Persistent Monitoring of Large Environments with Robot Deployment Scheduling in between Remote Sensing Cycles

Kizito Masaba, Monika Roznere, Mingi Jeong and Alberto Quattrini Li

***Abstract*— This paper proposes a novel decision-making framework for planning "when" and "where" to deploy robots based on prior data with the goal of persistently monitoring a spatio-temporal phenomenon in an environment. We specifically focus on large lake monitoring, where remote sensors, such as satellites, can provide a snapshot of the target phenomenon at regular cycles. Between these cycles, Autonomous Surface Vehicles (ASVs) can be deployed to maintain an up-to-date model of the phenomenon. However, deploying ASVs has a significant logistical overhead in terms of time and cost. It requires a team of people to go on site and spend a day or half to monitor the deployment. It is vital to not only be intentional about where to sample in the environment on a given day, but also determine the worth of deploying the ASVs that day at all. Therefore, we propose a persistent monitoring strategy that provides the days and locations of when and where to sample with the robots, by leveraging the remote sensing data as well as modeling the dynamics and changes in the variance via a Gaussian Process. Our approach minimizes the number of days and locations for sampling, while preserving the quality of estimates. Through simulation experiments using realistic spatio-temporal datasets, we demonstrate the benefits of our approach over traditional deployment strategies, including significant savings on the effort and operational cost of deploying the ASVs.**

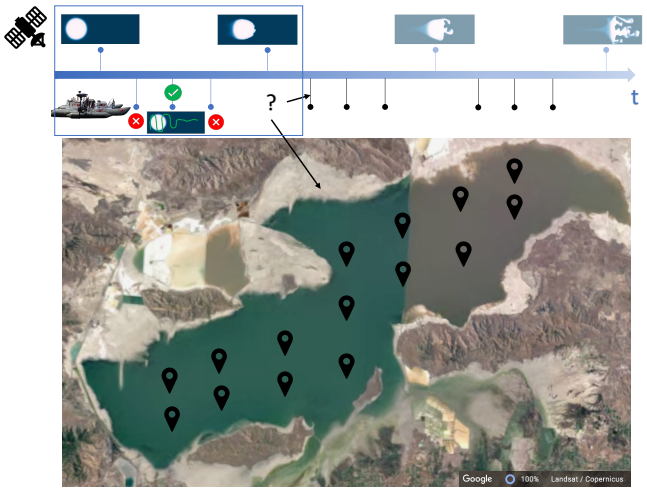

Fig. 1. Scenario: Given a region of interest with a spatio-temporal field, F, for which remote sensing data is available at regular intervals, and a robot, R (equipped with a sensor that can measure the field), which needs to coordinate with the remote sensor in order to persistently monitor F over a long period. On "Which days" and at "which locations" should R be deployed so that the uncertainty of the estimated spatial field remains low and the cost of operating R is minimized?

I. Introduction

We present a novel decision-making approach for planning "when" and "where" to deploy robots to sample, based on remote sensing data, to efficiently and accurately reconstruct a spatio-temporal phenomenon over a long-time horizon. The remote sensing data of the full environment is provided at fixed known intervals, while the robots sample smaller regions of interest during the remote sensing interludes. See Fig. 1 for the main idea.

Monitoring spatio-temporal phenomena is important for a number of high-impact applications, including environmental monitoring and precision agriculture [1]. A common sensing modality for such applications is hyperspectral imagery from various satellites [2]. While satellite flyovers occur at regular known intervals, the periods between data acquisition is somewhat large, e.g., LandSat's flyovers occur every two weeks. In order to persistently monitor a phenomenon of interest in the environment, it is crucial to utilize other data-collection technologies. For instance, water quality monitoring has been carried out with water quality measuring sondes carried by Autonomous Surface Vehicles (ASVs) [3]–[5] and images captured by a downward facing camera mounted on Unmanned Aerial Vehicles (UAVs) [6].

Most current research pursues strategies using robots to cover and sample regions of interest for one day's purpose [7]. However, many real-world phenomena in environments need to be monitored over long periods of time. Therefore, multiple robotic deployments are typically required over a 'season' of monitoring. Moreover, there is high logistical cost and overhead for deployments, such as traveling to the site, coordinating with team members, ensuring the robot safety, etc. Thus, it is generally difficult to conduct daily sampling campaigns.

Our goals are to reduce the on-site sampling burden and be more intentional on how the robotic deployment is conducted. We propose a strategy that exploits prior data, from both the remotely sensed and on-site sampled data, to decide whether a robot sampling deployment is necessary on a given day and, if so, where the robot(s) should sample – all of which to ensure an accurate daily estimate of the target spatio-temporal phenomena.

In particular, the main contributions include:

- A modeling framework based on a Mixture of Gaussian Process Experts (MoGPE) approach to quantify the uncertainty over time from the accumulated data, consisting of data from remote sensors and previous

The authors are in the Department of Computer Science, Dartmouth College, Hanover, NH, USA, 03755 {kizito.masaba.gr, monika.roznere.gr, mingi.jeong.gr, alberto.quattrini.li} @dartmouth.edu

robotic deployments.

- A novel decision-making framework, based on the aforementioned modeling framework, for spatio-temporal sampling over a long-time horizon.
- Analysis of the tradeoff between efficiency and accuracy – i.e., minimizing or maximizing the number of sampling performed – through experiments on synthetic and real-world phenomena.

Experimental results demonstrate that our method significantly reduces the number of sampling days while also preserving the quality of model reconstruction. This work represents the foundation of long-term spatio-temporal sampling to support scientists in their endeavor to protect our environment.

