

Depth-First Coupled Sensor Configuration and Path-Planning in Unknown Static Environments

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

Abstract—We address path-planning for a mobile agent in an unknown static environment. The environment is observed by a sensor network where each sensor has a configurable location and field of view. We propose a depth-first coupled sensor configuration and path-planning (DF-CSCP) iterative method, which iteratively finds an optimal sensor configuration (location and FoV), applies Gaussian Process Regression to construct a threat field estimate, and then finds a candidate optimal path with minimum expected threat exposure. The DF-CSCP method uses a two stage procedure, (1) Explore and (2) Exploit, to drive the uncertainty of the candidate path cost variance below a prespecified threshold. To maintain tractability of GPR with increasing number of measurements, we present a sparse-update scheme. The proposed method relies on novel task-driven information gain (TDIG) metrics, the maximization of which provides sensor configurations. The TDIG metric quantifies the importance of acquiring sensor data of highest relevance to the path-planning task. Through numerical studies, we demonstrate the technical results that the DF-CSCP algorithm finds near-optimal paths with significantly fewer sensor measurements compared to traditional information-maximization methods.

I. INTRODUCTION

We study the path-planning problem of finding an optimal path for a mobile agent in an unknown static environment comprised of threats. A sensor network takes noisy observations of the threat. In contrast to standard practice, we propose a *coupled* planning and sensing method that adapts the sensor configuration to best suit the planning task.

By sensor *configuration* we refer to the location and field of view (FoV), which we treat as tunable parameters of each sensor in the network. For example, in Fig. 1(a), a unmanned aerial vehicle (UAV) carries a camera and observes the environment from a specific location and with a specified FoV. An increase in the UAV's altitude provides a larger FoV, but at the expense of the measurement quality, namely, decreased image resolution. Conversely, decreasing the FoV improves the measurement quality.

The path-planning problem involves uncertainty in the cost of a candidate path. Sensor placement strategies in the existing literature do not directly address the reduction of this uncertainty. In this paper, we adopt a depth-first strategy within the coupled planning and sensing scheme to greedily find a path that meets predefined criteria.

Related Work: Path-planning for objectives [1] such as minimizing a traversal cost, maximizing utility, or avoiding obstacles may be categorized into classical and heuristic approaches. Classical methods include Voronoi diagrams,

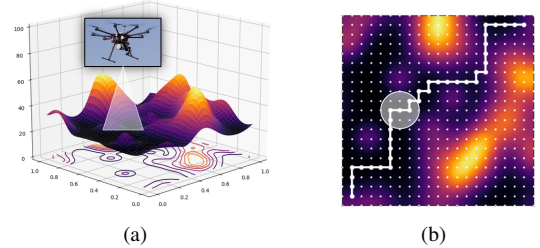


Fig. 1. (a) Threat field and sensor observation within its FoV. (b) Workspace with true optimal path (white line) and sensor FoV from (a).

artificial potential fields, and cell decomposition based on wavelet transforms, whereas heuristic approaches include A* and its variants, fuzzy logic, and genetic algorithms [2]–[6]. Probabilistic techniques for path-planning under uncertainty are addressed in the robotics literature [7], [8]. Reinforcement learning has been used for calibrating objective function weight parameters with a probabilistic policy and Boltzmann distribution for likely actions [9]. Reinforcement learning techniques based on Q-learning and SARSA have been used for global path-planning [10]. Common technical challenges include minimizing path length, ensuring completeness, robustness, and collision-avoidance [11].

The sensor placement literature is largely disjoint from path-planning. Performance metrics for sensor placement include entropy maximization, Kullback-Leibler (KL) divergence, Hellinger distance, mutual information, and the trace or determinant of the Fisher information matrix [12]. Objectives of sensor placement include uncertainty minimization [13], spatial coverage, sensor network lifetime, and communication range [14]. The pSPIEL algorithm [15] finds optimal sensor placements with constrained communication via Gaussian process regression (GPR).

GPR is a Bayesian nonparametric regression method that uses mean and kernel functions to define the structural properties of the fit, and thereby quantifies uncertainty of a model [16]–[18]. In this paper, we apply GPR for estimating the threat field from sensor measurements. The computational complexity of GPR is $\mathcal{O}(n^3)$ due to the matrix inversion, which can be reduced by methods such as reduced-rank approximations of the Gram matrix, the Nystrom method, and Bayesian committee machines [19]–[21].

The distinction between *task-driven* and information-driven performance metrics for sensor configuration are defined [22], where task-driven focuses on sensor placements of direct relevance to the planning task. Such a task-driven sensor placement method is exemplified in [23]. Although pointwise sensor measurements are assumed (i.e., FoV is not

[†]Mechanical Engineering Department.

