Breadth-First Coupled Sensor Configuration and Path-Planning in Unknown Environments

Chase St. Laurent[†] and Raghvendra V. Cowlagi,[*] *Senior Member, IEEE*

***Abstract*— We present a breadth-first sensor configuration strategy to find near-optimal placement and sensor field of view (FoV). The strategy couples the sensor configuration procedure directly with the decision making task of planning a path for an agent in an unknown static environment comprised of threats. This coupled sensor configuration and path-planning (CSCP) strategy iteratively uses Gaussian Process Regression to construct a threat field estimate and find a candidate optimal path with minimum threat exposure. The strategy utilizes a unique task-driven information gain (TDIG) metric, which yields the sensor configurations when maximized. Due to the non-convex and non-submodular nature of the problem, we present an approximation for the optimization of the TDIG metric. Finally, we discuss the performance of the breadth-first strategy in contrast to a standard and depth-first strategy as well as traditional information-maximization.**

I. Introduction

Autonomous agents such as aerial, land, and marine robots operate under a sequential sense-then-act principle. The planning operation, or *motion planning problem*, finds an optimal sequence of control inputs or actions that drive an *agent* to a terminal state while minimizing a *cost*. The sensing operation, or *information gathering problem*, collects pertinent data in an agent-centric environment with the objective of either maximizing information or minimizing the uncertainty. However, these two problems are typically solved independently, even in domains which they coexist and have a latent interdependence. The goal of this paper is to exploit this relationship by coupling the operations and solving them together in a *task-driven* manner.

The proposed coupled sensing and planning approach can be applied to autonomous agents that have a configurable sensor network for information gathering. For example, in a military application scenario, unmanned aerial vehicles (UAVs) can be configured to survey an operating environment for hazardous threats. This application may further require the determination of a safe route through this environment, and it would then be the task of the UAVs to survey the environment in an efficient manner. Instead of surveying the entire environment, which may be time-consuming or cost-prohibitive due to the sheer size of the environment, we are interested in configuring the UAV sensor network in way that provides information of *most relevance* to path-planning.

Related Work: Path-planning – in and of itself – is well-understood to the point of being considered a “solved problem.” Notable approaches include probabilistic roadmaps, artificial potential fields, geometric techniques and cell decomposition [1], [2], wavelet decomposition, A* and its variants, fuzzy logic, and the use of genetic algorithms [3]. Commonly addressed problems in path-planning involve minimizing the path length, collision avoidance, and maximizing utilities [4].

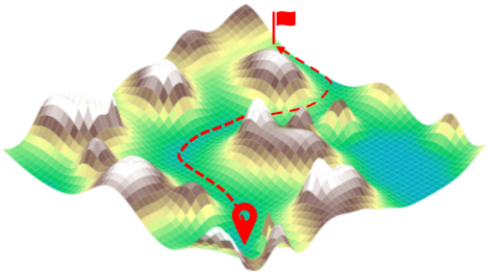


Fig. 1. Example threat field of a mountainous terrain and associated optimal path that minimizes the threat (elevation gain).

Sensor placement objectives include minimizing uncertainty and/or maximizing spatial coverage [5]. Performance metrics for sensor placement include entropy, Kullback-Liebler divergence, and mutual information [6]. Optimal sensor placement specifically for threat estimation has been studied for applications such as the dispersal of gases, specifically for volcanic ash, and with the use of distributed networks [7]–[9], but none considered a planning objective.

Optimal sensor placement using Gaussian Process (GP) models is studied in [10]. Field estimation is performed by Bayesian updates [11] of a mean function and a kernel [12]. Methods for reducing its computational burden [13]–[15] and for automatic kernel structure discovery and hyperparameter optimization [16], [17] are studied.

The concept of a *task-driven* sensor network configuration metric in contrast to an information-driven approach is defined in [18]. An interactive planning and sensing technique was developed for task-driven sensor placement [19]. This technique attempted to minimize the number of iterations to plan a path with some optimality guarantee.

Statement of Contributions: In this paper, we present a method to achieve coupling the sensor configuration and path-planning objectives. Specifically, we present a new breadth-first method for configuring sensor networks to near-optimal positions and field of view (FoV) to minimize the path-planning uncertainty. We assume that the sensor network is comprised of sensors that have a predefined mapping of observations to a notion of *threats* present within the environment. Furthermore, we consider that the threats present are unimodal, meaning that the sensory equipment

[†] Mechanical Engineering Department.

[*] Aerospace Engineering Department, Worcester Polytechnic Institute, Worcester, MA, USA. {clstlaurent, rvcowlagi}@wpi.edu.

amongst the sensor network produces a uniform and directly proportional mapping of sensor reading to threat value. Our method assumes the tradeoff between increasing sensor FoV at the expense of increased measurement noise while decreasing the FoV improves the measurement quality. Herein, we present a breadth-first approach for configuring our sensor network to greedily find a viable path plan that meets our predefined optimality standards.

Throughout this paper, we use the example of determining a path through an unknown mountainous region. The example field, the threat field, is shown in Fig. 1, along with the optimal route from our location to the goal location. In this example, we assume access to a network of UAVs with sensors for observing the environment elevation. We wish to configure these UAVs in a minimal number of iterations such that the uncertainty in the planned path cost falls below a prespecified threshold.

