Cooperative Mapping and Target Search over an Unknown Occupancy Graph using Mutual Information

Artur Wolek[1], Sheng Cheng[2], Debdipta Goswami[2], and Derek A. Paley[3]

***Abstract*— A cooperative mapping and target-search algorithm is presented for detecting a single moving ground target in an urban environment that is initially unknown to a team of autonomous quadrotors equipped with noisy, range-limited sensors. The target moves according to a biased random-walk model, and search agents (quadrotors) build a target state graph that encodes past and present target positions. A track-before-detect algorithm assimilates target measurements into the log-likelihood ratio and anisotropic kriging interpolation predicts the location of occupancy nodes in unexplored regions. Mutual information evaluated at each location in the search area defines a sampling-priority surface that is partitioned by a weighted Voronoi algorithm into candidate waypoint tasks. Tasks are assigned to each agent by iteratively solving a utility-maximizing assignment problem. Numerical simulations show that the proposed approach compares favorably to non-adaptive lawnmower and random coverage strategies. The proposed strategy is also demonstrated experimentally through an outdoor flight test using two real and two virtual quadrotors.**

I. Introduction

Autonomous mobile robots may be used to map uncertain environments while searching for targets and tracking those that are detected. Human teams also perform such tasks for military reconnaissance or search and rescue, frequently with the help of planning software such as SARTopo [1]. Both robotic and manned teams may benefit from using cooperative sampling strategies that assimilate noisy measurements to estimate the environment or location of a target and guide the collection of subsequent measurements. Many methods continue to be proposed for efficiently addressing the map [2], [3], search [4]–[7], and track [8] objectives independently, as well as balancing these priorities, for example, by combining searching and tracking [9]–[11].

This work investigates the use of autonomous quadrotors to concurrently map and search an urban area for a mobile ground target (see Fig. 1). The mapping objective is to build an occupancy graph of the environment with nodes that represent possible target locations and edges that represent target motions. The search objective is to detect a target on this graph. A uniform grid is imposed over the search area with cells that have one of three states: void, unoccupied, or occupied. Mapping aims to identify all non-void cells and search aims to find the cell occupied by the target. Thus, the cell states encode both of the objectives of the quadrotor team. Noisy mapping and target sensors measurements are assimilated to maintain the probability of each cell state using principled estimation techniques: recursive Bayesian estimation to detect occupancy nodes in newly explored cells, likelihood ratio detection and tracking (LRDT) for targets moving over the explored occupancy graph, and anisotropic kriging to predict the probability of nodes in unexplored cells. Mutual information between the cell state and the sensor measurements quantifies the reduction in uncertainty that is expected from sampling each cell to give a surface of sampling priority. A weighted Voronoi algorithm partitions the sampling-priority surface to define waypoint tasks and a fixed number of them are assigned to each agent by iteratively solving a utility-maximizing assignment problem that includes the agent dynamics and field of view.

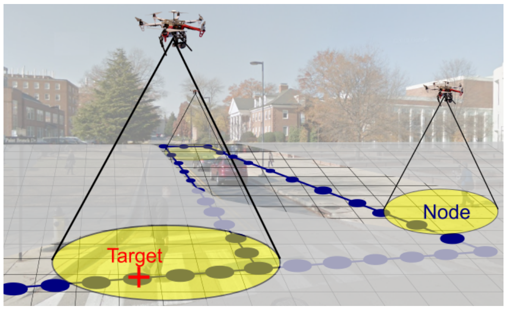


Fig. 1: Autonomous quadrotors with noisy and range-limited sensors explore an urban environment. Their objective is to simultaneously map an unknown occupancy graph (dark blue nodes and edges) and search for a ground target (red "+" marker) moving over this graph. The occupancy graph is embedded in a larger search grid. The state of a search grid cell that contains an occupancy graph node may be *occupied* or *unoccupied* (depending on the presence or absence of a target), whereas the state of cells with no occupancy graph nodes is always *void*.

Our approach is related to other strategies that use mutual information and information-theoretic methods to guide mapping [2], [12]–[14], search [4], [15], [16], and tracking [17]–[19]. However, we consider a variation of the problem in which the environment is represented by an occupancy graph and focus on concurrent mapping and search. Related work on surveillance and pursuit and evasion over graphs [20] or in continuous environments [21] provides search guarantees, but assumes perfect target or mapping sensors.

This work was supported by DARPA under Contract No. HR001118C0142. A. Wolek was supported by a postdoctoral fellowship from the Maryland Robotics Center at the University of Maryland.

[1] A. Wolek is with the Institute for Systems Research, University of Maryland, College Park, MD 20742 USA (e-mail: wolek@umd.edu).

[2] S. Cheng and D. Goswami are with the Department of Electrical and Computer Engineering, University of Maryland, College Park, MD 20742 USA (e-mail: cheng@terpmail.umd.edu, goswamid@umd.edu).

[3] D. A. Paley is with the Department of Aerospace Engineering and the Institute for Systems Research, University of Maryland, College Park, MD 20742 USA (e-mail: dpaley@umd.edu).

Target detection and tracking over occupancy graphs with noisy sensors is discussed in [22]–[25]. We extend these approaches to the case where the graph is not known initially and is instead constructed on the fly. Algorithms for incrementally building graphs [26] and occupancy grids [27] are not directly applicable to the target state graph introduced here, so a new method is proposed. The biased random-walk model is similar to [8], where a second-order Markov model reduces the probability of a target turning, however, we use the relative orientation of edges in an occupancy graph to predict target motion. To predict the location of occupancy nodes and guide mapping, we propose anisotropic kriging interpolation that uses the edge directions in an explored occupancy graph to adaptively determine the axes of anisotropy. A weighted Voronoi algorithm partitions the mutual information surface into a set of waypoints, and an assignment algorithm allocates a subset of them to each agent. This path-planning approach differs from other Voronoi-based strategies that partition the environment, typically with one partition per agent to minimize a coverage-performance cost [5], [28].

The contributions of this paper are (1) a principled approach to construct an occupancy graph and target state graph, which encodes past and present target locations, from noisy sensor measurements; (2) a biased random-walk target motion model for use in a likelihood ratio detector and tracker; (3) a comprehensive approach to generate sampling tasks that unifies mapping and search objectives using mutual information; and (4) a task-assignment strategy that assigns conflict-free and locally optimal sampling paths to each agent. Monte Carlo simulation results show favorable performance compared to non-adaptive lawnmower and random coverage strategies. The proposed strategy is also demonstrated experimentally using two real and two virtual quadrotors.

The paper is organized as follows. Section II describes the target occupancy graph, search grid, target and search agent motion, and sensor models. Section III presents the approach for building the occupancy graph on the fly, predicting occupancy nodes using kriging interpolation, and assigning waypoint tasks based on mutual information. Section IV provides results from simulations and describes the outdoor experiment. Lastly, Section V concludes the paper.