^{*}Aerospace Engineering Department, Worcester Polytechnic Institute, Worcester, MA, USA. {cstlaurent, rvcowlagi}@wpi.edu.

considered), [23] demonstrates near-optimal path-planning under uncertainty using a small number of measurements compared to information-maximizing methods.

In this paper, we propose an iterative depth-first method for coupled sensor configuration and path-planning (CSCP). At each iteration, the method finds an optimal sensor network configuration, estimates the threat field using GPR, and then finds a candidate optimal path with minimum estimated threat exposure. The sensor configuration is coupled with path-planning such that it aims to minimize the uncertainty in the path cost estimate. The iterations terminate when the path cost uncertainty reduces below a prespecified threshold. For GPR we use an anisotropic squared exponential kernel.

Statement of Contributions: There are three novel contributions of this work. First, we present a depth-first coupled sensor configuration method. This method quickly finds a near-optimal path by iteratively configuring sensors in either an exploratory or greedy manner. This sensor configuration method finds solutions to each sensor’s location and FoV, thus addressing a “quantity versus quality” FoV trade-off. Specifically, “quantity” is achieved through large FoV (noisy measurements over large areas) and “quality” is achieved through small FoV (precise measurements over small areas).

Second, we present a procedure to reduce the computation time of the field estimation procedure by utilizing a sparse observation procedure before GPR. We define a notion of sufficiently dense and strictly dense FoV observations, and leverage the latter to limit the amount of data required during the GPR. Third, we demonstrate that, in comparison to traditional information-maximization methods, the proposed depth-first coupled sensor configuration and path-planning method reduces by *orders of magnitude* the number of measured required to find near-optimal paths.

This paper is organized as follows. We provide the problem formulation in §II, present the proposed method in §III, demonstrate numerical results and discussion in §IV, and conclude the paper in §V.

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 $\mathbf{a} \in \mathbb{R}^N$, $\mathbf{a}[i]$ is the i^{th} element of $\mathbf{a}$ and $\text{diag}(\mathbf{a})$ denotes the $N \times N$ diagonal matrix with the elements of $\mathbf{a}$ 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. $\mathbf{I}_{(N)}$ 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 $i = 1, 2, \dots, N_g$. Consider a graph $\mathcal{G} = (V, E)$ whose vertices $V = [N_g]$ 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 $\mathbf{p}_i = (p_{ix}, p_{iy})$ 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. A *path* $\pi = (\pi[0], \pi[1], \dots, \pi[A])$ 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 $\pi[0] = i_{\text{start}}$ and $\pi[A] = i_{\text{goal}}$ for some $A \in \mathbb{N}$. We define the *path incidence vector* $\mathbf{v}_\pi \in \mathbb{R}^{N_g}$ such that $\mathbf{v}_\pi[i] = 1$ if $i = \pi[j]$ for $j \in [A] \setminus 0$ and $\mathbf{v}_\pi[i] = 0$ otherwise.

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

We cannot solve this problem as stated because the threat field is unknown. The threat field can be observed by a network of $N_s \in \mathbb{N}$ sensors. Each of these sensors measures the threat in a circular 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 $\mathbf{s}_k \in \mathcal{W}$ and $\varrho_k \in \mathbb{R}_{>0}$ for the k^{th} sensor, are parameters that we may choose for each $k \in [N_s]$. 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 $\mathbf{C} = \{\mathbf{s}_1, \varrho_1, \mathbf{s}_2, \dots, \varrho_{N_s}\}$.

Consider the set $V_{\mathcal{S}} := \{\cup_{k \in [N_s]} \mathcal{S}_k \cap V\}$ of vertices with grid points within the union of FoVs of all sensors. We define the *sensor cover incidence vector* $\boldsymbol{\nu} \in \mathbb{R}^{N_g}$ such that $\boldsymbol{\nu}[i] = 1$ if $i \in V_{\mathcal{S}}$ and $\boldsymbol{\nu}[i] = 0$ otherwise.

Within $\mathcal{S}_k$ the k^{th} sensor takes $M_k \in \mathbb{N}$ pointwise and noisy measurements of the threat field. The spatial points $\mathbf{x}_{km} \in \mathcal{S}_k$, for $m = 1, 2, \dots, M_k$, where these measurements are taken are uniformly distributed within $\mathcal{S}_k$. Each of these measurements is: $z_{km} = c(\mathbf{x}_{km}) + \eta_{km}$, $k \in [N_s]$, $m \in [M_k]$. The i.i.d measurement error $\eta_{km} \sim \mathcal{N}(0, \sigma_k^2)$ is normally distributed with $\sigma_k^2 := \frac{1}{2} \log(1 + \exp^{\pi \varrho_k^2}) - 0.1505$, where $|\mathcal{E}|$ denotes area and is monotonically increasing for $\varrho_k \geq 0$. We denote sensor measurements by $\mathbf{z} = (z_{11}, \dots, z_{N_s M_{N_s}})$.

