

Environmental Hotspot Identification in Limited Time with a UAV Equipped with a Downward-Facing Camera

Yoonchang Sung, Deeksha Dixit, and Pratap Tokekar

Abstract—Our work is motivated by environmental monitoring tasks, where finding the global maxima (*i.e.*, hotspot) of a spatially varying field is crucial. We investigate the problem of identifying the hotspot for fields that can be sensed using an Unmanned Aerial Vehicle (UAV) equipped with a downward-facing camera. The UAV has a limited time budget which it can use for learning the unknown field and identifying the hotspot. Our contribution is to show how this problem can be formulated as a novel multi-fidelity variant of the Gaussian Process (GP) multi-armed bandit problem. The novelty is two-fold: (i) unlike standard multi-armed bandit settings, the rewards of the arms are correlated with each other; and (ii) unlike standard GP regression, the measurements in our problem are images (*i.e.*, vector measurements) whose quality depends on the altitude of the UAV. We present a strategy for finding the sequence of UAV sensing locations and empirically compare it with several baselines. Experimental results using images gathered onboard a UAV are also presented and the scalability of the proposed methodology is assessed in a large-scale simulated environment in Gazebo.

I. INTRODUCTION

Robots are predominantly tasked with monitoring unknown environments but sometimes their limited sensing capabilities restrict them from observing the entire environment at once. It is thus of importance to actively explore the environment and learn the underlying characteristics of the environment. Given the limited resources (*e.g.*, operation time and fuel), the robot must carefully choose its actions to better estimate and predict states of the environment.



Fig. 1: A UAV exploring the environment to search for the plume in a lake in Blacksburg, Virginia [1].

This work is motivated by one such problem of monitoring hazardous plumes of pollutants released in water bodies, such as oil spills. The overarching project [2] is a collaboration

*This work was supported by the National Science Foundation under Grant No. 1637915. (Corresponding author: Yoonchang Sung.)

Y. Sung is with CSAIL, MIT, Cambridge, MA 02139, USA. yooncs8@mit.edu. D. Dixit and P. Tokekar are with the Department of Computer Science, University of Maryland, College Park, MD 20742, USA. {deeksha,tokekar}@umd.edu. The part of the work was completed when all authors were with the Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA 24061, USA.

with microbiologists interested in studying the transport of aerosolized pollutants from water bodies [3]. Figure 1 demonstrates our team of an Unmanned Aerial Vehicle (UAV) and Unmanned Surface Vehicle (USV) cooperatively monitoring a lake. The UAV observes regions in the lake with a downward-facing camera using which it can map out the concentration of the (visible) hazardous agent. To analyze the characteristics of toxic particulates, just mapping it with UAVs is not enough. Instead, we need physical samples that can be analyzed ex situ. The UAV can direct the USV to the location with the highest concentration. The USV can then collect the physical specimen at that location. The UAV acts as an explorer whereas the USV acts as a sampler. Our focus is on planning strategies for the UAV to find the location with the highest concentration in the limited battery life.

While the aforementioned application motivates our work, the problem we study is general enough to apply to many settings where UAVs with downward-facing cameras can be used for finding hotspots (*i.e.*, location of the global maxima) of unknown, spatially varying function. There exist many practical applications in the environmental monitoring literature, such as precision agriculture [4], wildlife habitat monitoring [5], plume tracking [1], where such a problem might arise.

Since the measurements are noisy, the accuracy of the estimate can be improved by increasing the number of measurements from a sensing location. However, spending too much time at one location is not beneficial since the UAV has a limited budget. This is the exploration-exploitation dilemma studied under the Multi-Armed Bandit (MAB) setting (we introduce several variants of MAB in Section II).

Our problem poses two major challenges that cannot be handled by existing MAB approaches. First, as the UAV can change its flight altitude (or, equivalently, the size of the camera footprint), we need different ways of evaluating the sensing performance at different altitudes. Second, there is no direct correspondence between the camera image and the measurements. Moreover, we need to define how good a particular sensing location is with respect to other locations.

Our approach is based on the Gaussian Process Upper Confidence Bound (GP-UCB) algorithm proposed by Srinivas *et al.* [6] that adopts the Gaussian Process (GP) to resolve spatially-correlated sensing locations. Our algorithm extends this to deal with the above challenges to be applicable in a 3D environment. In particular, we investigate the *multi-fidelity* aspect in GP to handle varying sensing performance at different altitudes. We then propose several heuristic planning strategies to qualitatively evaluate the proposed algorithm.

We further exploit sparse approximation studied in [7], [8] in our framework to overcome the cubic complexity of the GP regression. The sparse approximation methods are able to handle the large observation spaces induced by the image measurements.

The contributions of this paper are as follows: (1) We propose the Multi-Fidelity GP-UCB (MF-GP-UCB) algorithm for environmental monitoring in a 3D environment with a camera sensor that has noise proportional to the altitude. (2) We empirically show that the proposed strategies outperform numerous baselines through extensive simulations. (3) We demonstrate how our algorithms can be deployed on the actual hardware using real-world data and in large-scale simulated environments in Gazebo.

Our validation is based on real-world data we gathered from the field using a UAV and a physics simulation (*i.e.*, Gazebo) for large-scale experiments. We empirically show the performance of several planning algorithms through simulations. These results demonstrate the effectiveness of the Conditional Predictive Variance (CPV) that we propose in the MAB framework which reflects various flight altitudes of the UAV. In particular, we show that using the CPV resulted in approximately 10% improvement over the standard algorithms. We also find that using sparse GP techniques result in negligible performance drop but orders of magnitude lower computational times.

II. RELATED WORK

Multiple algorithms have been proposed for environmental monitoring. In this section, we briefly introduce some of the recent works. For survey results, see [9].

Although various UAV-based field estimation methods have been proposed (*e.g.*, [10]), our focus is to find global maxima or hotspot in unknown environments within a limited budget. Hotspot identification has been widely studied in the robotics community in various forms such as probabilistic classification of hotspots in a GP-based environment [11], source localization in a field of non-continuous, non-smooth gradients [12], and via a model-based approach to plume monitoring [13].

Offline algorithms compute a trajectory for the robot before the operation. Such coverage planning algorithms [14] need to know the size and shape of a given environment *a priori*. Either a predefined trajectory (*e.g.*, boustrophedon path [15]) or tree search algorithms, such as depth-first search, can be utilized to completely cover the environment.

If only partial information about the environment is given, the robot must be able to adaptively cope with unexpected situations and uncertain environments online. Many decision-making challenges arise from this context. In environmental monitoring, the robot has to decide what samples to collect if samples are available only for a specific time period [16], [17], or where to take measurements when constrained to a limited budget [1].

