LCD-RIG: Limited Communication Decentralized Robotic Information Gathering Systems

Abdullah Al Redwan Newaz[1], Paulo Padrao[2], Jose Fuentes[2], Tauhidul Alam[3], Ganesh Govindarajan[2] and Leonardo Bobadilla[2]

***Abstract*—Effective data collection in collaborative information-gathering systems relies heavily on maintaining uninterrupted connectivity. Yet, real-world communication disruptions often pose challenges to information-gathering processes. To address this issue, we introduce a novel method —a limited communication decentralized information gathering system for multiple robots to explore environmental phenomena characterized as unknown spatial fields. Our method leverages quadtree structures to ensure comprehensive workspace coverage and efficient exploration. Unlike traditional systems that depend on global and synchronous communication, our method enables robots to share local experiences within a limited transmission range and coordinate their tasks through pairwise and asynchronous communication. Information estimation is facilitated by a Gaussian Process with an Attentive Kernel, allowing adaptive capturing of crucial behavior and data patterns. Our proposed system is validated through simulated scalar field studies in non-stationary environments where multiple robots explore spatial fields. Theoretical guarantees ensure the convergence of distributed area coverage and the regret bounds of distributed online scalar field mapping. We also validate our method empirically in a water quality monitoring scenario featuring three Autonomous Surface Vehicles, tasked with constructing a spatial field.**

***Index Terms*—Limited communication, decentralized information gathering, Gaussian process, spatial fields.**

I. Introduction

ROBOTIC Information Gathering (RIG) is a process of optimizing an information-theoretic metric from efficient exploration of a continuous area of interest by robots with motion constraints while considering inferences from a probabilistic model within a limited mission time. A recent development has improved the uncertainty quantification of a probabilistic model using a nonstationary kernel (i.e., different locations have different degrees of variability) to identify critical locations for efficient information collection [1]. However, such a RIG planner only applies to a single robot application and fails to utilize a team of cooperative robots to collect spatial data from large-scale environments efficiently. Furthermore, recent decentralized robotic information gathering systems [2, 3] rely on strong connectivity with consensus-based cooperative coordination to map a spatial scalar field with confidence intervals. However, real-world conditions such as unstable networks, radio interference, and robot failures can easily disrupt this connectivity. In the event of disruptions, such systems become ineffective, especially when there is limited communication bandwidth available for transmitting information among a group of robots. One such scenario is underwater environments where traditional communication modalities such as radio frequency (RF) have only a limited reach, and acoustic-based communication devices have low ranges. Despite significant advancements in optical communication, their capabilities can be affected by water turbidity and line-of-sight disconnection events [4]. Moreover, in adversarial scenarios, RF communication can be intentionally jammed.

The coordination and communication among robots, along with their computational constraints, significantly impact system performance [5, 6]. For instance, centralized coordination methods [7, 8] suffer from the single point of failure problem. On the one hand, decentralized methods often necessitate the communication of entire past and anticipated trajectories or the coefficients of anticipated trajectories of each robot [9], which can result in computational burden. A state-of-the-art resource-efficient cooperative online field mapping method [5] requires periodical strong connectivity among all robots to enable distributed sparse Gaussian process regression. However, this assumption of perfect, stable connectivity is unrealistic in real-world scenarios where network conditions are unpredictable. The tight coupling between robots also limits flexibility and robustness, as even brief disruptions in connectivity can cause the complete breakdown of consensus-based coordination.

In contrast, decentralized systems are desired to operate robustly despite unreliable connectivity. If the network connectivity drops, robots continue to execute their local plans. Four key capabilities enable decentralized systems to gather information efficiently: (*i*) *Probabilistic models*: Each robot maintains a probabilistic model such as Gaussian process regression (GPR) to predict informa-

Manuscript received: May 29, 2024; Revised August 08, 2024; Accepted September 06, 2024.