Sensor observations are used to construct a stochastic estimate of the threat field, and in turn, use this estimate to find an optimal path that minimizes the *expected* cost. Conceptually, at each iteration $\ell = 1, 2, \dots, L$, the sensor configuration $\mathbf{C}_\ell^*$ is chosen, the threat field estimate is updated using the new measurements, and an optimal path is computed. The main problem of interest is as follows.

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

III. DEPTH-FIRST CSCP

The proposed method provides an alternative methodology to the iterative CSCP algorithm of [24], which provided a solution that finds the optimal sensor configuration along the *entire* path each iteration. We refer to this CSCP method as Direct CSCP to distinguish between the proposed approach. In contrast, the approach herein details an alternative that more aptly suits situations requiring quick locally optimal solutions. Specifically, it differs from the former approach by using a two stage approach: (1) *Exploration*, which focuses

DF-CSCP Algorithm

- 1: Initialization: set $\ell := 0$, $\mathcal{I} = \emptyset$, and π_0^* per §III-A.
- 2: **while** $\text{Var}_\ell(\pi_\ell^*) > \varepsilon$ and $\pi_\ell^* \notin \mathcal{I}$ **do**
- 3: Perform SENSOR CONFIGURATION.
- 4: Record measurements z and optionally combine sparse measurements per §III-C.
- 5: Increment iteration counter $\ell := \ell + 1$.
- 6: Find GPR-based threat field estimate f_ℓ and error covariance P_ℓ .
- 7: Use Dijkstra's algorithm to find path π_ℓ^* with minimum expected cost $\bar{\mathcal{J}}_\ell(\pi_\ell^*)$.

SENSOR CONFIGURATION

- 1: **if** $|\mathcal{I}_\pi^c| \geq N_s$ **then**
- 2: Exploration-based sensor configuration per §III-B.1.
- 3: **else**
- 4: Exploitation-based sensor configuration per §III-B.2.

Fig. 2. Pseudocode for an iterative algorithm to solve Problem 1.

sensor FoV along unidentified vertices, and (2) *Exploitation*, which assumes that a fully identified path is the true optimal path, and thus configures sensor FoVs accounting for potential future observations to drive the path variance below a termination threshold.

The proposed iterative method called “Depth-First” CSCP (DF-CSCP) follows three main steps: (1) finding optimal sensor configurations (location and FoV), (2) updating the threat field estimate and error covariance matrix, and (3) finding an optimal path, as shown in Fig. 2. At each iteration, the algorithm maintains a pointwise estimated mean threat $f_\ell \in \mathbb{R}^{N_s}$ at each grid point and an estimation error covariance matrix $P_\ell \in \mathbb{R}^{N_g \times N_g}$. The expected cost of any path π is $\bar{\mathcal{J}}_\ell(\pi) = \mathbb{E}[\mathcal{J}(\pi)] = \Delta p f_\ell^\top v_\pi$. An optimal path π_ℓ^* with minimum expected cost is computed using Dijkstra's algorithm. The path cost variance at iteration ℓ is

$$\text{Var}_\ell(\pi) := \mathbb{E}[(\mathcal{J}(\pi) - \bar{\mathcal{J}}_\ell(\pi))^2] = (\Delta p)^2 v_\pi^\top P_\ell v_\pi. \quad (1)$$

The method relies on GPR for estimating the threat field. For brevity, a description of GPR is omitted; the reader interested is referred to [16]. We employ an anisotropic squared exponential automatic relevance detection (SE-ARD) kernel.

A. Algorithm Initialization

The algorithm initializes “optimistically” by setting $f_0 = \mathbf{0}$. The initial uncertainty in the threat estimate is quantified by initializing the estimation error covariance matrix with uniformly large values, e.g., $P = \chi \mathbf{I}_{(N_g)}$, where $\chi \gg 1$ is an arbitrary large number. Due to this “optimistic” initialization, the initial optimal path π_0^* is of minimum length.

We define a set of vertices identified by $\mathcal{I} \in [N_g]$ and initialize it as an empty set $\mathcal{I} = \emptyset$. At each iteration it is updated to reflect sensor FoV configuration placements as $\mathcal{I}^{(\ell)} := \{\mathcal{I}^{(\ell-1)} \cup V_S\}$. Next we define the *identified path incidence vector* as $\mathcal{I}_\pi \in \mathbb{R}^{N_s}$ such that $\mathcal{I}_\pi[i] = 1$ if $i \in \mathcal{I}_\pi[j]$ for $j \in [A] \setminus 0$ and $\mathcal{I}_\pi[i] = 0$ otherwise. In contrast, the *unidentified path incidence vector* $\mathcal{I}_\pi^c := 1 - \mathcal{I}_\pi$.