The paper is structured as follows. The next section discusses related work specifically focusing on spatio-temporal monitoring. Section III formally states the sampling problem of focus. Section IV describes the modeling used for keeping track of changes in the uncertainty over time, which allows for decisions to be taken on when and where to sample, as presented in Section V. Experiment setup, results and concluding remarks are presented in Section VI, VII, and VIII respectively.

II. Related Work

Several approaches have been proposed for sampling in a region of interest, where robots have to collect spatial information in the environment. Such approaches typically fall in two categories, *coverage* [8]–[10] with robots that have to cover every point in the region of interest with the sensor footprint; and *adaptive sampling* [11]–[14] where robots will adapt to measurements taken during the mission. Some recent surveys on exploration and sampling include [7], [15]. The mainstream methods look at spatial sampling in static environments, neglecting the temporal components important in expeditionary science [16]. Here, we highlight the work that focuses on spatio-temporal monitoring.

Spatio-temporal monitoring can be achieved by having robots following preplanned missions [17] or offline optimized paths [18] and collecting data repeatedly. Some adaptation of preplanned missions to external factors can allow persistent monitoring in the ocean [19]. Reactive strategies like artificial potential field can also be found in the literature [20]. Graphs can represent the locations and their connections that the robots need to visit; when the objective function is submodular, i.e., when a sample from a location provides less utility because other closer locations have already been sampled, a greedy algorithm can have approximation guarantees [21]. Gaussian Processes (GPs) have been used for modeling the spatio-temporal map, where for example the robot moves following a simple behavior [22]. More commonly, GPs are used for finding paths that maximize the information gain and minimize the traveled distance [23]–[25]. Monte Carlo Tree Search (MCTS) is another technique used to balance exploration and exploitation and is shown to capture time dynamics [26], [27]. Caley and Hollinger [28] compared in simulation a number of different methods for sampling in ocean areas above a certain threshold ranging from boustrophedon [8]to sequential Bayesian optimization [29]. The latter, an adaptive sampling method, outperforms the others. Some methods model explicitly dynamics so that the exploration strategy can predict environment change and accordingly decide when to explore a specific area [30]. In a multirobot scenario, some work have looked at distributing the workload fairly by geometrically subdividing the region of interest [31]. A distributed multirobot strategy exploits a reduced-order model from sparse measurements in order to then estimate areas without measurements and accordingly reconfigure the sensing locations [32].

There have also been attempts on combining different sensor streams and proposing sampling strategies, such as mobile robots and static nodes [33], ASV with UAV [34], [35] and ASV with satellite [36]. Their goal was to enhance the efficiency of the sampling mission by compensating the weakness of one sensor stream and leveraging the strength of another sensor stream.

Differently from the current literature, our objective is to use prior data to take informed decisions on days when to deploy the robots and sample rather than making arbitrary deployments.

III. Problem Statement

Our aim is to estimate the state of an unknown spatio-temporal phenomenon F in a 2D environment $\mathscr{E} \subset \mathbb{R}^2$, during a monitoring period with a long-time horizon T. Specifically, $F = \bigcup_{t=1}^{T} F_t$, where F_t is a snapshot of F at some time step t.

We utilize two heterogeneous sensors, Z_R and Z_G. Z_R is a remote sensing tool (e.g., satellite) that observes F_t at fixed time intervals T_R, where $1 < T_R \leq T$, to obtain the data $D_R = \{D_{iT_R}\}, \forall i \in \{0, \ldots, \frac{T}{T_R}\}$. Z_G is a point measurement sensor (e.g., ASV) that can observe F_{t_D} if deployed at t_D to obtain the data $D_G = \{D_{t_D}\}$. Initially, t_D is unknown, as it represents the time steps of when to deploy Z_G.

We assume D_R and D_G can be transformed into a common space Z by some function $\Psi : D_* \mapsto Z$, such that $F_t = f(\Psi(D_*)) + \varepsilon \sim \mathcal{N}(0, \sigma^2)$, where ε is the Gaussian noise in $\Psi(D_*)$. Given Ψ, Z_R and Z_G do not need to collect data during the same time step. Since Z_R's time schedule is known and set, only Z_G's schedule needs to be optimized, by minimizing the cost of deployment and operation while preserving a predictive accuracy, Δ of F_t.

Thus, our goals are:

1) To determine the time t_D in the future of when to deploy Z_G while preserving the desired predictive accuracy, and
2) Given t_D, to identify the critical locations X_c that Z_G must sample.

IV. Spatio-Temporal Field Modelling

As our proposed approach is based on GPs and their properties, here we include a brief description of the GPs together with an analysis on the accuracy of the estimates.

Let $D = [X,Y]$ be a dataset, where X is a 3D vector with $(t,p,q) \in X$, with $t \in \mathscr{R}$ being time and $(p,q) \in \mathscr{E}$ a location in $\mathscr{E}$. Let Y be a vector of their corresponding measurements, $y = \Psi(D_*) + \varepsilon$ collected by any given sensor, such that for every $(t,p,q) \in X$, there is a corresponding $y \in Y$. Let X_* be a 3D vector of test inputs, whose corresponding measurements, Y_* are to be estimated. Then, a GP model f_* for estimating X_* is drawn from a normal distribution defined as

$$f_*|X,Y,X_* \sim \mathscr{N}(\mu(X,X_*),\Sigma), \tag{1}$$

where the mean vector $\mu(X,X_*)$ covariance matrix Σ are

$$\mu(X,X_*) = K(X_*,X)[K(X,X) + \sigma_n^2 I]^{-1} Y, \tag{2}$$

$$\Sigma = K(X_*,X_*) - K(X_*,X)[K(X,X) + \sigma_n^2 I]^{-1}]K(X,X_*) \tag{3}$$

The elements of the covariance matrix, $K(\cdot,\cdot)$ are given by a kernel function, which describes the spatio-temporal correlation between a pair of inputs. We apply a commonly used kernel because of its general applicability to different domains, the squared exponential (SE), [37], defined as

$$k_y(x_p,x_q) = \sigma_f^2 \exp(-\frac{(x_p - x_q)^2}{2l^2}) + \sigma_n^2 \sigma_{pq}, \tag{4}$$

where l is the length scale representing the function smoothness; σ_f^2 is signal variance determining the amplitude; σ_n^2 is the noise variance accounting for the estimate noise; and δ_{pq} is the Kronecker delta ($\delta_{pq} = 1$ if $p = q$, else $\delta_{pq} = 0$).