II. Problem Formulation

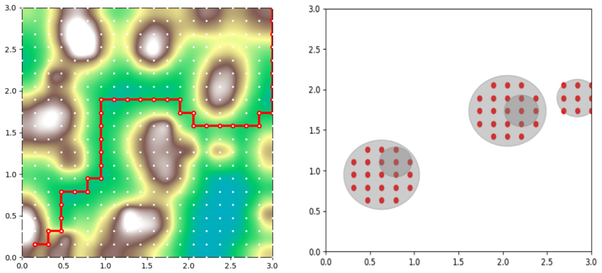
We denote by $\mathbb{R}$ and $\mathbb{N}$ the sets of real and natural numbers, respectively, and by $[N]$ the set $\{1, 2, \dots, N\}$ for any $N \in \mathbb{N}$. For any $\boldsymbol{a} \in \mathbb{R}^N$, $\boldsymbol{a}[i]$ is the i^{th} element of $\boldsymbol{a}$ and $\text{diag}(\boldsymbol{a})$ denotes the $N \times N$ diagonal matrix with the elements of $\boldsymbol{a}$ on the principal diagonal. For any matrix $A \in \mathbb{R}^{M \times N}$, $A[i, j]$ is the element in the i^{th} row and j^{th} column. $\boldsymbol{I}_{(N)}$ denotes the identity matrix of size N.

The agent operates in a prespecified closed square region called the *workspace* $\mathcal{W}$, which belongs to an *environment* $\mathcal{E}$ such that $\mathcal{W} \subset \mathcal{E} \subset \mathbb{R}^2$. Consider a uniformly-spaced grid of points labeled by integers $i = 1, 2, \dots, N_{\text{g}}$. Consider a graph $\mathcal{G} = (V, E)$ whose vertices $V = [N_{\text{g}}]$ are uniquely associated with these grid points, and whose edges E consist of pairs of geometrically adjacent grid points. In a minor abuse of notation, we label the vertices the same as grid points. We denote by $\boldsymbol{p}_i = (p_{ix}, p_{iy})$ the coordinates of the i^{th} grid point and by Δp the distance between adjacent grid points.

A *threat field* $c : \mathcal{E} \rightarrow \mathbb{R}_{>0}$ is a strictly positive temporally static scalar field. We are interested in a path-planning problem of minimizing the agent's threat exposure in $\mathcal{E}$. A *path* $\boldsymbol{\pi} = (\boldsymbol{\pi}[0], \boldsymbol{\pi}[1], \dots, \boldsymbol{\pi}[A])$ between prespecified initial and goal vertices $i_{\text{start}}, i_{\text{goal}} \in V$ is a finite sequence, without repetition, of successively adjacent vertices such that $\boldsymbol{\pi}[0] = i_{\text{start}}$ and $\boldsymbol{\pi}[A] = i_{\text{goal}}$ for some $A \in \mathbb{N}$. We define the *path incidence vector* $\boldsymbol{v}_{\boldsymbol{\pi}}$ such that $\boldsymbol{v}_{\boldsymbol{\pi}}[i] = 1$ if $i = \boldsymbol{\pi}[j]$ for $j \in [A] \backslash 0$ and $\boldsymbol{v}_{\boldsymbol{\pi}}[i] = 0$ otherwise. An example workspace and the true optimal path for the mountainous region example is depicted in Fig. 2(a).

The *cost* of this path $\boldsymbol{\pi}$ is the total threat exposure along the path: $\mathcal{J}(\boldsymbol{\pi}) := \Delta p \sum_{j=1}^{A} c(\boldsymbol{p}_{\boldsymbol{\pi}_j})$. The main problem of interest is to find a path $\boldsymbol{\pi}^*$ of minimum cost.

We cannot solve this problem as stated because the shape and intensity of the threat field is unknown. However, the threat field is observable by a network of $N_{\text{s}} \in \mathbb{N}$ sensors. Each sensor measures the threat field within in a circular and variable field of view (FoV), which is a subregion of $\mathcal{E}$. The center and radius of this circular FoV $\mathcal{S}_k \subset \mathcal{E}$, denoted $\boldsymbol{s}_k \in \mathcal{W}$ and $\varrho_k \in \mathbb{R}_{>0}$ for the k^{th} sensor, are parameters that we may choose for each $k \in [N_{\text{s}}]$. Maximum and minimum FoV radius constraints are specified as $\varrho^{\max}$ and $\varrho^{\min}$, respectively. The set of all sensor parameters is called a *configuration*, which we denote by $\boldsymbol{C} = \{\boldsymbol{s}_1, \varrho_1, \boldsymbol{s}_2, \dots, \varrho_{N_{\text{s}}}\}$.



(a)

(b)

Fig. 2. (a) 2D workspace representation and the true optimal path (red). (b) Example of sensor FoVs (some overlapping) and observations (red dots).

Consider the set $V_{\mathcal{S}_k} := \{\mathcal{S}_k \cap V\}$ of vertices with grid points within the union of FoVs of all sensors. We define the sensor *cover incidence vector* $\boldsymbol{\nu}_k$ such that $\boldsymbol{\nu}_k[i] = 1$ if $i \in V_{\mathcal{S}_k}$ and $\boldsymbol{\nu}_k[i] = 0$ otherwise. The cover incidence vector denotes the workspace vertices in which a sensor FoV intersects it. We further denote $\boldsymbol{\nu} := (\boldsymbol{\nu}_1 \vee \boldsymbol{\nu}_2 \vee ... \vee \boldsymbol{\nu}_k)$ as the sensor cover incidence of all sensors in the network.