B. Sensor Configuration

At each iteration $\ell = 0, 1, \dots, L$, we pose the sensor configuration problem as two conditional and separate stages for maximizing a *task-driven information gain (TDIG)*, which is a reduction in cost variance along the path π_ℓ^* . If $|\mathcal{I}_\pi^c| \geq N_s$, then we follow the procedure in *Exploration*, else we perform the *Exploitation* stage, described as follows.

1) *Exploration Stage*: We define an *Exploratory-TDIG* as:

$$h_{\text{explore}}(C_\ell) := (\Delta p)^2 (\mathcal{I}_\pi^c)^\top (P_\ell - P_{\ell+1}) \mathcal{I}_\pi^c. \quad (2)$$

We approximate the posterior covariance by $\hat{P}_{\ell+1}$, defined as follows. First consider the correlation matrix $\Gamma_\ell := D_\ell^{-1} P_\ell D_\ell^{-1}$, where D_ℓ is a diagonal matrix with $D_\ell[i, i] = \sqrt{P_\ell[i, i]}$. We define a reduction factor

$$\delta(C_\ell; i) := P_\ell[i, i]^{-1} + \sum_{k=1}^{N_s} \sigma_k^{-2}, \text{ for each } i \in [N_g]. \quad (3)$$

We define the *unidentified reduction incidence vector* $\mathbf{u}_\pi := (\nu \& \mathcal{I}_\pi^c)$, i.e., $\mathbf{u}_\pi[i] = 1$ if and only if $\nu[i] = 1$ and $\mathcal{I}_\pi^c[i] = 1$ for each $i \in [N_g]$. We define $Q_\pi \in \mathbb{R}^{N_g \times N_g}$ as:

$$Q_\pi := \begin{cases} \delta(C_\ell; i)^{-1}, & \text{if } \mathbf{u}_\pi[i] = 1, \\ 0 & \text{otherwise,} \end{cases} \quad (4)$$

We estimate the posterior diagonal variances as $\mathbf{q}_\pi[i] = P_\ell[i, i](1 - \mathbf{u}_\pi) + Q_\pi \mathbf{u}_\pi$. The posterior covariance matrix is then approximated as $\hat{P}_{\ell+1} := \text{diag}(\sqrt{\mathbf{q}_\pi}) \Gamma_\ell \text{diag}(\sqrt{\mathbf{q}_\pi})$. The approximation to the Exploratory-TDIG is:

$$\hat{h}_{\text{explore}}(C_\ell) := (\Delta p)^2 (\mathcal{I}_\pi^c)^\top (P_\ell - \hat{P}_{\ell+1}) \mathcal{I}_\pi^c. \quad (5)$$

The sensor configuration problem is to find C_ℓ^* maximizing $\hat{h}_{\text{explore}}$ subject to $s_k \in \mathcal{W}$ and $\varrho_k^{\min} \leq \varrho_k \leq \varrho_k^{\max}$, $k \in [N_s]$.

2) *Exploitation Stage*: The exploitation stage occurs whenever π_ℓ^* has strictly less unidentified vertices than there are available sensors. However, in this stage we iteratively add another batch of sensors to the total sensor count until we find the approximated terminal iteration $\hat{L}$. The procedure is demonstrated in Fig. 3. We define a *Sequential-TDIG* as:

$$h_{\text{seq}}(C_{\hat{L}}) := (\Delta p)^2 v_\pi^\top (P_\ell - P_{\hat{L}}) v_\pi. \quad (6)$$

Similar to the process leading to (5), we redefine

$$Q_\pi := \begin{cases} \delta(C_{\hat{L}}; i)^{-1}, & \text{if } \mathbf{r}_\pi[i] = 1, \\ 0 & \text{otherwise,} \end{cases} \quad (7)$$

and $\mathbf{r}_\pi$ as the *reduction incidence vector* $\mathbf{r}_\pi := (\nu \& v_\pi)$, i.e., $\mathbf{r}_\pi[i] = 1$ if and only if $\nu[i] = 1$ and $v_\pi[i] = 1$ for each $i \in [N_g]$. The estimated posterior diagonal variances are then $\mathbf{q}_\pi[i] = P_\ell[i, i](1 - \mathbf{r}_\pi) + Q_\pi \mathbf{r}_\pi$. The posterior covariance matrix is then approximated as $\hat{P}_{\hat{L}} := \text{diag}(\sqrt{\mathbf{q}_\pi}) \Gamma_\ell \text{diag}(\sqrt{\mathbf{q}_\pi})$. The final key difference is that the optimization stage is performed iteratively, using additional measurements until the Sequential-TDIG value reduces below the termination threshold ε .