Using the SE kernel, a GP model is parameterized by $\theta = (\sigma_f^2, l, \sigma_n^2)$, which are determined from the data using Maximum Likelihood Estimation (MLE) [37], by maximizing

$$\log p(Y|X,\theta) = -\frac{1}{2} Y^T \Sigma_y^{-1} Y - \frac{1}{2} \log|\Sigma_y| - \frac{n}{2} \log 2\pi \tag{5}$$

Suppose we are given a *GPR* estimator, $\hat{\theta}$ that estimates some random variable θ; the Mean Squared Error (MSE) [38] in the estimates is given by

$$\text{MSE}(\hat{\theta}) = E[(\hat{\theta} - \theta)^2] \tag{6}$$

We can express $E[(\hat{\theta} - \theta)^2]$ in terms of the GPR variance to obtain [38]

$$\text{MSE}(\hat{\theta}) = \text{Var}(\hat{\theta}) + \text{Bias}^2(\hat{\theta}) \tag{7}$$

where $\text{Var}(\hat{\theta})$ is the posterior variance of GPR and $\text{Bias}(\hat{\theta})$ is the model bias. Accordingly, for an unbiased estimator as a GP [39]

$$\text{Var}(\hat{\theta}) \leq \text{MSE}(\hat{\theta}) = \Delta \tag{8}$$

From Equation (8), we can define the accuracy of GPR estimates as a function of its posterior variance. Based on this formulation, we define a desired accuracy value, Δ, to determine the efficiency of the adaptive sampling mission.

Note that the running time and memory complexity of GP modeling is $O(N^3)$ and $O(N^2)$, respectively [40] – where N is the size of the training set D. This makes the GP model intractable for real-time exploration of large areas (e.g., $N >$ 100 on an embedded system). We'll show in the next section how we address this challenge.

V. Sensor Scheduling Algorithm

Suppose we have a time series of snapshots of a spatio-temporal field. Then, its future flow can be predicted using various models [37], [41], one of which is a Gaussian Process Regression (GPR) model described in IV. In this section, we describe how a GPR can be used to optimize the schedule of sensors utilized in a persistent monitoring task.

Initially, we need to strategically identify *hot-spot locations*, $X \in \mathscr{E}$, whose measurements at any given time t are sufficient for reconstructing the snapshot $F_t \subset F$ of the field. The goal is to minimize the number of data points required for modeling F and the cost of operating sensor Z_G, while also preserving the predictive accuracy of the estimates. One strategy for determining hot spots is to divide the environment into cells of a particular resolution and use the measurements from the center of each cell in the modeling. Note, the resolution can be empirically optimized for a given predictive accuracy Δ, using a GPR model.

Assuming the historical measurements of F, specifically $D_t = \bigcup_{i=-\infty}^{t} D_i$ taken up to t from X, can be used to accurately reconstruct F_t, the following properties are upheld:

1) *Spatial variability*: D_t captures the spatial variations (non-stationarity) that occurs in $F_t, \forall t = 1,\ldots,T$.
2) *Spatio-temporal variability*: D_t captures the spatio-temporal variations in F that occurs within the time window $[-\infty,t]$.

With the *spatial variability* property, the task of monitoring spatial variations in F_t can be reduced to monitoring *hot-spot locations* X at t. This minimizes the computational cost of modeling F_t and the operational cost of sampling with sensor Z_G. On the other hand, the *spatio-temporal variability* property allows us to predict temporal variations that may occur at every location $x \in X$ and, hence, identify those that may require sampling in the near future for proper planning. We leverage these properties in designing the proposed approach for scheduling the deployment of the on-demand sensor Z_G.

A. Predicting Future States

Given X, D_t (i.e., all data collected up to t,) and a desired predictive accuracy Δ, our goal is to predict snapshots $F_{T_c} = \{F_i, \forall i = t, t+1, \ldots, t+T_c\}$ for timestamped *hotspot* locations X_{T_c}, where T_c is the remaining number of time steps from t to the end of the current cycle for sensor Z_R. We focus on this particular time window because no new data is collected from any sensor in this period, making it the right time to decide on whether or not to deploy Z_G. To obtain F_{T_c}, we can train a GPR model with D_t and use it for prediction. However due to the high computational demands of GPR, this approach may become intractable as the size of D_t increases over time. To address this, we propose a mixture of GP experts approach outlined in Algorithm 1.