As illustrated in Fig. 2(b), within $\mathcal{S}_k$ the k^{th} sensing agent takes $M_k \in \mathbb{N}$ pointwise and noisy measurements, equal to the magnitude of $\boldsymbol{\nu}_k$. The measurements taken are formulated as $z_{km} = c(\boldsymbol{x}_{km}) + \eta_{km}$, where $k \in [N_{\text{s}}]$ and $m \in [M_k]$. The i.i.d measurement error $\eta_{km}^{(i)} \sim \mathcal{N}(0, \sigma_k^2)$ is normally distributed with the value of σ_k^2 dependent on each sensor's FoV radius ϱ_k and noise profile. We denote by $|M|$ the total number of measurements, and by $\boldsymbol{z} = \begin{bmatrix} z_{11} & \dots & z_{1M_1} & \dots & z_{N_{\text{s}} M_{N_{\text{s}}}} \end{bmatrix}^\intercal$ the measurements made by the collection of all sensors. We use $\boldsymbol{z}$ to construct a stochastic estimate of the threat field, and then find an optimal path that minimizes the *expected* cost.

We desire the uncertainty in the estimated path cost to be low. Specifically, the variance of the path cost must be less than a certain prespecified threshold. To do so, we must collect a sufficient number of measurements by repeatedly changing the sensor configuration over multiple iterations. Conceptually, at each iteration $\ell = 0, 1, \dots, L$, the sensor configuration $\boldsymbol{C}_\ell^*$ is chosen, the threat field estimate is updated using the new measurements, and an optimal path is recomputed. The main problem of interest is now reformulated as follows:

Problem 1. *Over finite iterations $\ell = 0, 1, \dots, L$, find sensor configurations $\boldsymbol{C}_\ell$ and a path $\boldsymbol{\pi}^*$ of minimum expected cost $\overline{\mathcal{J}}^* := \mathbb{E}[\mathcal{J}(\boldsymbol{\pi}^*)]$ that satisfies $\mathbb{E}[(\mathcal{J}(\boldsymbol{\pi}^*) - \overline{\mathcal{J}}^*)^2] \leqslant \varepsilon$, for a prespecified termination threshold $\varepsilon \in \mathbb{R}_{>0}$.*

III. Coupled Sensor Configuration and Planning

The coupled sensor configuration and path-planning algorithm executes three iterative steps: (1) find optimal sensor configurations (location and FoV), (2) update the threat field estimate and error covariance, and (3) find an optimal path.

Coupled Sensor Configuration and Path-Planning

```
Let $\ell = 0$, $\boldsymbol{f}_0 = \mathbf{0}, P_0 = \chi \boldsymbol{I}$, $\boldsymbol{\mathcal{I}} = \mathbf{0}$
Solve for $\boldsymbol{\pi}_0^*$
while $\mathrm{Var}_\ell(\boldsymbol{\pi}_\ell^*) > \varepsilon$ and $(\boldsymbol{v}_{\boldsymbol{\pi}} \wedge \boldsymbol{\mathcal{I}}) \neq \mathbf{0}$ do
    Perform a Sensor Configuration Strategy
    Record and combine measurements $\boldsymbol{z}$
    Increment iteration counter $\ell := \ell + 1$
    Find GPR-based threat field estimate $\boldsymbol{f}_\ell$ and error covariance $P_\ell$
    Mask the threat field estimate $\hat{\boldsymbol{f}}_\ell := \boldsymbol{f}_\ell \odot \boldsymbol{\mathcal{I}}^{(\ell)}$
    Use Dijkstra's algorithm to find path $\boldsymbol{\pi}_\ell^*$ with minimum expected cost $\overline{\mathcal{J}}_\ell(\boldsymbol{\pi}_\ell^*)$
end while
```

Fig. 3. Pseudocode for an iterative algorithm to solve Problem 1.

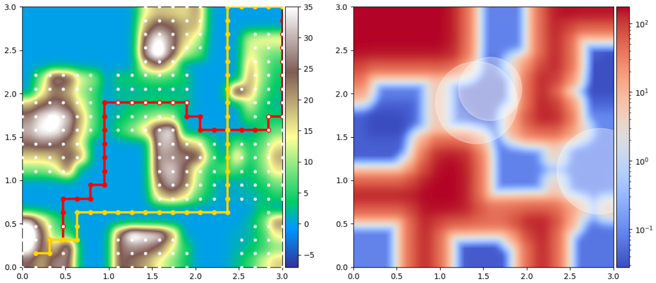


Fig. 4. Left: Estimated threat field, true optimal path (red), candidate path (yellow) of our terrain example. Right: Threat variance intensity values and sensor field of view (white circles).

At each iteration $\ell = 0, 1, \ldots, L$, the algorithm maintains at each of the N_g grid points a pointwise mean threat estimate $\boldsymbol{f}_\ell \in \mathbb{R}^{N_\mathrm{g}}$ and an estimation error covariance matrix $P_\ell \in \mathbb{R}^{N_\mathrm{g} \times N_\mathrm{g}}$. The expected cost of any path $\boldsymbol{\pi}$ is

$$\overline{\mathcal{J}}_\ell(\boldsymbol{\pi}) = \mathbb{E}[\mathcal{J}(\boldsymbol{\pi})] = \Delta p \ \boldsymbol{f}_\ell^\intercal \boldsymbol{v}_{\boldsymbol{\pi}}. \tag{1}$$

An optimal path $\boldsymbol{\pi}_\ell^*$ with minimum expected cost is computed using Dijkstra's algorithm, which has computational complexity $\mathcal{O}(|V| \log |E|)$ and is efficient for graphs with large counts of vertices and edges [20]. The path cost variance at iteration ℓ is calculated as:

$$\mathrm{Var}_\ell(\boldsymbol{\pi}) := \mathbb{E}[(\mathcal{J}(\boldsymbol{\pi}) - \overline{\mathcal{J}}_\ell(\boldsymbol{\pi}))^2] = (\Delta p)^2 \boldsymbol{v}_{\boldsymbol{\pi}}^\intercal P_\ell \boldsymbol{v}_{\boldsymbol{\pi}} \tag{2}$$

A. Algorithm Initialization

The algorithm initializes "optimistically" with $\boldsymbol{f}_0 = \mathbf{0}$. The large initial uncertainty in the threat estimate is quantified by $P = \chi \boldsymbol{I}_{(N_\mathrm{g})}$, where $\chi \gg 1$ is an arbitrary large number. Due to this "optimistic" initialization, the initial optimal path $\boldsymbol{\pi}_0^*$ is of minimum length.

