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

Yoonchang Sung, Deeksha Dixit, and Pratap Tokekar

***Abstract*— We are 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 must use for learning the unknown field and identifying the hotspot. Our first contribution is to show how this problem can be formulated as a novel variant of the Gaussian Process (GP) multi-armed bandit problem. The novelty is two-fold: (i) unlike standard multi-armed bandit settings, the arms ; and (ii) unlike standard GP regression, the measurements in our problem are image (*i.e.*, vector measurements) whose quality depends on the altitude at which the UAV flies. We present a strategy for finding the sequence of UAV sensing locations and empirically compare it with a number of baselines. We also present experimental results using images gathered onboard a UAV.**

I. Introduction

Robots are tasked with monitoring unknown environments but 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 so as to better estimate and predict states of the environment.

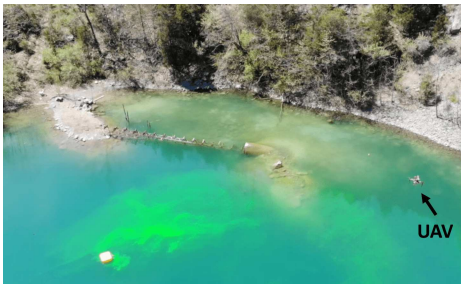


Fig. 1: A UAV exploring the environment to search for the plume in a lake [1].

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 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 physical specimen at that location. Thus, the UAV acts as an explorer whereas the USV acts as a sampler. In this paper, our focus is on planning strategies for the UAV to find the location with the highest concentration in the limited battery lifetime.

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 of unknown, spatially varying function. There exists many practical applications in the environmental monitoring literature, such as precision agriculture [4], wildlife habitat monitoring [5], plume tracking [1], where such a problem arises.

Due to noisy measurements, the more number of measurements from one sensing location, the more accurate the estimate. However, spending too much time at one location is not beneficial since the UAV has a limited budget. This is the famous exploration-exploitation dilemma which is 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 performance for sensing locations at different altitudes. Second, there is no clear interpretation between the camera image and measurements, and moreover, we need to define how good a sensing location is with respect to the other sensing location.

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. We then propose several heuristic planning strategies to qualitatively evaluate the proposed algorithm.

*This material is based upon work supported by the National Science Foundation under Grant No. 1637915.

Y. Sung is with the Department of Electrical and Computer Engineering, Virginia Tech, VA, 24061, USA. yooncs8@vt.edu.

D. Dixit and P. Tokekar were with the Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA 24061, USA when part of the work was completed. They are currently with the Department of Computer Science, University of Maryland, College Park, MD 20742, USA. {deeksha,tokekar}@umd.edu).

The contributions of this paper are as follows: (1) We propose an algorithm adapted from the MAB framework for environmental monitoring in a 3D environment with a camera sensor whose noise depends on the altitude. (2) We empirically compare the performance of the proposed heuristic strategies through extensive simulations and real-world data.

Our validation is based on real-world data we gathered from the field using a UAV. We empirically show the performance of several planning algorithms through simulations. These results demonstrate the effectiveness of the future variance that we propose in the MAB framework which reflects various flight altitudes of the UAV.

The rest of this paper is organized as follows. We begin by introducing the related work in Section II. We describe the problem setup in Section III. Our proposed algorithm is presented in Section IV. We present results from simulations in Section VI before concluding with a discussion of future work in Section VII.

II. Related Work

There have been various algorithms proposed for environmental monitoring. In this section, we briefly introduce some of recent works. For survey results, see [7].

Offline algorithms compute a trajectory for the robot before operation. Such coverage planning algorithms [8] need to know the environment a priori. Either a predefined trajectory (*e.g.*, boustrophedon path [9]) 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 environment online. Many decision-making challenges arise from this context. For environmental monitoring, the robot has to decide what samples to collect if samples are available only for specific time [10], [11], and where to take measurements constrained to a limited budget [1].

Information gathering is also an important task for learning the environment. In particular, entropy reduction [12], [13], or mutual information maximization [14] 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 [15] addresses this issue by balancing between exploration and exploitation. Previous works regarding MAB problems proposed algorithms to identify a near-optimal arm within a given budget. They also proved some guarantees on the regret bound (*i.e.*, a quantity implying how suboptimal reward the robot obtains).

