Noname manuscript No.
(will be inserted by the editor)

Physics-Inspired Motion Planning for Information-Theoretic Target Detection using Multiple Aerial Robots

Nitin Sydney · Derek A. Paley · Donald Sofge

Received: date / Accepted: date

Abstract This paper presents a motion-planning strategy for multiple, mobile sensor platforms using visual sensors with a finite field of view. Visual sensors are used to collect position measurements of potential targets within the search domain. Measurements are assimilated into a multi-target Bayesian likelihood ratio tracker that recursively produces a probability density function over the possible target positions. Vehicles are dynamically routed using a controller based on a concept from artificial physics, where vehicle motion depends on the target probability at their location as well as the distance to nearby agents. In this paradigm, the inverse log-likelihood ratio represents temperature, i.e., high likelihood corresponds to cold temperature and low likelihood corresponds to high temperature. Vehicles move at a temperature-dependent speed along the negative gradient of the temperature surface while interacting locally with other agents via a Lennard-Jones potential in order to emergently transition between the three states of matter — solid, liquid, and gas. We show that the gradient-following behavior corresponds to locally maximizing the mutual information between the measurements and the target state. The performance of the algorithm is experimentally demonstrated for visual measurements in a motion capture facility using quadrotor sensor platforms equipped with downward facing cameras.

N. Sydney
The MITRE Corporation, McClean, Virginia, 22102, USA
E-mail: nsydney@mitre.org

D. A. Paley
Department of Aerospace Engineering and Institute for Systems Research, University of Maryland, College Park, Maryland, 20742, USA
E-mail: dpaley@umd.edu

D. Sofge
Naval Research Laboratory, Washington, 4555 Overlook Avenue SW, Washington, D.C., 20375, USA
E-mail: don.sofge@nrl.navy.mil

1 Introduction

Ground target tracking is now readily available using low-cost aerial vehicles equipped with visual sensors. Example applications include search-and-rescue [1] and aerial surveillance [28]. The use of mobile platforms for aerial tracking involves several challenges, including bandwidth limitations of the communication network and intelligent routing of the vehicles to gain information. This paper describes a physics-inspired motion planning strategy based on the output of a Bayesian likelihood ratio tracker that collects noisy measurements of potential targets. The proposed motion planner has the advantage that vehicle motion is coupled to target detection in such a way that the mutual information between the target state and the measurements is locally maximized, while also maintaining a complexity that is linear in the number for agents. The algorithm represents a manifestation of the Dynamic Data-Driven Application Systems (DDDAS) [8] paradigm, in which sensor measurements are used to guide subsequent collection of more data.

There are many approaches to multiple target tracking in the literature. Probabilistic approaches such as [3, 6, 12, 26, 27] use Bayesian inference combined with maximum likelihood or multiple-hypothesis trackers to track multiple (moving) targets. Although the above approaches address how to combine sensor measurements to track and detect targets, only a few techniques [14, 20, 22] effectively move the mobile platforms to find targets and most of these techniques focus on the use of

a single sensor platform. Other approaches [5,13] focus on mapping or optimal sensor selection. Recently, several authors, e.g., [11,15], have used information-based approaches to design optimal sensor placement/control, but it is difficult to scale these approaches to a large number of vehicles due to the complexity in calculating mutual information [4, 10]. In particular, [4] presents work that approximates the mutual information over a finite time horizon for which vehicle paths are optimized to find a single moving target. While similar to our approach, the control used in [4] is better suited for tracking single vehicles, as opposed to the multi-target scenario considered here. Most similar to the work presented here are physics-inspired swarm controllers [2, 7, 25], where vehicles switch behaviors based on their "temperature" to improve the speed and quality of information gathering. However, the behavior in these approaches is explicitly imposed, leading to a large number of tuning parameters. Additionally, these strategies tend to be heuristic, and hence may not have performance guarantees.

In this paper, we describe a potential-based motion-planning strategy for multiple mobile sensor platforms to collaboratively search for multiple mobile targets using visual sensors with a finite field of view. The sensors give a noisy scalar measurement of target presence, e.g., the measurement signal increases if a target is within the sensor field of view. Sensor measurement data are assimilated by a Bayesian likelihood ratio tracker (LRT) that uses a recursive formulation to produce a probability density function over the set of possible target positions. In the LRT formulation, the probability builds in locations near potential targets until a predetermined threshold is reached and a detection is called. Multiple targets may be present; the LRT fuses all measurements into a single state space. The inverse of the state-space posterior, analogous to temperature, triggers emergent search behaviors corresponding to the states of matter: solid, liquid, and gas. The physics-inspired approach results in emergent behaviors that are intuitively plausible and therefore more amenable to human interaction with or control of the multi-vehicle team. An inverse log-likelihood formulation is used so that a higher target likelihood corresponds to a lower temperatures and measurements with a target present add likelihood, whereas measurements with no target present subtract likelihood.

The novelty of this target detection strategy lies in the following architecture. There are two main components to the target detection: (1) the LRT for detection, and (2) a potential-based, physics-inspired motion planner that guides vehicles to areas where the probability of detection is higher. Unlike other potential-based control strategies, the proposed motion planner avoids local minima by coupling the planner to the LRT detector. Vehicle motion is guided by the temperature surface according to artificial forces. The first force is along the negative gradient of the temperature surface; the speed along the gradient is proportional to the temperature. The second force is a spring-like interaction between vehicles via a Lennard-Jones potential [16], which is commonly used to model molecular dynamics between energetic particles. We show that the gradient-following behavior corresponds to locally maximizing the mutual information between the sensor measurements and the target state. The combination of these forces leads to three distinct behaviors corresponding to three states of physical matter: (a) a solid state, in which vehicles coalesce into a lattice structure around likely target locations; (b) a liquid state, in which the vehicles migrate towards colder regions likely to a harbor target; and (c) a gas state, in which solitary vehicles move quickly through hotter regions where targets are unlikely to be found. One key difference between this algorithm and other physics-inspired approaches is that the three states of matter are emergent, so that there are fewer tuning parameters. The presented algorithm, though implemented in a centralized system, is inherently decentralized under an all-to-all communication architecture, and thereby amenable to distributed computation.