Information gathering is also an important task for understanding the environment. In particular, *variance reduction* [18], [19] and *mutual information maximization* [20] are

two popular information metrics. However, these approaches are not suitable for identifying hotspots in the environment as they are interested in exploring the environment rather than exploiting the current knowledge of learned information.

Bayesian optimization [21] addresses this issue by balancing between exploration and exploitation. Previous works [22] on MAB problems proposed algorithms to identify a near-optimal arm within a given budget. They also proved some guarantees on the regret, although costs of traveling between arms (*i.e.*, switching cost) were not considered. Regret is a measure of suboptimality of the reward obtained by the robot given as the difference between the optimal reward and the actual reward obtained by the robot.

There exist variants of MAB considering switching costs, which can be useful in robotics because traveling cost is a bottleneck for the budget. Reverdy *et al.* [23] employed a block allocation scheme that restrains the robot from switching to other arms frequently. Guha and Munagala [24] developed an algorithm related to the orienteering problem [25] to minimize traveling costs while maximizing collected rewards. However, their algorithm cannot handle spatially correlated arms. Audibert and Bubeck [26] studied the problem of finding the best arm at a given confidence level by the minimal number of arm pulls. Their objective, however, is to optimize the unknown budget whereas in our problem the budget is given as an input. Those works [25], [26] used a terminal regret metric for evaluating their algorithms. We use terminal regret as our evaluation metric. This is particularly suitable in our setting since the maxima located will be where the USV will obtain a physical sample from. Hence, the final output at the end of the learning process is of interest, rather than cumulative regret.

To address varying degrees of accuracy of measurements in GP-UCB, multi-fidelity GP-UCB has recently been proposed [27], [28]. Compared to these works, in our setting, we obtain multi-fidelity measurements but we also have switching costs and the underlying geometry where the Field-Of-View (FOV) increases at lower fidelity (*i.e.*, higher altitude) levels. A similar setting is studied in [29] but their objective is multi-target search.

We implemented information-gathering approaches and a MAB variant with switching costs [23] introduced here for comparison analysis in Section VI-B.

III. PROBLEM DESCRIPTION

Let $\mathcal{E} \subseteq \mathbb{R}^2$ represent the 2D environment that contains the contamination and $\mathbf{x} \in \mathcal{E}$ be a point within the environment. The intensity of the contamination varies within \mathcal{E} but we assume that the intensity does not change during the course of the robot deployment.

The goal of this paper is to find a point in the 2D environment that has the highest intensity of contamination, denoted by $\mathbf{x}^{OPT} \in \mathcal{E}$. A UAV is deployed to explore the environment, constrained by the limited time allowed to search for this particular point. The total time allocated is denoted by budget $B \in \mathbb{R}_{>0}$. The time spent is computed as the combination of *sensing time* T_S and *traveling time*

function $T_T(\mathbf{x}, \mathbf{x}')$. We assume that sensing time T_S is the same for any points \mathbf{x} regardless of the UAV position. While T_S is a fixed constant value, $T_T(\mathbf{x}, \mathbf{x}')$ is a function that takes as input two locations of the UAV and outputs the time for traveling from \mathbf{x} to \mathbf{x}' , assuming that the UAV moves with unit speed. The highest contamination point inferred by the UAV at the end of the budget B will then be visited by the USV to physically collect a sample to be analyzed.

The UAV is mounted with a downward-facing camera to observe the contamination in the environment. We denote the state of the UAV at the k -th sensing location by $\mathbf{v}(k) \in \mathbb{R}^3$. The state is composed of the x , y , and z coordinates of the UAV. At every sensing location, the UAV collects an image of the environment from a limited FOV camera sensor.

Let $f(\cdot) : \mathbb{R}^2 \rightarrow \mathbb{R}_{\geq 0}$ be the true unknown intensity function. Each image gives a noisy observation of the true intensity function. We assume that the UAV can use image processing and other estimation techniques to fuse the data obtained from all images and form an estimate of $f(\mathbf{x})$ at all locations \mathbf{x} in the environment. Based on its own estimate of the point of highest intensity, the UAV must determine a point $\mathbf{x}^{ALG} \in \mathcal{E}$ that it believes to be the global maxima of f . We wish to find strategies that will minimize the regret $f(\mathbf{x}^{OPT}) - f(\mathbf{x}^{ALG})$ at the terminal time B . Since the UAV has a limited budget B , the number of images that can be collected by the UAV is also limited. Thus, sensing locations must be carefully chosen as the information about the unknown intensity is revealed online.

Without learning how the unknown intensities are varied over the environment, the UAV cannot identify the hotspot. Thus, the UAV explores unvisited regions to gather more information while carefully spending the assigned budget. To sum up, we propose the following problem.

Problem. (*Hotspot Identification*)

Let $f(x)$ be the true intensity function and \mathbf{x}^{OPT} be the location in \mathcal{E} where $f(\cdot)$ achieves the global maxima. Given the starting position of the UAV, $\mathbf{v}(0)$, find a strategy that produces outputs: (1) a sequence of sensing locations, $\mathbf{v}(1), \mathbf{v}(2), \dots, \mathbf{v}(k)$; and (2) a point, $\mathbf{x}^{ALG} \in \mathcal{E}$, to minimize a terminal regret $f(\mathbf{x}^{OPT}) - f(\mathbf{x}^{ALG})$ subject to the constraint that $\sum_{k=1}^k (T_T(\mathbf{v}(k), \mathbf{v}(k+1)) + T_S) \leq B$.

IV. 3D MF-GP-UCB ALGORITHM

In this section, we show how to model this problem as a novel multi-fidelity variant of the MAB problem. In a typical MAB, we are given a set of arms with initial unknown reward distributions. The goal is to find the arm with the highest expected reward.