There exist variants of MAB which consider switching costs. These variants can be useful in robotics because a traveling cost is a bottleneck for robots. Reverdy *et al*. [16] employed a block allocation scheme that restrains the robot from switching to other arm frequently. Guha and Munagala [17] developed an algorithm and related with the orienteering problem [18] to minimize traveling costs while maximizing collected rewards. However, their algorithm cannot handle spatially correlated arms. Their approach and Audibert and Bubeck [19] used a terminal metric for evaluating their algorithms, which is considered in this paper.

III. Problem Description

Let $\mathcal{E} \subseteq \mathbb{R}^2$ represent the 2D environment that contains the contamination. 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 environment (denoted by $\mathbf{x}^{OPT} \in \mathcal{E}$) that has the highest intensity of contamination. We deploy a UAV to explore the environment, constrained by limited time allowed for searching for this particular point. We denote the total allowed time by budget B. We compute the time spent as the combination of sensing time T_S and traveling time T_T. While T_S is a fixed constant value, T_T is the time measured for traveling from one location to the next, assuming that the UAV moves with unit speed. The highest contamination point inferred by the UAV 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 timestep k by $\mathbf{v}(k) \in \mathbb{R}^3$. At every sensing location, the UAV collects an image of the environment from a limited FOV camera sensor.

Let $f(\cdot)$ be the true 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 $f(\mathbf{x})$. Based on its estimate, the UAV must determine, $\mathbf{x}^{ALG}$, its own estimate of the point of highest intensity at the end of its path. We wish to find strategies that will minimize $f(\mathbf{x}^{OPT}) - f(\mathbf{x}^{ALG})$. Since the UAV has a limited budget B, however, the number of images the UAV can collect is also limited. Thus, sensing locations must be carefully chosen as the information on the unknown intensity is revealed online.

Without learning the unknown intensity, the UAV cannot identify the highest contamination point. Thus, the UAV explores unvisited regions to gather more information while carefully spending its given budget.

To sum up, we propose the following problem.

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

IV. Algorithm

A. Arm Locations

We call sensing locations as *arm locations* since that is the standard terminology in the multi-armed bandit literature. We create a 3D grid where each grid location is an arm (Figure 2). We denote the arm locations by $\mathrm{A} = \{\mathbf{a}_1, ..., \mathbf{a}_i, ..., \mathbf{a}_N\} \subseteq \mathbb{R}^3$ where N is the total number of arm locations. We place the grid such that every point in the environment will be in the FOV of at least one arm at the lowest altitude.

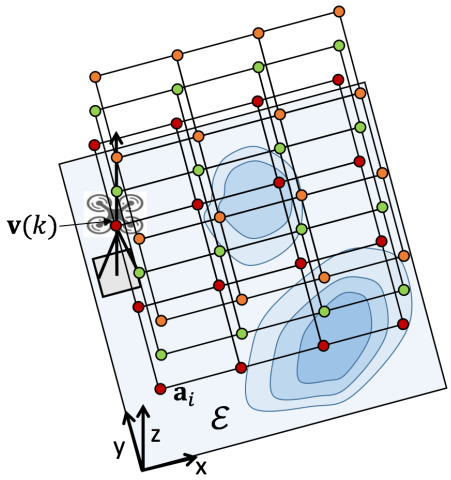


Fig. 2: Illustration of our problem setting where arms at three different altitudes are placed in a given environment $\mathcal{E}$.

B. Sensor Model

At each arm location, the UAV obtains a camera image from the downward-facing camera. The camera footprint will be larger from higher altitudes.

A single image measurement yields M pixel measurements. 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 [20].

C. Reward

We assume that there is a function that takes as input an image and produces a noisy estimate of the true intensity function for every pixel. The reward is a function of concentration of the contamination in the area covered by that pixel. In the simulation section, we describe the reward function we use.

We denote the reward value of the j-th measurement by $y_j \in \mathbb{R}_{\geq 0}$. We assume that the robot obtains noisy estimates of the true reward function:

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

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 the larger y represents a higher contamination.

At each timestep, the UAV obtains M measurements where one pixel corresponds to one measurement. This measurement is the noisy version of the true intensity function at the location at the center of the footprint of the pixel on the ground. Therefore, at timestep k, we have a collection of kM measurements, which we denote by Y. The corresponding kM measurement locations are denoted by X. Note that these kM locations denoted by X are not the same as the k sensing locations where the k images are obtained from.