There are several papers in the literature, e.g. [9,24], that use potential-based approaches similar to the kind used in this work, however, they suffer from several shortcomings. The primary shortcoming of potential-based algorithms is that vehicles may fall into local minima of the potential. In [9], the authors decrease the probability of remaining in a local minima by introducing an excitation factor, which is a heuristic method for hill climbing. One of the main benefits of the algorithm presented in this paper is that it avoids local minima by coupling vehicle control to target detection via the evolving LRT surface. Another advantage is that the LRT surface generated on-the-fly (and recursively updated) doesn't incur the excessive computational burden imposed by explicit calculation of the mutual information, which usually scales exponentially.

We demonstrate the proposed cooperative detector first in simulation, and then in a motion-capture experimental testbed, utilizing the Laboratory for Autonomous Systems Research at the Naval Research Laboratory in Washington, D.C. The experimental testbed is composed of two Ascending Technologies Pelican quadrotors with downward facing Point Grey Chameleon cameras and three Sphero robots—rolling targets programmed to perform a random walk. Although visual sensors are

described here, any sensor characterized by its probability of detection and probability of false alarm may be employed in this algorithm.

The primary contribution of this paper is a physics-based motion planner for a multi-vehicle, aerial sensor network that uses a Bayesian likelihood ratio tracker incorporating position information from onboard cameras to track multiple moving targets. One of the novelties of the DDDAS paradigm is that it directly couples the motion planning to the estimation scheme, so gathered information directly influences the subsequent movement of the platforms. As a secondary contribution, we show that the gradient-following behavior of the planner locally maximizes the mutual information between the measurement and the target state. Unlike other information-based techniques, the algorithm developed in this paper scales linearly with the number of agents as a result of the rapid drop-off of the Lennard-Jones potential. Additionally, the three states of matter are emergent, based on a combination of the speed control, Lennard-Jones potential, and gradient-descent behaviors. Vehicles are deployed to detect an unknown number of moving targets in a computationally efficient manner that locally maximizes the probability of detecting targets.

The rest of the paper is organized as follows. Section II summarizes the likelihood-ratio detection and tracking calculations and the model used for the quadrotors' visual sensors. Section III presents the physics-based motion-planning algorithm for a team of mobile sensors. Section IV shows experimental results from the quadrotor testbed. Section V summarizes the paper and ongoing work.

2 Likelihood-ratio detection and tracking

This section reviews the likelihood ratio tracker [26] and formulates sensor and motion models for the vehicles and targets, respectively. The output of the tracker is used in the physics-inspired motion planner presented in Section 3.

2.1 Likelihood-ratio tracking

A likelihood ratio tracker is used to estimate the positions of possibly multiple targets. This framework is sometimes called "track-before-detect" [26], because it is based on recursive Bayesian estimation. A Bayesian filter is a probabilistic approach for assimilating noisy measurements into a probability density function over the target state space, in this case is their two-dimensional position. The filter implementation comprises the discrete steps of predicting and updating. Let $\theta_k = (x_k, y_k) \in \Omega$ denote the target position (state) at time step k and z_k denote the measurement at time k. The predict step involves computing the conditional probability [26]

$$p(\theta_k|z_{k-1}) = \int_\Omega p(\theta_k|\theta_{k-1})p(\theta_{k-1}|z_{k-1})d\theta_{k-1}. \quad (1)$$

The measurement update is proportional to the product of the measurement likelihood $p(z_k|\theta_k)$ and the predicted state [26]:

$$p(\theta_k|z_k) = \frac{p(z_k|\theta_k)p(\theta_k|z_{k-1})}{p(z_k|z_{k-1})}, \quad (2)$$

where

$$p(z_k|z_{k-1}) = \int_\Omega p(z_k|\theta_k)p(\theta_k|z_{k-1})d\theta_k$$

is the integral of the numerator. Note that measurements from multiple sensors are assimilated by executing multiple, independent update steps.

A likelihood-ratio tracker replaces the measurement likelihood with the measurement likelihood ratio, i.e., the ratio of two likelihood functions. The numerator of the likelihood ratio represents the conditional probability of the measurement given that the target is present (θ_k^+), whereas the denominator represents the conditional probability of the measurement given that the target is not present (θ_k^-). Thus, the log-likelihood ratio is

$$\log \mathcal{L}(z_k|\theta_k) = \log \frac{p(z_k|\theta_k^+)}{p(z_k|\theta_k^-)}$$
$$= \log(p(z_k|\theta_k^+)) - \log(p(z_k|\theta_k^-)). \quad (3)$$

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

$$P(\theta_k|z_k) = \log \frac{\mathcal{L}(z_k|\theta_k)p(\theta_k|z_{k-1})}{p(z_k|z_{k-1})}$$
$$= P(z_k|\theta_k^+) - P(z_k|\theta_k^-) + P(\theta_k|z_{k-1}) - P(z_k|z_{k-1}). \quad (4)$$

The first term in (4) represents the new, positive information (likelihood the target is present), whereas the second term represents the new, negative information (likelihood the target is not present). The third term represents the prior information and the fourth term is a normalization constant. When the log-likelihood ratio surpasses a pre-defined threshold, the target is declared detected; otherwise, the sub-threshold target probabilities are maintained as hypotheses. Note, the inverse log-likelihood-ratio posterior represented by temperature is $-P(\theta_k|z_k)$.

2.2 Predict step: Integrating the diffusion equation

The predict step (2) involves updating the likelihood ratio in the absence of measurements. We allow targets to move randomly; we do not estimate their velocities. The target motion model is a random walk, which can be described mathematically by the diffusion equation with constant diffusivity, also called the heat equation.