II. Problem Formulation

A. Target Occupancy Graph and Search Grid

A finite and undirected [29] target occupancy graph $\mathcal{G}_O = (\mathcal{V}_O, \mathcal{E}_O)$ represents a network of navigable passages in an urban environment (for example, roads or sidewalks as shown in Fig. 2). The set $\mathcal{V}_O = \{v_1, v_2, \cdots, v_{N_O}\}$ contains N_O nodes that represent feasible target locations. An edge, denoted by a pair $(i, j) \in \mathcal{E}_O$, where $v_i, v_j \in \mathcal{V}_O$, indicates that nodes are adjacent and a target may transition from one to the other over a fixed time interval. While targets are constrained to move along the occupancy graph, the search agents are free to move anywhere over a subset of the plane that contains a grid of M cells. Each cell is of equal length and width Δ and contains at most a single occupancy graph node. The location of the vth node is given by $\psi(v) = (\psi_x(v),\ \psi_y(v)) \in \mathbb{R}^2$. Edges $(i, j) \in \mathcal{E}_O$ may exist only between nodes contained in adjacent cells (either directly above or below, to the left or right, or diagonally adjacent). If a cell contains a node, it is classified as either occupied or not occupied (depending on the presence of a target at a given time); otherwise it is void for all time. Thus, a cell may have one of three states: $C = \{O, U, V\}$, which indicate whether the cell is occupied (O), unoccupied (U), or void (V).

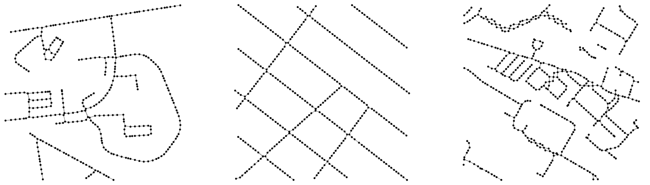

(a) Suburban (b) Urban, regular (c) Urban, irregular

Fig. 2: Example occupancy graphs adapted from [30]

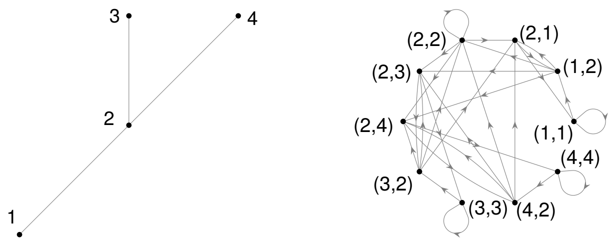
B. Biased Random-Walk Target Motion Model

Target motion is governed by a discrete, biased random-walk model in which a target either transitions to an adjacent node or remains at its current node after each fixed time interval $T_s = t_k - t_{k-1}$. As a target moves along the occupancy graph, its state is denoted by a pair of nodes $s = (p, c) \in S$, where $c \in \mathcal{V}_O$ is the target's current node and $p \in \mathcal{V}_O$ is its previous node. If the target transitions to a new state $s' = (p', c')$, then $p' = c$. For a given occupancy graph, the target state space is $S = \{(p, c) \subseteq \mathcal{V}_O \times \mathcal{V}_O :\ (p, c) \in \mathcal{E}_O, (c, p) \in \mathcal{E}_O \text{ or } p = c\}$. Each state is enumerated to give the nodes $\{s_1, s_2, \cdots, s_{N_S}\} = S$ and define a directed target state graph $\mathcal{G}_S = (S, \mathcal{E}_S)$, where the edges $\mathcal{E}_S$ represent possible transitions between target states (see Fig. 3).

The benefit of expressing the target state as a pair of nodes, $s = (p, c)$, is that it allows a heading angle θ to be defined as $\tan\theta(s) = [\psi_y(c) - \psi_y(p)]/[\psi_x(c) - \psi_x(p)]$ for $c \neq p$. If $c = p$, then we refer to $s \in R$ as a rest state, where $R = \{(p, c) \in S\ :\ p = c\}$, and the heading is undefined. The heading is used to develop a target motion model that captures the tendency of targets (e.g., people or cars) to move in a constant direction along the occupancy graph. A fixed probability $0 \leq p_s < 1$ describes transitioning (for a single time step) to a rest state. The remaining probability $1 - p_s$ is distributed among neighboring states $\mathcal{N}(s)$ to bias motion in the direction $\theta(s)$. The probability of transitioning from state $s' \in S \setminus R$ to $s \in S$ is thus

$$q_{\text{walk}}(s|s') = \begin{cases} p_s & \text{if } s = (c, c) \\ (1 - p_s) w(s, s') & \text{if } s \in \mathcal{N}(s') \\ 0 & \text{otherwise}\ , \end{cases} \tag{1}$$

where $w(s, s')$ is a bias function that satisfies $\sum_{s \in \mathcal{N}(s')} w(s, s') = 1$. Various bias functions may be



(a) Occupancy graph (b) Target state graph

Fig. 3: An occupancy graph with four nodes and the corresponding target state graph.

used to model target motion; we use

$$w(s,s') \propto \frac{(c_d - c_m/\pi)\Delta\theta(s,s') + 1 + c_m}{1 + c_d\Delta\theta(s,s')} \ , \qquad (2)$$

where $\Delta\theta(s,s')$ is the smallest difference in heading between two states, $c_m \geq 0$ and $c_d > 0$ are scale and shape parameters, and $\propto$ denotes proportionality. Other bias functions are possible; for example, $w(s,s')$ could be parametrized by the location of nearby search agents to model evasive target motion. In the special cases that $s' \in R$, a uniform probability is assigned for transitioning to all neighboring states; if there are no neighbors, the target remains stationary.

C. Search-Agent Motion and Sensor Models

The motion of each search agent is modeled by a planar, double-integrator system with input saturation and linear damping. Given a desired waypoint, $\mathbf{w}_i = (x_d^i, y_d^i)$, a proportional-derivative controller determines the acceleration input for the ith agent, where $i = \{1, \ldots, N\}$. Each agent is equipped with both a mapping sensor and a target sensor that report scalar measurements G_m^i and Z_m^i, respectively, for each cell $m \in \{1, \cdots, M\}$ whose center is within a sensing range ρ. In the following, the superscript indicating the ith agent and the subscript for the mth cell are suppressed for brevity. The performance of the mapping and target sensors is characterized by the sensitivities [31] m_G and m_Z, and the sensors are quantized and take on values $G \in \{g_1, \cdots, g_{n_g}\}$ and $Z \in \{z_1, \cdots, z_{n_z}\}$, respectively. When reporting noise, the sensors output scalar values drawn from a unit-variance, zero-mean Gaussian distribution. When reporting a signal, the measurements are shifted by the sensitivity, either m_G or m_Z. The mapping sensor identifies cells that are not void (i.e., that contain occupancy graph nodes) and the target sensor identifies cells that are occupied by a target. Thus, the probability of $G = g_l$, where $l \in \{1, \cdots, n_g\}$, is conditioned upon the cell state as follows:

$$g_{l|V} \triangleq p(g_l|V) \propto \exp\left(-g_l^2/2\right) \qquad (3)$$

$$g_{l|U,O} \triangleq p(g_l|\{U,O\}) \propto \exp\left(-(g_l - m_G)^2/2\right) \ . \qquad (4)$$

Also, the probability of $Z = z_f$, where $f \in \{1, \cdots, n_z\}$, is

$$z_{f|V,U} \triangleq p(z_f|\{V,U\}) \propto \exp\left(-z_f^2/2\right) \qquad (5)$$

$$z_{f|O} \triangleq p(z_f|O) \propto \exp\left(-(z_f - m_Z)^2/2\right) \ . \qquad (6)$$

In practice, these sensors may be RGB or thermal cameras coupled with a data-processing algorithm that reports a scalar signal indicating the presence or absence of a node or target.

