Multi-Robot Adaptive Sampling based on Mixture of Experts Approach to Modeling Non-Stationary Spatial Fields

Kizito Masaba and Alberto Quattrini Li

Abstract— This paper presents an adaptive sampling strategy for a team of robots to model *non-stationary* spatial fields – i.e., fields with uneven variations – in large environments, with a desired *predictive accuracy*. Modeling non-stationary heterogeneous fields is essential for many applications, like monitoring air quality or contamination level in lakes. Mainstream adaptive sampling strategies assume stationarity of the environmental phenomenon and use a single model to explain such fields, resulting in inaccurate characterization of unique localized variations. In this paper, we model a non-stationary field as a collection of (infinite, in theory) non-overlapping layers of stationary homogeneous subfields. This approach allows for modeling non-stationary fields using a mixture of *experts*, where each expert is assigned a particular homogeneous subregion to map. This approach decomposes the environment into smaller homogeneous regions, which allows real-time modeling of large environments. We design a data-driven approach to adaptively identify each stationary layer and define its relationship to other layers. We model the relationship between various subregions as a network of experts – the rapidly-sampling adaptive graph. Each robot incrementally builds its own sub-network of experts, which is used to determine where to sample. Several experiments in realistic simulation demonstrate competitive accuracy and sampling efficiency compared to other state-of-the-art methods.

I. INTRODUCTION

We tackle the problem of multi-robot adaptive sampling for large data collection and modeling of a *non-stationary spatial field* with predefined *predictive accuracy*. In contrast to stationary fields that exhibit a uniform variation across the environment, non-stationary fields have uneven variations – a property that is more representative of real world phenomena, e.g., wildlife [1] and phytoplankton [2] distribution. Spatial field modeling is important for many applications, such as environmental monitoring and precision agriculture [3].

Mainstream adaptive sampling strategies commonly exploit Gaussian Process Regression (GPR) [4] – a non-parametric Bayesian model, to estimate the distribution of an environmental phenomena from its samples – because of its effectiveness in modeling spatial fields and providing an uncertainty measure for identifying informative regions [5]–[9]. However, GPRs typically assume a homogeneous distribution of the spatial field of interest, an assumption that breaks when modeling non-stationary fields as there tends to be heterogeneous spatial distributions [10]. In addition,
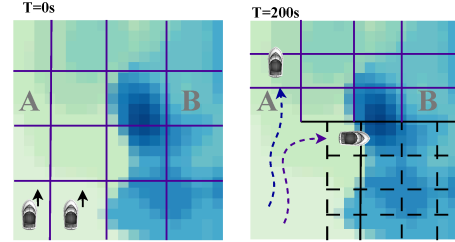
Fig. 1. Sample scenario of the problem considered in the paper: Given an unknown non-stationary field with unique regions, *A* and *B*, and multiple robots that are tasked to collect data to reconstruct the spatial field. Is it possible to adapt the sampling path to ensure a certain predictive accuracy (e.g., left vs. right grid) and at the same time minimize the task cost?

due to the cubic time computational complexity of GPR models, their application in environments that require a large number of measurements is still prohibitive. To address these limitations, some work has been done to enhance the performance of GPRs and its flexibility in modeling uneven variations. Examples include sparse GPRs [11], [12] that regulate the input data while maintaining high performance; multi-kernel learning [13] that combine several kernels to enhance the flexibility of the covariance function in modeling uneven distributions; methods that project the non-stationary field into a different space so that the stationary assumption can hold [14]; and a mixture of experts (MOE) [15]–[17] approach. Despite these efforts, the practical use of such enhanced modeling methods for adaptive sampling with a desired predictive accuracy is still an open challenge.

This paper proposes a scalable adaptive sampling strategy for multiple robots, which aims at modeling non-stationary spatial fields with a specific predictive accuracy, while minimizing traveled distance. To achieve these goals, taking inspiration from MOE, we propose a network of GPR experts, which are incrementally created by a team of robots as they sample the environment. Each expert is associated with a particular non-overlapping sub-region that it is tasked to model. Each robot creates its own sub-network of experts and uses it to determine where to collect samples and when a given expert can model its corresponding sub-region. Robots coordinate by restricting themselves to visiting a particular subarea of the environment through sharing the regions covered by the sub-network. A robot creates a new expert by visiting and sampling data at a location that is not covered by any expert. Depending on the variation in the data collected in a given locality, a sparse or dense network of experts will be created, as shown in Fig. 1. Regions with less variance in the data will be modeled by fewer experts than those with high variance. Consequently, some regions are sampled more densely than others, which makes the sampling process

adaptive to the non-stationarity in the environment.

Overall, this paper makes the following contributions:

1) a novel distributed MOE-based approach to modeling non-stationary spatial fields with a desired predictive accuracy.

2) a novel adaptive sampling strategy that reasons on such a model, allowing robots to identify informative sampling locations to explore.

3) a ROS implementation and an experimental analysis with realistic simulations comparing other methods used for environmental monitoring. Results show that enforcing a desired predictive accuracy enhances modeling and overall adaptive sampling performance.