In this approach, D_t is partitioned into clusters and each cluster is used to train a GPR model that makes local predictions of X_{T_c}, denoted as $\hat{Y}_c$ (lines 2-6). We assume that the resulting clusters categorize D_t into i.i.d. subsets that can

Algorithm 1 Predict Future States, F_{T_c}

Input: t, D, X_{T_c}, T; Time step, Prior Data up to t, Timestamped hot spot locations, Time horizon
Output: μ_{Y_c}, Σ_{T_c}; Estimates for X_t

1: $C, C_D, \leftarrow clusterData(D)$; Cluster ids, Clustered data
2: $\hat{Y}, \leftarrow \{\}$; Cluster model estimates for X_t
3: **for all** $c \in C$ **do**
4: $\quad \hat{Y}_c \leftarrow Predict(C_D[c], X_{T_c})$
5: $\quad \hat{Y}[c] \leftarrow \hat{Y}_c$
6: **end for**
7: $\mu_{T_c}, \Sigma_{T_c}, \leftarrow \emptyset, \emptyset$
8: **for all** $x_i \in X_{T_c}$ **do**
9: $\quad \mu_{x_i}, \sigma^2_{x_i} \leftarrow argmin_{\hat{Y}_c^{x_i}} \{\sigma^2_{x_i} \in \hat{Y}_c^{x_i}, \forall c \in C\}$
10: $\quad$ Add μ_{x_i} to μ_{T_c}
11: $\quad$ Add $\sigma^2_{x_i}$ to Σ_{T_c}
12: **end for**
13: **return** μ_{T_c}, Σ_{T_c}

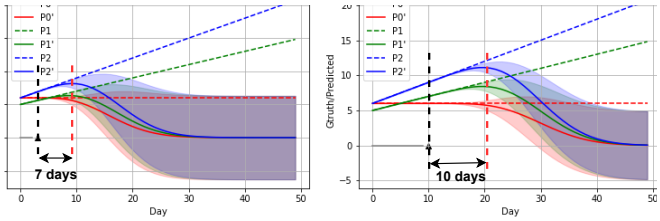

Fig. 2. GPR estimates for locations $P0$, $P1$, and $P2$, where their measurements are defined by functions $f(t) = 6 + \varepsilon$, $f(t) = 0.2t + 5 + \varepsilon$, and $f(t) = 0.3t + 8 + \varepsilon$, respectively. Left: GPR Estimates after training the model with historical data up to day 3. The model accurately estimates the field up to 7 days in the future without requiring more samples. Right: GPR estimates after training the model with historical data up to day 10. The model accurately estimates the field up to 10 days in the future without requiring more samples.

be modeled by a GPR, and the trained GPR can estimate new inputs that are within its cluster. Hence, training a local GPR model for each cluster allows for finding optimal parameters that representative of each cluster. Moreover, we use the DBSCAN [42], [43] technique for clustering because of its superior performance on dynamic data [44] and its ability to optimize the number of clusters. Hence, we predict F_{T_c} by finding the most optimal estimate $\hat{Y}_c^{x_i} \in Y_c$ for each instance $x_i \in X_{T_c}$ among all of its local estimates, such that

$$F_{T_c} = \bigcup_{\forall x_i \in X_{T_c}} \hat{Y}_c^{x_i}. \tag{9}$$

We assume that the most optimal estimate is the one with the smallest predictive accuracy (line 9). Note that F_{T_c} has two components: the posterior mean, μ_{T_c}, and variance, Σ_{T_c}. We obtain the snapshot F_t from μ_{T_c} and exploit Σ_{T_c} to determine the next *critical time* t_D as well as the corresponding *critical locations* X_c for scheduling Z_G, as described in the next section.

B. Scheduling Next Deployment

At any given time t we can compute the *critical time* t_D and *critical locations* X_c by analyzing Σ_{T_c}, obtained from the model as described in Section V-A. Let snapshot $\Sigma_c^{t_D} \subseteq \Sigma_{T_c}$ be the predictive accuracy for input $X_{t_D} \subseteq X_{T_c}$ (i.e., X at time step t_D). We consider $\Sigma_c^{t_D}$ to be critical if its maximum value exceeds Δ and if t_D is the closest time step to t. We refer critical time as t_D, where $t_D > t$, and critical locations as $X_c \subseteq X_{t_D}$, corresponding to values that exceed Δ. (See Algorithm 2, line 15).

We demonstrate the application of the proposed algorithm with a toy example shown in Fig. 2. In this example, X contains 3 unique locations, *P0*, *P1* and *P2*. The spatio-temporal field F exhibits unique variations at each location, such that the variation at *P0*, *P1* and *P2* are defined by linear functions $f(t) = 6 + \varepsilon$, $f(t) = 0.2t + 5 + \varepsilon$ and $f(t) = 0.3t + 8 + \varepsilon$, respectively, where $e \sim \mathcal{N}(0, \sigma^2)$ is Gaussian noise. In this example, we define $F = \{f(t)\}, \forall t = 1, \ldots, T = 50$ (days). The goal is to predict the appropriate time for taking new samples, such that a desired predictive accuracy is maintained. Using data from Day 0-3 (D_3) and from Day 0-10 (D_{10}), collected by both sensors Z_R and Z_G, we train GPR models and then predict the progression of F in subsequent time-steps. With D_3, we observe that the model can accurately predict F_t for 7 consecutive days in the future without new samples of F. Similarly, we observe that D_{10} can accurately predict up to 10 consecutive days without new data. Consequently in both scenarios, sampling is only necessary at Day 10 and Day 20 respectively. Hence, we can deploy Z_G at these time steps (i.e., *critical time*) to locations $X_c \subseteq X_{t_D}$ that need sampling (i.e., *critical locations*).

Note that if Z_R is scheduled within the 7-day or 10-day period, then deployment of Z_G is unnecessary, as an update of the model with new data from Z_R may extend the accurate prediction further out into the future. Hence, we optimize the deployment schedule of Z_G (t_D and X_{t_D}) by iteratively optimizing the predictive accuracy bound of the GPR model whenever new data is introduced. On deployment, Z_G samples *critical locations* using an off-the-shelf path planner, such as boustrophedon [8] or TSP [45].