We also define a set of vertices identified by $\boldsymbol{\mathcal{I}} \in [N_\mathrm{g}]$ that are covered by at least one sensor in at least one iteration. The set $\boldsymbol{\mathcal{I}}$ helps to ensure that the field estimate is not underfit in regions away from training data or when we have an insufficient amount of training data, and enables the use of a depth-first sensor configuration scheme. The importance of both these items will be detailed in a later step of the algorithm. At each iteration, $\boldsymbol{\mathcal{I}}$ is updated to reflect sensor configurations as $\boldsymbol{\mathcal{I}}^{(\ell)} := (\boldsymbol{\nu} \vee \boldsymbol{\mathcal{I}}^{(\ell-1)})$. Furthermore we define the *identified path incidence* vector as $\boldsymbol{\mathcal{I}}_{\boldsymbol{\pi}} \in \mathbb{R}^{N_\mathrm{g}}$ such that $\boldsymbol{\mathcal{I}}_{\boldsymbol{\pi}} := (\boldsymbol{\pi}_\ell \wedge \boldsymbol{\mathcal{I}}^{(\ell)})$. The *unidentified path incidence* vector is the complement $\boldsymbol{\mathcal{I}}_{\boldsymbol{\pi}}^c = 1 - \boldsymbol{\mathcal{I}}_{\boldsymbol{\pi}}$. Figure 4(a) shows an example of the estimated threat field at an intermediate iteration, with the white dots representing identified vertices.

B. Sensor Network Configuration

At each iteration $\ell = 0, 1, ..., L$, we calculate a sensor network configuration, denoted $\boldsymbol{C}_\ell = \{\boldsymbol{s}_{1\ell}, \varrho_{2\ell}, \boldsymbol{s}_{2\ell} \ldots, \varrho_{N_\mathrm{s}\ell}\}$. Figure 4(b) shows an example configuration of three sensors covering regions of the diagonal entries of the covariance matrix P_ℓ. To find optimal sensor configurations, we present a task-driven information gain (TDIG) metric. The optimal sensor configuration $\boldsymbol{C}_\ell^*$ is found by maximizing the TDIG metric, which is the reduction in variance along a *region of interest* (ROI) $\boldsymbol{\tau} \in \mathbb{R}^{N_\mathrm{g}}$. The TDIG metric is defined as:

$$h(\boldsymbol{C}_\ell, \boldsymbol{\tau}) := (\Delta p)^2 \boldsymbol{\tau}^\intercal (P_\ell - P_{\ell+1}) \boldsymbol{\tau} \tag{3}$$

Each element of $\boldsymbol{\tau}$ takes binary values in $\{0, 1\}$.

The posterior covariance $P_{\ell+1}$ cannot be directly computed, so we require an approximation $\hat{P}_{\ell+1}$. To this end, we assume that the correlation between vertices is kept fixed, and we use the proportional relationship between correlation and covariance to approximate the covariance matrix after configuring the sensor network.

First, we obtain the inverse of diagonal entries of the covariance matrix $P_\ell[i,i]^{-1}$. Next, we compute the correlation matrix as $\Omega_\ell := D_\ell P_\ell D_\ell$, where D_ℓ is a diagonal matrix with $D_\ell[i,i] = \sqrt{P_\ell[i,i]^{-1}}$. Next, define $\boldsymbol{q}$ as:

$$\boldsymbol{q} := \left(P_\ell[i,i]^{-1} + \textstyle\sum_{k=1}^{N_\mathrm{s}} \boldsymbol{\nu}_k / \sigma_k^2 \right)^{-1} \tag{4}$$

Let $\boldsymbol{r} := (\boldsymbol{\nu} \wedge \boldsymbol{\tau})$ be the sensor network reduction incidence vector. We estimate the posterior diagonal variances as $\boldsymbol{q}_{\boldsymbol{\pi}} := \boldsymbol{r}^c \odot P_\ell[i,i] + \boldsymbol{r} \odot \boldsymbol{q}$, where $\odot$ is the element-wise product. Finally, we can obtain the approximate posterior covariance matrix $\hat{P}_{\ell+1} := \mathrm{diag}(\sqrt{\boldsymbol{q}_{\boldsymbol{\pi}}}) \Omega_\ell \mathrm{diag}(\sqrt{\boldsymbol{q}_{\boldsymbol{\pi}}})$. This approximation can then be used to find a sensor network configuration $\boldsymbol{C}_\ell^*$ that maximizes the TDIG subject to the constraints $\boldsymbol{s}_k \in \mathcal{W}, \varrho^{\min} \leqslant \varrho_k \leqslant \varrho^{\max}$, for each $k \in [N_\mathrm{s}]$.

A sensor configuration strategy then amounts to choosing the ROI $\boldsymbol{\tau}$. Next, we discuss three sensor network configuration strategies including the new breadth-first approach.