II. RELATED WORK

Different approaches have been proposed for the sampling problem, where robots have to collect spatial information in the environment. Some recent surveys on exploration and sampling include [18], [19]. These methods can be categorized into two main families: *non-adaptive* and *adaptive*.

In non-adaptive approaches, informative paths are computed offline using a previously trained model, if available, and shared among robots. The problem addressed by these approaches is also called *coverage problem* [20]–[22], where the robots are assumed to have a sensor footprint and they need to cover every point in the workspace. Such coverage algorithms typically decompose the environment into cells, abstract them through a graph, and solve the related problem of visiting every node – please see a comprehensive survey from Galceran *et al.* [22] for full details. Some offline methods have looked at how to determine informative trajectories for modeling spatial distributions, for example, using a sequential allocation approach to find paths to allocate to robots, with mutual information to measure their informativeness [23]; an asymptotically optimal RRT-based approach [24]; or trained models based on data collected beforehand to find offline informative paths with a certain minimum accuracy [25].

Non-adaptive approaches have the advantage that plans can be computed beforehand without being constrained by on-board computational power availability. In fact, they are the main approach used in environmental monitoring missions. The main limitation of these methods is that computed plans do not adapt according to the collected data during the mission, which may introduce inefficiencies. In contrast, our method explicitly adapts the plan according to the data collected by the robots at a given locality in the environment.

In adaptive approaches, robots learn the spatial field by continuously updating their belief of the field using the collected samples and re-planning informative paths. Various works [6], [26]–[33] used different metrics to measure uncertainty (e.g., variance, entropy) and proposed information-based acquisition functions that enable robots to select informative locations to sample, which are locations that maximize a given acquisition function.

These approaches are capable of adapting the plan to the collected data and drive robots to areas of higher uncertainty.

They typically use a single model for the entire environment. In practice, the choice of using a single model over the whole environment has two limitations: 1) the scalability in the size of the environment might be limited and 2) there might be variations in the environment that would be better captured by an ensemble of models.

The scalability of the model has been addressed by some methods [7], [34], [35] that reason on smaller partitions of the environment. Some works [36], [37] proposed data fusion strategies that minimize the data used in training the GP. No guarantee on the predictive accuracy is provided by adaptive methods. Our work proposes a solution that focuses on maintaining a defined predictive accuracy based on the real-time data stream.

A specific type of model that can help towards improving the predictive accuracy is the Mixture of Gaussian Processes (MGP) that was introduced by Tresp *et al.* [15] as a model for active learning – another name for adaptive sampling – [38] problems. These methods model large spatial fields by aggregating local approximations obtained from a collection of models. For a comprehensive review of MGPs, please see the surveys from Liu *et al.* [39] and Yuksel *et al.* [17]. Here, we highlight some of the existing work that used MGPs in multi-robot adaptive sampling. For instance, Ouyang *et al.* [8] proposed a decentralized MGP approach for multirobot active learning of non-stationary phenomena, where each robot represents a Gaussian Process Expert (GPE) and data points are assigned to each expert following a Dirichlet Process [40]. Luo *et al.* [9] presented a distributed mixture of GPs that aim at improving the tractability of GPs in a multirobot system by eliminating the need for sharing raw data between robots. This is achieved by enabling each robot to model its own collected data and only exchange the resulting optimal GP parameters with other robots. Each robot creates one GP expert that is linked to other experts through their respective optimal parameters. Similarly, Kemna *et al.* [7] and Fung *et al.* [41] developed a decentralized mixture of GP models, where each robot has one model. To improve coverage and tractability of the GP models, robots independently divide the environment into Voronoi partitions [42], [43], which are then assigned to each robot. These approaches use information-theoretic acquisition functions, such as entropy, mutual information [6], [44], Integrated Variance Reduction (IVR) [28] and Upper Confidence Bound (UCB) [44] to identify hot-spots for sampling.

Our work differs from the existing work in 3 main aspects: 1) we define a new acquisition strategy whose utility is a function of a network of GP experts, which cover the explored region. This eliminates the data redundancy problem that results from the use of most acquisition functions such as IVR, UCB, and entropy [45], and greatly reduces the cost of computing sampling locations; 2) the number of experts required to model and accordingly partition a given workspace is independent of the size of the robot team – it only depends on the variation in the target environmental phenomenon; 3) we enforce a desired predictive accuracy on

the local estimates of every sub-region.

III. Background: Gaussian Process Regression

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 X be a 2D vector of locations and y a vector of their corresponding measurements collected by the robot. X_* is a vector of test locations, whose measurements are to be estimated. Then, a GP model f_* for estimating X_* is drawn from a normal distribution defined as

$$f_*|X, y, X_* \sim \mathcal{N}(\mu(X, X_*), \Sigma), \qquad (1)$$

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

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

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

The elements of the covariance matrix, $K(\cdot, \cdot)$ are given by a kernel function, which describes the spatial correlation between a pair of locations. We use a commonly used kernel because of its general applicability to different domains, the squared exponential (SE) [4], defined as

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

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) [4], by maximizing

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

Suppose we are given a *GPR* estimator, $\hat{\theta}$ that estimates some random variable θ; the Mean Squared Error (MSE) [46] in the estimates is given by

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

We can express $E[(\hat{\theta} - \theta)^2]$ in terms of the GPR variance to obtain [46]

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

where $\mathrm{Var}(\hat{\theta})$ is the posterior variance of GPR and $\mathrm{Bias}(\hat{\theta})$ is the model bias. Accordingly, for an unbiased estimator as a GP [25]

$$\mathrm{Var}(\hat{\theta}) \leq \mathrm{MSE}(\hat{\theta}) = \Delta \qquad (8)$$

From Equation (8), we can define the predictive accuracy of GPR estimates as a function of its posterior variance. Based on this formulation, we define a threshold, Δ, that ensures the corresponding desired predictive accuracy.