D. Node and Target Detection

Assuming a single data fusion center, let the probabilities $\mathfrak{v} \triangleq \Pr\{C = V\}$, $\mathfrak{u} \triangleq \Pr\{C = U\}$, and $\mathfrak{o} \triangleq \Pr\{C = O\}$ represent the agents' collective belief of a cell's state and $\mathfrak{v} + \mathfrak{u} + \mathfrak{o} = 1$. Assume the measurements G and Z are independent given the cell state C. Mapping and target sensor measurements are assimilated to update $(\mathfrak{v}, \mathfrak{u}, \mathfrak{o})$ for cells that are in view using Bayes' law $p(C|G,Z) = p(G|C)p(Z,C)/p(G,Z)$ so that

$$\mathfrak{v}' \triangleq p(C = V|g_l, z_f) = \mathfrak{v}(g_{l|V})(z_{f|V,U})/p(g_l, z_f) \qquad (7)$$

$$\mathfrak{u}' \triangleq p(C = U|g_l, z_f) = \mathfrak{u}(g_{l|U,O})(z_{f|V,U})/p(g_l, z_f) \qquad (8)$$

$$\mathfrak{o}' \triangleq p(C = O|g_l, z_f) = \mathfrak{o}(g_{l|U,O})(z_{f|O})/p(g_l, z_f) \ , \qquad (9)$$

where

$$p(g_l, z_f) = \sum_{c \in C} p(g_l, z_f|c)p(c) = \sum_{c \in C} p(g_l|c)p(z_f|c)p(c)$$
$$= \mathfrak{v} g_{l|V} z_{f|V,U} + \mathfrak{u} g_{l|U,O} z_{f|V,U} + \mathfrak{o} g_{l|U,O} z_{f|O} \ .$$

To detect a node or a target from the probabilities $(\mathfrak{v}, \mathfrak{u}, \mathfrak{o})$, a likelihood ratio test [32, Sec. 15.3] is used. Define the likelihood ratio $\Lambda = p(H_1)/p(H_0)$ as a ratio of probabilities, where H_1 denotes a positive hypothesis (e.g., a cell contains a node or target) and H_0 denotes the negative hypothesis. According to the Neyman-Pearson lemma, the likelihood ratio test $\Lambda > \bar{\Lambda}$ accepts the hypothesis H_1 with a fixed false-alarm rate. The threshold may be defined for a given confidence level p_T as $\bar{\Lambda} = p_T/(1 - p_T)$ [33, Sec. 7.3.2]. For node detection, the likelihood ratio is the ratio of probabilities of node presence to absence $(1 - \mathfrak{v})/\mathfrak{v}$. For target detection, it is $\mathfrak{o}_i/(1 - \mathfrak{o}_i)$. We refer to this detection approach as the cell-wise binary hypothesis test (CBHT).

III. Cooperative Mapping and Search

The following introduces a method to build the occupancy and target state graph on the fly and use it to more accurately detect targets and predict the location of nodes in unexplored regions. The mutual information between the resulting $(\mathfrak{v}, \mathfrak{u}, \mathfrak{o})$ cell probabilities and the sensor model yields a sampling-priority surface for cooperative path planning.

A. Constructing the Target State Graph

Let the explored occupancy graph known to the agents at time t_k be denoted $\mathcal{G}_O(t_k) = (\mathcal{V}_O(t_k), \mathcal{E}_O(t_k))$. Initially, $\mathcal{V}_O(t_0)$ and $\mathcal{E}_O(t_0)$ are empty sets. When a node v_{new} is detected at time t_k, it is appended to the set of explored vertices $\mathcal{V}_O(t_k)$ and corresponding adjacent edges are formed. The following proposition provides an expression for the difference in the number of nodes and edges between $\mathcal{G}_S(t_k)$ and $\mathcal{G}'_S(t_k)$ that results from the addition of v_{new} to $\mathcal{G}_O(t_k)$. The proof (omitted for brevity) considers the construction of edges in the target state graph that are of the form $(x,y) \to (y,z)$, where (x,y) and (y,z) are target states with either

x, y, and/or z equal to v_{new}. The set $\mathcal{V}_O(t_k)$ is partitioned into four disjoint sets $\mathcal{V}_O(t_k) = X_1 \cup X_2 \cup X_3 \cup X_4$, with $X_1 = v_{\text{new}}$, $X_2 = \mathcal{N}(v_{\text{new}})$, $X_3 = \{v \subset \mathcal{V}_O \mid v \in \{\mathcal{N}(i) \setminus \{v_{\text{new}}\}\} \setminus \mathcal{N}(v_{\text{new}}) \quad \text{for} \quad i \in \mathcal{N}(v_{\text{new}})\}$, $X_4 = V \setminus \{X_1 \cup X_2 \cup X_3\}$, and the edges are enumerated for each case $x \in \{X_1, X_2, X_3, X_4\}$.

Proposition 1: Let $\mathcal{G}_O(t_k) = (\mathcal{V}_O(t_k), \mathcal{E}_O(t_k))$, $\mathcal{G}_S(t_k) = (S(t_k), E_S(t_k))$ be the (partially explored) occupancy and target state graph at time t_k, and $\mathcal{G}_O = (\mathcal{V}_O, \mathcal{E}_O)$ be the (fully explored) occupancy graph. If a detected vertex v_{new} is added to $\mathcal{G}_O(t_k)$ to form $\mathcal{G}'_O(t_k) = (\mathcal{V}'_O(t_k), \mathcal{E}'_O(t_k))$, where $\mathcal{V}'_O(t_k) = \mathcal{V}_O(t_k) \cup \{v_{\text{new}}\}$ and $\mathcal{E}'_O(t_k) = \mathcal{E}_O(t_k) \cup \{(i, v_{\text{new}}) \in \mathcal{E}_O \mid i \in \mathcal{V}_O\}$, then the corresponding augmented target state graph $\mathcal{G}'_S(t_k) = (S'(t_k), E'_S(t_k))$ is obtained from $\mathcal{G}_S(t_k)$ by the addition of $2\text{deg}(v_{\text{new}}) + 1$ nodes and $\text{deg}(v_{\text{new}})^2 + 3\text{deg}(v_{\text{new}}) + 1 + 2\sum_{i \in \mathcal{N}(v_{\text{new}})} \text{deg}(i)$ edges, where $\text{deg}(\cdot)$ denotes the node degree in $\mathcal{G}'_O(t_k)$.