VI. Experiments

We implemented the proposed approach in simulation and evaluated it using two unique spatio-temporal field datasets, *SF1* and *SF2*. Each dataset is a 100-frame video, where *SF1* simulates a realistic spread of a barium cloud and *SF2* simulates variation of water temperature over time. We obtained *SF1* from [35] and generated *SF2* synthetically with GSTOOLS [46]. We assume each video frame shows an accurate distribution of the field for a day, resulting in 100 days of ground truth data. Some snapshots from *SF1* and *SF2* are shown in Fig. 3. Both *SF1* and *SF2* are in a 400 m x 150 m environment with 200 hot-spot locations X, distributed at a resolution of 20 m.

Furthermore, we assume the remote sensor Z_R can accurately measure both *SF1* and *SF2* at fixed cycles T_R (in days). Whereas, Z_G can take in-situ measurements for both fields whenever it is deployed. We test different remote sensing cycles, 2, 5 and 10. For simplicity, we define $\Psi(z) = z + \varepsilon$, where z is a ground truth measurement and ε is Gaussian noise with 0 mean and variance, equivalent to 5% of the max value of the corresponding spatio-temporal field.

Algorithm 2 Sensor scheduling Algorithm

Input: X, T, T_R; Hotspot locations, Time horizon, Sampling cycle of sensor Z_R
Output: $F = \{F_t, \forall t = 1, ..., T\}$; Daily estimates of F across T

$S, D, t, t_D \leftarrow \{\}, \{\}, 0, -1$; locations scheduled for sampling, collected sensor data, time step, predicted time step for deployment of sensor Z_G
$F \leftarrow \emptyset$; Estimated Snapshots of F
while $t < T$ **do**
  $S[t+1] \leftarrow \emptyset$
  if $t\%T_R = 0$ or $t = t_D$ **then**
    if $t\%T_R = 0$ **then**
      $D_R \leftarrow$ fetch data with sensor Z_R
      $D[t] \leftarrow \Psi(D_R)$
    else
      $D_G \leftarrow$ Deploy sensor Z_G to sample at locations $S[t_D]$
      $D[t] \leftarrow \Psi(D_G)$
    end if
    $X_{T_c} \leftarrow$ generate timestamped X from t to $t + T_c$
    $\mu_{T_c}, \Sigma_{T_c} \leftarrow predictFutureStates(t, D_t, X_{T_c}, T)$
    $t_D, X_c \leftarrow scheduleNextDeployment(t, \Sigma_{T_c}, X, T_R)$
    $S[t_D] \leftarrow X_c$
    $F_t \leftarrow$ Extract estimates of X_t from μ_{T_c}
  else
    $F_t \leftarrow$ Compute Estimate for X_t from D_t
  end if
  Add F_t to F
  $t \leftarrow t + 1$
end while
return F

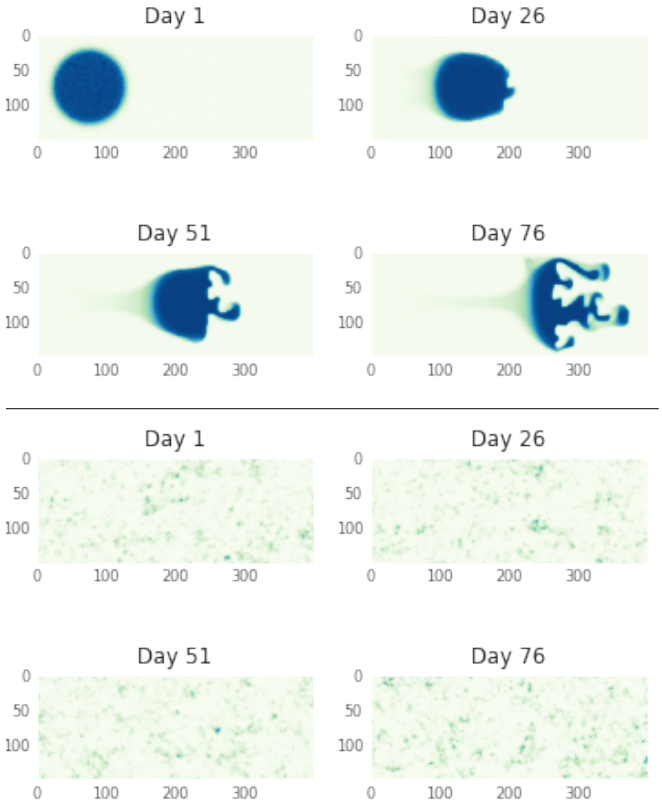

Fig. 3. Snapshots of the spatio-temporal fields, *SF1* (top) and *SF2* (bottom). *SF1* represents a barium cloud flow field, where a mass flows from the left to right, while dispersing over time. *SF2* represents the variation of water temperature over time.

Our goal is to obtain accurate daily estimates of *SF1* and *SF2* at the hot-spot locations, while minimizing the number of deployments and visited locations, $X_c \subseteq X$ of Z_G, throughout the 100-day period.

To achieve this, we consider four persistent monitoring strategies: 1) and 2) The proposed strategy is where Z_G is deployed to collect samples when necessary, and estimates are made based on both the collected data and the latest remote sensing data. This approach is further evaluated based on the adaptability of sample locations whenever Z_G is deployed. We denote the approach where all locations are sampled at every deployment as *ADAPT_S* and the approach where only critical hot-spot locations are sampled as *ADAPT_D*. 3) The *DAILY* strategy is where Z_G is deployed daily to collect samples, and estimates are made based on both the collected data and the latest remote sensing data. 4) The *REMOTE* strategy is where Z_G is never deployed, but estimates are made based on the previously collected remote sensing data. For all strategies, estimates are made from data using a GPR model described in Section IV.