Consider the two-dimensional heat equation with diffusivity α, i.e.,

$$\frac{\partial P}{\partial t} = \alpha \left(\frac{\partial^2 P}{\partial x^2} + \frac{\partial^2 P}{\partial y^2} \right).$$

The alternating direction implicit (ADI) method [18] is a fractional-step method for numerically integrating the heat equation. ADI employs the difference operators A_x and A_y representing the spatial derivatives in the x and y directions, respectively. For example, if probability P is discretized over $n \times n$ grid points, then $A_x P$ is a vector of length $n \times n$ with elements

$$\frac{P_{i+1,j} - 2P_{i,j} + P_{i-1,j}}{\Delta x^2}, \quad i = 1, \ldots, n, \quad j = 1, \ldots, n.$$

Let I be the $n \times n$ identity matrix. The solution is obtained from $P(\theta_{k-1}|z_{k-1})$ using matrix inversion to solve the following equation for $P(\theta_k|z_{k-1})$:

$$\left(I - \tfrac{\alpha \Delta t}{2} A_x\right)\left(I - \tfrac{\alpha \Delta t}{2} A_y\right) P(\theta_k|z_{k-1}) = \left(I + \tfrac{\alpha \Delta t}{2} A_x\right)\left(I + \tfrac{\alpha \Delta t}{2} A_y\right) P(\theta_{k-1}|z_{k-1}).$$

To ensure that no information crosses the boundary of the search domain, we enforce Neumann boundary conditions around the perimeter of Ω by specifying that the normal component of the gradient is zero.

This motion model diffuses likelihood into locations that are not visited. It promotes visitation of areas of low likelihood to ensure that a target has not moved to that location. Note that while a random walk motion model is used in this paper, it is not necessary for the use of the proposed control strategy; any motion model for updating target position may be used.

2.3 Update step: The sensor measurement model

Consider a measurement data model based on an imperfect sensor with finite field of view. Suppose the criterion location for a positive response is chosen such that targets within range ρ_{target} are detected with probability P_d and false alarms occur with probability P_f per time step. The sensitivity m of each sensor is [17]

$$m = z(P_d) - z(P_f), \tag{5}$$

where $z(\cdot)$ represents the z-transformation into standard deviation units given by the quantile function [17]

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

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 $z_k = w_k$, whereas when the target is present the measurement data is $z_k = m + w_k$. Assuming a zero-mean Gaussian sensor model [23] yields

$$p(z_k|\theta_k^-) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{z_k^2}{2}\right) \tag{6}$$

$$p(z_k|\theta_k^+) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(z_k - m)^2}{2}\right). \tag{7}$$

(The sensor noise variance is absent because z_k and m are already assumed to be in standard deviation units.) The log-likelihood ratio (3) becomes

$$\log \mathcal{L}(z_k|\theta_k) = -\frac{(z_k - m)^2}{2} + \frac{z_k^2}{2} = m\left(z_k - \frac{m}{2}\right), \tag{8}$$

where m is a function of the sensor P_d and P_f given by (5).

While the methodology above would apply to any sensor characterized by its P_d and P_f, in this paper the sensor is a camera with a known field of view (FOV) and resolution. Using standard image processing techniques [19], the camera places a bounding box around targets within the FOV. When there is a target present, (8) with $z_k = m + w_k$ is applied to the prior uniformly inside a disc of radius ρ_{target} located at the center of the bounding box. In areas of the FOV where there is no target, (8) with $z_k = w_k$ is applied to the prior uniformly. P_d and P_f for a camera are complicated functions of the camera resolution, the size of the object in the FOV, and the quality of the image processing algorithm. For the experiments presented in this paper, P_d and P_f were treated as tuning parameters for the tracker and were chosen to improve the performance of the estimation and control strategy.

3 Physics-Inspired Motion Planning

The agent motion is driven by a combination of two artificial forces. The first force guides the vehicle down the gradient of the inverse log-likelihood surface, i.e., the temperature surface, in a manner that resembles a flowing liquid. The speed of movement down the gradient is determined by the temperature at the vehicle location: colder temperatures correspond to slower

speeds. The second force is the gradient of a Lennard-Jones potential [16] between the agents. The Lennard-Jones potential is a common approximation used in physical chemistry to model molecular gas dynamics. Figure 1 shows the vehicles moving like gas molecules in areas where there are no targets and gathering like solid molecules with other agents where there are likely to be targets. In this algorithm, as opposed to other physics-inspired swarm controllers, the transitions between states of matter (solid, liquid, gas) are emergent.

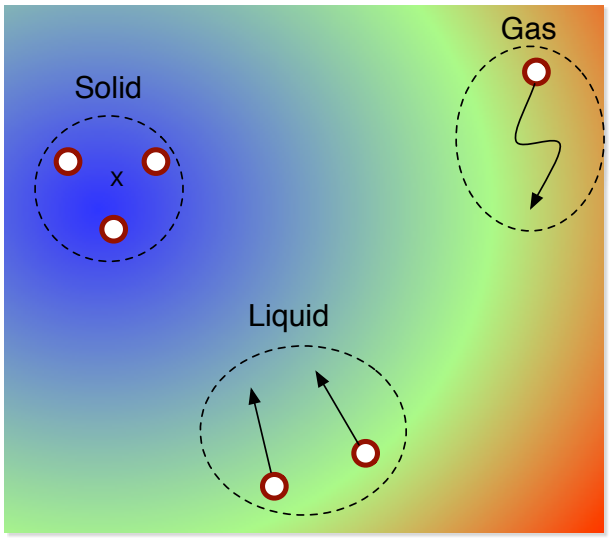


Fig. 1 Diagram depicting the three emergent behaviors of the proposed algorithm. Vehicles in cold areas near a target form a crystalline formation with nearby agents. Vehicles flow like a liquid from warm areas to cold areas. Since speed is proportional to temperature, vehicles in hot areas travel quickly like gas molecules.

Let r_k^j denote the position of agent $j \in \{1, \ldots, N\}$ at time step k. The Lennard-Jones potential is [16]

$$V_k^j = \sum_{i \neq j}^{N} 4\epsilon \left(\sigma^{12} ||r_k^j - r_k^i||^{-12} - \sigma^6 ||r_k^j - r_k^i||^{-6} \right), \quad (9)$$

where ϵ is the depth of the well and σ is the distance at which the potential between two agents is zero. An example of the Lennard-Jones potential between two agents is shown in Figure 2. Note that the strength of the potential drops quickly to zero for large distances. Thus, the interaction between two agents only happens when they are close to one another.