A. Arm Locations

We call sensing locations as *arm locations* since that is the standard terminology in the MAB literature. We create a 3D grid where each grid location is an arm (Figure 2 (a)). We denote the arm locations by $\mathbf{A} = \{\mathbf{a}_1, \dots, \mathbf{a}_N | \mathbf{a}_i \in \mathbb{R}^3\}$ where N is the total number of arm locations. We use i to denote the index of an arbitrary arm. We place the grid such

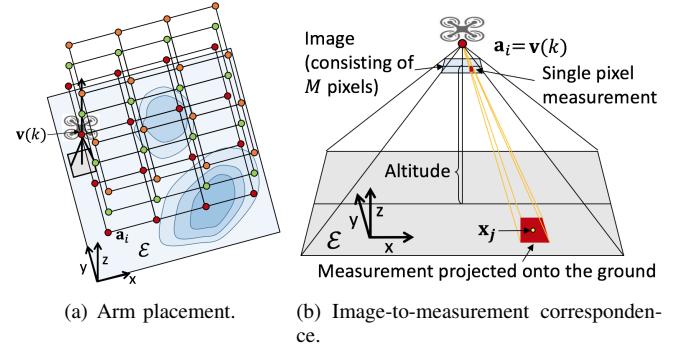


Fig. 2: Illustration of our problem setting. In (a), arms (small circles) at three different altitudes (colored red, green, and orange for increasing altitude) are placed in a given 2D environment \mathcal{E} . In (b), how the obtained image can be converted into measurements from an arm i is described.

that every point in the environment will be in the FOV of at least one arm at the lowest altitude.

B. Sensor Model

At each arm location, the UAV obtains a camera image from the downward-facing camera. The size of the camera footprint is proportional to the altitude of the UAV, *i.e.*, the camera footprint will be larger from higher altitudes.

A single image measurement yields M pixel measurements over \mathcal{E} . By using the camera projection equations with the known intrinsic, extrinsic parameters and known camera height, we can compute points in \mathcal{E} that corresponds to M measurements [30]. We will provide an example of this transformation in Section VI-C.

C. Reward

We assume that there is a function that takes an image as input and produces a noisy estimate of the true intensity function $f(\cdot)$ for every pixel. The reward is a non-negative real-valued function of the concentration of the contamination in the area covered by that pixel. In Section VI, we describe the reward function we use.

We denote the reward value of the j -th measurement by $y_j \in \mathbb{R}_{\geq 0}$. The reward value can be computed from noisy estimates of the true intensity function $f(\cdot)$:

$$y_j = f(\mathbf{x}_j) + \epsilon, \quad (1)$$

where $\epsilon \sim \mathcal{N}(0, \sigma_N^2)$ is additive i.i.d. Gaussian noise. The point \mathbf{x}_j corresponds to the j -th measurement position. Note again that larger the value of f higher the contamination.

At each sensing location, the UAV obtains M measurements where one pixel corresponds to one measurement. This measurement is the noisy version of the true intensity function of the location at the center of the footprint of the pixel on the ground (see Figure 2 (b)). Therefore, after taking k images, we have a collection of observed intensity values of kM measurements, which we denote by $\mathbf{Y} \subseteq \mathbb{R}^{kM \times 1}$. The corresponding kM measurement locations are denoted by \mathbf{X} . Note that these kM locations denoted by $\mathbf{X} \subseteq \mathbb{R}^{kM \times 2}$ are not the same as the k sensing locations (*i.e.*, arm locations) where the k images are obtained from.

Notice that the footprint of a pixel is not a point but an area, as the red quadrangle shown in Figure 2 (b). The size of the footprint of a pixel gets larger as the altitude of the UAV becomes higher. The larger footprint size would not give an accurate reward value from a particular point \mathbf{x} compared to the smaller footprint size. The measurement noise variance σ_N^2 is proportional to the altitude of the UAV. Instead of using a fixed σ_N^2 , we use $\sigma_N^2(\mathbf{a}_i)$ that can vary depending on the arm altitude. We assume that the proportion of $\sigma_N^2(\mathbf{a}_i)$ with respect to the arm altitude is linear.

D. Multi-Fidelity Gaussian Process

In this subsection, we discuss how the UAV proceeds with obtained measurements and maintains its belief of the contamination density distribution in the environment.

We use GP to represent the belief over time (*i.e.*, $f \sim \mathcal{GP}$). Unlike conventional GP, measurements we obtain have various noise levels based on which altitude the measurements were observed. This is related with the noise variance $\sigma_N^2(\mathbf{a}_i)$ at an arm i (one of hyperparameters of the GP [31]), as explained in Section IV-C. Therefore, we adapt GP to our case that can take into account various noise levels.

When we learn the hyperparameters (length-scale, signal variance, and noise variance) offline by using the log marginal likelihood, $\sigma_N^2(\mathbf{a}_i)$ is larger for data obtained from higher altitude than the one from lower altitude, *e.g.*, $\sigma_N^2(\mathbf{a}_i) > \sigma_N^2(\mathbf{a}_j)$ if the altitude of arm i is higher than that of arm j . We define the noise variance matrix given by:

$$Q(\mathbf{X}) = \begin{bmatrix} \sigma_N^2(\mathbf{a}_i) & 0 \\ 0 & \ddots \\ 0 & \sigma_N^2(\mathbf{a}_i) \end{bmatrix}, \quad (2)$$

which is a diagonal matrix. Each diagonal element of $Q(\mathbf{X})$, $\sigma_N^2(\mathbf{a}_i)$, has a unique value based on the altitude of the i -th arm, allowing diagonal elements of $Q(\mathbf{X})$ to have various values, but we set $\sigma_N^2(\mathbf{a}_i)$ for arms at the same altitude to have the same value. The dimension of $Q(\mathbf{X})$ is proportional to the total number of measurements accumulated during the flight, *i.e.*, $Q(\mathbf{X}) \subseteq \mathbb{R}^{kM \times kM}$ at sensing location k .

Given a 2D grid of the environment, let L_i be the number of grid cells that fall in the camera footprint of arm i which we call test points. Let I_i be the index set containing indices of test points observed from an arm i . We denote the set of L_i test points by $X_i^* = \{\mathbf{x}_{I_i(1)}^*, \dots, \mathbf{x}_{I_i(L_i)}^*\} \subset \mathcal{E}$. Alternatively, $X_i^* = \mathbf{x}_{I_i}^*$ for an arm i . As we have N arms, the total number of all test points from all arms becomes $L = \sum_{i=1}^N L_i$. The set of all test points from all arms is $X^* = \cup_{i=1}^N X_i^* = \{\mathbf{x}_1^*, \dots, \mathbf{x}_L^* | \mathbf{x}_l^* \in \mathcal{E}\}$. The prior of the GP then becomes:

$$\begin{bmatrix} \mathbf{Y} \\ f_* \end{bmatrix} \sim \mathcal{N} \left(\mathbf{0}, \begin{bmatrix} \mathcal{K}(\mathbf{X}, \mathbf{X}) + Q(\mathbf{X}) & \mathcal{K}(\mathbf{X}, \mathbf{X}^*) \\ \mathcal{K}(\mathbf{X}^*, \mathbf{X}) & \mathcal{K}(\mathbf{X}^*, \mathbf{X}^*) \end{bmatrix} \right), \quad (3)$$

where $\mathcal{K}(\cdot, \cdot)$ is a covariance function (or kernel). In this work, we use the squared exponential covariance function [31]. The predictive mean μ and covariance P are:

$$\mu = \mathcal{K}(\mathbf{X}^*, \mathbf{X}) (\mathcal{K}(\mathbf{X}, \mathbf{X}) + Q(\mathbf{X}))^{-1} \mathbf{Y}, \quad (4)$$

$$P = \mathcal{K}(\mathbf{X}^*, \mathbf{X}^*) - \mathcal{K}(\mathbf{X}^*, \mathbf{X}) (\mathcal{K}(\mathbf{X}, \mathbf{X}) + Q(\mathbf{X}))^{-1} \mathcal{K}(\mathbf{X}, \mathbf{X}^*). \quad (5)$$