Remark 2: We use Proposition 1 to generate the additional target states and target state graph edges that arise from each newly detected occupancy graph node.

B. Likelihood Ratio Target Detection and Tracking

Rather than using the cell-wise binary hypothesis test described in Section II-D, the target state graph is exploited to improve detection performance using a likelihood ratio detection and tracking (LRDT) framework. Let s_k be the true state of the target at time step t_k, for $k \geq 1$. An LRDT approach maintains the target likelihood ratio $\Lambda(s, t_k) = p(s, t_k)/p(\phi, t_k)$ for each node in $\mathcal{G}_S$, where $p(s, t_k)$ is the probability that s_k is equal to some $s \in S$, $p(\phi, t_k)$ is the probability that s_k is equal to the null state ϕ, and $p(\phi, t_k) + \sum_{s \in S} p(s, t_k) = 1$. The null state ϕ represents the target being absent from the search area. Thus, $s_k \in S^+ = S \cup \phi$, the augmented target state space. While many tracking methods only update the target estimate $p(s, t_k)$ when a target is in view, the LRDT approach continuously performs a Bayesian recursion to update $p(s, t_k)/p(\phi, t_k)$ regardless of whether or not the target is visible (i.e., the LRDT incorporates both positive and negative information). To declare a target present in the state space S, the likelihood ratio is accumulated across all states at time t_k, which yields the integrated likelihood ratio $\tilde{\Lambda}(t_k) = \sum_{s \in S} \Lambda(s, t_k)$. A target detection occurs when $\tilde{\Lambda}(t_k)$ surpasses a threshold $\bar{\Lambda}$. The approach is suitable for low signal-to-noise ratios scenarios in which a single glimpse of the target may be insufficient to detect it. The LRDT simultaneously performs detection and tracking and is therefore a track-before-detect method [34, Ch. 10]. The null state has dynamics that are defined by the constants $q(\phi|s)$, $q(s|\phi)$, and $q(\phi|\phi)$, which give the probability of transitioning into, out of, and remaining in, respectively, the null state. The probability $q(\phi|s)$ models a target disappearing or leaving the search area. Conversely, $q(s|\phi)$ is a constant probability of a target appearing at an arbitrary state in S. Therefore, $q(\phi|\phi) = 1 - \sum_{s \in S} q(s|\phi) = 1 - |S| q(s|\phi)$. The target motion model in (1) is used to produce the transition probability $q(s|s') = (1 - q(\phi|s)) q_{\text{walk}}(s|s')$ in the augmented state space. The motion update applied to $p(s, t_{k-1})$ and $p(\phi, t_{k-1})$ is [34]

$$p^-(s, t_k) = q(s|\phi) p(\phi, t_{k\text{-}1}) + \sum_{s' \in S} q(s|s') p(s', t_{k\text{-}1}) \quad (10)$$

$$p^-(\phi, t_k) = q(\phi|\phi) p(\phi, t_{k\text{-}1}) + \sum_{s' \in S} q(\phi|s') p(s', t_{k\text{-}1}) \; . \quad (11)$$

In the absence of a measurement update, (10) and (11) lead $\Lambda(s, t_k)$ to decay towards an equilibrium value such that the integrated likelihood ratio $\bar{\Lambda}(t_k) \rightarrow q(s|\phi)/q(\phi|s)$ as $k \rightarrow \infty$. The rate of decay is governed by the relative magnitudes of $q(s|\phi)$ and $q(\phi|s)$. We propose to set $q(\phi|s) = \alpha \bar{p}_\phi$ and $q(s|\phi) = \alpha(1 - \bar{p}_\phi)$ so that, in the absence of measurements, the integrated likelihood ratio decays to $(1 - \bar{p}_\phi)/\bar{p}_\phi$, where $\bar{p}_\phi$ is a user-defined probability that a target is absent from the search area, and $\alpha \in (0, 1]$ is a tuning parameter that controls the rate of decay. This approach models increasing uncertainty over time for nodes that are no longer in view.

The measurement likelihood functions $L(z_k|s)$ and $L(z_k|\phi)$ produce the likelihood of receiving measurement z_k if the true target state is s or ϕ, respectively. The posterior probabilities arise from the expressions for a recursive Bayesian estimator [34]

$$p(s, t_k) = L(z_k|s) p^-(s, t_k)/C_k \quad (12)$$

$$p(\phi, t_k) = L(z_k|\phi) p^-(\phi, t_k)/C_k \; , \quad (13)$$

where C_k is a normalizing constant that ensures the probability sums to one. The recursions (12) and (13), re-written in terms of the target likelihood ratio Λ and measurement likelihood ratio $\mathcal{L}(s, t_k) = L(z_k|s)/L(z_k|\phi)$, are

$$\Lambda(s, t_k) = \mathcal{L}(s, t_k) \Lambda^-(s, t_k) \; , \quad (14)$$

where $\Lambda^-(s, t_k) = p^-(s, t_k)/p^-(\phi, t_k)$. We express (14) as the natural log of the target likelihood ratio, i.e.,

$$\underbrace{\log \Lambda(s, t_k)}_{\text{posterior}} = \underbrace{\log \mathcal{L}(s, t_k)}_{\text{measurement update}} + \underbrace{\log \Lambda^-(s, t_k)}_{\text{motion update}} \quad (15)$$

$$= \log L(z_k|s) - \log L(z_k|\phi) + \log \Lambda^-(s, t_k) \; .$$

As described in Section III-A, the target state space S grows as new occupancy nodes are discovered. At a given time t_k, the nodes of $\mathcal{G}_S(t_k)$ yield the (explored) target state space $S(t_k)$ over which the likelihood ratio $\Lambda(s, t_k)$ is defined. The expression (15) in vector form is $\log \boldsymbol{\Lambda}(t_k) = m_Z \left(z_k - \frac{m_Z}{2}\right) + \log \boldsymbol{\Lambda}^-(t_k)$, where $\boldsymbol{\Lambda}^-(t_k) \triangleq [\Lambda^-(s_1, t_k), \cdots, \Lambda^-(s_{|S(t_k)|}, t_k)]^{\text{T}}$, $\boldsymbol{\Lambda}(t_k) \triangleq [\Lambda(s_1, t_k), \cdots, \Lambda(s_{|S(t_k)|}, t_k)]^{\text{T}}$, the log is applied element-wise, and the measurement update is simplified using the sensor model (5)–(6). The motion update is applied to all nodes, whereas the measurement update is only applied to the nodes in the field of view of each agent.

For cells that contain nodes of the explored occupancy graph, the LRDT gives the likelihood ratio Λ_c that is the sum of likelihood over states $s = (p, c)$ in $\mathcal{G}_S$ with c corresponding to the occupancy graph node whose cell is being updated. Given Λ_c the cell probability is set to $(\mathfrak{v}, \mathfrak{u}, \mathfrak{o}) = (0, 1/(\Lambda_c + 1), \Lambda_c/(\Lambda_c + 1))$.