Let $T_k^j = -P(\theta_k | z_k, j)$ in (4) be the temperature at vehicle j at time k and V_k^j be the Lennard-Jones potential (9) for the jth vehicle. We prescribe the desired velocity for each vehicle by a feedback controller. The desired velocity for the jth vehicle is

$$\dot{r}_k^j = -\min(1 + e^{T_k^j}, v_{max}) k_P \nabla T_k^j - k_V \nabla V_k^j, \quad (10)$$

where k_P and k_V are control gains and v_{max} is the maximum speed of the agent (the same for all agents). The dependence on temperature causes the vehicles to slow down when they are near a possible target in order to collect more measurements in that area. This behavior, along with the sensor aggregation caused by the Lennard-Jones potential, seeks to increase the chances that a target will be detected. The algorithm works well in simulations when the gain on the Lennard-Jones potential is one order of magnitude smaller than the gradient-following gain. Additionally, σ is set to ensure that the vehicles do not collide with each other, which depends on the scale of the vehicle. Consequently, there are only three parameters to tune: either k_P or k_V, ϵ (the well depth), and T_0 (the detection threshold).

As seen in equation (10), we multiply the gradient of the temperature potential by a heuristic scaling factor. This scaling factor, in combination with the Lennard-Jones potential, is what causes the states-of-matter behavior to be emergent. In the scaling factor, the speed of the vehicle is scaled by the temperature such that the speed increases when the temperature is high, and decreases when the temperature is slow. This causes vehicles to "freeze" in place when they are near an area of high probability and, conversely, to speed up when they are in areas of low probability. The full architecture (in pseudocode form) is shown in Algorithm 1.

Unlike other works in the field, the architecture shown in Algorithm 1 avoids local minima in an informative, non-heuristic fashion. A vehicle may fall into a local minima if it is surrounded by hills of low target likelihood. However, if there is no target present under-

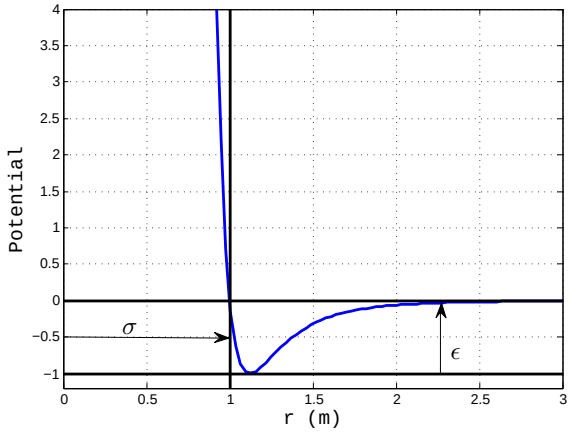


Fig. 2 Example of Lennard-Jones potential; $\epsilon = \sigma = 1.0$.

Algorithm 1 Physics-Inspired Target Search and Detection Algorithm

Require: $\mathrm{N}, \epsilon, \sigma,\ k_P,\ k_V$
Initialize log-likelihood ratio $P_0(\theta|z)$
for $k = 1$ till end of task **do**
  for $j = 1, \ldots, N$ **do**
    Get measurement z_k^j
    Calculate likelihood ratio $\log \mathcal{L}(z_k^j|\theta_k)$ using (8)
    Update: $P_k(\theta|z) = P_{k-1} + \log \mathcal{L}(z_k^j|\theta_k)$
  end for
  Normalize likelihood ratio: $P_k(\theta|z) = P_k(\theta|z) - \text{sum}((P_k(\theta|z)))$
  Get r_k for every vehicle
  for $j = 1 \ldots N$ **do**
    Compute $\nabla V_k^j(r_k)$ and $\nabla P(\theta_k|z_k)$
    Calculate desired velocity $\dot{r}_k^j$
    Apply control input to platform to achieve $\dot{r}_k^j$
  end for
end for

neath the vehicle, the temperature will increase (from negative information) to the point where the vehicle is pushed away from its location. While this does not guarantee convergence to a global minimum, it does ensure that vehicles to not remain in areas of high target probability when a target has moved out of the region. Note, this behavior is completely emergent and is provided inherently by connecting the detection scheme to the vehicle control. In addition to informatively handling local minima, we have the following theorem.

Theorem 1 *The mutual information between sensor measurement z_k and target location θ_k using the sensor model given by* (6)–(7) *is locally maximized when the vehicle moves along the gradient of the prior target distribution $p(\theta_k)$.*

Proof Let $\xi_k = (x_k, y_k)$ be the position of the kth quadrotor. For readability we drop the temporal subscript on all the variables in this proof. The mutual information between z and θ is

$$I(z,\theta) = \int_{z^+,z^-} \int_{\Omega} p(\theta, z|\xi) \log(p(z|\theta,\xi) d\theta dz - \int_{z^+,z^-} p(z|\xi) \log p(z|\xi) dz, \tag{11}$$

where $p(z|\xi)$ is the marginal probability density function of the observation in observation space. Since the sensor model is Gaussian, the logarithm in the first term on the right hand side of (11) evaluates to a constant. Hence, the double integral is independent of ξ. Thus, the mutual information is maximized by maximizing the second term, which is the entropy of the observation, $H(z|\xi)$, given the vehicle location. The marginal probability $p(z|\xi)$ can be written as

$$p(z|\xi) = \int_{\theta \in D} p(z|\theta,\xi) p(\theta) d\theta. \tag{12}$$

Therefore, $H(z|\xi)$ can be expressed as

$$H(z|\xi) = -\int_{z^+,z^-} \int_{\theta \in D} p(z|\theta,\xi) p(\theta) d\theta \times \log\left(\int_{\theta \in D} p(z|\theta,\xi) p(\theta) d\theta \right) dz. \tag{13}$$

Evaluating the integral over z yields

$$\begin{aligned} H(z|\xi) = &- \int_{\theta \in D} p(z^+|\theta,\xi) p(\theta) d\theta \times \log\left(\int_{\theta \in D} p(z^+|\theta,\xi) p(\theta) d\theta \right) \\ &- \int_{\theta \in D} p(z^-|\theta,\xi) p(\theta) d\theta \log\left(\int_{\theta \in D} p(z^-|\theta,\xi) p(\theta) d\theta \right). \end{aligned} \tag{14}$$

Both terms on the right hand side of (14) have the form $f(x) = x \log(1/x)$, which is an increasing function of x as long as $x \leq 1/e \approx 0.36$. In (14), the argument is the integral of $p(\theta)$ scaled by either $p(z^+|\theta,\xi)$ or $p(z^-|\theta,\xi)$. Since these are both less than $1/e$ (see (6)–(7)) and the integral of $p(\theta)$ is less than one in domain D, $H(z|\xi)$ is guaranteed to be an increasing function of the integral of $p(\theta)$.