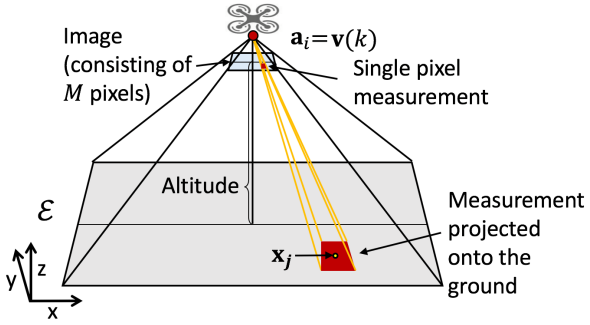


Fig. 3: Description of how the obtained image can be converted into measurements from an arm location.

Notice that the footprint of a pixel is not a point but an area, as shown in Figure 3. 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. This means that 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. GP

This subsection discusses about 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, however, measurements we obtain have various noise levels based on which altitude the measurements were observed. This is related with the noise variance $\sigma_N^2(\mathbf{a}_i)$ (one of hyperparameters of the GP), as explained in Section IV-B. Therefore, we adapt the 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. We define the noise variance matrix given by:

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

which is a diagonal matrix. Each diagonal element of $Q(\mathrm{X})$, $\sigma_N^2(\mathbf{a}_i)$, has a unique value based on the altitude of $\mathbf{a}_i$, but

we set $\sigma_N^2(\mathbf{a}_i)$ for arms at the same altitude to have the same value. The dimension of $Q(\mathrm{X})$ corresponds to the total number of measurements accumulated during the flight, *i.e.*, $Q(\mathrm{X}) \subseteq \mathbb{R}^{kM \times kM}$ at timestep 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 $\mathbf{a}_i$ which we call test points. We denote the set of L_i test points by $\mathrm{X}_i^* = \{\mathbf{x}_1^*, ..., \mathbf{x}_{L_i}^*\} \subset \mathcal{E}$. 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 $\mathrm{X}^* = \cup_{i=1}^{N} \mathrm{X}_i^* = \{\mathbf{x}_1^*, ..., \mathbf{x}_l^*, ..., \mathbf{x}_L^*\} \subset \mathcal{E}$. The prior of the GP then becomes:

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

where $\mathcal{K}(\cdot,\cdot)$ is a covariance function (or kernel). In this work, we use the squared exponential covariance function [21]. The predictive mean and covariance are:

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

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

E. Pseudo-code

The pseudo-code in Algorithm 1 demonstrates the entire 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 timestep k, the UAV updates the GP mean and variance functions (Lines 10 and 13) by using measurements collected up to and including the previous timestep (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 UAV spends the time more than B (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

In this section, we present several heuristic planning strategies that can be applied in Algorithm 1. Our main contribution is the strategy that is presented in Section V-B. However, before we present our strategy, we explain the standard GP-UCB algorithm and then present the two changes we make to the algorithm (future variance and dynamic windows).

We present the objective function (Line 17 of Algorithm 1) that is decision-making of the UAV to decide which arm to 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 according to 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 offline through simulations.

Algorithm 1: GP-UCB with Future Variance

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: $\mathrm{A} = \{\mathbf{a}_1, ..., \mathbf{a}_i, ..., \mathbf{a}_N\}$, test set: $\mathrm{X}^* = \{\mathbf{x}_1^*, ..., \mathbf{x}_l^*, ..., \mathbf{x}_L^*\}$, measurement set: $\mathrm{X} = \emptyset$, reward value set: $\mathrm{Y} = \emptyset$.