Experiments were performed as a cron job on a Ubuntu machine with an Intel i7 CPU and 32GB RAM. For every deployment scheduled by the cron job, we spawned a team of four differential drive robots in Stage [47] simulator to traverse all hotspot locations scheduled for sampling. On deployment, we used the *m Traveling Sales person (mTSP)* [48] method, where $m = 4$, to compute subtours for each robot. However, other path planning methods can be implemented. In addition, we limited prior data used in training the model to a window of 10 days for computational tractability purposes.

For evaluation, we report the root mean square error (RMSE) of the daily estimates made by each strategy, the percentage of days Z_G is deployed across the entire monitoring duration, and the percentage of locations visited during each deployment. These metrics indicate the efficiency of persistent monitoring and the cost of operating Z_G.

In summary, we have two spatio-temporal fields, *SF1* and *SF2*, and we would like to monitor them using sensor Z_R and Z_G. Z_R samples the fields every 2, 5, and 10 days, whereas Z_R is deployed using three different strategies: *ADAPT_S*, *ADAPT_D*, and *DAILY*. In the next section, we evaluate the performance of the monitoring task, evaluating our methods against the alternative strategies.

VII. RESULTS AND DISCUSSION

We evaluate the effect of monitoring *SF1* and *SF2* in the hot-spot locations using *ADAPT_S*, *ADAPT_D*, *REMOTE*, and *DAILY* strategies.

Fig. 4 shows the aggregated *daily error* in estimates of *SF1* and *SF2* with each strategy. We observe that *DAILY* outperforms *ADAPT_** and *REMOTE* throughout the entire monitoring period, for both *SF1* and *SF2*. This is due to *DAILY*'s ability to collect daily data, which improves accuracy in all its estimates. However, *ADAPT_S* and *ADAPT_D* follow closely in performance since, unlike, *DAILY*, they collect data only when it is deemed necessary. The *REMOTE*

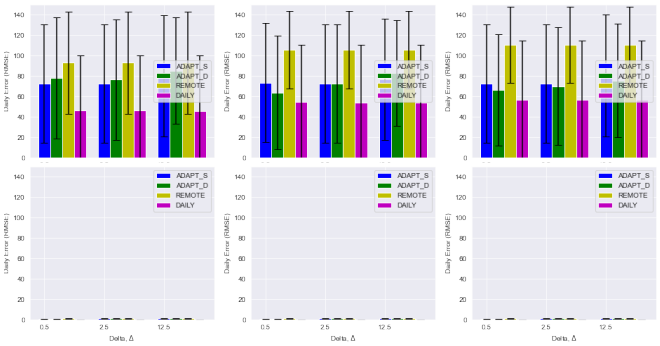

Fig. 4. Aggregated error (RMSE) in daily estimates of $F_t \subset F$, with Z_R acquiring data at 2 day (left), 5 day (center) and 10 day (right) intervals. Overall, *DAILY* (magenta) approach has the smallest error, closely followed by *ADAPT_S* (blue) and *ADAPT_D* (green), for both *SF1* (top row) and *SF2* (bottom row). *Remote* (yellow) has the highest error across all scenarios.

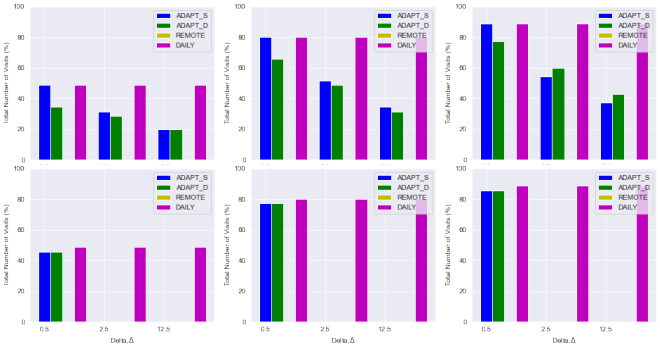

Fig. 5. Percentage of Z_G deployments based on each method across the time horizon, T, for cycles 2 days (left), 5 days (center), and 10 days (right). For both *SF1* (top row) and *SF2* (bottom row), *DAILY* (magenta) has the highest percentage of visits and *REMOTE* (yellow) has 0 across all cycles. *ADAPT_S* (blue) reports the second highest percentage of deployments followed by *ADAPT_D* (green).

strategy has the highest *daily error* due to its reliance on only remote sensing data.

Fig. 5 reports the number of days on which Z_G is deployed as a percentage of days within the monitoring period. We note that *DAILY* has the highest percentage of visits, for both *SF1* and *SF2*, followed by *ADAPT_S* and *ADAPT_D*. This high percentage of visits makes it impractical for implementing the DAILY strategy, especially in persistent environment monitoring, where technicians and domain experts need to be present on every monitoring mission. On the other hand, the relatively low visits when applying either *ADAPT* method is due to the implementation of the optimized deployment schedules. Moreover, when implementing either *ADAPT* strategy, Z_G visits less number of locations than under the *DAILY* strategy across all scenarios, as shown in Fig. 6. We also observe that Z_G visits less locations under *ADAPT_D* than under *ADAPT_S*. This indicates *ADAPT_D*'s ability to not only identify the right time to deploy Z_G but also select critical locations for sampling. This adaptive behavior is beneficial in minimizing the operational costs of Z_G. Overall, these advantages can be useful in optimizing sensor deployment schedules, thereby minimizing operational costs