1) Standard Strategy: We choose $\boldsymbol{\tau} = \boldsymbol{v}_{\boldsymbol{\pi}}$, due to which the maximization of the TDIG entails reducing uncertainty along the candidate optimal path $\boldsymbol{\pi}_\ell^*$.

2) Depth-First Strategy: The depth-first (DF) strategy is a two stage approach that utilizes the identified path vertices to dictate whether to "explore" the environment or "exploit" the candidate optimal path. The *exploration* state occurs when $|\boldsymbol{\mathcal{I}}_{\boldsymbol{\pi}}^c| \geqslant N_\mathrm{s}$, and it sets $\boldsymbol{\tau} = \boldsymbol{\mathcal{I}}_{\boldsymbol{\pi}}^c$. Sensors are placed in unexplored regions outside of the candidate optimal path. The *exploitation* state occurs whenever $|\boldsymbol{\mathcal{I}}_{\boldsymbol{\pi}}^c| < N_\mathrm{s}$, and it sets $\boldsymbol{\tau} = \boldsymbol{v}_{\boldsymbol{\pi}}$. Unlike the exploration state, we iteratively add batches of our sensor network, effectively configuring them to make multiple observations in a single iteration ℓ, until the approximated TDIG is below ε. Therefore, in a single iteration, the sensor network might take multiple observations prior to obtaining the threat field estimate.

3) Breadth-First Strategy: The breadth-first (BF) strategy samples the approximate posterior threat estimate. Using these samples, we construct an ensemble of N_a threat fields, and for each threat field we find an optimal path. Then, we define a path weight vector based on the resulting ensemble of incidence vectors of these optimal paths:

$$\boldsymbol{w}_{\boldsymbol{\pi}} := 2\boldsymbol{a}_{\boldsymbol{\pi}} - \tfrac{1}{N_a+1}\left(\boldsymbol{v}_{\boldsymbol{\pi}} + \textstyle\sum_{i=1}^{N_a}\boldsymbol{a}_i\right). \tag{5}$$

Here $\boldsymbol{a}_{\boldsymbol{\pi}}$ is the intersection of all path incidences and the estimated optimal path incidence. The optimal sensor network configuration is found by maximizing the TDIG with $\boldsymbol{\tau} = \boldsymbol{w}_{\boldsymbol{\pi}}$. This strategy enforces exploration at the "fringe" until the alternate paths converge after several iterations and the method performs similar to the standard strategy.

C. GPR-based Field Estimation

Gaussian Process regression (GPR) is a supervised machine learning technique [21] that uses function values $\boldsymbol{z}$ at training points $\boldsymbol{X}$ (sensor observations) to estimate function values at test points $\boldsymbol{X}_*$ (workspace vertices). As the sensors measure points each iteration at covered workspace vertices tracked by $\boldsymbol{\nu}$, we use a weighted update procedure as in [22] to combine multiple sensor observations at the same vertex. This leads to at most N_g training data points and N_g test points. Therefore, the computational efficiency is a function of our workspace resolution.

We perform the GPR calculations "optimistically" assuming mean $\boldsymbol{0}$ because stationary kernels exhibit $\kappa(\boldsymbol{x}, \boldsymbol{x}_*) \to 0$ as $\|\boldsymbol{x} - \boldsymbol{x}_*\| \to \infty$. This property and the initialization $\boldsymbol{f}_0 = \boldsymbol{0}$ provide optimistic field estimates in regions away from the training data, thereby avoiding suboptimal solutions. We utilize the squared exponential with automatic relevance determination (SE-ARD) kernel and from the posterior of the GPR we obtain the mean field estimate $\boldsymbol{f}_\ell$ and threat covariance matrix P_ℓ. We find two length scale hyperparameters and one data height variation height hyperparameter by maximizing the marginal log-likelihood equation.

The field estimate is further made "optimistic" by masking the current field estimate with identified vertices to create the optimistic threat field estimate $\hat{\boldsymbol{f}}_\ell := \boldsymbol{f}_\ell \odot \boldsymbol{\mathcal{I}}^{(\ell)}$. Upon obtaining $\hat{\boldsymbol{f}}_\ell$, we may find a candidate optimal path $\boldsymbol{\pi}_\ell^*$ using Dijkstra's algorithm. Figure 5(a) depicts the initial threat field estimate produced by the GPR method, while Fig. 5(b) shows the third iteration and Fig. 5(c) the final iteration.

D. Algorithm Termination

Upon finding a candidate path $\boldsymbol{\pi}_\ell^*$ we compute the path cost variance $\text{Var}_\ell(\boldsymbol{\pi}_\ell^*)$ as in (2). If $\text{Var}_\ell(\boldsymbol{\pi}_\ell^*) < \varepsilon$, then the algorithm terminates. The termination threshold $\varepsilon > 0$ is a small user-specified threshold that sets the minimum desired confidence in the estimated path cost. Decreasing the value of ε increases the desired confidence.