C. Predicting Occupancy Nodes via Kriging

We refer to cells as explored if they have been updated using (7) at any preceding time instant. The void probabilities of explored cells are used to predict the void probabilities of unexplored cells using anisotropic kriging interpolation [35]. Since edges of an occupancy graph in an urban environment tend to be oriented along a few dominant directions, e.g., due to parallel streets and sidewalks, we propose the following kernel for kriging: $K(\mathbf{h}) = \sum_{p=1}^{P} \omega_p K_e\left(\mathbf{T}\mathbf{R}_p\mathbf{h}\right)$, where $\mathbf{T} = \mathrm{diag}([1/a_{\max}\ 1/a_{\min}])$ is a constant scaling matrix, $\mathbf{R}_p$ is a rotation matrix defined by an angle φ_p, $\mathbf{h}$ is the spatial lag vector, ω_p is a set of weights such that $\sum_{p=1}^{P} \omega_p = 1$, and $K_e(\mathbf{x}) = \exp\left(-||\mathbf{x}||\right)$ is the exponential kernel. The proposed kernel rotates and scales the spatial lag vector to yield higher correlation among occupancy nodes that lie along the P dominant directions $\{\varphi_1, \cdots, \varphi_P\}$. The correlations are scaled by the weights ω_p to account for some edge directions being more common. The dominant directions may be known a priori or can be determined adaptively by finding peaks of a histogram of edge directions in the explored occupancy graph (as done here). Unexplored cells are updated with $\mathfrak{v}' = \mathfrak{v}_{\mathrm{krig}}$, where $\mathfrak{v}_{\mathrm{krig}}$ is the interpolated value and the remaining probability $1 - \mathfrak{v}_{\mathrm{krig}}$ is allocated as $\mathfrak{u}' = \eta(1 - \mathfrak{v}_{\mathrm{krig}})$ and $\mathfrak{o}' = (1 - \eta)(1 - \mathfrak{v}_{\mathrm{krig}})$, where $\eta = \bar{p}_\phi / M_{\mathrm{unexplored}}$ is the ratio of the probability that no target is present to the number of remaining unexplored cells.

D. Task Generation via Mutual Information

Mutual information $I(C; GZ)$ is the expected reduction in uncertainty of the cell state C that results from obtaining measurements G and Z in that cell. Regions of high mutual information are used to define sampling tasks (waypoints to visit) for the search agents. The uncertainty in cell state is given by the entropy [36]

$$H(C) = \sum_{c \in \{V,U,O\}} -p(c) \log p(c) = -\log(\mathfrak{v}^{\mathfrak{v}} \mathfrak{u}^{\mathfrak{u}} \mathfrak{o}^{\mathfrak{o}}) \ , \quad (16)$$

where the log is taken with base two to give units of bits. Under the local Markov property assumptions (i.e., G and Z are independent given C) [36],

$$H(C|G,Z) = -\sum_{q=1}^{n_z} \sum_{l=1}^{n_g} \sum_{c \in C} k_c \log[k_c / p(g_l, z_f)] \ , \quad (17)$$

where $k_V = z_{f|V,U} g_{l|V} \mathfrak{v}$, $k_U = z_{f|V,U} g_{l|U,O} \mathfrak{u}$, and $k_O = z_{f|O} g_{l|U,O} \mathfrak{o}$. Thus, mutual information [36]

$$I(C; GZ) = H(C) - H(C|G,Z) \ , \quad (18)$$

is computed using (16) and (17).

Tasks are generated and assigned periodically every T_{replan} seconds. To generate tasks, mutual information is evaluated for each cell. Then the search area is partitioned into n_t regions using a variant of Lloyd's algorithm [28] that uses a non-uniform density function (i.e., the mutual information). A waypoint task is defined at the center of mass of each region to produce the tasks $\mathcal{W}$ that are used for path planning, as described next.

E. Path Planning via Step-wise Greedy Task Assignment

A path is constructed for each agent that consists of n_w waypoints: $\mathbf{w}_i(1) \rightarrow \cdots \rightarrow \mathbf{w}_i(n_w)$, where $\mathbf{w}_i(\tau) \in \mathcal{W}$ represents a waypoint at step τ along the path that is assigned to agent i. The initial waypoint $\mathbf{w}_i(1)$ is set to the ith agent's current location. An assignment problem is solved iteratively to assign the remaining $n_w - 1$ waypoints. At a given step $\tau \geq 2$, agent i considers moving to a waypoint in a candidate set $\mathcal{W}^i_{\mathrm{cand}}(\tau) \subset \mathcal{W}$ that consists of its n_b nearest neighbors [37]. Waypoints already assigned at an earlier step are excluded from $\mathcal{W}^i_{\mathrm{cand}}(\tau)$. The union of candidate waypoints across all agents is $\mathcal{W}_{\mathrm{cand}}(\tau) = \bigcup_{i=1}^{N} \mathcal{W}^i_{\mathrm{cand}}(\tau)$. A utility-maximizing assignment problem is solved using the Hungarian algorithm [38] with the $N \times |\mathcal{W}_{\mathrm{cand}}(\tau)|$ utility matrix $\mathbf{R}$ whose entry on row i and column j is

$$\mathbf{R}(i,j) = \begin{cases} \frac{I_{\max}(\mathbf{w}_i(\tau-1), \mathbf{w}_j)}{T(\mathbf{w}_i(\tau-1), \mathbf{w}_j)} & \text{if } \mathbf{w}_j \in \mathcal{W}^i_{\mathrm{cand}}(\tau) \ , \\ 0 & \text{otherwise} \ , \end{cases} \quad (19)$$

where $I_{\max}(\mathbf{w}_i(\tau-1), \mathbf{w}_j)$ is the maximum cell-wise mutual information (18) accumulated in the ith agent's field of view as it moves from waypoint $\mathbf{w}_i(\tau - 1)$ to waypoint $\mathbf{w}_j \in \mathcal{W}^i_{\mathrm{cand}}(\tau)$ and $T(\mathbf{w}_i(\tau - 1), \mathbf{w}_j)$ is the associated transit time. Zero utility is set for non-neighboring waypoints to penalize such assignments. The number of waypoints n_w is set to be sufficiently large so that agents do not idle at their final waypoint before re-planning occurs. After each assignment step, the mutual information surface is updated to remove the information along the assigned path. The procedure is conflict-free and step-wise locally optimal since the Hungarian algorithm guarantees a unique assignment and maximizes total utility [38].

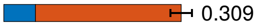
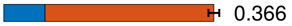
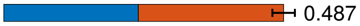
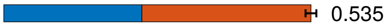
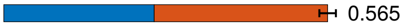
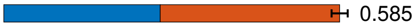
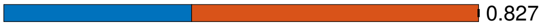
TABLE I: (top) Parameters used in Monte Carlo simulation; (bottom) parameters used by the mutual information strategy.