The inversion of a high-dimensional matrix in Equations (4) and (5) makes the algorithm infeasible to run in real-time. This is a well-known difficulty in using GP and in our case the size of the inversion matrix scales as $\mathcal{O}(k^3 M^3)$. To overcome the computational bottleneck, various techniques have been proposed to develop a sparse GP approximation [7], [8]. These approaches generate a small set of S inducing (or support) points to approximate the original GP such that $S \ll kM$, eventually reducing the time complexity to $\mathcal{O}(SkM)$. We empirically show the power of linear complexity owing to the sparse GP in Section VI. By choosing a small enough S , one can adapt the algorithm for real-time applications. Specifically, we show an order of magnitude reduction in computational time without affecting the performance.

E. Pseudo-code

The pseudo-code in Algorithm 1 demonstrates all the steps of hotspot identification. The UAV starting from an initial position $\mathbf{v}(0) = \mathbf{a}_i(0)$ is given a budget B and initializes GP with zero mean. At every sensing location k , the UAV updates the GP mean and variance functions (Lines 10 and 13) by using measurements collected up to and including the previous sensing locations (Line 20). To compute the objective function (Line 17), we calculate the average mean and the average variance (Lines 14 and 15) as the number of test points is not 1. If the budget B assigned to the UAV is exhausted (Line 2), the algorithm terminates and the UAV finds \mathbf{x}^{ALG} from learned GP for the USV to sample from this location.

V. PLANNING STRATEGIES

We present several heuristic planning strategies that can be used as a subroutine in Algorithm 1. We explain the MF-GP-UCB algorithm first and then present the two changes we make to the algorithm for better accuracy, *i.e.*, *CPV* and *dynamic window approach*.

We present the objective function (Line 17 of Algorithm 1) that decides which arm the UAV should visit next. The weighted combination of mean and variance is the conventional functional form in MAB, however, β values proposed by the UCB algorithms as well as the GP-UCB algorithm [6] cannot directly be applied to our problem because of the challenges explained in Section I. Thus, we propose the following exponential functional form for β :

$$\beta = \gamma e^{(\lambda k)}, \quad (6)$$

where γ and λ are hyperparameters that control the decreasing rate or the increasing rate of β . We tune them as well as the inherent GP hyperparameters offline using data collected in simulations.

GP-UCB we use is originated from Srinivas *et al.* [6] but extended to take into account our problem setting (*i.e.*, MF-GP-UCB). Moreover, with Equations (4) and (5), MF-GP-UCB can handle various noise levels for collected measurements. Algorithm 1 without having Line 12 is this MF-GP-UCB version. Taking out Line 12 implies that the variance of test points is estimated without worrying about where the current measurements are observed. Since the flight altitude of the UAV affects the sensing credibility, we tackle this in the following strategy.

Algorithm 1: 3D MF-GP-UCB with CPV

```

Input : Initial position of the UAV:  

 $\mathbf{v}(0) = \mathbf{a}_i(0) \in A$ , GP prior:  $\mu(0) = 0$ ,  $\sigma(0)$ ,  

 $\mathcal{K}$ , budget:  $B$ , arm set:  

 $A = \{\mathbf{a}_1, \dots, \mathbf{a}_i, \dots, \mathbf{a}_N\}$ , test set:  

 $X^* = \{\mathbf{x}_1^*, \dots, \mathbf{x}_l^*, \dots, \mathbf{x}_L^*\}$ , measurement set:  

 $X = \emptyset$ , reward value set:  $Y = \emptyset$ .  

1 for  $k = 1, 2, \dots$  do  

2   if  $T_T(\mathbf{v}(k), \mathbf{v}(k-1)) + T_S k > B$  then  

3      $\mathbf{x}^{ALG} = \arg \max_{\mathbf{x}} \mu$ .  

4     TerminateExploration( $\mathbf{x}^{ALG}$ ).  

5   end  

6   if  $k = 1$  then  

7     | Choose a nearest arm  $i$ , i.e.,  $\mathbf{a}_i(1) = \mathbf{v}(1)$ .  

8   end  

9   else  

10     $\mu = \mathcal{K}(X^*, X)(\mathcal{K}(X, X) + Q(X))^{-1}Y$ .  

11    for  $i = 1, 2, \dots, N$  do  

12       $X' = X \cup \mathbf{x}_{I_i}^*$ .  

13       $P|\mathbf{a}_i =$   

14       $\mathcal{K}(\mathbf{x}_{I_i}^*, \mathbf{x}_{I_i}^*) - \mathcal{K}(\mathbf{x}_{I_i}^*, X')(\mathcal{K}(X', X') +$   

15       $Q(X')^{-1}\mathcal{K}(X', \mathbf{x}_{I_i}^*)$  where  

16       $P \ni \sigma_l^2 \forall l \in I_i$ .  

17       $\bar{\mu}_i = (\sum_{l \in I_i} \mu_l)/L_i$ .  

18       $\bar{\sigma}_i^2 = (\sum_{l \in I_i} \sigma_l^2)/L_i^2$ .  

19    end  

20     $\mathbf{a}_i(k) = \arg \max_i \{\bar{\mu}_i + \beta \bar{\sigma}_i\}$ .  

21  end  

22   $\mathbf{a}_i(k) \leftarrow \text{MoveUAV}(\mathbf{v}(k-1))$ .  

23   $\{X, Y\} \leftarrow \{X, Y\} \cup \text{GetMeasurements}(\mathbf{a}_i(k))$ .  

24 end

```

Although the predictive covariance P in Equation (5) considers various noise levels for the training points (*i.e.*, accumulated measurements), this aspect is not addressed for predicting the covariance at test points in Equation (5). That is, we predict the covariance at a test point as if we were to observe the test point from the altitude of the corresponding arm. We achieve this by adopting CPV, *i.e.*, $P|A$.