Proposition 1. *The CSCP algorithm terminates in a finite number of iterations $L \in \mathbb{N}$ for $N_\text{s} > 0$.*

Proof. First, we note that the diagonal entries along P_ℓ have infimum 0 and supremum χ. At each iteration $N_\text{s} > 0$ sensors are placed along ROI $\boldsymbol{\tau}$, which is a subset of the workspace, due to our optimization of the TDIG metric. Due to this and the GPR-based field estimation the diagonal entries along P_ℓ are monotonically decreasing. By our definition, a path $\boldsymbol{\pi}$ is a sequence of vertices in the workspace and thus a subset. Since the workspace vertices variance are bounded and monotonically decreasing each iteration by consequence the subset along $\boldsymbol{\pi}$ is as well. Therefore, by the monotone convergence theorem, $\inf\{\text{Var}(\boldsymbol{\pi}_L^*)\} = 0 < \varepsilon$. □

Proposition 2. *The CSCP algorithm solves Problem 1.*

Proof. We note that the path $\boldsymbol{\pi}_L^*$ satisfies the conditions stated in Problem 1, namely, that is has minimum expected cost $\overline{\mathcal{J}}^* = \overline{\mathcal{J}}_L(\boldsymbol{\pi}_L^*)$ and $\mathbb{E}[(\mathcal{J}_L - \overline{\mathcal{J}}^*)^2] = \text{Var}_L(\boldsymbol{\pi}_L^*) < \varepsilon$ per the termination criterion enforced by Line 3 in Fig. 3. □

Corollary 1. *A path $\boldsymbol{\pi}_L^*$ is near-optimal as follows:*

$$\mathbb{P}\left[|\overline{\mathcal{J}}^* - \mathcal{J}_L| \leqslant 3\sqrt{\varepsilon}\right] \geqslant 0.9973$$

Corollary 2. *The path $\boldsymbol{\pi}_L^*$ is near-optimal in the following sense. Let $\mathcal{J}^*$ denote the cost of the true optimal path. Then:*

$$\mathbb{P}\left[|\overline{\mathcal{J}}^* - \mathcal{J}^*| \leqslant 3\sqrt{\varepsilon}\right] \geqslant 0.9973$$

Proof. Due to the GPR threat field model and linearity in (1), the path cost is normally distributed, and both results follow immediately from the standard normal table. □

IV. Results and Discussion

In this section we numerically study simulated results of the CSCP algorithm, specifically the BF strategy and alternative sensor configuration strategies. We then compare them all to an information driven approach and discuss the performance improvements in terms of iterations, observations, and computation times.

A. Results of Mountainous Terrain Example

In the motivating example presented in §I, we wanted to find an optimal path through a mountainous terrain that avoided high elevation. The depictions in Fig. 6 are the end result of using the BF strategy with three sensors and a termination threshold of $\varepsilon = 1.0$. The candidate optimal path (yellow) is spatially close to the true optimal path, and the path cost variance satisfies ε. Using the CSCP algorithm with BF yielded a solution in 9 iterations. However, DF took 13 iterations to find a similar path that satisfied ε. When we increased our desired confidence in the path to be $\varepsilon = 0.01$, we observed that the BF strategy took 15 iterations while the DF strategy took only 14 iterations. Thus, the qualitative property of the BF strategy is that we can find paths in fewer iterations but with a higher termination threshold. Conversely, as we emphasize in the following numerical study, the DF strategy requires fewer

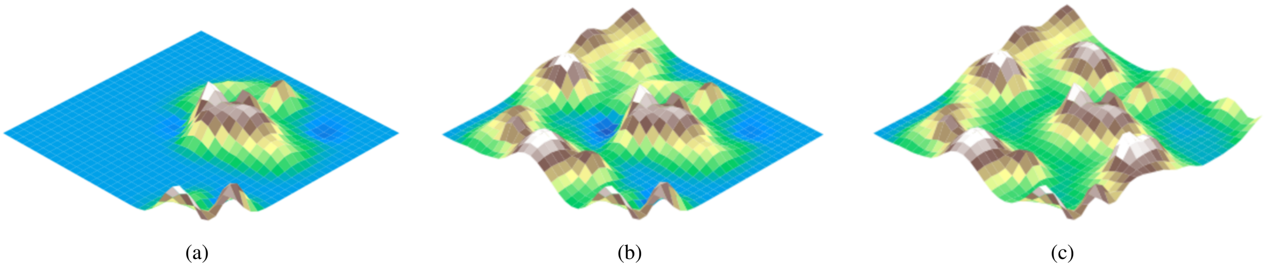


Fig. 5. (a) Threat field estimate at $\ell = 1$ (b) Threat field estimate at $\ell = 3$ (c) Final threat field estimate at $\ell = 9$.

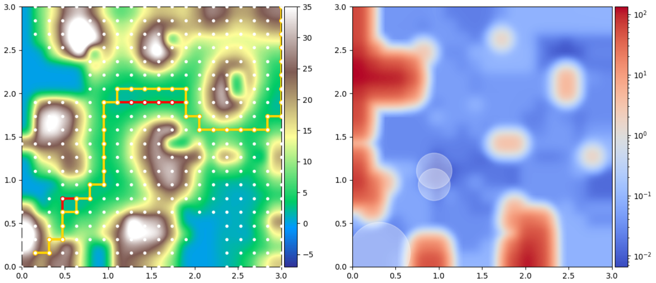


Fig. 6. Left: Final estimated threat field, true optimal path (red), candidate path (yellow). Right: Threat variance intensity values and final sensor configuration (white circles).

iterations when the termination threshold is small. The choice of strategy is therefore application dependent, but as we show next, *any* of the three task-driven strategies outperforms existing traditional approaches *by orders of magnitude.*

B. Performance Analysis on Randomly Generated Fields

To assess the performance of the CSCP algorithm with the various strategies we established a study that randomly generated, for each experiment, 100 threat fields of the form $c(\boldsymbol{x}) = \sum_{n=1}^{N_p} \theta_n \phi_n(\boldsymbol{x})$. The value N_p represents the number of radial basis functions ϕ_n that cover $\mathcal{E}$. The threat intensity θ_n is a coefficient that embodies the magnitude of each threat parameter. For these experiments we fixed the number of threat parameters to be $N_p = 50$ with $\theta_n = 10$.