Because the mutual information is an increasing function of the integral of $p(\theta)$, moving along the gradient of the integral will maximize the mutual information at the next time step. As a first-order approximation, assume that $p(\theta)$ is given by the first-order Taylor expansion

$$p(\theta) \approx a_0 + a_x x + a_y y, \tag{15}$$

where it is assumed that coordinates in D are given by the pair (x, y). Then we have

$$\int_{\theta \in D} p(\theta) d\theta \approx \int_{x_k - \frac{s}{2}}^{x_k + \frac{s}{2}} \int_{y_k - \frac{s}{2}}^{y_k + \frac{s}{2}} (a_0 + a_x x + a_y y) dy dx, \tag{16}$$

where we have assumed that D is a square region with side length s for simplicity. The gradient of (16) yields

$$\nabla \left(\int_{\theta \in D} p(\theta) d\theta \right) \approx a_x s^2 \hat{e}_x + a_y s^2 \hat{e}_y, \tag{17}$$

where $\hat{e}_x$ and $\hat{e}_y$ are unit vectors in the x and y directions respectively. Taking the gradient of $p(\theta)$ directly shows that

$$\nabla p(\theta) \approx a_x \hat{e}_x + a_y \hat{e}_y \tag{18}$$

$$= \frac{1}{s^2} \nabla \left(\int_{\theta \in D} p(\theta) d\theta \right). \tag{19}$$

Thus, moving along the gradient of $p(\theta)$ is locally the same as moving along the gradient of its integral, which completes the proof.

Theorem 1 shows that the sensor model coupled with a gradient control allows each vehicle to greedily maximize the expected gain in information without explicitly having to calculate it. This property has significant benefit on the ability of the search strategy to scale with numbers of vehicles and targets, as compared to the calculation of mutual information, which generally scales exponentially. (Note, Theorem 1 only holds locally.) Since the Lennard-Jones potential drops off dramatically with distance, each vehicle need only consider the position of other nearby agents. The first term in (10) scales with the calculation of the temperature surface, which scales linearly with the number of vehicles. The second term, the Lennard-Jones potential, also scales linearly with the number of vehicles. Therefore, since the two terms add in the controller, the algorithm scales linearly with the number of vehicles. Note that if the algorithm was distributed, the complexity would be fixed by local vehicle density under a distance-based communication topology.

Algorithm performance is evaluated via simulation in the operating domain $\Omega = [-5 \quad 5] \times [-5 \quad 5]$ m. Agents and targets are constrained to remain in Ω at all times. Parameter definitions and values used in the simulation are provided in Figure 3. The local temperatures of the agents and targets are obtained via linear interpolation between the grid points. The overall temperature surface is normalized by subtracting its mean every time step. To avoid overheating in the temperature surface, the individual sensor updates are scaled by N, i.e., the number of agents. This scaling has the effect of making the collective detection performance in terms of the cumulative number of targets detected over time roughly independent of N.

Figure 4 shows the results of the simulation. Figures 4(a) and 4(b) depict snapshots of the vehicles, targets, and temperature (inverse log-likelihood) surface at time steps $k = 125$ and 425 respectively. Vehicles are shown as red circles, undiscovered targets as white crosses, and discovered targets as red crosses. At any moment, there are only three undiscovered targets in the domain. Once a target is detected, it turns red and another target appears with uniform probability somewhere in the domain. The heat map, which describes the temperature, is blue in areas of low temperature (high likelihood) and red in areas of high temperature (low likelihood).

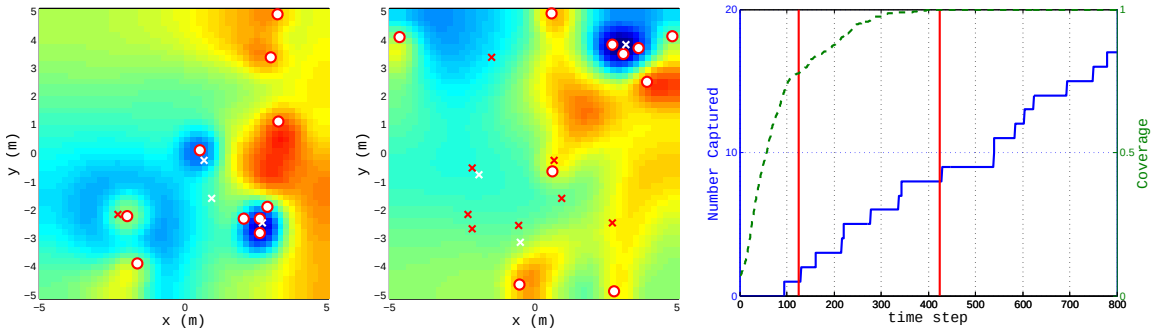
Parameter	Value	Definition
N	10	number of agents
M	3	instantaneous number of targets
α	0.5	target diffusivity
σ	1.5 m	repulsive threshold
ϵ	0.2	Lennard-Jones potential depth
T_0	-15 °	target detection threshold
K_P	5	gradient-following gain
K_V	0.05	Lennard-Jones gain
$v_{\max}$	0.3 m/s	maximum agent speed
P_d	0.95	probability of detection
P_f	0.10	probability of false alarm
ρ_{target}	α	Sensor detection range
n	50	number of grid points in each dimension
Δx	$2/n$	grid spacing
Δt	0.03 s	time step

Fig. 3 Parameter values and definitions

In Figure 4(a), one target has been detected. There are several vehicles scanning the warmer regions like gas molecules, while others are clustering around potential targets in a semi-solid configuration. Figure 4(c) shows the cumulative number of targets captured during the simulation (solid line) and the amount of time it takes for the vehicles to fully cover the domain once (dashed line). The vertical red lines indicate the times the snapshots were taken. This algorithm follows a typical coverage curve, with approximately linear growth in coverage early on, followed by slower asymptotic convergence to full coverage. Also note the target capture rate is approximately linear.

