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 [55]. Each altitude was simulated for both of the tracking algorithms over 50 trials in a custom Matlab simulation environment.

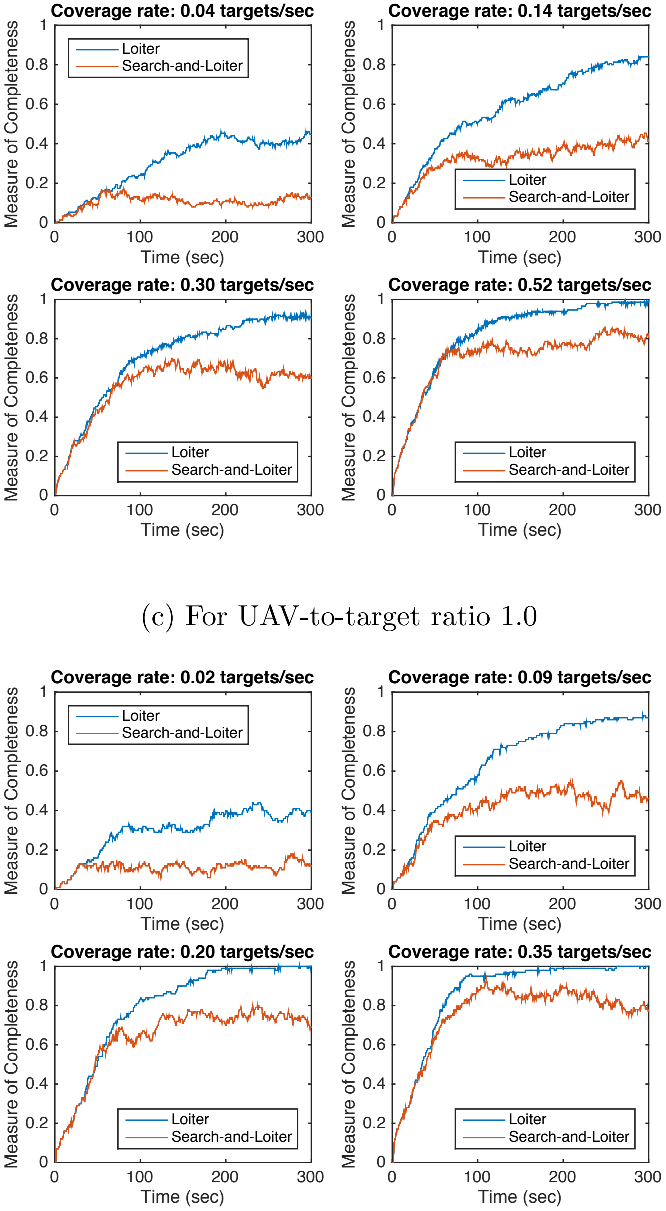
Figure 6.5 shows the change in MOC vs. time for four different $\tilde{O}$'s and a variety of different coverage rates. For all four UAV-to-target ratios, both algorithms are characterized by two sequences: the (mostly) positive linear aggregation of information about the targets and road network, and 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. The loiter algorithm consistently performs better in tracking all of the targets than the search-and-loiter algorithm, but not in detecting new targets. As coverage rate increases, the time required to reach the plateau in performance decreases and the steady-state performance increases.

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 well.



(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 6.5: MOC vs. time for increasing coverage rate.

The loiter and search-and-loiter algorithms perform relatively well for different UAV-to-target ratios, but with some key distinctions. 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% for both loiter and search-and-loiter. However, while loiter is compelling the UAV to focus on keeping one tracker accurate rather than tracking all possible targets, Search-and-loiter has similar performance in terms of MOC, but is actually glimpsing more targets as it searches and returns to loiter. This result indicates that search-and-loiter is actually performing better and is highlighted in case studies provided in Figure 6.6, which is discussed later.

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. As anticipated, searching for additional targets (for search-and-loiter) rather than simply loitering over detected targets underperforms in the case where there are more UAVs than targets.

For another perspective, let the percentage of valid trackers PVT be

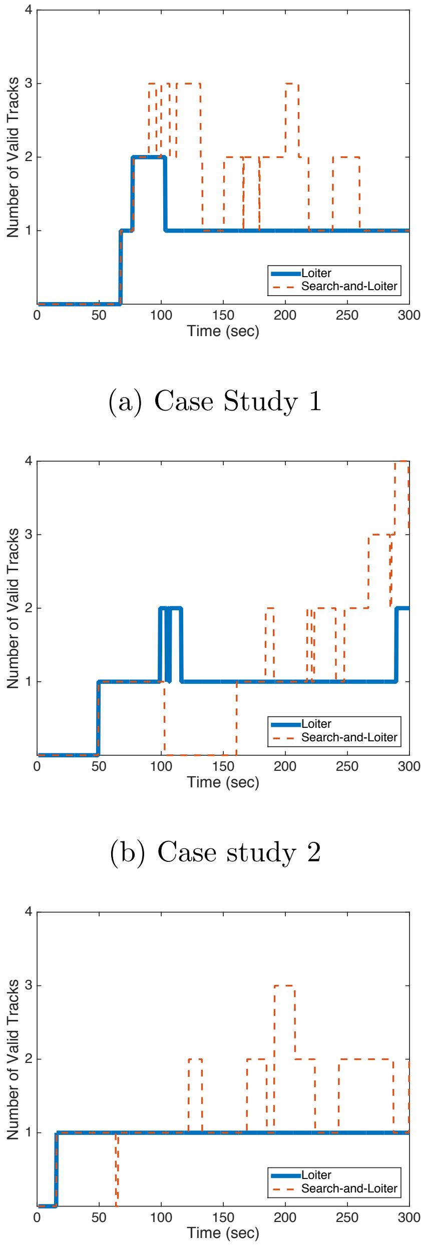
$$PVT = \frac{NVT}{NVT + NST}. \tag{6.8}$$