Proposition 1. *The Sequential-TDIG procedure terminates in a finite number of iterations with $N_s > 0$.*

Proof. With $N_s > 0$, at least one sensor covers a path vertex such that ν has at least one nonzero entry. Additionally,

Sequential-TDIG Procedure

- 1: Initialization: set $\hat{L} = \ell$.
- 2: **while** $h_{seq} > \varepsilon$ **do**
- 3: Increment batch counter $\hat{L} := \hat{L} + 1$.
- 4: Perform sensor configuration with $N_s := N_s(\hat{L} - \ell)$ sensors as per §III-B.2.

Fig. 3. Pseudocode for determining the approximated terminal iteration and the maximization of the Sequential-TDIG metric.

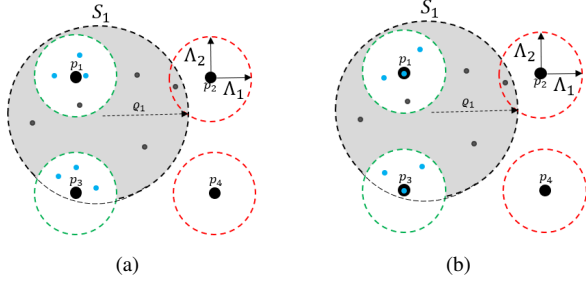


Fig. 4. M_k^α (blue dots), M_k^β (gray dots), S_k (gray region), identified vertex region (green-dotted region), unidentified vertex region (red-dotted region). (a) Sufficiently dense measurements. (b) Strictly dense measurements.

at each iteration with $\hat{h}_{seq}\varepsilon$, we add another batch of N_s sensors. We note that the path cost variance is monotonically decreasing in this case and is bounded by 0 and the prior path cost variance $\text{Var}_\ell(\pi_\ell^*)$, therefore by monotone convergence theorem we converge to $\inf\{\hat{h}_{seq}\} = 0 < \varepsilon$. $\square$

Initial guesses for optimization of the TDIG are found using a data partition technique common in unsupervised learning applications known as k -means [25]. We partition the path into N_s equally sized clusters. The farthest distance from the center of a cluster to its furthest member becomes $\varrho_{k_{\text{Guess}}}$ and the center of each cluster $s_{k_{\text{Guess}}}$. Therefore, the resulting initial configuration becomes $\mathcal{C}_{\text{Guess}} := \{s_{1,\text{Guess}}, \varrho_{1,\text{Guess}}, \dots, s_{N_s,\text{Guess}}, \varrho_{N_s,\text{Guess}}\}$.

C. Observation Sparsification

The computational complexity of GPR is $\mathcal{O}(n^3)$, where n is the number of training data points. If we keep aggregating sensor observations each iteration, the number of training data points grows quickly. To alleviate this problem, we propose a data sparsification method. First, we must define the notion of *sufficiently dense* observations.

Definition 1. The measurements M_k drawn from S_k are said to be *sufficiently dense* if $\forall i \in V_S, \exists m \in [M_k]$ s.t. $\|p_i - x_{km}\| \leq \min \Lambda$, where Λ is the characteristic length scale matrix of the SE-ARD kernel.

Informally, for an n -D ball around a vertex, there exists at least one measurement contained within that region (Fig. 7(a)). This definition may be assumed in practice since sensors such as cameras obtain compact rectangular pixel-based observations of an entire area, and there is always at least one pixel intersecting this region. Similarly:

Sparse-Update Procedure

- 1: **Input:** $M_k^\alpha, M_k^\beta \in \mathbb{N}$, given S_k .
- 2: Obtain measurements satisfying *strictly dense* assumption with prespecified counts M_k^α, M_k^β .
- 3: **for all** $V_S \cap [M_k^\alpha]$ at position x **do**
- 4: $z_x = \frac{z_{km}\sigma_x^2 + z_x\sigma_k^2}{\sigma_x^2 + \sigma_k^2}$
- 5: $\sigma_x^2 = \frac{\sigma_x^2\sigma_k^2}{\sigma_x^2 + \sigma_k^2}$
- 6: Combine measurements z and create sensor noise covariance matrix R .

Fig. 5. Pseudocode for the sparse-update procedure.

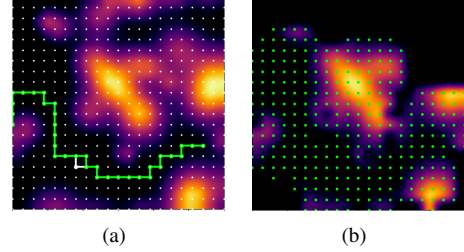


Fig. 6. Final iteration example: (a) estimated optimal path (green), true optimal (white); (b) threat field estimate and identified vertices (green).