1 **for** $k = 1, 2, ...$ **do**
2 **if** $T_T\big(\mathbf{v}(k), \mathbf{v}(k\text{-}1)\big) + T_S k > B$ **then**
3 $\mathbf{x}^{ALG} = \arg\max_{\mathbf{x}} \mu$.
4 `TerminateExploration`$\Big(\mathbf{x}^{ALG}\Big)$.
5 **end**
6 **if** $k = 1$ **then**
7 Choose a nearest arm $\mathbf{a}_i(1) = \mathbf{v}(1)$.
8 **end**
9 **else**
10 $\mu = \mathcal{K}(\mathrm{X}^*,\mathrm{X})\Big(\mathcal{K}(\mathrm{X},\mathrm{X}) + Q(\mathrm{X})\Big)^{-1}\mathrm{Y}$.
11 **for** $i = 1, 2, ..., N$ **do**
12 $\mathrm{X}' = \mathrm{X} \cup \mathbf{x}_{I_i}^*$.
13 $P|\mathbf{a}_i = \mathcal{K}(\mathbf{x}_{I_i}^*, \mathbf{x}_{I_i}^*) - \mathcal{K}(\mathbf{x}_{I_i}^*, \mathrm{X}')\Big(\mathcal{K}(\mathrm{X}', \mathrm{X}') + Q(\mathrm{X}')\Big)^{-1}\mathcal{K}(\mathrm{X}', \mathbf{x}_{I_i}^*)$ where $P \ni \sigma_l^2 \ \forall l \in \mathrm{I}_i$.
14 $\overline{\mu}_i = \big(\sum_{l \in \mathrm{I}_i} \mu_l\big)/L_i$.
15 $\overline{\sigma}_i^2 = \big(\sum_{l \in \mathrm{I}_i} \sigma_l^2\big)/L_i^2$.
16 **end**
17 $\mathbf{a}_i(k) = \arg\max_i \Big\{\overline{\mu}_i + \beta\overline{\sigma}_i\Big\}$.
18 **end**
19 $\mathbf{a}_i(k) \leftarrow$ `MoveUAV`$\Big(\mathbf{v}(k\text{-}1)\Big)$.
20 $\big\{\mathrm{X}, \mathrm{Y}\big\} \leftarrow \big\{\mathrm{X}, \mathrm{Y}\big\} \cup$ `GetMeasurements`$\Big(\mathbf{a}_i(k)\Big)$.
21 **end**

A. GP-UCB

GP-UCB we use is originated from Srinivas *et al*. [6] but extended to take into account our problem setting. Moreover, with Equations (4) and (5), GP-UCB can handle various noise levels for collected measurements. Algorithm 1 without having Line 12 is this 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 next strategy.

B. GP-UCB with Future Variance

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 conditional variance, *i.e.*, $P|\mathrm{A}$.

Let I_i be the index set containing indices of test points from X^* which can be observed from arm $\mathbf{a}_i$. Thus, the set of test points by arm i is represented by $\mathbf{x}^*_{I_i}$. We decompose the conditional variance $P|\mathrm{A}$ into N conditional variances, *i.e.*, $P|\mathbf{a}_i$ for each arm. Then, the test points and the training points become $\mathbf{x}^*_{I_i}$ and $\mathrm{X}' = \mathrm{X} \cup \mathbf{x}^*_{I_i}$, respectively, for arm i. The decomposed conditional variance for the i-th arm can be computed as:

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

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

C. DWA GP-UCB

The previous strategies are not concerned with minimizing the total traveling cost of the UAV. If an algorithm makes the UAV keep moving 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 a budget B.

We employ the Dynamic Window Approach (DWA) where the 3D window is defined centered on the current position of the UAV. The UAV is only allowed to visit near-neighboring arms to observe at each timestep. To do that in Algorithm 1, the UAV considers neighboring arms (*i.e.*, a subset of A) with respect to the current UAV position $\mathbf{v}(k)$ from Line 10 to Line 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 the same maximum intensity value (*i.e.*, 50) so as to have the same $f(\mathbf{x}^{OPT})$ for all cases) and ran 10 cases for each environment. We set three altitudes for the UAV to fly at, *i.e.*, 10, 40, and 70 meters. Each image consists of 9 pixels and the camera footprint size at the lowest altitude is 1×1 square meters. We considered the total budget of 100 in all cases.

We conducted a comparison analysis of planning strategies in Section V 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.

We define two performance metrics that are the percent of how close $f(\mathbf{x}^{ALG})$ is to $f(\mathbf{x}^{OPT})$ (from perspective of a point) and that of how close $\sum_{\mathbf{a}_i^{ALG}} f(\mathbf{x})$ is to $\sum_{\mathbf{a}_i^{OPT}} f(\mathbf{x})$ (from perspective of an arm). 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.