4 Experimental Results

As a demonstration of the proposed technique we implemented the algorithm in hardware using a quadrotor testbed at the Laboratory for Autonomous Systems Research located at the Naval Research Laboratory. Note that the experiments conducted in this paper are for proof-of-concept and not intended to serve as a rigorous verification of the algorithm. The testbed is composed of Ascending Technologies Pelican quadrotors operated in a facility with 115 Vicon motion capture cameras. Figure 5 shows the prototyping high bay where the experiments were conducted. The quadrotors are equipped with active LED motion capture markers in order to be seen by the Vicon cameras.



(a) Time step $k = 125$ (b) Time step $k = 425$ (c)

Fig. 4 (a)(b) Snapshot of the simulation; (c) The number of targets captured throughout the simulation and the amount of time taken to cover the entire domain once. Vehicles cluster like solid molecules in cold regions near the target and explore randomly like gas particles in hot regions, while flowing down the temperature gradient from hot to cold.

The quadrotors each have a downward facing Point Grey Chameleon 1.3 megapixel camera with a resolution of 1296 × 964 pixels. The camera is equipped with a Tamron CCTV manual lens to adjust zoom and focus. The camera has a global shutter to minimize motion blur and adjustable white balance and shutter speed. The quadrotors have an onboard Linux computer that processes camera frames at 15 Hz, which is limited by how fast images can be captured by the camera. A binary measurement signal sent to the LRT represents whether or not a blob of a predetermined color is in the image. For the experiments, the targets were Sphero robots, a spherical rolling toy with an orange cover. Figure 6 shows an example of the blob-tracking software used to detect two Spheros in the field of view of the Point Grey camera.

Pose and target information from each vehicle is sent to a laptop that uses the Robot Operating System (ROS) [21] to process the data and compute control commands in a control node. ROS is an open-source software architecture that contains software libraries and tools to build robotic applications. The binary signal is generated by using a standard blob segmentation algorithm [19] in a camera processing ROS node onboard the vehicle. The vehicle also has a ROS node to handle passing measurements to the control node running on a laptop as well as receiving position and control commands from the external laptop and motion capture system. Control commands are sent to the vehicle through a WiFi connection to the message processing node at approximately 30 Hz. The full architecture for the experiment is shown in Figure 7.

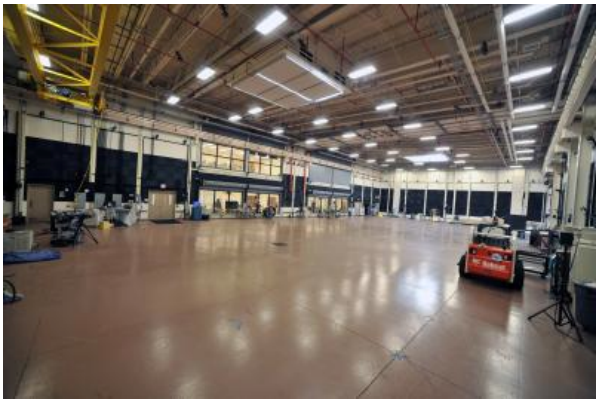


Fig. 5 Prototyping high bay in the Laboratory for Autonomous Systems Research at the Naval Research Laboratory.

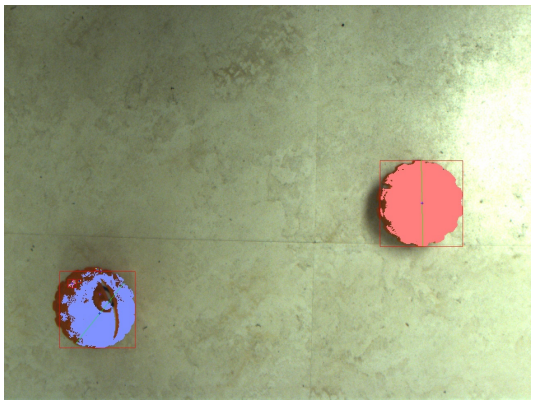


Fig. 6 Processed camera image showing two orange Spheros identified using a color blob detector.

4.1 Single-Vehicle, Single-Target Experiment

To verify the algorithm using the hardware testbed, experiments were first conducted with a single vehicle and a single moving target. The Sphero served as an ideal target once covered with a protective colored

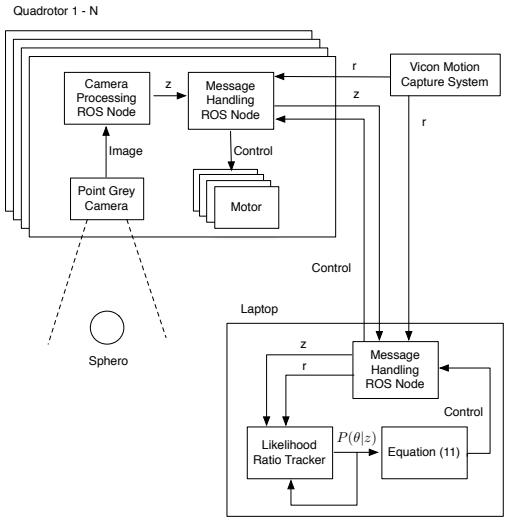


Fig. 7 Hardware architecture for the experiments. The quadrotors run two ROS nodes to process camera data and handle incoming position measurements and desired velocity. An offboard laptop receives position and blob detection measurements to compute the temperature surface and calculate the desired velocity.

cover, since a color-blob detection algorithm reliably located it within the image frame of the camera. We commanded the Sphero speed but not its direction, yielding a random walk behavior due to the cover's bumpiness.

Figure 8 shows a snapshot of an experiment. The targets are surveilled using the Point Grey camera; color blobs (and false alarms) were found in each image frame. The blob detections are fed into the LRT to calculate the inverse log-likelihood surface, from which the desired control is calculated. Once a target is detected, any blob within a 0.5 m radius of the detected-target location is ignored by the LRT tracker. The Sphero was commanded to move in a random walk, so multiple detections of the same Sphero were possible.