Definition 2. The measurements M_k drawn from S_k are *strictly dense* if $\forall i \in V_S, \exists m \in [M_k]$ s.t. $\|p_i - x_{km}\| = 0$.

Using this definition, we can partition the measurements into two groups to reduce data volume. We define M_k^α and M_k^β as the number of measurements taken within the bounds of Λ , and the number of uniformly distributed measurements taken within the entirety of S_k , respectively. Thus, for $M_k^\alpha > 0$ we can satisfy that our strictly dense assumption holds.

This assumption reduces the number of measurements required by a sensor to characterize the region it measures with $M_k := M_k^\alpha \cup M_k^\beta$. This enables a GPR update for identified vertices each iteration in Fig. 5. In a slight abuse of notation, z_x and σ_x^2 refer to location specific observation and noise at location x . It enforces that the upper bound of our training data is $N_g + \ell(M_k^\alpha - |\nu| + M_k^\beta)$, since we encapsulate the measurements taken at each vertex each iteration. If we take $M_k^\alpha = 1$ and $M_k^\beta = 0$ this upper bound becomes N_g .

D. Algorithm Termination

After finding the optimal path π_ℓ^* , the path cost variance $\text{Var}_\ell(\pi_\ell^*)$ is computed per (1). If $\text{Var}_\ell(\pi_\ell^*) < \varepsilon$ and $\pi_\ell^* \subseteq \mathcal{I}$, then the algorithm terminates. Here $\varepsilon > 0$ is an arbitrarily small user-specified threshold that indicates the desired confidence in the estimated path cost.

An example near-optimal path was found in 6 iterations as shown in Fig. 6(a) in green, and is close to fully overlapping the true optimal path in white. The estimated threat field and identified vertices are shown in Fig. 6(b). Only 64.4% of the vertices had to be identified.

The properties of the CSCP algorithm are summarized in the following technical results.

Proposition 2. *The DF-CSCP algorithm terminates in a finite number of iterations $L \in \mathbb{N}$ for $N_s > 0$.*

Proof. Each iteration, two scenarios are possible: (1) we place at least one sensor at unidentified vertices along π_ℓ^* or (2) we place at least one sensor anywhere along π_ℓ^* . In either case, the path cost variance is bounded by 0 and the prior path cost variance $\text{Var}_\ell(\pi_\ell^*)$ and we note that it is monotonically decreasing. Thus, we can say by monotone convergence theorem we converge to $\inf\{\text{Var}_L(\pi_L^*)\} = 0 < \varepsilon$. By the optimization of the sensor configuration, if the path cost variance does converge, the individual elements must also converge, and consequentially a sensor configuration must cover an unidentified workspace vertex. $\square$

Proposition 3. *The DF-CSCP algorithm solves Problem 1.*

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

Corollary 1. *The path π_L^* is near-optimal in the following sense. Let $\mathcal{J}^*$ denote the cost of the true optimal path. Then:*

$$\mathbb{P}\left[|\bar{\mathcal{J}}^* - \mathcal{J}^*| \leq 3\sqrt{\varepsilon}\right] \geq 0.9973.$$

Proof. Due to GPR-based field estimation and linearity of the path cost, the path cost is normally distributed, and the result follows from the standard normal table. $\square$

IV. RESULTS AND DISCUSSION

In this section, we provide a sampling of results of numerical simulations of the DF-CSCP algorithm and its comparison to the Direct CSCP approach, an information-maximization approach, and random sensor configuration.

A. Performance Analysis on Randomly Generated Fields

We conducted a study with randomly generated threat fields in the series form $c(\mathbf{x}) = \sum_{n=1}^{N_p} \theta_n \phi_n(\mathbf{x})$, where ϕ_n are radial basis functions that cover $\mathcal{E}$. Threat intensity θ_n values were set to 100 for all basis functions N_p . The number of bases N_p is indicative of the “richness” in spatial variations in the threat field: fields with small N_p have a few peaks and several flat regions whereas fields with very large N_p have closely spaced peaks, which may cause several flat regions. To better compare the DF-CSCP to pointwise information maximization sensor placement methods, we assumed that $M_{k_\beta} = 0$ with $M_{k_\alpha} = 1$. From our results in [24], increasing workspace resolution has a near linear effect on iterations, so we choose a set $N_g = 21^2$ for this experimentation. The results also showed that the performance scales proportionally to the termination threshold, so for these experiments we consider a fixed termination threshold $\epsilon = 0.05$. To simulate the minimum and maximum heights for UAVs, we constrained the sensor FoV to $\varrho_{min} = 0.01\text{km}$ and $\varrho_{max} = 1\text{km}$. The parameters used in this experiment are shown in Table I.

Results in this paper were obtained in part using a high-performance computing system acquired through NSF MRI grant DMS-1337943 to Worcester Polytechnic Institute.