We decompose the CPV $P|A$ into N CPVs, *i.e.*, $P|\mathbf{a}_i$ for each arm i . Then, the test points and the training points become $\mathbf{x}_{I_i}^*$ and $X' = X \cup \mathbf{x}_{I_i}^*$, respectively, for an arm i . The

decomposed CPV for the i -th arm can be computed as:

$$P|\mathbf{a}_i = \mathcal{K}(\mathbf{x}_{I_i}^*, \mathbf{x}_{I_i}^*) - \mathcal{K}(\mathbf{x}_{I_i}^*, X') \times \\ (\mathcal{K}(X', X') + Q(X'))^{-1} \mathcal{K}(X', \mathbf{x}_{I_i}^*). \quad (7)$$

We call this planning strategy as MF-GP-UCB with CPV. Similarly, we call the standard MF-GP-UCB as MF-GP-UCB with current variance.

The above strategies are not concerned with minimizing the total traveling cost of the UAV with respect to the traveling time function $T_T(\mathbf{x}, \mathbf{x}')$ as the algorithm greedily chooses the best arm at each time. If an algorithm keeps moving the UAV from one end of the environment to the other end, the UAV cannot gather much information and may output a low intensity point due to the limited budget B .

We employ the dynamic window approach as a heuristic where the 3D window is defined centered on the current position of the UAV. The UAV is only allowed to visit neighboring arms to observe at each sensing location. To do that in Algorithm 1, the UAV considers neighboring arms (*i.e.*, a subset of A) with respect to the current UAV position $\mathbf{v}(k)$ in Lines 10-17 to decide which arm to visit next.

VI. SIMULATIONS

We implemented Monte Carlo simulations using MATLAB to verify the performance of Algorithm 1. We randomly generated 40 environments (20×20 square meters, having different distributions of multiple local maxima but the same global maximum intensity value, *i.e.*, 50, so as to have the same $f(\mathbf{x}^{OPT})$ for all environments) and ran 10 instances for each environment. We set three altitudes for the UAV to fly at 10, 40, and 70 meters. Each image consists of 9 pixels and the camera footprint size is $1 \times 1 m^2$ at the lowest altitude, $4 \times 4 m^2$ at the middle altitude, and $7 \times 7 m^2$ at the highest altitude. We considered the total budget of 100 seconds in all cases because this amount prevents the UAV from visiting all arms, otherwise resulting in a situation where careful decision-making is not required. Taking an image requires a total of 2 seconds, *i.e.*, $T_S = 2$ seconds. This time can be used to make a stop to take images without blur as well as to plan for subsequent actions.

We conducted an ablation study on planning strategies in Section V and comparison analysis with baseline algorithms. We then compared 3D planning with 2D planning where the UAV was allowed to fly at a fixed altitude. We also tested how the performance of algorithms varies depending on the amount of budget. Lastly, we studied the effect of sparsity to alleviate the computational complexity of the GP regression.

We define two terminal regrets as performance metrics: the *point performance metric* to measure how close $f(\mathbf{x}^{ALG})$ is to $f(\mathbf{x}^{OPT})$ by $\frac{f(\mathbf{x}^{ALG})}{f(\mathbf{x}^{OPT})} \times 100\%$, and the *arm performance metric* to measure how close $\sum_{\mathbf{a}_i^{ALG}} f(\mathbf{x})$ is to $\sum_{\mathbf{a}_i^{OPT}} f(\mathbf{x})$ by $\frac{\sum_{\mathbf{a}_i^{ALG}} f(\mathbf{x})}{\sum_{\mathbf{a}_i^{OPT}} f(\mathbf{x})} \times 100\%$. The point performance metric is relevant when a USV will go and collect a physical sample at \mathbf{x}^{OPT} . The arm performance metric is relevant when, instead of a USV, a UAV will go and collect a final image measurement at the best arm location.

A. Ablation Study

We evaluated eight heuristic planning algorithms where we consider the decreasing and increasing rates of β as well as with and without the dynamic window approach. From randomly generated environments, we tuned the hyperparameters (γ and λ) of β (Equation (6)) offline through simulations for both the increasing rate and the decreasing rate. We used the following exponential functional forms:

- $\beta = 1.5e^{-0.05k}$ for MF-GP-UCB with current variance of decreasing β .
- $\beta = 10e^{-0.05k}$ for MF-GP-UCB with CPV of decreasing β .
- $\beta = -0.5e^{-0.05k} + 0.5$ for MF-GP-UCB with current variance of increasing β .
- $\beta = -10e^{-0.05k} + 10$ for MF-GP-UCB with CPV of increasing β .

From Table I, we observe that taking into account the CPV consistently results in approximately 10% improvement. This can be observed by comparing CV with CPV and DCV with DCPV. We notice similar improvements in all versions of the algorithm. Also, with the dynamic window approach, we observe approximately 25% improvement on average.

Algorithms	Performance metrics (%)	
	Point	Arm
MF-GP-UCB with CV ($\beta--$)	47.65 \pm 4.52	46.31 \pm 4.51
MF-GP-UCB with CPV ($\beta--$)	57.40 \pm 4.83	57.49 \pm 5.17
MF-GP-UCB with CV ($\beta++$)	49.39 \pm 4.81	50.62 \pm 4.83
MF-GP-UCB with CPV ($\beta++$)	59.46 \pm 5.07	58.49 \pm 5.14
MF-GP-UCB with DCV ($\beta--$)	62.94 \pm 4.89	63.53 \pm 4.82
MF-GP-UCB with DCPV ($\beta--$)	71.28 \pm 4.82	71.96 \pm 4.59
MF-GP-UCB with DCV ($\beta++$)	61.49 \pm 4.72	62.19 \pm 4.59
MF-GP-UCB with DCPV ($\beta++$)	71.60 \pm 4.73	74.51 \pm 4.52

TABLE I: Ablation study on eight heuristic planning strategies. The values represent the percents of how close the best-estimated value is to the true best value with the one-sigma error. Acronyms in this table are: CV (Current Variance), DCV (Dynamic window approach Current Variance), DCPV (Dynamic window approach CPV). $\beta++$ and $\beta--$ imply β with the increasing rate and with the decreasing rate, respectively.