Note that the running time and memory complexity of GP modeling is $O(N^3)$ and $O(N^2)$, respectively [47] – where N is the size of the training set (X, y). This makes the GP model intractable for real-time exploration of large areas (e.g., $N > 100$ on an embedded system as a Raspberry Pi).

IV. Problem Statement

Given a 2D environment, $\mathcal{E}$ with an unknown non-stationary spatial field, $\mathcal{S}$, such that $s = [x \in \mathcal{E}, z \in \mathcal{R}] \in \mathcal{S}, \forall x \in \mathcal{E}$; a team of K differential drive robots, $R = \{1, ..., K\}$ that can communicate with each other through long-range WiFi or radio devices, and are equipped with a sensor that measures z. The robots' task is to adaptively sample from the environment to reconstruct $\mathcal{S}$ with a minimum desired predictive accuracy Δ and minimum traveled distance.

Although $\mathcal{S}$ is unknown, we can assume that it is composed of an unknown collection of homogeneous sub-fields so that spatially heterogeneous fields can be modeled. Hence, $\mathcal{E}$ can be expressed as $\mathcal{E} = \bigcup_{i=1}^{\infty} v_i$, where $v_i \subseteq \mathcal{E}$ is a sub-region centered at x_i within a radius r_i, containing a homogeneous subfield, $S_i \subseteq \mathcal{S}$. For simplicity, we truncate v_i such that it is bounded by a square that is circumscribed by the disc of radius r_i. This simplification allows us to work with grids, a popular discretization approach for coverage and exploration problems. Therefore, v_i is bounded by a square shaped region of length, $L_i = r_i\sqrt{2}$. Future work will look at other discretizations.

With sufficient measurements, $M_i \subseteq \mathcal{S}_i$ collected from within v_i, $\mathcal{S}_i$ can be reconstructed by a regression model, m_i with a desired predictive accuracy, Δ. However given that the boundaries of $\mathcal{S}_i$ are unknown and are potentially non-linear, it is also necessary to include measurements collected from the neighborhood of v_i. This allows for accurate modeling of boundary regions of $\mathcal{S}_i$.

We define a neighbor, v_j of v_i as any sub-region such that some of its measurements are within distance $\frac{L_i}{2}$ with at least one location, $x_i \in v_i$ (i.e., $\exists x_i \in v_i, x_j \in v_j; ||x_i - x_j|| \leq \frac{L_i}{2}$). To model $\mathcal{S}_i$, all locations in v_i must have a neighbor sub-region. Henceforth, we refer to this condition as the *neighborhood constraint.* For example, if all sub-regions are approximately of equal area (as in grids), then v_i will need to have measurements collected around the center of its surrounding sub-regions (between 3 and 8 neighbors of approximately equal sized grid cells) in order to satisfy the neighborhood constraint.

Identifying all the sub-regions that make up $\mathcal{S}$ along with their neighbors such that each sub-region satisfies its *neighborhood constraint* allows the robots to achieve the adaptive sampling task.

V. Rapidly-sampling Adaptive Graph

We devise the Rapidly-sampling Adaptive Graph (RAG), a connected undirected graph $\mathcal{G} = (V, E)$ (with V nodes and E edges), that encodes the predictive ranges of models used for estimating various subfields and the boundary neighborhood of these subfields. A subfield is a contiguous, homogeneous subset of a spatial field. A node $v_i = (x_i, L_i, m_i) \in V$ is a subfield defined by a tuple consisting of a center location, $x_i \in \mathcal{E}$, length, $L_i \in \mathcal{R}$ of a boundary within which the subfield is enclosed and a regression model, m_i that can accurately estimate the same subfield with a desired predictive accuracy, Δ. In addition, $(v_i, v_j) \in E$ is an edge
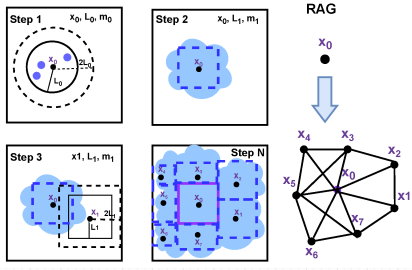
Fig. 2. Illustration of the steps for RAG generation. **Step 1**: RAG creation process starts with sampling data (blue dots) within radius L_0 from initial location x_0 (i.e., within circle with solid border). The data is used to train a GPR model which in turn is used to predict a region that is $2L_0$ radius from x_0 (i.e., region within circle with discrete border). **Step 2**: From the GPR estimates, a region whose posterior variance satisfies the Δ constraint is identified and a RAG Node is created within this region using the new parameters (L_1 and m_1). The new RAG node is added to the graph (top right). **Step 3**: The robot finds a new location, x_1 in the uncovered region and creates another RAG node using the parameters of the RAG Node closest to x_1. **Step N**: The process continues until the entire neighborhood of each node is covered. Whenever a node meets the neighborhood constraint, its region is modeled (e.g., x_0).

of $\mathcal{G}$ if and only if v_i and v_j are neighbors as defined in the previous section.