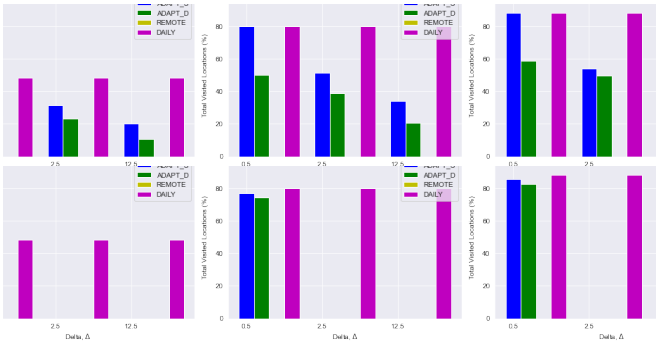

Fig. 6. Percentage of locations sampled by Z_G based on each method across the time horizon, T, for cycles 2 days (left), 5 days (center), and 10 days (right). Overall, Z_G samples the largest percentage of locations with the *DAILY* (magenta) strategy, followed by *ADAPT_S* (blue) then *ADAPT_D* (green) across all cycles, for both *SF1* (top row) and *SF2* (bottom row). Note, the *REMOTE* (yellow) method will never deploy Z_G.

and logistical demands associated with running a persistent monitoring mission with portable sensors.

VIII. Conclusions

We presented a decision-making framework that plans for when and where on-demand robot(s) should sample in an environment, based on previously collected data of past robotic deployments and remote sensing data. Analysis of temporal variance changes in a mixture of GP experts model estimates allows the decision-making framework to preserve a predictive accuracy over time, while minimizing the number of deployments. Our experiments with synthetic and real-world spatio-temporal phenomena demonstrated the effectiveness of our proposed approach compared to baseline strategies.

We plan to extend this work by determining how many and which types of robots, possibly equipped with different sensor quality tools, would be best to deploy. We expect to conduct a large scale experimental campaign over the summer to monitor cyanobacterial blooms in lakes.

Ultimately, our novel decision-making framework has the potential to significantly reduce the logistics and cost of data collection in the field, a vital requirement for operating various lake and ocean monitoring missions.

Acknowledgement

This work is supported in part by the Burke Research Initiation Award and NSF CNS-1919647, 2144624, OIA-1923004.

References

[1] 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.

[2] A. Erkkilä and R. Kalliola, "Patterns and dynamics of coastal waters in multi-temporal satellite images: Support to water quality monitoring in the archipelago sea, finland," *Estuarine, Coastal and Shelf Science*, vol. 60, no. 2, pp. 165–177, 2004.

[3] Y. Huang, Y. Yao, J. Hansen, *et al.*, "An autonomous probing system for collecting measurements at depth from small surface vehicles," *OCEANS 2021: San Diego – Porto*, 2021.

[4] H. Cao, Z. Guo, S. Wang, H. Cheng, and C. Zhan, "Intelligent wide-area water quality monitoring and analysis system exploiting unmanned surface vehicles and ensemble learning," *Water*, vol. 12, no. 3, p. 681, 2020.

[5] G. Ferri, A. Manzi, F. Fornai, F. Ciuchi, and C. Laschi, "The hydronet asv, a small-sized autonomous catamaran for real-time monitoring of water quality: From design to missions at sea," *IEEE Journal of Oceanic Engineering*, vol. 40, no. 3, pp. 710–726, 2014.

[6] D. Karimanzira, M. Jacobi, T. Pfützenreuter, *et al.*, "First testing of an auv mission planning and guidance system for water quality monitoring and fish behavior observation in net cage fish farming," *Information Processing in Agriculture*, vol. 1, no. 2, pp. 131–140, 2014.

[7] S. Bai, T. Shan, F. Chen, L. Liu, and B. Englot, "Information-driven path planning," *Current Robotics Reports*, pp. 1–12, 2021.

[8] H. Choset and P. Pignon, "Coverage path planning: The boustrophedon cellular decomposition," in *Field and Service Robotics*, A. Zelinsky, Ed. London: Springer London, 1998, pp. 203–209.

[9] A. Xu, C. Viriyasuthee, and I. Rekleitis, "Efficient complete coverage of a known arbitrary environment with applications to aerial operations," English, *Autonomous Robots*, vol. 36, no. 4, pp. 365–381, 2014.

[10] 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.

[11] M. Rahimi, R. Pon, W. J. Kaiser, G. S. Sukhatme, D. Estrin, and M. Srivastava, "Adaptive sampling for environmental robotics," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, IEEE, vol. 4, 2004, pp. 3537–3544.

[12] P. F. Lermusiaux, T. Lolla, P. J. Haley Jr, *et al.*, "Science of autonomy: Time-optimal path planning and adaptive sampling for swarms of ocean vehicles," *Springer Handbook of Ocean Engineering*, pp. 481–498, 2016.

[13] J. L. Nguyen, N. R. Lawrance, R. Fitch, and S. Sukkarieh, "Real-time path planning for long-term information gathering with an aerial glider," *Autonomous Robots*, vol. 40, pp. 1017–1039, 2016.

[14] 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. 4873–4880.

[15] A. Quattrini Li, "Exploration and mapping with groups of robots: Recent trends," *Current Robotics Reports*, pp. 1–11, 2020.

[16] V. Preston, G. Flaspohler, A. P. Michel, J. W. Fisher III, and N. Roy, "Robotic planning under uncertainty in spatiotemporal environments in expeditionary science," *arXiv preprint arXiv:2206.01364*, 2022.

[17] M. Dunbabin and A. Grinham, "Quantifying spatiotemporal greenhouse gas emissions using autonomous surface vehicles," *Journal of field robotics*, vol. 34, no. 1, pp. 151–169, 2017.

[18] J. Yu, M. Schwager, and D. Rus, "Correlated orienteering problem and its application to persistent monitoring tasks," *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1106–1118, 2016.