We further assumed that our configurable sensor network with either $N_\mathrm{s} = 3,\ 5,$ or 9 UAV-attached sensors, with minimum and maximum altitudes simulated by assigning constraints $\varrho_{\min} = 0.01$ km and $\varrho_{\max} = 0.5$ km. Additionally, the sensor noise is modeled as $\sigma_k^2 = \frac{1}{2}\log(1 + \exp^{\pi \varrho_k^2}) - 0.1505$, which is monotonically increasing for $\varrho_k \geqslant 0$.

For the experiments we fixed the workspace resolution to have either $N_\mathrm{g} = 225$ or 400 and the environment axis size to be either 3km or 5km for a total area of $|\mathcal{E}| = 9\text{km}^2$ or $|\mathcal{E}| = 25\text{km}^2$, respectively. For the BF strategy, we varied the number alternate paths as $N_a = 25,\ 50,$ or 100. These we abbreviate as BF-25, BF-50, and BF-100, respectively. Finally, we set our termination threshold $\varepsilon = 0.01$.

1) Task-Driven vs. Information-Maximization Results: The experimental results across the varied number of sensors, grid points, and environment area are depicted in Fig. 7. Info-Max represents the baseline comparison between the task-driven approaches and the traditional information-maximization (IM) approach. From this data, we observe that the task-driven approaches increasingly outperform IM approaches as the number of available sensors decrease. IM approaches become competitive only when increasingly many sensors are available.

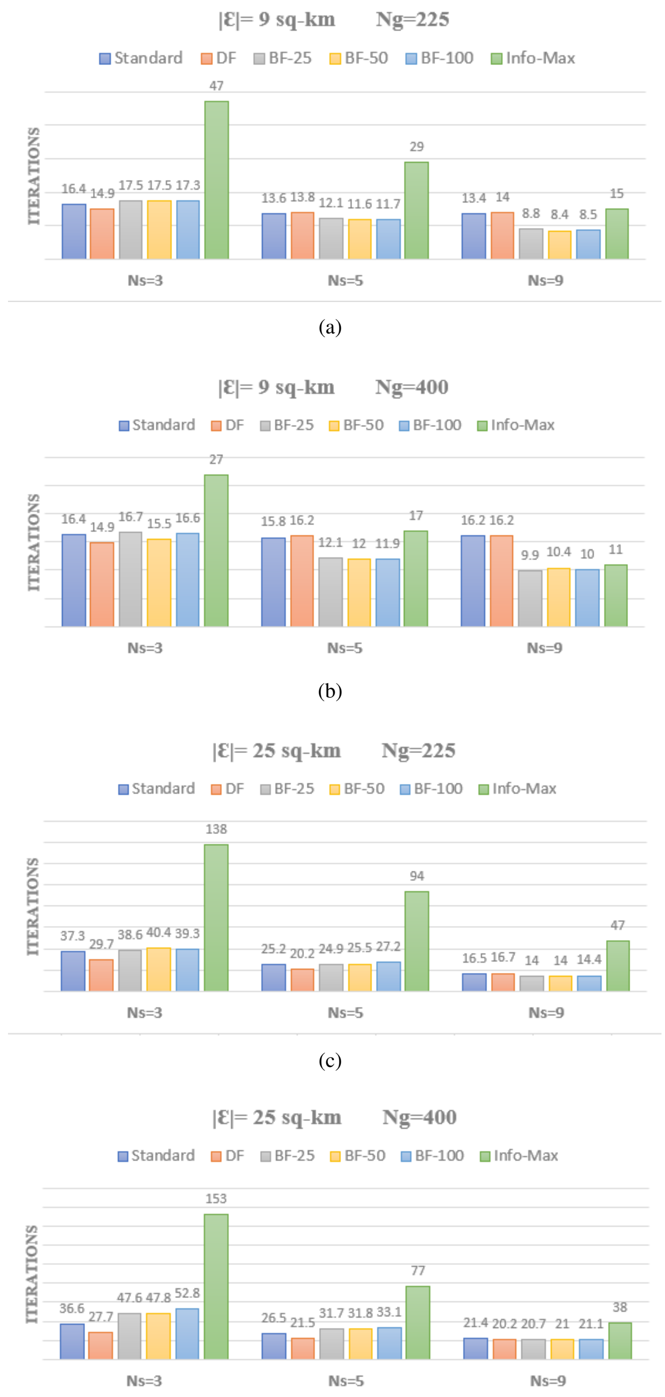
2) Breadth-First vs. Other Task-Driven Strategies: The breadth-first strategy, in comparison to the depth-first and standard task-driven strategies, is shown to minimize the number of iterations until convergence as more sensors are available. Conversely, fewer available sensors led to the breadth-first method to perform to a lesser degree than depth-first and standard task-driven approaches. This same pattern held true for all environment areas and workspace resolutions, however it was more pronounced when the environment area was lower.

From this data, we can qualitatively assess the nature in which we should select either the breadth-first, depth-first, or standard task-driven sensor configuration strategies. Firstly, the performance between the standard and depth-first varied little. Thus, we would suggest using the standard approach only when there is little information about which strategy to select. From the data, however, the main observation to be made is that the depth-first approach is better suited when fewer sensors are available while breadth-first is better suited when more are available. Likewise, when the environment area we wish to survey is relatively smaller in proportion to our maximal sensing radius $\varrho^{\max}$, the breadth-first approach is more proficient. In larger areas, it may require more and more sensors until the breadth-first approach performs better than depth-first. The workspace resolution parameter did not contain any apparent distinction in performance between the depth-first and breadth-first strategies.