	Point	Arm
CV (β--)	$47.65 \pm 4.52\%$	$46.31 \pm 4.51\%$
FV (β--)	$57.40 \pm 4.83\%$	$57.49 \pm 5.17\%$
CV (β++)	$49.39 \pm 4.81\%$	$50.62 \pm 4.83\%$
FV (β++)	$59.46 \pm 5.07\%$	$58.49 \pm 5.14\%$
DCV (β--)	$62.94 \pm 4.89\%$	$63.53 \pm 4.82\%$
DFV (β--)	$71.28 \pm 4.82\%$	$71.96 \pm 4.59\%$
DCV (β++)	$61.49 \pm 4.72\%$	$62.19 \pm 4.59\%$
DFV (β++)	$\mathbf{71.60 \pm 4.73\%}$	$\mathbf{74.51 \pm 4.52\%}$
Block UCL	$35.13 \pm 1.84\%$	$11.91 \pm 0.45\%$
BH	$35.86 \pm 3.16\%$	$3.40 \pm 0.00\%$
BM	$34.81 \pm 2.74\%$	$3.40 \pm 0.00\%$
BL	$28.73 \pm 2.83\%$	$3.40 \pm 0.00\%$
D2DH (β--)	$65.81 \pm 4.81\%$	$69.58 \pm 4.52\%$
D2DH (β++)	$66.93 \pm 4.70\%$	$72.12 \pm 4.50\%$
D2DM (β--)	$63.89 \pm 5.00\%$	$69.08 \pm 3.93\%$
D2DM (β++)	$69.23 \pm 4.71\%$	$71.22 \pm 4.12\%$
D2DL (β--)	$44.48 \pm 4.67\%$	$45.88 \pm 4.60\%$
D2DL (β++)	$47.56 \pm 5.00\%$	$48.31 \pm 4.91\%$

TABLE I: Comparison with eight heuristic planning strategies, two baseline algorithms, and 2D exploration algorithms. The values represent the percents of how close the best estimated value is to the true best value with one-sigma error. Acronyms in this table are: CV (Current Variance), FV (Future Variance), DCV (DWA Current Variance), DFV (DWA Future Variance), BH (Boustrophedon at the Highest altitude), BM (Boustrophedon at the Middle altitude), BL (Boustrophedon at the Lowest altitude), D2DH (DWA 2D exploration at the Highest altitude), D2DM (DWA 2D exploration at the Middle altitude), and D2DL (DWA 2D exploration at the Lowest altitude). β++ and β-- imply β with the increasing rate and the decreasing rate, respectively.

A. Comparison Analysis

We evaluated 8 heuristic planning algorithms where we also consider the decreasing rate and the increasing rate of β and compared with 10 other baselines (see the details of all algorithms in the caption of Table I).

From randomly generated environments, we tune the hyperparameters (γ and λ) of β (Equation (6)) offline through simulations for both the increasing rate and the decreasing rate. We use the following exponential functional forms:

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

1) Comparison with baseline algorithms: We introduce two baseline algorithms that can also be applied to our problem: (1) the boustrophedon algorithm [9] and (2) the block Upper Credible Limit (UCL) algorithm [16]. The boustrophedon algorithm is coverage planning where the entire environment is completely covered by the camera footprint of the UAV but gathered information online is not exploited. 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 obtained within a budget, but it also minimizes the number of switching to other arms, which is related to minimizing a traveling cost.

From Table I, the proposed algorithms outperform 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. Due to the allocation scheme used by the block UCL, the algorithm requires sensing at an arm a large number of times before moving to the next arm. This resulted in exploring only a small portion of the environment.

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 I that flying at a specific altitude is not beneficial in comparison with 3D exploration.

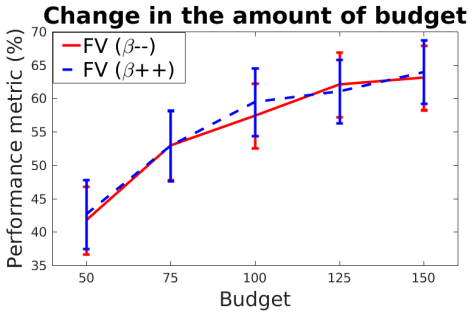


Fig. 4: Relationship between the budget and the performance metric for a point in case of GP-UCB with current and future variances.

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 4, the performance metric increases logarithmically as the budget increases.