B. Comparison Analysis

1) *Comparison with baseline algorithms:* We introduce four baseline algorithms that can also be applied to our problem: (1) the boustrophedon algorithm [15], (2) gradient ascent method, (3) information-theoretic approach [32], and (4) the block Upper Credible Limit (UCL) algorithm [23]. The boustrophedon algorithm is coverage planning where the entire environment is completely covered by the camera footprint of the UAV but information gathered online is not exploited. The gradient ascent method computes the gradient of the intensity function with respect to the current UAV position and the UAV moves in the direction of the positive gradient. We implemented two variations of the information-theoretic approach: variance reduction and mutual information maximization, introduced in Section II. The block UCL algorithm is a Bayesian approach that addresses the exploration-exploitation dilemma. The unique feature of this algorithm is that it not only maximizes the expected reward

Algorithms	Performance metrics (%)	
	Point	Arm
MF-GP-UCB with DCPV ($\beta++$)	71.60 \pm 4.73	74.51 \pm 4.52
Boustrophedon at high altitude	35.86 \pm 3.16	3.40 \pm 0.00
Boustrophedon at middle altitude	34.81 \pm 2.74	3.40 \pm 0.00
Boustrophedon at low altitude	28.73 \pm 2.83	3.40 \pm 0.00
Gradient ascent at high altitude	14.61 \pm 7.55	10.11 \pm 5.87
Gradient ascent at middle altitude	14.03 \pm 10.52	11.06 \pm 7.60
Variance reduction	53.70 \pm 1.34	52.49 \pm 1.31
Gradient ascent at low altitude	13.69 \pm 9.56	11.84 \pm 7.77
Mutual information maximization	52.92 \pm 1.32	48.60 \pm 1.22
Block UCL	35.13 \pm 1.84	11.91 \pm 0.45
MF-GP-UCB with D2DH ($\beta--$)	65.81 \pm 4.81	69.58 \pm 4.52
MF-GP-UCB with D2DH ($\beta++$)	66.93 \pm 4.70	72.12 \pm 4.50
MF-GP-UCB with D2DM ($\beta--$)	63.89 \pm 5.00	69.08 \pm 3.93
MF-GP-UCB with D2DM ($\beta++$)	69.23 \pm 4.71	71.22 \pm 4.12
MF-GP-UCB with D2DL ($\beta--$)	44.48 \pm 4.67	45.88 \pm 4.60
MF-GP-UCB with D2DL ($\beta++$)	47.56 \pm 5.00	48.31 \pm 4.91

TABLE II: Comparison with eight heuristic planning strategies, four baseline algorithms, and 2D exploration algorithms. We leave our algorithm with the best performance here as a comparison standard. Acronyms in this table are: D2DH (Dynamic window approach 2D exploration at the Highest altitude), D2DM (Dynamic window approach 2D exploration at the Middle altitude), and D2DL (Dynamic window approach 2D exploration at the Lowest altitude).

obtained within a budget, but it also minimizes the number of switches to other arms, which is related to minimizing the traveling cost.

We observe from Table II that the proposed algorithm outperforms baseline algorithms in all cases. For the case of boustrophedon algorithms, the budget of 100 was not sufficient to cover the entire environment and consequently ended up with poor performance. The gradient ascent methods at all altitudes performed the worst by getting trapped in local optima. Both variations of information methods outperformed other baselines, but not ours, as they explored the entire environment sufficiently while not concentrating on high-intensity regions. An environment having a peak on its boundary might be a bad case for information methods. Due to the allocation scheme used by the block UCL, the algorithm requires sensing at an arm numerous times before moving to the next arm. This results in the exploration of only a small portion of the environment.

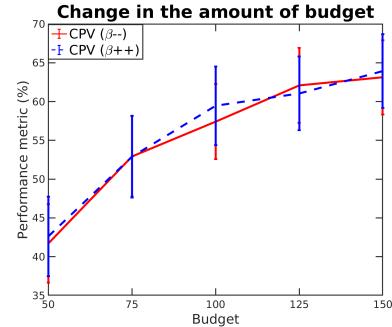


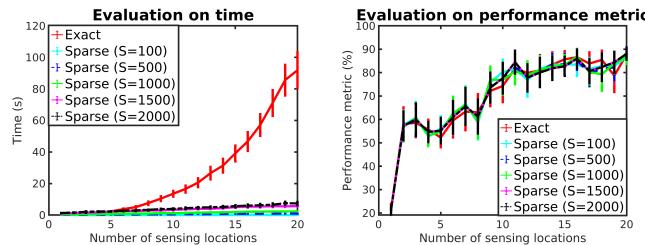
Fig. 3: Relationship between the budget and the performance metric for a point in case of MF-GP-UCB with CPV.

2) *Comparison with 2D planning:* We compared with 2D planning algorithms where we fixed the altitude at which the UAV can fly. It can be seen from Table II that flying at a fixed altitude is not beneficial in comparison with 3D exploration.

3) *Effect of the amount of budget*: We tested how the performance metric for a point varies with respect to the change in the amount of budget given to the UAV. As shown in Figure 3, the performance metric increases logarithmically as the budget increases.

4) *Sparse GP evaluation*: We compared the sparse GP approximation with the original GP to analyze how sparsity affects the performance of the proposed algorithm. We generated ten different environments to run algorithms 10 times in each environment. At each sensing location, the robot received 646 measurements. Plots in Figure 4 were obtained by computing the mean and standard deviation from all runs for each sensing location. We used different numbers of inducing points (*i.e.*, S) for various levels of sparsity.

As shown in Figure 4 (a), the time taken by the original GP increases exponentially due to the exact inference while the times taken by the sparse GPs grow linearly. However, we observe in Figure 4 (b) that there is little difference in performance metric values among the original GP and sparse approximations. From our analysis, the sparse GP yields orders of magnitude lower time than the original GP without any sacrifice of performance. Thus, we must take advantage of sparsity for real-time applications while introducing S as an additional hyperparameter.



(a) The result of efficiency analysis. (b) The result of performance analysis.

Fig. 4: Comparison of time taken by the exact inference (*i.e.*, the original GP) and the performance metric values with sparse approximations that have different numbers of inducing points S .

C. Real-World Data

In our experimental setup, we used the UAV (Figure 5) with a single onboard-PC which has Linux 16.04 and ROS Kinetic [33] installed. It is equipped with a GPS sensor, a compass, and a downward-facing camera sensor (GoPro Hero4), which can communicate with the UAV over WiFi. The GPML MATLAB toolbox [31] is used to perform GP regression along with the ROS Toolbox package.