3) Effect of Varying Number of Alternate Paths: Given the data, we can observe the number of alternate paths N_a played almost no role on the performance of the breadth-first strategy for CSCP. Given this, it becomes apparent that we can utilize few alternate paths and still get good performance.

V. Conclusions

In this paper we presented the breadth-first sensor configuration approach applied to the task-driven information gain metric. We detailed the coupled sensor configuration and path-planning algorithm and the various sensor configuration strategies for selecting a region of interest. We outlined the



(a)

(b)

(c)

(d)

Fig. 7. Numerical study results on the sensor configuration strategies.

procedure for finding a set of alternate paths and devising a weighted path incidence for the breadth-first approach of defining the region of interest. Ultimately, we provide a method that is capable of finding a minimum cost path in minimal iterations that satisfies our optimality standards. Numerically, we show the superiority of the task-driven sensor configuration over traditional information-maximization. Further, we provided numerical backing explaining the best suited application for the breadth-first strategy over depth-first and standard task-driven sensor configuration.

Acknowledgement: This research is funded by AFOSR grant #FA9550-17-1-0028 and NSF grant #1646367.

References

[1] S. M. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.

[2] A. S. H. H. V. Injarapu and S. K. Gawre, “A survey of autonomous mobile robot path planning approaches,” in *2017 International Conference on Recent Innovations in Signal Processing and Embedded Systems (RISE)*, Oct. 2017, pp. 624–628.

[3] T. T. Mac, C. Copot, D. T. Tran, and R. De Keyser, “Heuristic approaches in robot path planning: A survey,” *Robotics and Autonomous Systems*, vol. 86, pp. 13–28, Dec. 2016.

[4] S. Aggarwal and N. Kumar, “Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges,” *Computer Communications*, vol. 149, pp. 270–299, Jan. 2020.

[5] D. Ramsden, “Optimization Approaches To Sensor Placement Problems,” Ph.D. dissertation, Rennselaer Polytechnic Institute, 2009.

[6] D. Cochran and A. O. Hero, “Information-driven sensor planning: Navigating a statistical manifold,” in *2013 IEEE Global Conference on Signal and Information Processing*, Dec. 2013, pp. 1049–1052.

[7] M. A. Demetriou and D. Uciński, “State estimation of spatially distributed processes using mobile sensing agents,” in *Proceedings of the 2011 American Control Conference*, Jun. 2011, pp. 1770–1776.

[8] R. Madankan, S. Pouget, P. Singla, M. Bursik, J. Dehn, M. Jones, A. Patra, M. Pavolonis, E. Pitman, T. Singh, and P. Webley, “Computation of probabilistic hazard maps and source parameter estimation for volcanic ash transport and dispersion,” *Journal of Computational Physics*, vol. 271, pp. 39–59, Aug. 2014.

[9] D. Uciński, “Sensor network scheduling for identification of spatially distributed processes,” *International Journal of Applied Mathematics and Computer Science*, vol. 22, no. 1, pp. 25–40, Mar. 2012.

[10] A. Krause, C. Guestrin, A. Gupta, and J. Kleinberg, “Near-optimal sensor placements: Maximizing information while minimizing communication cost,” in *Proceedings of the 5th international conference on Information processing in sensor networks*, 2006, pp. 2–10.

[11] G. Andrew, J. Carlin, H. Stern, D. Dunson, A. Vehtari, and D. Rubin, *Bayesian Data Analysis, Third Edition*. Boca Raton, FL: Chapman & Hall/CRC, 2014.

[12] C. E. Rasmussen and K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, 2005.

[13] S. Fine and K. Scheinberg, “Efficient svm training using low-rank kernel representations,” *Journal of Machine Learning Research*, vol. 2, no. Dec, pp. 243–264, 2001.

[14] V. Tresp, “A Bayesian Committee Machine,” *Neural Computation*, vol. 12, no. 11, pp. 2719–2741, Nov. 2000.

[15] C. K. I. Williams and M. Seeger, “Using the Nyström Method to Speed Up Kernel Machines,” in *Advances in Neural Information Processing Systems 13*, T. K. Leen, T. G. Dietterich, and V. Tresp, Eds. MIT Press, 2001, pp. 682–688.

[16] D. Duvenaud, J. R. Lloyd, R. Grosse, J. B. Tenenbaum, and Z. Ghahramani, “Structure Discovery in Nonparametric Regression through Compositional Kernel Search,” p. 9.

[17] A. B. Abdessalem, N. Dervilis, D. J. Wagg, and K. Worden, “Automatic Kernel Selection for Gaussian Processes Regression with Approximate Bayesian Computation and Sequential Monte Carlo,” *Frontiers in Built Environment*, vol. 3, 2017.

[18] S. Thrun, “Learning occupancy grid maps with forward sensor models,” *Auton. Robots*, vol. 15, no. 2, pp. 111–127, 2003.

[19] B. S. Cooper and R. V. Cowlagi, “Interactive planning and sensing in unknown static environments with task-driven sensor placement,” *Automatica*, vol. 105, no. 7, pp. 391–398, July 2019.

[20] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, “Section 24.3: Dijkstra’s algorithm,” *Introduction to Algorithms*, pp. 595–601, Jan. 2001.

[21] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*, ser. Adaptive Computation and Machine Learning. Cambridge, Mass: MIT Press, 2006.

[22] H. Durrant-Whyte and T. C. Henderson, “Multisensor data fusion,” in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Cham: Springer International Publishing, 2016, pp. 867–896.