B. Real-world Data

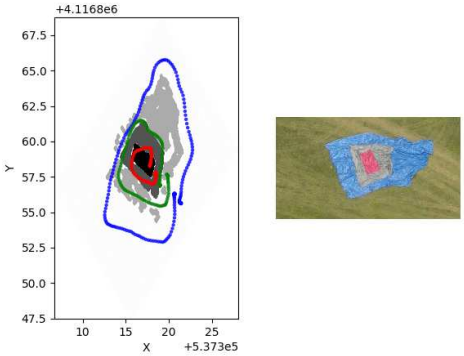
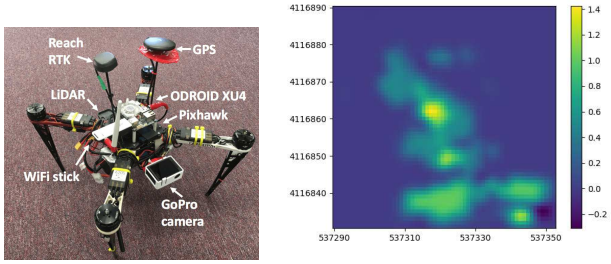


Fig. 5: Image on the right shows the tarps used as proxy of the regions with different concentrations. Image on the left shows the concentration for the image on the right along with the ground truth boundaries.

In our experimental setup we used the UAV (Figure 6) with a single onboard-PC which has Linux 16.04 and ROS kinetic [22] 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. We used three tarps of blue, gray, and red colors as a proxy of regions with different concentration of toxicants. We



(a) The UAV platform. (b) The resultant GP learned by the UAV from real-world data.

Fig. 6: Simulation results using real-world data.

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). We gathered 346 images spanned over these altitudes. Figure 5 shows the ground truth. For each image in our dataset, we assigned the concentration value $f(\mathbf{x})$ for each color on a per-pixel basis. We have three discrete concentration values of 3, 2, and 1 associated with HSV color ranges of the red, gray, and blue tarps, respectively. To generate the concentration matrix, we created masks for each of the aforementioned colors. These masks are morphologically processed for smoothening, and depending on which mask a particular pixel belongs to, it is assigned a concentration value. Left image in Figure 5 shows the visualization of the concentration values in Universal Transverse Mercator (UTM) coordinates. The darker region represents higher coordinate value.

In simulation the original resolution of the image 240×432 was reduced to 19×34. Thus, the number of pixel measurements from an image (*i.e.*, M) was 646. The camera footprint at an arm depends on the yaw of the UAV as well as $\mathbf{v}$. To use the concentration matrix with GP-UCB, we do the mapping from image coordinates to real-world 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 each pixel position of a given image in UTM coordinates, we rotate the frame of reference to make it align with the direction of UTM coordinate axis at that point. We achieve this by rotating each pixel, hence the entire image in a direction opposite to the UAV heading. Consequently, in the rotated frame of reference the UAV heading aligned with the north direction and hence the UTM coordinate axis. This makes it possible to use the geometric formula for translation in x- and y-axis along with UTM coordinates of the image center to interpolate the UTM location of each pixel.

Once we have the rotated frame of reference and the UTM coordinates of the image center, the translation values in x- and y-direction, respectively, can be used to compute the

Camera Parameters	Focal Length	Sensor Height	Sensor Width
Value	$6.21mm$	$4.04mm$	$5.87mm$

TABLE II: Intrinsic camera parameters obtained from camera calibration.

UTM coordinates of the pixel [20]. The intrinsic parameters of the camera sensor are obtained from camera calibration (Table II). It can be seen from Figure 5 that our estimates of the UTM locations aligns well with the ground-truth boundaries.

Figure 6 (b) shows the resultant GP by applying the proposed algorithm to real-world data. $\mathbf{x}^{OPT}$ is $(537317, 4116858)$ and $\mathbf{x}^{ALG}$ is $(537317.78, 4116858.29)$. The performance metric we obtained is 100% as both $f(\mathbf{x}^{OPT})$ and $f(\mathbf{x}^{ALG})$ are 3.

VII. Conclusion

In this paper, we propose the 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.

Immediate future work would be to conduct more rigorous real-world experiments using the proposed scheme. Also, since we may suffer from explosively increasing number of measurements, we may need to use sparse GP. Analyzing 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 International Conference on Machine Learning*. Omnipress, 2010, pp. 1015–1022.

[7] 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.

[8] 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.

[9] H. Choset, "Coverage of known spaces: The boustrophedon cellular decomposition," *Autonomous Robots*, vol. 9, no. 3, pp. 247–253, 2000.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] J.-Y. Audibert and S. Bubeck, "Best arm identification in multi-armed bandits," 2010.

[20] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[21] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer School on Machine Learning*. Springer, 2003, pp. 63–71.

[22] 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.