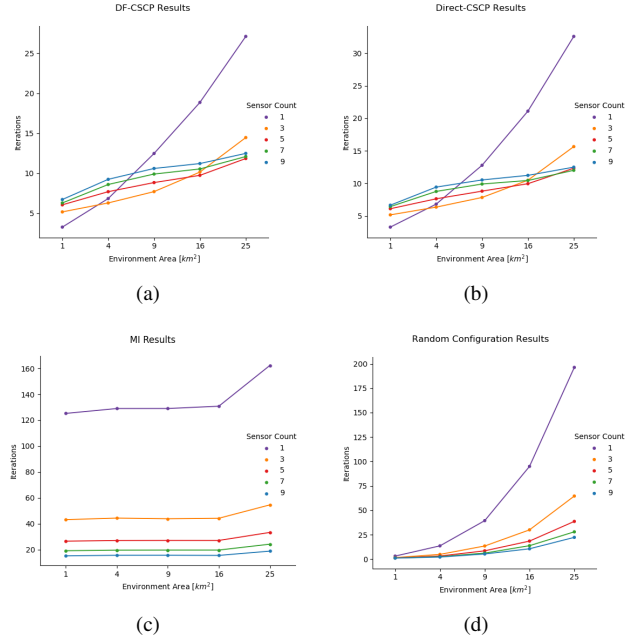


Fig. 7. Average iterations given environment area for each sensor count and for each sensor configuration method.

TABLE I
SET OF PARAMETERS USED IN NUMERICAL PERFORMANCE ANALYSIS.

Parameter	Set of simulated values
# basis functions N_p	$\{25, 50, 75, 100\}$
# sensors N_s	$\{1, 3, 5, 7, 9\}$
Environment Area $ \mathcal{E} $	$\{1\text{km}^2, 4\text{km}^2, 9\text{km}^2, 16\text{km}^2, 25\text{km}^2\}$

The average number of iterations to converge with DF-CSCP was 10.174 and it took 49.239 observations on average. The estimation error and incurred error were 0.33% and 0.061% respectively, enforcing our claim of near-optimality.

B. Comparison with Direct CSCP

We studied the performance of Direct CSCP on the same controlled environments. Notably, the DF-CSCP requires less iterations, shown in Fig. 7(a), on average in larger $|\mathcal{E}|$, but the Direct CSCP, shown in Fig. 7(b), has a slight edge in very small $|\mathcal{E}|$, given our aforementioned sensor constraints.

DF-CSCP takes less iterations than Direct CSCP on average over all experiments and significantly less runtime as shown in Table II. Both exhibit nearly identical observation count, indicating DF-CSCP sensor configuration efficiency given lower iteration count. Both methods maintain around only 76% field identification required on average.

C. Comparison to an Information-Maximization Approach

We compare the proposed approach to a sensor placement strategy based on maximization of mutual information [15]. Therein, pointwise measurements are assumed, i.e., sensor FoV is not considered. Optimization relies on submodularity, which cannot be guaranteed when considering overlapping sensor configurations as considered in this paper. To develop a fair comparison, we append the approach of [15] with

TABLE II
AVERAGE RESULTS OF THE NUMERICAL STUDY

	Iterations	Observations	Estimation Error %	Incurred Error %	Field ID %	Runtime [sec]
Direct CSCP	10.581 ± 6.280	48.874 ± 32.393	0.314 ± 0.911	0.058 ± 0.246	76.241 ± 12.666	671.147 ± 774.227
DF-CSCP	10.174 ± 5.205	49.239 ± 32.235	0.330 ± 0.995	0.061 ± 0.262	76.266 ± 12.656	446.274 ± 506.172
MI	49.161 ± 46.709	140.231 ± 28.709	0.579 ± 5.225	2.033 ± 41.33	96.725 ± 6.437	647.795 ± 669.255
Random	24.973 ± 45.303	70.868 ± 78.536	0.696 ± 1.915	0.755 ± 6.547	96.463 ± 6.093	951.652 ± 1730.718

a fixed sensor FoV of $\varrho_k = \Delta p$. The main distinction is that DF-CSCP configures sensors to maximize knowledge of the path region whereas information-maximization attempts to place sensors to maximize knowledge of the entire threat field. Results in Fig. 7(c) show that the method requires many sensors to even begin matching the performance of Direct or DF-CSCP as it does not explicitly consider sensor FoV optimization. In addition, Table II shows that the method requires 96.725% of the field to be observed, more than either CSCP approach.

D. Comparison to Random Placement

Random configuration was trialed with randomly generated locations and FoV within aforementioned constraints. We use the maximum between ϱ_{min} and the minimum radius required to guarantee intersection with a workspace vertex. Results in Fig. 7(d) show the number of iterations exponentially increase with $|\mathcal{E}|$. Table II shows that random placement takes nearly 2.5 times more iterations and significantly longer computational runtime.