Parameter	Symbol	Value
Max. speed, acceleration	$(v_{\max}, u_{\max})$	(3 m/s, 2 m/s^2)
Sensing radius	ρ	12.5 m
Sampling time	T_s	1 s
Target motion	(c_m, c_d, p_s)	(1.0, 0.1, 0.5)
Grid cell width	Δ	5 m
Detection threshold	$\bar{\Lambda}$	19 (95% conf.)
Sensor sensitivities	(m_Z, m_G)	(4, 4)
Likelihood decay rate	α	0.1
Anisotropy scaling	$(a_{\max}, a_{\min})$	(5 m, 0.5 m)
No. Voronoi cells	n_t	100
Path length	n_w	4 waypoints
No. neighbors	n_b	15
Replan time	T_{replan}	5 s

IV. Performance Evaluation

Monte Carlo simulations were conducted to compare the performance of the proposed approach to non-adaptive lawnmower and random coverage strategies. Three hundred trials were conducted in which $N = 4$ agents mapped and searched the three 200 m × 200 m urban environments in Fig. 2 for four minutes (a total of 900 trials per strategy). The initial positions of the target and agents were randomly

TABLE II: Monte Carlo simulation performance under various planning and estimation strategies. The search and map components of the metric (20), E_{targ} and E_{nodes}, are shown in blue and red, respectively, in the bar graph results. Error bars indicate standard deviation of E_{nodes}. *Case 7 uses a mutual information strategy with a fully explored occupancy graph ($E_{\text{nodes}} = 0.5$) so that only the E_{targ} metric varied with each simulation.

Cases	Planning			Estimation				Results
	Non-adaptive		Adaptive	Target Detection		Node Prior		$E = E_{\text{targ}} + E_{\text{nodes}}$ ($E = 0$ implies no targets/nodes detected for all mission time) ($E = 1$ implies all targets/nodes detected immediately)
	Random	Lawnmower	Mutual Info.	CBHT	LRDT	Uniform	Kriging	
1	X			X		X		 0.309
2		X		X		X		 0.366
3	X				X	X		 0.487
4		X			X	X		 0.535
5			X		X	X		 0.565
6			X		X		X	 0.585
7			X		X	known map*		 0.827

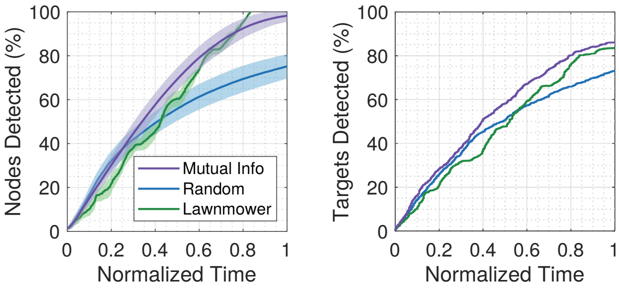
varied. Agents were constrained to start along the search perimeter, except for the lawnmower strategy, for which the agent initial positions were fixed to each corner so that an efficient coverage path could be constructed. Simulation parameters are listed in Table I.

To illustrate the impact of the proposed planning and estimation schemes, several variants of the algorithms were also tested (see Table II). Mapping performance was evaluated by computing the mean and standard deviation of the fraction of occupancy graph nodes detected, $\text{nodes}(t)$, with time $t \in [0, 1]$ normalized by the mission duration. Target-search performance was evaluated by recording the detection time in each trial (if a target was detected) and aggregating the results across all trials to give the fraction of targets detected with time, $\text{targets}(t)$. Overall performance is summarized by the metric

$$E = \underbrace{\frac{1}{2}\int_0^1 \text{targets}(t)\text{d}t}_{0 \leq E_{\text{targ}} \leq 0.5} + \underbrace{\frac{1}{2}\int_0^1 \text{nodes}(t)\text{d}t}_{0 \leq E_{\text{nodes}} \leq 0.5}\,, \qquad (20)$$

which captures both the timeliness and numerosity of node and target detections. The minimum value $E = 0$ represents a total lack of knowledge (no nodes and targets detected for all time), whereas the maximum $E = 1$ represents complete and immediate knowledge of all nodes and targets.

As outlined in Table II, the random and lawnmower strategies were evaluated with either the cell-wise binary hypothesis test or the LRDT approach for target detection (Cases 1–4). Since the LRDT incorporates knowledge of the occupancy graph and target motion, it improves search performance—even with non-adaptive planning. The mutual information strategy was evaluated with a uniform prior for nodes in unexplored regions and the kriging method (Sec. III-C) in Cases 5 and 6, respectively. Since kriging predicts the location of nodes, it guides agents toward exploring the frontiers of the occupancy graph to encounter targets and new nodes more quickly. Lastly, to establish a performance upper bound, the mutual information approach was evaluated assuming a known map (Case 7 in Table II).



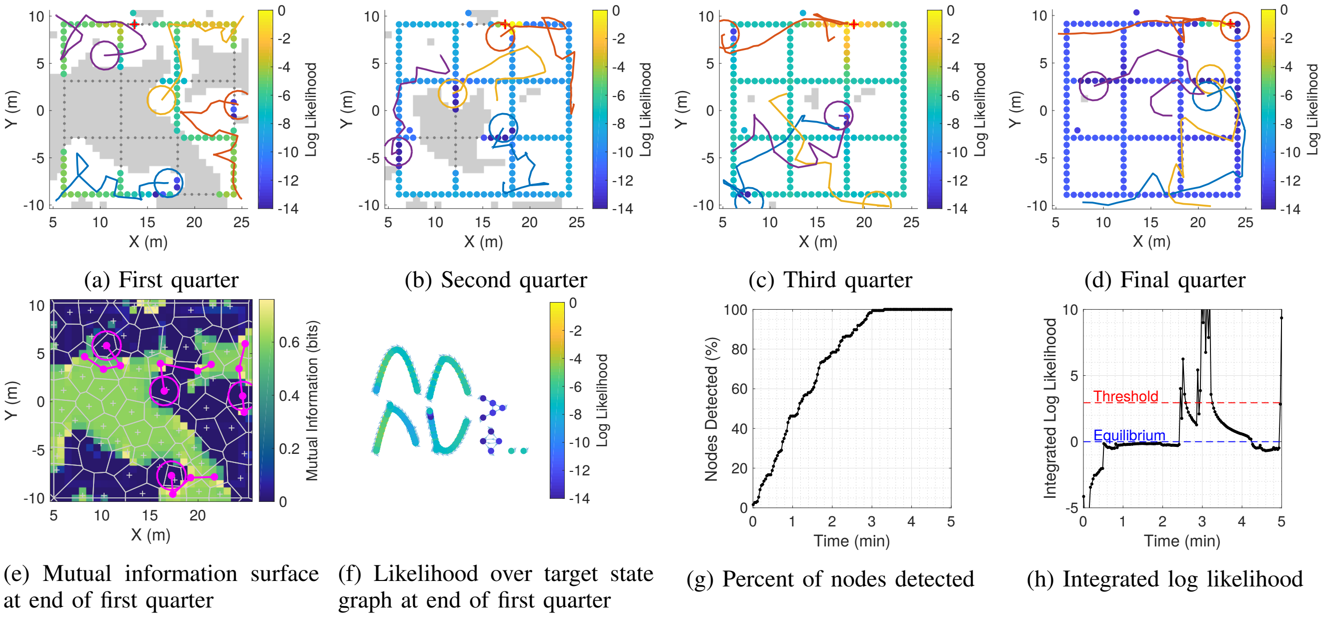
(a) Mapping performance

(b) Search performance

Fig. 4: Monte Carlo results comparing Cases 3, 4, and 6 from Table II. (a) $\text{nodes}(t)$: The mean percent of nodes detected over time (solid lines) with corresponding standard deviation (shaded region) for each coverage strategy; and (b) $\text{targets}(t)$: the percent of targets detected with time. (Time is normalized by the duration of each simulation.)