This paper was recommended for publication by Editor Chao-Bo Yan upon evaluation of the Associate Editor and Reviewers' comments. This work is supported in part by NSF grants IIS-2024733, IIS-2331908, the Office of Naval Research grant N00014-23-1-2789, the U.S. Dept. of Homeland Security grant 23STSLA00016-01-00, the U.S. Department of Defense (DoD) grant 78170-RT-REP, and the FDEP grant INV31.

[1]A. A. R. Newaz is with the Department of Computer Science, University of New Orleans, New Orleans, LA 70148, USA (email: aredwann@uno.edu).

[2]P. Padrao, J. Fuentes, G. Govindarajan, and L. Bobadilla are with the School of Computing and Information Sciences, Florida International University, Miami, FL 33199, USA (email:{plope113@, jfuent099@, ggovinda@, bobadilla@cs.}fiu.edu).

[3]T. Alam is with the Department of Computer Science, Lamar University, Beaumont, TX 77710, USA (email: talam1@lamar.edu).

Digital Object Identifier (DOI): see top of this page.

tive locations. Well-calibrated uncertainty allows accurate prediction with limited samples. (*ii*) *Information-theoretic planning*: Robots plan trajectories locally to maximize information gain based on their uncertainty models. This drives efficient exploration without global oversight. (*iii*) *Reactive control*: Robots reactively avoid collisions with neighbors while exploring regions of high uncertainty. This maintains coordination locally. (*iv*) *Task allocation*: Robots asynchronously communicate to allocate tasks and avoid redundant work. This cooperation prevents conflicts without the need for long-term planning.

In this context, this paper proposes a Limited Communication Decentralized Robotic Information Gathering (LCD-RIG) system to improve the robustness and flexibility of cooperative online scalar field mapping. Each robot maintains a local representation of the target area using a quadtree data structure. The quadtree dynamically subdivides the area as the robot explores, effectively directing future actions toward unvisited regions and minimizing interference between robots. Communication between robots occurs when they come within a transmission range in a pairwise manner. This ad-hoc interaction enables efficient experience sharing and task coordination while avoiding redundant computations. The absence of periodic global communication [5] also empowers individual robots to operate with greater autonomy. This decentralized architecture improves robustness against network disruptions while facilitating efficient parallel exploration. To manage the inherent complexity of decentralized information gathering, we propose the implementation of behavior trees.

The key contributions of this paper are: (*i*) We introduce a novel LCD-RIG system that removes the assumption for periodical strong connectivity among robots [5]. Our method enables robots to share local experiences and coordinate tasks through pairwise communication when within a defined transmission range, reducing reliance on network connectivity, as explained in Section IV. (*ii*) By implementing the LCD-RIG using behavior trees, our system benefits from a modular architecture. This effectively handles the complexities associated with decentralized operation, such as constraints on time, memory, and communication, as shown in Table I. (*iii*) We provide theoretical proofs for the convergence of the proposed algorithm and establish bounds on its regret, as illustrated in Section V. (*iv*) Through physical experiments with autonomous surface vehicles (ASVs), we demonstrate the real-world applicability and performance of the LCD-RIG system in Section VI.

II. Related Work

In multi-robot information gathering, Gaussian mixtures [10] and Gaussian Processes (GP) [6, 11] enable efficient mapping, and their cycle consistency improves data association [12]. Collaboration schemes conserve battery life, as indicated in [13]. Swarm gradient algorithms explore unknown environments [14], although they fall short in providing precise mapping. However, existing methods overlook inconsistent collaboration under intermittent communication.

To speed up Gaussian process regression, sparse [15, 16] and parallel [17, 18] methods have been proposed. However, these methods entail a central node with access to global data. Distributed aggregation [19, 20] offers a way of reducing communication, and aggregation techniques that rely on average consensus are proposed in [21, 22] for decentralized networks. Applications of online distributed GP strategies for scalar field mapping are proposed in [23]. However, these strategies often lack bounded error guarantees or sparsity. Alternatively, local methods [6, 24] involve evaluating partial data to reduce computational burden and communication between robots.