A *full RAG* covers the entire environment and every node satisfies the *neighborhood constraint*. In theory, an unknown spatial field in a continuous environment, V is assumed to have infinite nodes as every possible location can have a unique spatial field property. Whereas, for a known homogeneous spatial field, V could have only 1 node. In practice, a RAG usually has a finite set of nodes that varies with the size of the workspace and the predictive power of the model.

VI. MULTI-ROBOT SPATIAL FIELD ADAPTIVE SAMPLING

Our proposed adaptive sampling strategy with a desired predictive accuracy, Δ, constructs a *full RAG* and estimates the distribution of each node with its corresponding model. The overall process is depicted in Fig. 2 and the formalization of the algorithm is in Algorithm 1. In the following, we describe in detail the algorithm.

Algorithm 1 RAG Sampling Algorithm

Input: x_0, L_0; Initial Center and Boundary Length
1: $R, x, L \leftarrow \emptyset, x_0, L_0$
2: **while** x != -1 **do**
3:     Navigate to x
4:     $A \leftarrow$ Extract Sub-region of square length $2L$ centered at x
5:     $D \leftarrow$ Get measurements in square of length L centered at x
6:     $v \leftarrow$ CreateRAGNode(x, D, A)
7:     AddNodeToRAG(R, v)
8:     v.status $\leftarrow$ COVERED
9:     ModelCompleteNodes(R)
10:    ShareRAGNode(v)
11:    $x \leftarrow$ FindNextCenter(R, x)
12:    $L \leftarrow$ GetNextBoundaryLength(R, x)
13: **end while**

A. RAG Construction

RAG construction is an iterative two-step process composed of RAG node generation and RAG edge generation, until the full RAG is constructed – see Fig. 2.

1) RAG Node generation: Suppose we have a location x in the environment. To obtain the length of the boundary L, a GPR model is used to estimate the measurements of every location in some arbitrary area, A that is centered at x using data, D collected from within A. The output of a GPR prediction is the measurement estimates (posterior mean, $\mu_{A|D}$) and the predictive uncertainty (posterior variance, $\sigma^2_{A|D}$) of each location in A – as discussed in Section III. From these outputs we use $\sigma^2_{A|D}$ of each location to compute L as shown in the following:

$$L = \sqrt{2}\max[\min(x - x_D)\forall x \in A, \forall x_D \in V] \ s.t. \ \sigma^2_{A|D}(x) \leq \Delta] \quad (9)$$

which is the maximum distance of every location in A to its closest measurement. The rationale of this formulation is based on the property of the SE kernel, which we apply in GPR modeling. With a SE kernel, posterior variance/correlation of any location, x, to other estimates and measurements is only dependent on the hyper-parameters obtained from D and the distance between x and these locations, which reduces exponentially as the distance increases. Therefore, the estimate at a given location is highly dependent on its closest measurement. We hereby attribute this relationship to the predictive power of GPR, which is how wide it can accurately estimate a measurement at an unobserved location in A. Hence, we can observe that a GPR model that is trained with data in some region A, centered at x can predict a certain subset of the locations within A with a desired predictive accuracy, Δ.

Although the area of A can be arbitrary, its size has to be carefully selected to balance the trade-off between performance and predictive power of the model, which inherently balances the exploration-exploitation trade-off. In this work, we consider A to be a square region of length $2L$.

2) RAG Edge Generation: When a new RAG node is created, it induces new edges on the RAG depending on its proximity to other nodes and the boundary length of each node. For any new node, v_i that is introduced into the RAG and an already existing RAG node, v_j, (v_i, v_j) is an edge on the RAG if and only if $||x_i - x_j|| \leq L_i$. Likewise, another node on RAG will seek to establish an edge with v_i if it satisfies this proximity condition.

As discussed earlier, the purpose of these neighbors is to avail more data to the model of a node, such that it is able to accurately predict the entire sub-region within its boundaries.

B. Path Planning

With the environment initially unexplored, we propose the following strategy for a robot in pose x_r, to find the next location to sample:

$$x_{\text{opt}} = \arg\min_x(||x_r - x||I(x)), \forall x \notin V \quad (10)$$

The overall objective is to minimize traveled distance and at the same time to ensure that created RAG nodes quickly satisfy their neighborhood constraints and get modeled while extending to unexplored regions. Thus, we define the informativeness $I(x)$ of x in an unexplored (i.e., it is not covered by the RAG) sub-region of the workspace, as $I(x) = |\{v_i : ||x - x_i|| \leq L_i \land x \notin v_i\}|, \forall i = 1,...,|V|$, which is the

number of RAG nodes that are close enough to x that they can form a RAG edge with a node created at x. Specifically, we need to find a location in the unexplored sub-region at which a new node can be created and linked to at least one RAG node.