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

$$PVA = \frac{NVA}{NVA + NFA}. \tag{6.9}$$

Figure 6.7 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



(a) Case Study 1

(b) Case study 2

(c) Case Study 3

Figure 6.6: Case study of valid tracks vs. time for 4 targets and 1 UAV at an altitude of 1828.8 m.

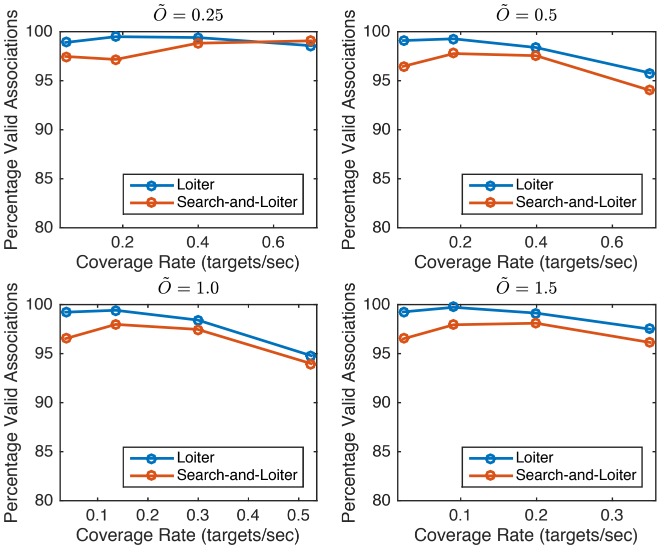


Figure 6.7: The percentage of associations that are valid vs. coverage rate.

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.

The performance of the third search-and-reacquire algorithm was not tested in simulation due to time constraints on this thesis, but suggestions for determining its effectiveness through simulation is provided in Section 7.2.

Chapter 7: Conclusion

7.1 Summary of Contributions

The variety of mission requirements for surveillance operations make the search and track problem difficult and computationally intensive. We present a 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 mover's distance. Utilizing artificial potentials, three motion-planning strategies were created to balance finding undetected targets with keeping trackers accurate.

In simulation, 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 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.

7.2 Suggestions for Future Research

The search-and-reacquire strategy was not tested in simulation and could be improved if more computational time is allowed. The full iterated local search heuristic provided a near optimal solution to the orienteering problem when an additional shake step was included to perturb locally optimal solutions [51]. Simulations comparing performance of the locally optimal and near optimal solutions could be performed in parallel to compare the computational time and relative effectiveness of reacquiring targets in the same scenarios.

Additionally, the effectiveness of the baseline models could be improved with changes in the revisit threshold for UAVs tracking targets and the motion model for targets. The revisit threshold was determined heuristically and is by no means optimal. Further investigation of the variation of this threshold could provide heightened performance for all three algorithms. Likewise, providing a better model for the target motion in the target trackers would provide the UAVs with more time to search for new targets and could provide opportunities to revise how pairings between UAVs and trackers is determined. Dynamic pairing could be an option, wherein UAVs could be paired with multiple trackers and update them as needed, as well as provide more opportunities for heightened cooperation among UAVs.

Finally, interesting extensions of this work involve changing how the sensor platform onboard the UAV is modeled. The first interesting case would be providing UAVs with heterogeneous sensor footprints such as a fixed forward-looking camera (a cone) and gimbaled cameras (movable cone) and studying how performance changes for searching and tracking targets. The second extension is spatially varying probabilities of detection and false alarm. As UAVs move around a dense urban environment, buildings, bridges, and other objects obfuscate targets and make detection hard if not impossible. With height maps becoming available for more major cities, the effect of buildings on probability of detection and false alarm can be modeled and studied in target tracking situations.

Bibliography

[1] Capacitated vrp with time windows instances. http://neo.lcc.uma.es/vrp/vrp-instances/capacitated-vrp-with-time-windows-instances/. Accessed: 2017-04-26.

[2] Department of Defense. Unmanned Systems Roadmap: 2007-2032. Technical report, 2007.

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

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

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

[6] Y. Rubner, C. Tomasi, and C. J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

[7] L. Hong, N. Cui, M. Bakich, and J. R. Layne. Multirate interacting multiple model particle filter for terrain-based ground target tracking. *Proc. IEEE Control Theory and Applications*, 153(6):721–731, Nov 2006.