In this experiment, the moving Sphero was detected three times as it performed the random walk. The locations where the target was detected are indicated by the red crosses. Figures 8(a) and 8(b) show snapshots of the experiment at time steps $k = 120$ after the first detection and $k = 220$ after the third detection. The number of detections over time is shown in Figure 8(c), where the vertical red lines indicate the times of the snapshot shown in Figures 8(a) and 8(b). In a real target-tracking application, once the target is detected using Algorithm 1 the vehicle would switch to another controller to maintain a line-of-sight track of the detected target such that multiple detections are unnecessary. For illustrative purposes, the vehicle in this experiment continued using Algorithm 1 even after a detection was called. The single-vehicle experiments used the same gains that were used in simulation; the Lennard-Jones potential was omitted, as it is pertinent only in the multi-vehicle experiment described next.

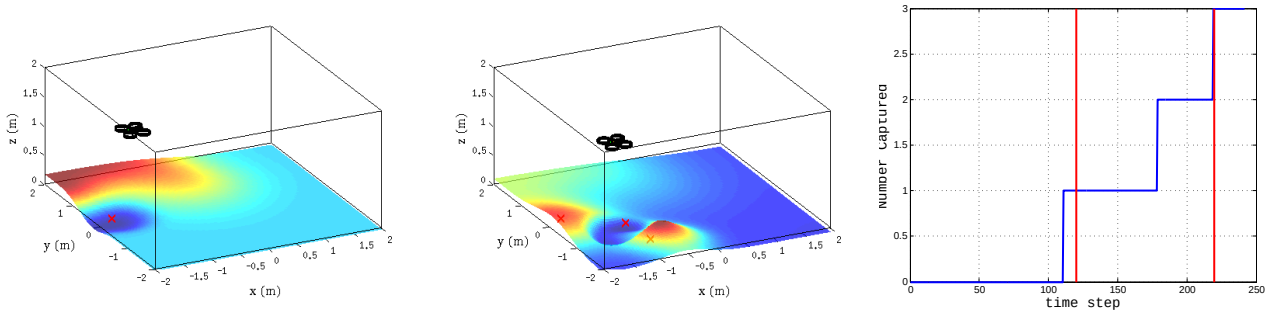
4.2 Multi-Vehicle, Multi-Target Experiment

A multi-vehicle experiment was also conducted with stationary targets located at (-0.25, -0.35), (1.60, 0.40), and (1.90, -1.00) meters. Figures 9(a) and 9(b) show snapshots of the inverse log-likelihood surface at time steps $k = 60$ and 155 with the position of the undetected targets shown as black crosses and detected targets shown as red crosses. At the depicted time step, there is one target that has been recently detected, one that has yet to be discovered, and a third that in the process of being detected. Figure 9(c) shows the number of targets captured and the area covered over time. With two vehicles, all targets were captured within 200 time steps and the vehicles covered the full domain in approximately 400 time steps, which corresponds to 40 seconds.

Comparing Figure 4(c) to Figure 9(c), we see that the coverage rate in the experiment follows the same patterns as the simulated experiments, i.e., initial linear growth followed by asymptotic convergence. It was also determined that, in general, the vehicles must have lower gains than in simulation. As observed by comparing Figure 4(c) to Figure 9(c), lower gains result in a slower capture rate. With higher gains, the vehicles tended to overshoot the targets before the temperature was cool enough to slow them down. The discrepancy between simulation and experiment can be explained by observing that the dynamics of the quadrotors are inherently second-order, whereas the desired velocity stated in (10) is for vehicles with first-order dynamics. The Pelicans approximate this control by treating the output of (10) as a desired velocity to be achieved by the onboard flight controller.

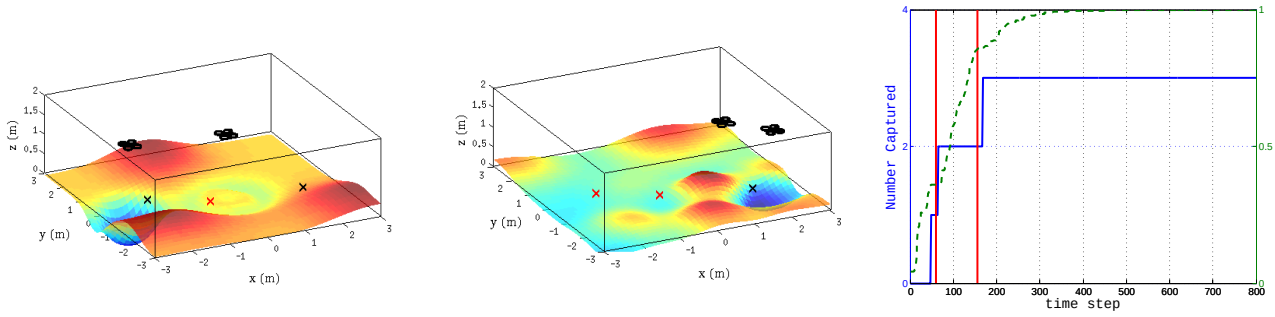
5 Conclusion

We present a physics-inspired target search algorithm for multiple agents and multiple moving targets. The agents assimilate noisy measurements from onboard sensors to produce log-likelihood ratio estimates of target positions over the operating domain. The inverse log-likelihood surface plays the role of temperature so that agents speed up and slow down along the temperature (information) gradient. Additionally, vehicles are influenced by a Lennard-Jones potential between other local vehicles. The Lennard-Jones potential provides collision avoidance as well as local collaboration with



(a) Time step $k = 120$ (b) Time step $k = 220$ (c)

Fig. 8 (a)(b) Snapshot of the single-vehicle experiment; (c) The number of targets captured throughout the experimetn and the amount of time taken to cover the entire domain once.



(a) Time step $k = 60$ (b) Time step $k = 155$ (c)

Fig. 9 a)(b) Snapshot of the multi-vehicle experiment; (c) The number of targets captured throughout the experiment and the amount of time taken to cover the entire domain once.

other vehicles. The approach is amenable to heterogeneous agents and to distributing the computation of the log-likelihood surface over the agents. Multi-vehicle experiments verify the vehicle routing algorithm. In ongoing work, we are conducting experiments with more quadrotors and moving targets. We are also modifying the proposed algorithm for distributed computation.