C. Modeling RAG Nodes

A sub-region of a given RAG node is modeled when the *neighborhood constraint* is satisfied by the node. A RAG node that satisfies such a constraint is called a *complete node*. A complete RAG Node, v_i has atleast one neighbor on the RAG, and each neighbor has its own model and data.

To model v_i, we need to compare the data obtained from its neighbors against its own data in order to ascertain if they are all homogeneous (i.e., have similar distributions) and thus determine whether its model needs retraining. If data from the neighborhood share a similar distribution with that of v_i, then v_i model needs no retraining. Otherwise, it will be retrained with the neighboring data in order to learn the patterns in the neighborhood, before the neighboring data is incorporated into the modeling. We perform a pairwise Student's t-test [48] to determine the similarity between v_i's data and that from each of its neighbors.

The new modeling dataset D_i^* for the v_i model is the data from v_i and all its neighbors:

$$D_i^* = D_i \cup \bigcup_{j=1}^{P} D_j \qquad (11)$$

Using D_i^*, the v_i model is used to estimate the distribution of every location within the v_i node, which is the ultimate spatial field estimate of this sub-region. The status of the modeled node is then changed to *MODELED*.

D. Robot Coordination

The RAG allows multirobot task allocation, restricting robots to particular regions of the environment. Initially, robots can be deployed in arbitrary locations. However, as they seek to construct their respective RAG, they encounter boundaries along the way.

We define boundaries as regions, which have been explored, yet they are not covered by the RAG. These regions are flagged by other robots through communication. A robot shares new nodes of its RAG with communicating robots to increase the situation awareness of its surrounding robots. Note that the algorithm does not require full communication since robots can continue expanding the RAG even if they are unable to communicate.

Note that RAG construction is restricted to unexplored regions and the planning objective function prioritizes locations that are closer to a robot's RAG nodes, thus pushing robots within their boundaries.

VII. EXPERIMENTAL RESULTS AND DISCUSSION

Setup. We implemented our method in Python and the Robot Operating System (ROS) [49] to enable portability across different robotic platforms. We ran several experiments with teams of differential drive autonomous surface
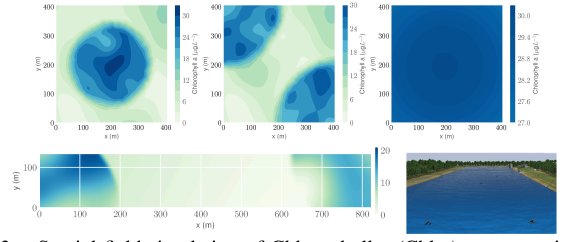

Fig. 3. Spatial field simulation of Chlorophyll a (Chl-a) concentration (in $\mu g L^{-1}$) as an indicator of the presence of cyanobacteria. SF1 (top left): The lateral region of the lake has lower Chl-a concentration; whereas the pelagic region has higher Chl-a concentration SF2 (top center): High Chl-a concentration along two opposite shorelines and low Chl-a concentration in the rest of the lake. SF3 (top right): homogeneous region. SF4 (bottom left) has two inlets (both ends) with high variation of Chl-a concentration, whereas the center regions exhibit low variation. Example of Gazebo simulation with 4 ASVs (bottom right).

vehicles (ASVs) in 2D and 3D simulators – Stage [50] and Gazebo [51] respectively – that have actuators and localization noise. In addition, Gazebo includes water dynamics.

We simulated four different environments, each with a generated unique non-stationary spatial field (SF) (Fig. 3), representing realistic variations of Chlorophyll-a concentration as an indicator of the presence of Cyanobacteria [52] in lakes. The purpose for these variations is to evaluate the adaptiveness of the proposed method in environments with various phenomena. Environments with *SF1*, *SF2* and *SF3* cover an area of size 400×400 m whereas the environment with *SF4* cover an area of size 823×134 m. We implemented a simulated sensor providing SRF measurements with Gaussian noise to account for the sensor error, such as the YSI EXO 2 sonde with chlorophyll-a probe [53], so that we can analyze the robustness to noisy data.

We spawned ASV teams of 3 different sizes $2, 4, 6$. We ran experiments with three (3) different accuracy thresholds, Δ to assess its effect on the overall performance, namely; $0.5, 1.0$ and 1.5. Each experiment was repeated 5 times. The experiments are run on a Ubuntu 20.04 computer with an Intel i7 CPU with 32GB RAM.

For comparison with our method (labeled *RAG* with corresponding Δ), we included both non-adaptive – being a commonly used strategy – and adaptive alternatives. For non-adaptive, we implemented a lawn-mower way point profile (labeled as *LM*), with five (5) different inter-lap spacing, namely; 8 m, 10 m, 20 m, 30 m and 40 m. We set 8 m resolution because it has been shown to perform better than higher resolutions [25] when a boustrophedon [54] waypoint pattern is applied. For adaptive, we implemented the dynamic partitioning approach (labeled as *DPart* with corresponding Δ) [7], where a robot team iteratively divides the environment into Voronoi partitions and each robot samples one unique partition. Instead of using only time as the termination criterion, we modify *DPart* to terminate whenever a desired Δ is achieved. We also cap the mission time to 3500 s as applied in [7] to guarantee termination.