The Monte Carlo simulation results are further illustrated in Fig. 4 by comparing the curves of $\text{nodes}(t)$ and $\text{targets}(t)$ for Cases 3 and 4 (random and lawnmower with LRDT) to Case 7 (mutual information with kriging). The results indicate that on average, the mutual information strategy detected the most occupancy graph nodes during the middle 25–75% of the mission duration. Mutual information leads to a quick initial exploration of the occupancy graph since regions near frontiers of the graph exhibit high sampling priority. However, this strategy becomes less efficient once the environment is mostly explored and agents backtrack to reach the remaining unexplored cells. As indicated by Fig. 4b, mutual information performs similarly to a random strategy during the first half of the search but outperforms all strategies during the latter half. Since mutual information guides agents to revisit high likelihood regions and to move along the occupancy graph, targets are encountered sooner.

The proposed strategy was also experimentally demonstrated using a team of two real and two virtual quadrotors to map and search a simulated urban environment for five minutes. The quadrotors were constrained to fly in



(a) First quarter

(b) Second quarter

(c) Third quarter

(d) Final quarter

(e) Mutual information surface at end of first quarter

(f) Likelihood over target state graph at end of first quarter

(g) Percent of nodes detected

(h) Integrated log likelihood

Fig. 5: Summary of outdoor flight test demonstration. Panels (a)–(d) show the paths of the quadrotors and the explored occupancy graph with likelihood during each quarter of the mission (real quadrotors in blue/orange and virtual quadrotors in purple/yellow). The field of view of each agent at the end of the quarter is indicated by a circle. Unexplored cells are shown in gray with an overlay of unexplored occupancy graph nodes as darker gray points. The position of the target is a red "+" marker. Panel (e) shows the mutual information surface at the end of the first quarter with Voronoi partitioning (gray cells) and planned waypoint paths (magenta line segments). Panel (f) gives the likelihood over the (explored) target state graph at the end of the first quarter of the mission, corresponding to (a). The target state space contains six connected subgraphs and two isolated target states at this time instant. Panel (g) and (h) indicate mapping and search progress over time; 100% of nodes are detected after 3 min. and the target is detected after 2.5 min.

a 20 m × 20 m area within the Fearless Flight Facility [39], an outdoor netted flight test site at the University of Maryland. The environment was a simplified three-city block occupancy graph with a single target. The open-source version of the Multi-Agent Cooperative Engagement framework (OpenMACE [40]) interfaced with a MATLAB simulation that generated synthetic sensor data based on the quadrotor positions and executed the estimation and planning strategy. (The parameters from Table I were slightly modified for the experimental demonstration to reduce computation time and to account for a smaller search area.)

The results of the flight test are summarized in Fig. 5. The quadrotors generally followed the edges of the occupancy graph (Figs. 5a–5d) since the Voronoi algorithm often placed waypoints near frontiers where cells have high mutual information (Fig. 5e). Since paths were re-planned every 20 seconds, agents had a delayed response to discovered frontiers and exhibited a zig-zag motion that re-directed them back towards the occupancy graph. Half way through the experiment (at 2.5 minutes) about 90% of the occupancy graph nodes are discovered (Fig. 5g) and one of the real quadrotors (colored orange) traverses over the virtual target (Fig. 5b). The integrated likelihood ratio surpasses the threshold, and the target is detected (Fig. 5h). As the agent leaves the targets vicinity to continue mapping, the likelihood diffuses and entropy around the target increases. This increase in entropy prompts the agent to revisit the target twice more: at about 3 minutes and near the end of the mission at 5 minutes. During the final 30 seconds the orange quadrotor in Fig. 5d searches possible routes that the target may have taken before locating it in the north-east corner. The experiment concludes with the quadrotor landing on top of the target.

V. Conclusion

This paper presents a cooperative-control algorithm that guides a team of autonomous quadrotors to simultaneously build an occupancy graph of an uncertain environment and detect a single target moving on this graph.

An outdoor flight test experiment involving two real and two simulated quadrotors demonstrates the feasibility of the proposed strategy, and numerical simulations compare map and search performance to non-adaptive lawnmower and random coverage strategies. The results suggest that using mutual information leads to nodes and targets being detected sooner in comparison to the non-adaptive methods. Mutual-information-based planning is thus most appropriate for scenarios in which gathering information is time sensitive (e.g., in search and rescue applications). Initially, when the environment is completely unknown, random motion that covers new ground is just as effective as mutual information. Conversely, given sufficient time to exhaustively cover a search area, lawnmower coverage surpasses mutual information in mapping performance and performs nearly as well in search performance. Mutual information is beneficial in an intermediate regime—when the team has been deployed long enough to estimate and exploit the environment, but there is insufficient time to exhaustively cover the area. Inefficiencies later in the mission arise when the environment is mostly explored and agents must backtrack to reach the remaining unexplored regions.

While the focus of this work is cooperative mapping and search, the adaptive sampling framework could be used in other applications by defining the cell state to encode different or additional objectives (e.g., classifying targets or mapping an additional scalar field). Moreover, varying the dynamics and sensing modalities of heterogeneous agents could be accommodated when computing mutual information and utility for task assignment. Ongoing work aims to consider multiple targets, develop dedicated Bayesian trackers for detected targets, and incorporate spatially varying sensor performance to model occlusions.

VI. Acknowledgments

We thank Kenneth Kroeger and Patrick Nolan of Heron Systems, Inc. for their helpful discussions and assistance in integrating the OpenMACE software into our outdoor multi-vehicle flight testbed.

References

[1] "SARTopo – Backcountry Mapping Evolved," 2019. [Online]. Available: https://sartopo.com/

[2] B. J. Julian, S. Karaman, and D. Rus, "On mutual information-based control of range sensing robots for mapping applications," *Int. J. Robotics Research*, vol. 33, no. 10, pp. 1375–1392, 2014.

[3] S. Thrun, "Robotic mapping: A survey," in *Exploring Artificial Intelligence in the New Millennium*, G. Lakemeyer and B. Nebel, Eds. Morgan Kaufmann Publishers, 2003, pp. 1–31.

[4] M. J. Kuhlman, M. W. Otte, D. Sofge, and S. K. Gupta, "Multipass target search in natural environments," *Sensors*, vol. 17, no. 11, pp. 1–36, 2017.