We used three tarps of blue, gray, and red colors as a proxy of regions with different intensities of toxicants. We collected our real-world data by flying the UAV at 3 different altitudes of 10, 20, and 30 meters in a boustrophedon pattern (3.25×4.73 , 6.51×9.45 , 9.76×14.12 square meters, respectively, in the camera footprint size) in the environment of $30 \times 30 \times 30$ cubic meters. We gathered 346 images spanned over these altitudes. Figure 5 shows the ground truth. For each image in our dataset, we assigned the intensity value $f(\mathbf{x})$ for each

color on a per-pixel basis. We have three discrete intensity values of 3, 2, and 1 associated with HSV color ranges of the red, gray, and blue tarps, respectively. The left image in Figure 5 shows the visualization of the intensity values in Universal Transverse Mercator (UTM) coordinates. Darker regions represent a higher intensity value.

In the simulation, the original resolution of the image 240×432 was reduced to 19×34 resulting in 646 pixel measurements. Note that down-sampling is not required for GP computation, owing to the use of sparse GP, but required for fast data transmission. The camera footprint at an arm depends on the yaw of the UAV as well as the UAV position \mathbf{v} . To use the intensity matrix with MF-GP-UCB, we do the mapping from image coordinates to real-world coordinates (*i.e.*, UTM coordinates). To estimate UTM coordinates for each pixel, we make three main assumptions: (1) The principal point of the camera aligns with the image center. (2) Consequently, the UTM northing and easting of the UAV can be projected directly onto the image center. (3) Roll and pitch errors of the UAV are negligible. To estimate world coordinates of each pixel position, we rotate the frame of reference to align it with the UTM coordinate axis. Then, we apply the Pinhole camera model [30] using the intrinsic parameters of the camera sensor obtained from camera calibration.

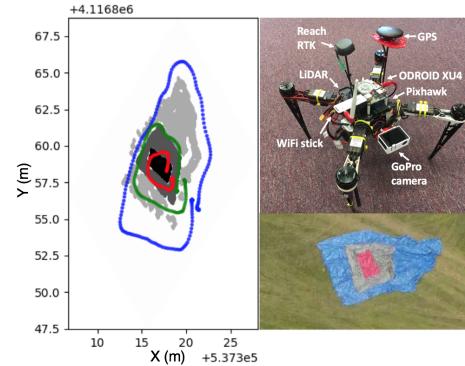


Fig. 5: Image on the top right shows the UAV platform. The bottom right image shows the tarps obtained by the UAV and used as a proxy of the regions with different intensities (red:3, gray:2, blue: 1). The image on the left shows the intensity for the image on the bottom right along with the ground truth boundaries.

Figure 6 shows the terminal states of GP by applying the proposed algorithm to real-world data. \mathbf{x}^{OPT} is $(537318m, 4116861m)$ and \mathbf{x}^{ALG} is $(537316m, 4116859m)$ in UTM coordinates. The point performance metric we obtained is 100% as both $f(\mathbf{x}^{OPT})$ and $f(\mathbf{x}^{ALG})$ are 3 while the arm performance metric is 66.7%.

D. Large-Scale Simulated Experiments

To evaluate the scalability of our algorithm in a large-scale environment, we used the Gazebo simulator (Figure 7). We used the hector_quadrotor [34] package in Gazebo to simulate the UAV model. To emulate a real-world scenario, an image of oil spill in San Francisco bay [35] is used as the ground plane in Gazebo to model the plume spread over

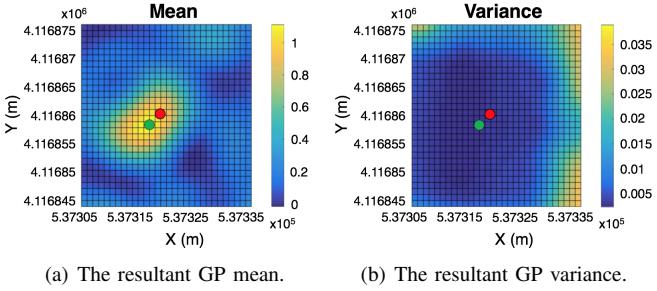


Fig. 6: Simulation results using real-world data. The red solid circle and the green dotted circle denote \mathbf{x}^{OPT} and \mathbf{x}^{ALG} , respectively.

an area of 200×200 square meters. We randomly chose five initial locations of the UAV within the environment and compared the performance with baselines (*i.e.*, Boustrophedon, Gradient ascent, and variance reduction). The three altitudes were set to be 10, 15, and 20 meters. The number of arms at each altitude was 400.

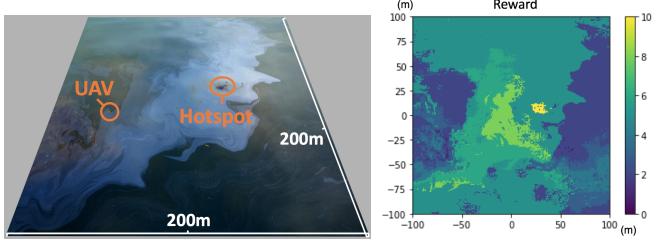


Fig. 7: Image on the left shows a snapshot of the Gazebo simulation environment. The image of oil spill in San Francisco is used as ground plane in Gazebo to model the plume. Image on the right shows the Ground-truth reward map for the plume.

We computed the average performance values over both point and arm performance metrics from all five runs when the budget of 5000 was given. The average performance values we obtained are $83.05 \pm 3.16\%$ for MF-GP-UCB, $64.14 \pm 2.84\%$ for Boustrophedon, $32.86 \pm 3.34\%$ for Gradient ascent, and $70.62 \pm 5.00\%$ for variance reduction. We also observed that if a relatively small amount of budget is given, MF-GP-UCB (as well as any strategies) might not succeed as the robot does not explore the enough environment. This result supports our comparison analysis result as long as a certain amount of environment exploration is guaranteed by assigning a reasonable budget.

VII. CONCLUSION

In this paper, we propose an exploration algorithm that can find a hotspot in an unknown environment in limited time. We show how the UAV equipped with a downward-facing camera can be adopted to MAB and present simulation results to verify the performance of the algorithm. In particular, our proposed algorithm outperforms numerous baselines. Our empirical results indicate that incorporating CPV results in almost 10% improvement in performance and sparse GPs results in orders of magnitude lower computational time without sacrificing any performance.

Immediate future work would be to conduct more rigorous real-world experiments using the proposed scheme. Analyz-

ing the regret bound or suboptimality with respect to the optimal reward would be promising.