Qualitative Analysis. As sample representatives, Fig. 4 shows the reconstructed fields of each environment, for *RAG0.5*, *LM_08*, *LM_40* and *DPart0.5* with a team of 6 robots respectively. We observe that all methods except *DPart0.5* accurately modeled *SF1* and *SF2*. However, only *RAG0.5*
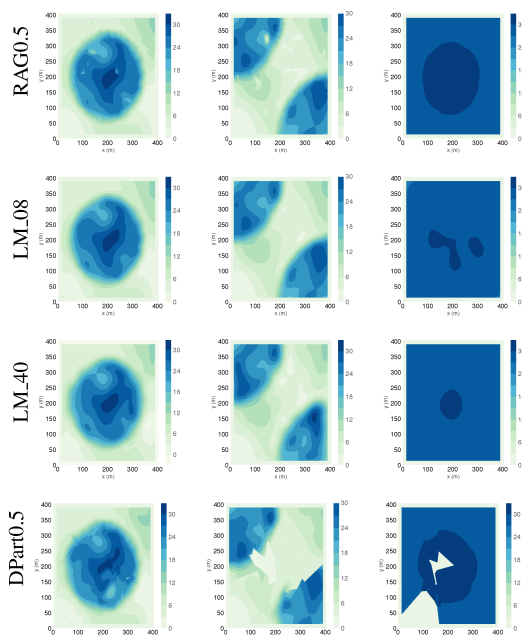
Fig. 4. Sample reconstruction of spatial fields using 6 robots – *SF1* (left), *SF2* (Center), *SF3* (Right) – while the methods indicated at each row. *RAG* accurately reconstructs all the spatial fields, while DPart leaves some gaps. Lawn mower generally is able to reconstruct the spatial fields with some inaccuracies dependent on the interlap spacing.
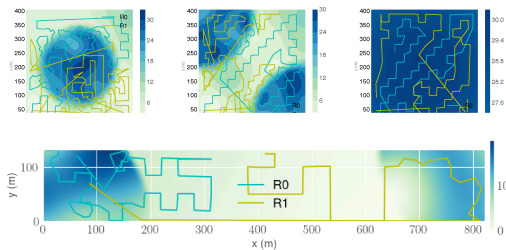

Fig. 5. Trajectories with *RAG* of a 2 robot team after the mission in *SF1* (Top Left), *SF2* (Top Center), *SF3* (Top Right) and *SF4* (Bottom) environments. Robots sample more densely at the *boundary* of the two homogeneous regions than in the homogeneous areas. This demonstrates the robustness of our method in sampling non-stationary fields.

managed to reconstruct *SF3* accurately. The failure for both *LM_** and *DPart0.5* is due to the areas that are not sampled by the robots. *LM* paths are planned with a non-adaptive interlap spacing, which limits the informativeness of the path and might miss for example the higher spot in SF3. On the other hand, *DPart* dynamically partitions the environment such that each robot focuses on a particular partition. As a result, some subregions of the environment can be ignored. These two shortfalls do not affect *RAG* as it ensures that all regions of the environment are sufficiently sampled in order to achieve the desired predictive accuracy.

Indeed, looking at the robot trajectories in the four environments – Fig. 5 – we observe that robots sample the fields more densely along the boundary of the two heterogeneous regions compared to the homogeneous regions. This is due to the high variance in the data collected in this area.

Quantitative Analysis. The quantitative evaluation is in terms of predictive accuracy measured as average posterior variance (aPVAR), reconstruction error measured as Root Mean Squared Error (*RMSE*) [55], average distance traveled
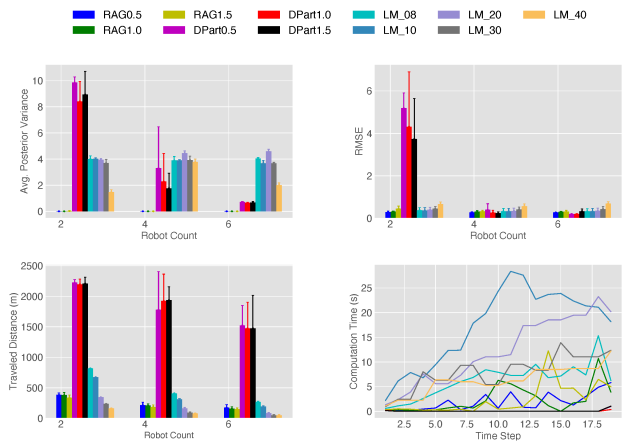

Fig. 6. Modeling SF2: Average Posterior variance (top left), RMSE (top right), traveled distance (bottom left), computation time (bottom right).

by the robots, and computation time to assess the feasibility of a given method for use in online adaptive sampling. As a representative example (other environments overall follow similar trends) we show the results for SF2 in Fig. 6.