[5] J. Hu, L. Xie, K. Y. Lum, and J. Xu, "Multiagent information fusion and cooperative control in target search," *IEEE Trans. Control Systems Technology*, vol. 21, no. 4, pp. 1223–1235, 2013.

[6] J. McMahon, H. Yetkin, A. Wolek, Z. J. Waters, and D. J. Stilwell, "Towards real-time search planning in subsea environments," in *IEEE Int. Conf. Intelligent Robots and Systems*, 2017, pp. 87–94.

[7] Y. Yang, A. A. Minai, and M. M. Polycarpou, "Evidential map-building approaches for multi-UAV cooperative search," in *American Control Conf.*, 2005, pp. 116–121.

[8] H. Yu, K. Meier, M. Argyle, and R. W. Beard, "Cooperative path planning for target tracking in urban environments using unmanned air and ground vehicles," *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 2, pp. 541–552, 2015.

[9] M. Senanayake, I. Senthooran, J. C. Barca, H. Chung, J. Kamruzzaman, and M. Murshed, "Search and tracking algorithms for swarms of robots: A survey," *Robotics and Autonomous Systems*, vol. 75, pp. 422–434, 2016.

[10] C. Robin and S. Lacroix, "Multi-robot target detection and tracking: taxonomy and survey," *Autonomous Robots*, vol. 40, no. 4, pp. 729–760, 2016.

[11] T. H. Chung, G. A. Hollinger, and V. Isler, "Search and pursuit-evasion in mobile robotics: A survey," *Autonomous Robots*, vol. 31, no. 4, pp. 299–316, 2011.

[12] E. Kaufman, K. Takami, T. Lee, and Z. Ai, "Autonomous exploration with exact inverse sensor models," *J. Intelligent and Robotic Systems*, vol. 92, no. 3-4, pp. 435–452, 2018.

[13] T. O. Fossum, J. Eidsvik, I. Ellingsen, M. O. Alver, G. M. Fragoso, G. Johnsen, R. Mendes, M. Ludvigsen, and K. Rajan, "Information-driven robotic sampling in the coastal ocean," *J. Field Robotics*, vol. 35, pp. 1101–1121, 2018.

[14] N. Atanasov, J. Le Ny, K. Daniilidis, and G. J. Pappas, "Decentralized active information acquisition: Theory and application to multi-robot SLAM," in *IEEE Int. Conf. Robotics and Automation*, 2015, pp. 4775–4782.

[15] D. Thakur, V. Kumar, G. J. Pappas, N. Atanasov, and B. Schlotfeldt, "Anytime planning for decentralized multirobot active information gathering," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1025–1032, 2018.

[16] N. Sydney, D. A. Paley, and D. Sofge, "Physics-inspired motion planning for information-theoretic target detection using multiple aerial robots," *Autonomous Robots*, vol. 41, no. 1, pp. 231–241, 2017.

[17] B. Charrow, V. Kumar, and N. Michael, "Approximate representations for multi-robot control policies that maximize mutual information," *Autonomous Robots*, vol. 37, no. 4, pp. 383–400, 2014.

[18] G. M. Hoffmann and C. J. Tomlin, "Mobile sensor network control using mutual information methods and particle filters," *IEEE Trans. Automatic Control*, vol. 55, no. 1, pp. 32–47, 2010.

[19] B. Grocholsky, J. Keller, V. Kumar, and G. Pappas, "Cooperative air and ground surveillance," *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, pp. 16–26, 2006.

[20] A. Kolling and S. Carpin, "Pursuit-evasion on trees by robot teams," *IEEE Trans. Robotics*, vol. 26, no. 1, pp. 32–47, 2010.

[21] A. Kolling and A. Kleiner, "Multi-UAV motion planning for guaranteed search," in *12th Int. Conf. Autonomous Agents and Multiagent Systems*, 2013, pp. 79–86.

[22] B. E. Barkley and D. A. Paley, "Multi-target tracking and data association on road networks using unmanned aerial vehicles," in *IEEE Aerospace Conf.*, 2017, pp. 1–11.

[23] ——, "Cooperative Bayesian target detection on a real road network using aerial vehicles," in *Int. Conf. Unmanned Aircraft Systems*, 2016, pp. 53–61.

[24] P. Skoglar, U. Orguner, D. Tornqvist, and F. Gustafsson, "Road target search and tracking with gimballed vision sensor on an unmanned aerial vehicle," *Remote Sensing*, vol. 4, no. 7, pp. 2076–2111, 2012.

[25] C. C. Bidstrup, J. J. Moore, C. K. Peterson, and R. W. Beard, "Tracking multiple vehicles constrained to a road network from a UAV with sparse visual measurements," in *Proc. American Control Conf.*, 2019, pp. 3817–3822.

[26] G. Dudek, M. Jenkin, E. Milios, and D. Wilkes, "Robotic exploration as graph construction," *IEEE Trans. Robotics and Automation*, vol. 7, no. 6, pp. 859–865, 1991.

[27] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *IEEE Computer*, vol. 22, no. 6, pp. 46–57, 1989.

[28] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Trans. Robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.

[29] R. Diestel, "The basics," in *Graph Theory*. New York, NY, USA: Springer, 2000, pp. 1–28.

[30] "OpenStreetMap," 2019. [Online]. Available: https://www.openstreetmap.org

[31] N. A. Macmillan and C. D. Creelman, *Detection Theory: A User's Guide*, 2nd ed., Mahwah, NY, USA, 2005.

[32] M. Richards, J. Scheer, and W. Holm, *Principles of Modern Radar, Vol. I: Basic Principles*. SciTech, 2010.

[33] L. D. Stone, "Chapter 7: Likelihood Ratio Detection and Tracking," in *Bayesian Multiple Target Tracking*. Artech House, 2013, pp. 239–284.

[34] ——, "A Bayesian approach to multiple-target tracking," in *Handbook of Multisensor Data Fusion: Theory and Practice*, 1st ed., D. L. Hall and J. Llinas, Eds. CRC Press, 2001, vol. 3, ch. 10, pp. 1–30.

[35] Z. Sen, "Ordinary Kriging," in *Spatial Modeling Principles in Earth Sciences*. Springer, 2009, pp. 239–245.

[36] T. M. Cover and J. A. Thomas, "Entropy, relative entropy, and mutual information," in *Elements of Information Theory*. John Wiley & Sons, 1991, pp. 12–50.

[37] J. H. Friedman, J. Bentely, and R. A. Finkel, "An algorithm for finding best matches in logarithmic expected time," *ACM Trans. Mathematical Software*, vol. 3, no. 3, pp. 209–226, 1977.

[38] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistics*, vol. 2, no. 1-2, pp. 83–97, 1955.

[39] "Fearless Flight Facility (F3)," 2019. [Online]. Available: https://aero.umd.edu/research/fearless-flight-facility-f3

[40] "OpenMACE – Multi-Agent Cooperative Engagement," 2019. [Online]. Available: https://github.com/heronsystems/OpenMACE