REFERENCES

- [1] Y. Sung and P. Tokekar, “A competitive algorithm for online multi-robot exploration of a translating plume,” in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 3391–3397.
- [2] “Nri: Coordinated detection and tracking of hazardous agents with aerial and aquatic robots to inform emergency responders,” Oct 2016. [Online]. Available: https://nsf.gov/awardsearch/showAward?AWD_ID=1637915
- [3] C. Powers, R. Hanlon, and D. Schmale, “Tracking of a fluorescent dye in a freshwater lake with an unmanned surface vehicle and an unmanned aircraft system,” *Remote Sensing*, vol. 10, no. 1, p. 81, 2018.
- [4] P. Tokekar, J. Vander Hook, D. Mulla, and V. Isler, “Sensor planning for a symbiotic uav and ugv system for precision agriculture,” *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1498–1511, 2016.
- [5] O. M. Cliff, R. Fitch, S. Sukkarieh, D. L. Saunders, and R. Heinsohn, “Online localization of radio-tagged wildlife with an autonomous aerial robot system,” in *Robotics: Science and Systems*, 2015.
- [6] N. Srinivas, A. Krause, S. Kakade, and M. Seeger, “Gaussian process optimization in the bandit setting: No regret and experimental design,” in *Proceedings of the 27th International Conference on Machine Learning*, no. CONF. Omnipress, 2010.
- [7] J. Quiñonero-Candela and C. E. Rasmussen, “A unifying view of sparse approximate gaussian process regression,” *Journal of Machine Learning Research*, vol. 6, no. Dec, pp. 1939–1959, 2005.
- [8] M. Titsias, “Variational learning of inducing variables in sparse gaussian processes,” in *Artificial Intelligence and Statistics*, 2009, pp. 567–574.
- [9] M. Dunbabin and L. Marques, “Robots for environmental monitoring: Significant advancements and applications,” *IEEE Robotics & Automation Magazine*, vol. 19, no. 1, pp. 24–39, 2012.
- [10] X. Lan and M. Schwager, “Rapidly exploring random cycles: Persistent estimation of spatiotemporal fields with multiple sensing robots,” *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1230–1244, 2016.
- [11] K. H. Low, J. Chen, J. M. Dolan, S. Chien, and D. R. Thompson, “Decentralized active robotic exploration and mapping for probabilistic field classification in environmental sensing,” in *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*. Citeseer, 2012, pp. 105–112.
- [12] H. Hajieghrary, M. A. Hsieh, and I. B. Schwartz, “Multi-agent search for source localization in a turbulent medium,” *Physics Letters A*, vol. 380, no. 20, pp. 1698–1705, 2016.
- [13] M. Silic and K. Mohseni, “Field deployment of a plume monitoring uav flock,” *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 769–775, 2019.
- [14] E. Galceran and M. Carreras, “A survey on coverage path planning for robotics,” *Robotics and Autonomous systems*, vol. 61, no. 12, pp. 1258–1276, 2013.
- [15] H. Choset, “Coverage of known spaces: The boustrophedon cellular decomposition,” *Autonomous Robots*, vol. 9, no. 3, pp. 247–253, 2000.
- [16] G. Flaspohler, N. Roy, and Y. Girdhar, “Near-optimal irrevocable sample selection for periodic data streams with applications to marine robotics,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–8.
- [17] S. Manjanna, A. Q. Li, R. N. Smith, I. Rekleitis, and G. Dudek, “Heterogeneous multi-robot system for exploration and strategic water sampling,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–8.
- [18] K.-C. Ma, Z. Ma, L. Liu, and G. S. Sukhatme, “Multi-robot informative and adaptive planning for persistent environmental monitoring,” in *Distributed Autonomous Robotic Systems*. Springer, 2018, pp. 285–298.
- [19] G. A. Hollinger and G. S. Sukhatme, “Sampling-based robotic information gathering algorithms,” *The International Journal of Robotics Research*, vol. 33, no. 9, pp. 1271–1287, 2014.
- [20] J. Binney, A. Krause, and G. S. Sukhatme, “Optimizing waypoints for monitoring spatiotemporal phenomena,” *The International Journal of Robotics Research*, vol. 32, no. 8, pp. 873–888, 2013.
- [21] J. Snoek, H. Larochelle, and R. P. Adams, “Practical bayesian optimization of machine learning algorithms,” in *Advances in neural information processing systems*, 2012, pp. 2951–2959.

- [22] A. Slivkins *et al.*, “Introduction to multi-armed bandits,” *Foundations and Trends® in Machine Learning*, vol. 12, no. 1-2, pp. 1–286, 2019.
- [23] P. B. Reverdy, V. Srivastava, and N. E. Leonard, “Modeling human decision making in generalized gaussian multiarmed bandits,” *Proceedings of the IEEE*, vol. 102, no. 4, pp. 544–571, 2014.
- [24] S. Guha and K. Munagala, “Multi-armed bandits with metric switching costs,” in *International Colloquium on Automata, Languages, and Programming*. Springer, 2009, pp. 496–507.
- [25] P. Vansteenwegen, W. Souffriau, and D. Van Oudheusden, “The orienteering problem: A survey,” *European Journal of Operational Research*, vol. 209, no. 1, pp. 1–10, 2011.
- [26] J.-Y. Audibert and S. Bubeck, “Best arm identification in multi-armed bandits,” 2010.
- [27] K. Kandasamy, G. Dasarathy, J. Oliva, J. Schneider, and B. Poczos, “Multi-fidelity gaussian process bandit optimisation,” *Journal of Artificial Intelligence Research*, vol. 66, pp. 151–196, 2019.
- [28] J. Song, Y. Chen, and Y. Yue, “A general framework for multi-fidelity bayesian optimization with gaussian processes,” in *The 22nd International Conference on Artificial Intelligence and Statistics*, 2019, pp. 3158–3167.
- [29] L. Wei, X. Tan, and V. Srivastava, “Expedited multi-target search with guaranteed performance via multi-fidelity gaussian processes,” 2020.
- [30] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.
- [31] C. E. Rasmussen, “Gaussian processes in machine learning,” in *Summer School on Machine Learning*. Springer, 2003, pp. 63–71.
- [32] A. Krause, “Sfo: A toolbox for submodular function optimization,” *The Journal of Machine Learning Research*, vol. 11, pp. 1141–1144, 2010.
- [33] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, “Ros: an open-source robot operating system,” in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.
- [34] ROS-hector_quadrotor, “Wiki.” [Online]. Available: http://wiki.ros.org/hector_quadrotor
- [35] B. Inaglory, “Oil spill in san francisco bay.jpg.” Dec 2007. [Online]. Available: https://commons.wikimedia.org/wiki/File:Oil_spill_in_San-Francisco_bay.jpg