The GPR estimates in *RAG* variants have the least aPVAR across all robot teams and environments because of the Δ. This is followed by *LM* and then *DPart* variants. Both *RAG* and *LM* variants record lower *reconstruction error* across all robot teams than *DPart* especially with a team of 2 robots. Observing its trajectories, *DPart*'s approach of partitioning the environment tends to leave out some unexplored regions. Overall, *RAG* variants achieve the least traveled distance across all robot teams, except for *LM_30* and *LM_40*, which however have the worst learning accuracy. The low traveled distance of our method is attributed to the adaptiveness of the data collection process as shown in the qualitative analysis. The computation time results show a constant time for RAG.

VIII. Conclusion

This work proposes an efficient multirobot active learning strategy for accurately reconstructing spatial fields. We propose an approach that allows for continuous modeling of the desired phenomenon with locally collected data to achieve local optimality based on a predefined desired predictive accuracy, Δ. The locally optimal regions are interconnected to account for local discontinuities, such that a given local region is modeled only when its entire neighborhood has been sampled. We model this process as a rapidly sampling adaptive graph, where its nodes are the locally optimal regions and edges encode the neighborhood of each locally optimal region. Experiments and results show a superior performance of the proposed method over non-adaptive and adaptive strategies from the literature.

In future work, we plan to incorporate kinodynamic constraints to account for other robot types for generating efficiently reachable neighbors, as well as explicitly modeling communication constraints to enable robots to take decisions on when to share information with other robots. In addition, we plan to study long-term sampling where temporal resolution is another important factor to decide when to deploy the robots. Finally, we intend to test the

proposed strategy in the real world, e.g., lake, to study algal blooms with a team of ASVs and contribute towards high-impact applications, such as the study of climate change.

REFERENCES

[1] P. E. Osborne, G. M. Foody, and S. Suárez-Seoane, "Non-stationarity and local approaches to modelling the distributions of wildlife," *Diversity and Distributions*, vol. 13, no. 3, pp. 313–323, 2007.

[2] S. Wu, Z. Du, Y. Wang, T. Lin, F. Zhang, and R. Liu, "Modeling spatially anisotropic nonstationary processes in coastal environments based on a directional geographically neural network weighted regression," *Science of the Total Environment*, vol. 709, p. 136 097, 2020.

[3] 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.

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

[5] C. Rasmussen and Z. Ghahramani, "Infinite mixtures of Gaussian process experts," *Advances in neural information processing systems*, vol. 14, 2001.

[6] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies.," *Journal of Machine Learning Research*, vol. 9, no. 2, 2008.

[7] S. Kemna, J. G. Rogers, C. Nieto-Granda, S. Young, and G. S. Sukhatme, "Multi-robot coordination through dynamic Voronoi partitioning for informative adaptive sampling in communication-constrained environments," in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2017, pp. 2124–2130.

[8] R. Ouyang, K. H. Low, J. Chen, and P. Jaillet, "Multi-robot active sensing of non-stationary Gaussian process-based environmental phenomena," in *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, Association for Computing Machinery (ACM), 2014.

[9] W. Luo, C. Nam, G. Kantor, and K. Sycara, "Distributed environmental modeling and adaptive sampling for multi-robot sensor coverage," in *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, 2019, pp. 1488–1496.

[10] S. Sharp, A. Forrest, K. Bouma-Gregson, Y. Jin, A. Cortés, and S. Schladow, "Quantifying Scales of Spatial Variability of Cyanobacteria in a Large, Eutrophic Lake Using Multiplatform Remote Sensing Tools. Front," *Environ. Sci*, vol. 9, p. 612 934, 2021.

[11] E. Snelson and Z. Ghahramani, "Sparse Gaussian processes using pseudo-inputs," *Advances in neural information processing systems*, vol. 18, 2005.

[12] S. Kim and J. Kim, "GPmap: A unified framework for robotic mapping based on sparse Gaussian processes," in *Field and service robotics*, 2015, pp. 319–332.

[13] A. Melkumyan and F. Ramos, "Multi-kernel Gaussian processes," in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2011.

[14] A. M. Schmidt and A. O'Hagan, "Bayesian inference for non-stationary spatial covariance structure via spatial deformations," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 65, no. 3, pp. 743–758, 2003.

[15] V. Tresp, "Mixtures of Gaussian processes," *Advances in neural information processing systems*, vol. 13, 2000.

[16] J. W. Ng and M. P. Deisenroth, "Hierarchical mixture-of-experts model for large-scale Gaussian process regression," *arXiv preprint arXiv:1412.3078*, 2014.

[17] S. E. Yuksel, J. N. Wilson, and P. D. Gader, "Twenty years of mixture of experts," *IEEE transactions on neural networks and learning systems*, vol. 23, no. 8, pp. 1177–1193, 2012.

[18] S. Bai, T. Shan, F. Chen, L. Liu, and B. Englot, "Information-Driven Path Planning," *Current Robotics Reports*, pp. 1–12, 2021.

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

[20] 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.

[21] 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.

[22] 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.