V. CONCLUSIONS

We propose a depth-first strategy to coupled sensor configuration and path-planning (DF-CSCP) in unknown static environments. This method depends on a two-stage approach of “explore” and “exploit”, each having specific task-driven information gain (TDIG) metrics and approximation methods. We also provide a sparse-update scheme for reducing computational runtime. By utilizing DF-CSCP we achieve near-optimal path plans and reduce the number of required iterations and runtime. We numerically demonstrate the benefits of DF-CSCP over Direct CSCP in large environmental areas. We demonstrate its superiority over traditional information-maximization approaches and random placement.

Acknowledgement: This research is funded in part by AFOSR grant #FA9550-17-1-0028.

REFERENCES

- [1] S. Anavatti, S. Francis, and M. Garratt, “Path-planning modules for Autonomous Vehicles: Current status and challenges,” Oct. 2015, pp. 205–214.
- [2] S. M. LaValle, “Motion Planning,” *IEEE Robotics Automation Magazine*, vol. 18, no. 1, pp. 79–89, Mar. 2011.
- [3] A. S. H. H. V. Injrapu 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.
- [4] 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.
- [5] R. V. Cowlagi, “Multiresolution path-planning with traversal costs based on time-varying spatial fields,” in *53rd IEEE Conference on Decision and Control*, Dec. 2014, pp. 3745–3750.
- [6] B. Patle, G. Babu L, A. Pandey, D. Parhi, and A. Jagadeesh, “A review: On path planning strategies for navigation of mobile robot,” *Defence Technology*, vol. 15, no. 4, pp. 582–606, Aug. 2019.
- [7] M. Mohanan and A. Salgoankar, “A survey of robotic motion planning in dynamic environments,” *Robotics and Autonomous Systems*, vol. 100, pp. 171–185, Feb. 2018.
- [8] P. Mohajerin Esfahani, D. Chatterjee, and J. Lygeros, “Motion Planning for Continuous-Time Stochastic Processes: A Dynamic Programming Approach,” *IEEE Transactions on Automatic Control*, vol. 61, no. 8, pp. 2155–2170, Aug. 2016.
- [9] H. Igarashi, “Motion planning of a mobile robot as a discrete optimization problem,” in *Proceedings of the 2001 IEEE International Symposium on Assembly and Task Planning (ISATP2001). Assembly and Disassembly in the Twenty-First Century. (Cat. No.01TH8560)*, May 2001, pp. 1–6.
- [10] V. N. Sichkar, “Reinforcement Learning Algorithms in Global Path Planning for Mobile Robot,” in *2019 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM)*, Mar. 2019, pp. 1–5.
- [11] 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.
- [12] 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.
- [13] J. Ranieri, A. Chebira, and M. Vetterli, “Near-optimal sensor placement for linear inverse problems,” *IEEE Transactions on Signal Processing*, vol. 62, no. 5, pp. 1135–1146, 2014.
- [14] D. Ramsden, *Optimization approaches to sensor placement problems*. Citeseer, 2009.
- [15] A. Krause, A. Singh, and C. Guestrin, “Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies,” *Journal of Machine Learning Research (JMLR)*, vol. 9, pp. 235–284, Feb. 2008.
- [16] C. E. Rasmussen and K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, 2005.
- [17] D. J. MacKay, “Introduction to Gaussian processes,” *NATO ASI Series F Computer and Systems Sciences*, vol. 168, pp. 133–166, 1998.
- [18] C. K. Williams and C. E. Rasmussen, “Gaussian processes for regression,” in *Advances in Neural Information Processing Systems*, 1996, pp. 514–520.
- [19] S. Fine, “Efficient SVM Training Using Low-Rank Kernel Representations,” p. 22.
- [20] V. Tresp, “A Bayesian Committee Machine,” *Neural Computation*, vol. 12, no. 11, pp. 2719–2741, Nov. 2000.
- [21] 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.
- [22] S. Thrun, “Learning occupancy grid maps with forward sensor models,” *Autonomous robots*, vol. 15, no. 2, pp. 111–127, 2003.
- [23] B. S. Cooper and R. V. Cowlagi, “Interactive planning and sensing in unknown static environments with task-driven sensor placement,” *Automatica*, vol. 105, pp. 391–398, Jul. 2019.
- [24] C. St. Laurent and R. V. Cowlagi, “Coupled Sensor Configuration and Path-Planning in Unknown Static Environments [In Review],” in *American Control Conference*, New Orleans, Louisiana, USA.
- [25] J. MacQueen, “Some methods for classification and analysis of multivariate observations,” in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*. Berkeley, Calif.: University of California Press, 1967, pp. 281–297.