[19] R. N. Smith, M. Schwager, S. L. Smith, B. H. Jones, D. Rus, and G. S. Sukhatme, "Persistent ocean monitoring with underwater gliders: Adapting sampling resolution," *Journal of Field Robotics*, vol. 28, no. 5, pp. 714–741, 2011.

[20] P. P. Neumann, S. Asadi, A. J. Lilienthal, M. Bartholmai, and J. H. Schiller, "Autonomous gas-sensitive microdrone: Wind vector estimation and gas distribution mapping," *IEEE robotics & automation magazine*, vol. 19, no. 1, pp. 50–61, 2012.

[21] 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.

[22] T. M. Sears and J. A. Marshall, "Mapping of spatiotemporal scalar fields by mobile robots using gaussian process regression," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2022, pp. 6651–6656.

[23] 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: The 13th International Symposium*, Springer, 2018, pp. 285–298.

[24] T. O. Fossum, G. M. Fragoso, E. J. Davies, *et al.*, "Toward adaptive robotic sampling of phytoplankton in the coastal ocean," *Science Robotics*, vol. 4, no. 27, eaav3041, 2019.

[25] S. Garg and N. Ayanian, "Persistent monitoring of stochastic spatio-temporal phenomena with a small team of robots," *arXiv preprint arXiv:1804.10544*, 2018.

[26] R. Marchant, F. Ramos, S. Sanner, *et al.*, "Sequential bayesian optimisation for spatial-temporal monitoring," in *UAI*, Citeseer, 2014, pp. 553–562.

[27] W. Chen and L. Liu, "Multi-objective and model-predictive tree search for spatiotemporal informative planning," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, IEEE, 2019, pp. 5716–5722.

[28] J. A. Caley and G. A. Hollinger, "Data-driven comparison of spatio-temporal monitoring techniques," in *OCEANS 2015-MTS/IEEE Washington*, IEEE, 2015, pp. 1–7.

[29] P. Morere, R. Marchant, and F. Ramos, "Sequential bayesian optimization as a pomdp for environment monitoring with uavs," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2017, pp. 6381–6388.

[30] J. M. Santos, T. Krajník, J. P. Fentanes, and T. Duckett, "Lifelong information-driven exploration to complete and refine 4-d spatio-temporal maps," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 684–691, 2016.

[31] Y.-H. Kim and D. A. Shell, "Distributed robotic sampling of non-homogeneous spatio-temporal fields via recursive geometric subdivision," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2014, pp. 557–562.

[32] T. Salam, D. Kularatne, E. Forgoston, and M. A. Hsieh, "Adaptive sampling and energy efficient navigation in time-varying flows," in *Autonomous Underwater Vehicles*, Institution of Engineering and Technology, 2020, pp. 493–537.

[33] R. Graham and J. Cortés, "A cooperative deployment strategy for optimal sampling in spatiotemporal estimation," in *2008 47th IEEE Conference on Decision and Control*, IEEE, 2008, pp. 2432–2437.

[34] K. Cheng, S. Chan, and J. H. Lee, "Remote sensing of coastal algal blooms using unmanned aerial vehicles (uavs)," *Marine Pollution Bulletin*, vol. 152, p. 110 889, 2020.

[35] T. Salam and M. A. Hsieh, "Heterogeneous robot teams for modeling and prediction of multiscale environmental processes," *arXiv preprint arXiv:2103.10383*, 2022.

[36] D. J. Lary, D. Schaefer, J. Waczak, *et al.*, "Autonomous learning of new environments with a robotic team employing hyper-spectral remote sensing, comprehensive in-situ sensing and machine learning," *Sensors*, vol. 21, no. 6, p. 2240, 2021.

[37] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer school on machine learning*, Springer, 2003, pp. 63–71.

[38] H. Pishro-Nik, *Introduction to probability, statistics, and random processes*. QUBES Educational Resources, 2016.

[39] V. Suryan and P. Tokekar, "Learning a spatial field in minimum time with a team of robots," *IEEE Transactions on Robotics*, vol. 36, no. 5, pp. 1562–1576, 2020.

[40] B. Mikhail, B. Evgeny, and K. Yermek, "Exact Inference for Gaussian Process Regression in case of Big Data with the Cartesian Product Structure," in *ICML workshop on New Learning Frameworks and Models for Big Data*, 2014.

[41] 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)*, 2018, pp. 4873–4880.

[42] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "Dbscan revisited, revisited: Why and how you should (still) use dbscan," *ACM Transactions on Database Systems (TODS)*, vol. 42, no. 3, pp. 1–21, 2017.

[43] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *kdd*, vol. 96, 1996, pp. 226–231.

[44] S. Chakraborty, N. K. Nagwani, and L. Dey, "Performance comparison of incremental k-means and incremental dbscan algorithms," *arXiv preprint arXiv:1406.4751*, 2014.

[45] J. D. Little, K. G. Murty, D. W. Sweeney, and C. Karel, "An algorithm for the traveling salesman problem," *Operations research*, vol. 11, no. 6, pp. 972–989, 1963.

[46] S. Müller, L. Schüler, A. Zech, and F. Heße, "`GSTools` v1.3: A toolbox for geostatistical modelling in python," *Geoscientific Model Development*, vol. 15, no. 7, pp. 3161–3182, 2022.

[47] R. Vaughan, "Massively multi-robot simulation in stage," *Swarm intelligence*, vol. 2, no. 2, pp. 189–208, 2008.

[48] A. E. Carter and C. T. Ragsdale, "A new approach to solving the multiple traveling salesperson problem using genetic algorithms," *European journal of operational research*, vol. 175, no. 1, pp. 246–257, 2006.