[23] A. Singh, A. Krause, C. Guestrin, and W. J. Kaiser, "Efficient informative sensing using multiple robots," *Journal of Artificial Intelligence Research*, vol. 34, pp. 707–755, 2009.

[24] 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.

[25] 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.

[26] B. Zhang and G. S. Sukhatme, "Adaptive sampling for estimating a scalar field using a robotic boat and a sensor network," in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2007, pp. 3673–3680.

[27] A. Krause and C. Guestrin, "Nonmyopic active learning of gaussian processes: an exploration-exploitation

approach," in *International Conference on Machine learning (ICML)*, 2007, pp. 449–456.

[28] A. Blanchard and T. Sapsis, "Informative path planning for anomaly detection in environment exploration and monitoring," *Ocean Engineering*, vol. 243, p. 110 242, 2022.

[29] A. Blanchard and T. Sapsis, "Bayesian optimization with output-weighted optimal sampling," *Journal of Computational Physics*, vol. 425, p. 109 901, 2021.

[30] Y. Yang, A. Blanchard, T. Sapsis, and P. Perdikaris, "Output-weighted sampling for multi-armed bandits with extreme payoffs," *Proceedings of the Royal Society A*, vol. 478, no. 2260, p. 20 210 781, 2022.

[31] J. P. Kleijnen and W. V. Beers, "Application-driven sequential designs for simulation experiments: Kriging metamodelling," *Journal of the operational research society*, vol. 55, no. 8, pp. 876–883, 2004.

[32] D. J. MacKay, "Information-based objective functions for active data selection," *Neural computation*, vol. 4, no. 4, pp. 590–604, 1992.

[33] D. Cohn, "Neural network exploration using optimal experiment design," *Advances in neural information processing systems*, vol. 6, 1993.

[34] G. P. Kontoudis and D. J. Stilwell, "Decentralized nested Gaussian processes for multi-robot systems," in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, pp. 8881–8887.

[35] L. Wu, M. Á. García García, D. Puig Valls, and A. Solé Ribalta, "Voronoi-based space partitioning for coordinated multi-robot exploration," *Journal of Physical Agents*, 2007.

[36] K.-C. Ma, L. Liu, and G. S. Sukhatme, "Informative planning and online learning with sparse gaussian processes," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2017, pp. 4292–4298.

[37] D. Jang, J. Yoo, C. Y. Son, D. Kim, and H. J. Kim, "Multi-robot active sensing and environmental model learning with distributed Gaussian process," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5905–5912, 2020.

[38] A. T. Taylor, T. A. Berrueta, and T. D. Murphey, "Active learning in robotics: A review of control principles," *Mechatronics*, vol. 77, p. 102 576, 2021.

[39] H. Liu, Y.-S. Ong, X. Shen, and J. Cai, "When Gaussian process meets big data: A review of scalable GPs," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 11, pp. 4405–4423, 2020.

[40] H. Föllmer, "Dirichlet processes," in *Stochastic integrals*, Springer, 1981, pp. 476–478.

[41] N. Fung, J. Rogers, C. Nieto, H. I. Christensen, S. Kemna, and G. Sukhatme, "Coordinating multi-robot systems through environment partitioning for adaptive informative sampling," in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2019, pp. 3231–3237.

[42] M. Tanemura, T. Ogawa, and N. Ogita, "A new algorithm for three-dimensional Voronoi tessellation," *Journal of Computational Physics*, vol. 51, no. 2, pp. 191–207, 1983.

[43] Y. Shi, N. Wang, J. Zheng, *et al.*, "Adaptive informative sampling with environment partitioning for heterogeneous multi-robot systems," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2020, pp. 11 718–11 723.

[44] R. Marchant and F. Ramos, "Bayesian optimisation for informative continuous path planning," in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2014, pp. 6136–6143.

[45] Y. T. Tan, A. Kunapareddy, and M. Kobilarov, "Gaussian process adaptive sampling using the cross-entropy method for environmental sensing and monitoring," in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2018, pp. 6220–6227.

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

[47] 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.

[48] D. W. Zimmerman, "Comparative power of Student t test and Mann-Whitney U test for unequal sample sizes and variances," *The Journal of Experimental Education*, vol. 55, no. 3, pp. 171–174, 1987.

[49] M. Quigley, K. Conley, B. Gerkey, *et al.*, "Ros: An open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, 2009, p. 5.

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

[51] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *IEEE/RSJ international conference on intelligent robots and systems (IROS)*, IEEE, vol. 3, 2004, pp. 2149–2154.

[52] I. Bertani, C. E. Steger, D. R. Obenour, *et al.*, "Tracking cyanobacteria blooms: Do different monitoring approaches tell the same story?" *Science of the Total Environment*, vol. 575, pp. 294–308, 2017.

[53] YSI, *The EXO Sonde Platform*, Accessed on Feb 1, 2023.

[54] H. Choset and P. Pignon, "Coverage path planning: The boustrophedon cellular decomposition," in *Field and service robotics*, Springer, 1998, pp. 203–209.

[55] C. J. Willmott and K. Matsuura, "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance," *Climate research*, vol. 30, no. 1, pp. 79–82, 2005.