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

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

[10] W.R. Blanding, P.K. Willett, and Y. Bar-Shalom. Multiple target tracking using maximum likelihood probabilistic data association. In *Proc. IEEE Aerospace Conference*, pages 1–12, March 2007.

[11] D.E. 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*, 152(5):327–334, October 2005.

[12] C. Huang and S. 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*, 9(1):16–30, Jan 2012.

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

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

[15] 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.*, volume 5986, pages 59860J–59860J–10, 2005.

[16] F. Rafi, S. Khan, K. Shafiq, and M. Shah. Autonomous target following by unmanned aerial vehicles. In *Defense and Security Symposium*, pages 623010–623010. International Society for Optics and Photonics, 2006.

[17] Xiao-Rong Li and Y. Bar-Shalom. Multiple-model estimation with variable structure. *IEEE Transactions on Automatic Control*, 41(4):478–493, Apr 1996.

[18] 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*, 36(1):26–46, Jan 2000.

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

[20] 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*, pages 1–8, July 2007.

[21] C. Yang, M. Bakich, and E. Blasch. Nonlinear constrained tracking of targets on roads. In *8th Conf. Proc. on Information Fusion*, volume 1, pages 235–242, July 2005.

[22] 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.

[23] M. Ulmke and W. Koch. Road-map assisted ground moving target tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 42(4):1264–1274, October 2006.

[24] U. Orguner, T.B. Schon, and F. Gustafsson. Improved target tracking with road network information. In *Proc. IEEE Aerospace Conference*, pages 1–11, March 2009.

[25] S. Gattein, B. Pannetier, and P. Vannoorenberghe. Analysis and integration of road projection methods for multiple ground target tracking. In *Proc. 8th Conference on Information Fusion*, volume 1, pages 227–234, July 2005.

[26] 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*, volume 1, July 2005.

[27] 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*, 41(3):937–954, July 2005.

[28] S. M. Tonissen and R. J. Evans. Peformance of dynamic programming techniques for track-before-detect. *IEEE Transactions on Aerospace and Electronic Systems*, 32(4):1440–1451, Oct 1996.

[29] S. M. Tonissen and Y. Bar-Shalom. Maximum likelihood track-before-detect with fluctuating target amplitude. *IEEE Transactions on Aerospace and Electronic Systems*, 34(3):796–809, Jul 1998.

[30] C. Jauffret and Y. Bar-Shalom. Track formation with bearing and frequency measurements in clutter. *IEEE Transactions on Aerospace and Electronic Systems*, 26(6):999–1010, Nov 1990.

[31] B.E. Barkley and D.A. Paley. Multi-target tracking and data association on road networks using unmanned aerial vehicles. In *Proc. IEEE Aerospace Conference*, March 2017.

[32] A. Orych. Review of methods for determining the spatial resolution of uav sensors. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 40(1):391, 2015.

[33] R. Gayle, W. Moss, M.C. Lin, and D. Manocha. Multi-robot coordination using generalized social potential fields. In *Proc. IEEE Conference on Robotics and Automation*, pages 106–113, May 2009.

[34] M. Schwager, D. Rus, and J Slotine. Unifying geometric, probabilistic, and potential field approaches to multi-robot deployment. *The International Journal of Robotics Research*, 30(3):371–383, 2011.

[35] B. Bollobás. *Modern graph theory.* Graduate texts in mathematics. Springer, 1998.

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

[37] M. Reuter, S. Biasotti, D. Giorgi, G. Patane, and M. Spagnuolo. Discrete Laplace-Beltrami operators for shape analysis and segmentation. *Computers and Graphics*, 33(3):381–390, 2009. Proc. IEEE Conference on Shape Modelling and Applications.

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

[39] M. Haklay and P. Weber. Openstreetmap: User-generated street maps. *IEEE Pervasive Computing*, 7(4):12–18, October 2008.

[40] I. Filippidis. Openstreetmap interface. https://github.com/johnyf/openstreetmap, 2016.

[41] E.L. Cussler. *Diffusion.* Cambridge Series in Chemical Engineering. Cambridge University Press, 1997.

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

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

[44] 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*, volume 2, pages 251–256 vol.2, 2001.

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

[46] 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*, 79(3):497–516, 1957.

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

[48] 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*, 106(738):463–477, 1924.

[49] M.G. Kantor and M.B. Rosenwein. The orienteering problem with time windows. *Journal of the Operational Research Society*, 43(6):629–635, Jul 1992.

[50] M.M. Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35(2):254–265, 1987.

[51] P. Vansteenwegen, W. Souffriau, G. Vanden Berghe, and D. Van Oudheusden. Iterated local search for the team orienteering problem with time windows. *Computers and Operations Research*, 36(12):3281 – 3290, 2009.

[52] J. Dill. Measuring network connectivity for bicycling and walking. In *83rd Annual Meeting of the Transportation Research Board*, pages 11–15, 2004.

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

[54] C. 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.

[55] 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*, pages 1–8, 2011.