Acknowledgements The authors would like to acknowledge Keith Sullivan for the Sphero computer vision tracking software and Tom Apker and Frank Lagor for discussions pertaining to the design of the control algorithm. This work was performed at the Naval Research Laboratory and was funded by the Office of Naval Research under grant number N0001413WX21045, "Mobile Autonomous Navy Teams for Information Surveillance and Search (MANTISS)", and the Air Force Office of Scientific Research under DDDAS Grant No. FA95501310162. The views, positions and conclusions expressed herein reflect only the authors' opinions and expressly do not reflect those of the Office of Naval Research, Air Force Office of Scientific Research, or the Naval Research Laboratory.

References

1. Almurib, H., Nathan, P., Kumar, T.: Control and path planning of quadrotor aerial vehicles for search and rescue. In: Proceedings of SICE Annual Conference (SICE), pp. 700–705 (2011)
2. Apker, T.B., Potter, M.A.: Robotic swarms as solids, liquids and gasses. In: AAAI Fall Symposium Series: Human Control of Bioinspired Swarms, AAAI Technical Report FS-12-04 (2012)
3. Blanding, W., Willett, P., Bar-Shalom, Y.: Multiple target tracking using maximum likelihood probabilistic data association. In: 2007 IEEE Aerospace Conference, pp. 1–12 (2007)
4. Charrow, B., Kumar, V., Michael, N.: Approximate representations for multi-robot control policies that maximize mutual information. Autonomous Robots, 37(4), 383-400 (2015).
5. Charrow, B., Liu, S., Kumar, V., Michael, N.: Information-Theoretic Mapping Using Cauchy-Schwarz Quadratic Mutual Information. In IEEE Int. Conf. on Robotics and Automation (ICRA) (2015).
6. Clark, D., Bell, J.: Bayesian multiple target tracking in forward scan sonar images using the phd filter. IEEE Proceedings of Radar, Sonar and Navigation 152(5), 327–334 (2005). DOI 10.1049/ip-rsn:20045068
7. Cuevas, E., Echavarria, A., Ramirez-Ortegon, M.A.: An optimization algorithm inspired by the states of matter that improves the balance between exploration and exploitation. Applied Intelligence 40(2), 256–272 (2014)

8. Darema, F.: Dynamic data driven applications systems: A new paradigm for application simulations and measurements. In: Computational Science-ICCS, Volume 3038 of Lecture Notes in Computer Science, pp. 662-669 (2004)
9. Gayle, R., Moss, W., Lin, M.C., Manocha, D.: Multi-robot coordination using generalized social potential fields. In: Proc. IEEE Conf. Robotics and Automation, pp. 106–113, (2009)
10. Julian, B. J., Angermann, M., Schwager, M., Rus, D. Distributed robotic sensor networks: An information-theoretic approach. The International Journal of Robotics Research, 31(10), 1134-1154 (2012)
11. Hoffmann, G.M., Tomlin, C.: Mobile sensor network control using mutual information and particle filters. Automatic Control, IEEE Transactions on 55(1), 32–47 (2010)
12. Huang, C.C., Wang, S.J.: A Bayesian hierarchical framework for multitarget labeling and correspondence with ghost suppression over multicamera surveillance system. Automation Science and Engineering, IEEE Transactions on 9(1), 16–30 (2012)
13. Isler, V., Magdon-Ismail, M.: Sensor selection in arbitrary dimensions. Automation Science and Engineering, IEEE Transactions on 5(4), 651–660 (2008)
14. Kim, J., Kim, Y.: Moving ground target tracking in dense obstacle areas using UAVs. In: Proceedings of the 17th IFAC World Congress, pp. 8552-8557 (2008)
15. Kreucher, C.M., Kastella, K.D., Hero III, A.O.: Information-based sensor management for multitarget tracking. In: Optical Science and Technology, SPIE's 48th Annual Meeting, pp. 480–489 (2003)
16. Lennard-Jones, J.: On the determination of molecular fields. Proc. R. Soc. Lond. 106(738), 463–477 (1924)
17. Macmillan, N.A., Creelman, C.D.: Detection Theory: A User's Guide, 2nd edn. Psychology Press (2008)
18. Moin, N.: Fundamentals of Engineering Numerical Analysis, 2nd edn. Cambridge (2010)
19. nán, C.C.L.: cvBlob. http://cvblob.googlecode.com. URL http://cvblob.googlecode.com
20. Peot, M.A., Altshuler, T.W., Breiholz, A., Bueker, R.A., Fertig, K.W., Hawkins, A.T., Reddy, S.: Planning sensing actions for UAVs in urban domains. Proc. SPIE 5986, Unmanned/Unattended Sensors and Sensor Networks II, 59860J (2005)
21. Quigley, M., Conley, K., Gerkey, B., F., J., Foote, T., Leibs, J., Wheeler, R., Ng, A.: ROS: an open-source robot operating system. ICRA Workshop on Open Source Software 3(3.2) (2009)
22. Rafi, F., Khan, S., Shafiq, K., Shah, M.: Autonomous target following by unmanned aerial vehicles Proc. SPIE 6230, Unmanned Systems Technology VIII, 623010 (2006)
23. Richards, M.A.: Fundamentals of Radar Signal Processing. McGraw-Hill (2005)
24. Schwager, M., Slotine, J.J., Rus, D.: Unifying geometric, probabilistic, and potential field approaches to multi-robot coverage control. In: Robotics Research, vol. 70, pp. 21–38. Springer Berlin Heidelberg (2011)
25. Spears, W.M., Spears, D.F., Hamann, J.C., Heil, R.: Distributed, physics-based control of swarms of vehicles. Autonomous Robots 17(2-3), 137–162 (2004)
26. Stone, L.D., Barlow, C.A., Corwin, T.L.: Bayesian Multiple Target Tracking. Artech House (1999)
27. Willett, P., Ruan, Y., Streit, R.: Pmht: problems and some solutions. IEEE Transactions on Aerospace and Electronic Systems 38(3), 738–754 (2002)
28. Yang, H.C., AbouSleiman, R., Sababha, B., Gjoni, E., Korff, D., Rawashdeh, O.: Implementation of an autonomous surveillance quadrotor system. Proceedings of AIAA Infotech, AIAA 2009-2047, pp. 1–7 (2009)