Moreover, recent efforts have combined Bayesian optimization and motion planning [25, 26]. However, these approaches often assume knowledge of environments and dynamics, limiting their applicability in real-world scenarios [1, 27]. Previous reinforcement learning approaches [28] cannot handle partial observability or multi-robot systems. Methods handling partial observability face trade-offs between optimality, speed, and scalability [29, 30]. Therefore, key open problems remain in achieving reliable decentralized planning under incomplete knowledge and limited perception.

III. Preliminaries

We consider a team of n robots, $\mathcal{A}^1, \ldots, \mathcal{A}^n$, tasked with exploring an environmental phenomenon.

Definition 1 (Environmental phenomenon)**.** *The environmental phenomenon is defined as a spatial field with an unknown function ϕ such that $\phi : \mathcal{W} \to \mathbb{R}$. It is spatially distributed over a shared $2D$ workspace $\mathcal{W}$ such that $\mathcal{W} \subseteq \mathbb{R}^2$.*

Let $\mathcal{X}^i$ be the state space of a robot $\mathcal{A}^i$ such that $\mathcal{X}^i \subseteq \mathbb{R}^n$, $\mathcal{U}^i$ be the action space such that $\mathcal{U}^i \subseteq \mathbb{R}^u$, and $f^i : \mathcal{X}^i \times \mathcal{U}^i \to \mathcal{X}^i$ be the state transition function. A joint state space is defined as the states of all robots, $\mathcal{X} = \mathcal{X}^1 \times \mathcal{X}^2 \times \cdots \times \mathcal{X}^m$. Each robot is equipped with onboard sensors to measure ϕ in a point-wise manner. Therefore, to accomplish the above task, at time t, the i^{th} robot at a state x_t^i (i.e., p_t^i represents the robot's location in $\mathcal{W}$) takes an action u_t^i, obtains a noisy measurement y_t^i, and updates its state to a new state x_{t+1}^i based on its state transition function $f^i(x_t, u_t)$ in such a way that optimizes an objective function. Each robot must avoid inter-robot collisions to safely collect measurements $y_{1:t}^i$ from $\mathcal{W}$.

Definition 2 (Inter-robot collisions)**.** *Inter-robot collisions can be characterized in terms of a dynamic obstacle region. Formally, a dynamic obstacle region $\mathcal{X}_{obs}^{ij}$ is a subset $\mathcal{X}_{obs}^{ij} \subseteq \mathcal{X}^i$ that corresponds with the i^{th} robot state x_t^i in collision with the j^{th} robot state x_t^j as:*

$$\mathcal{X}_{obs}^{ij} = \left\{ x_t \in \mathcal{X}^i | \mathcal{A}^i(p_t^i) \cap \mathcal{A}^j(p_t^j) \neq \emptyset \right\}, \tag{1}$$

where $\mathcal{A}^i(p)$ denotes the set of points exclusively surrounding the rigid body of the robot $\mathcal{A}^i$ at a position p. Any

obstacle intersecting this set is considered to be colliding with the robot.

Each robot independently observes the field $\phi : \mathcal{W} \to \mathbb{R}$ and gets the measurements $y_t^i \in \mathbb{R}$ with zero-mean Gaussian noise e_t^i at time t.

Definition 3 (Observation model)**.** *The observation model for each robot i is given by*

$$y_t^i = \phi(f^i(x_t^i, u_t^i)) + e_t^i, \;\; e_t^i \sim \mathcal{N}(0, \sigma^2). \tag{2}$$

Each robot $\mathcal{A}^i$ has a local dataset D_t^i at time t, containing observations $\mathcal{X}_t^i = \{x_k^i\}_{k=1}^t$ and $\mathcal{Y}_t^i = \{y_k^i\}_{k=1}^t$ from the start up to time t. To effectively gather information, the robot $\mathcal{A}^i$ uses Gaussian process regression on its local training data $D_t^i = (\mathcal{X}_t^i, \mathcal{Y}_t^i)$ to predict the value of the latent function $\phi(x)$ at a new input $x \in \mathcal{X}$. This assumes that ϕ follows a Gaussian process. The regression model provides probabilistic predictions for $\phi(x)$ using the robot's current observations.

Definition 4 (Gaussian Process Regression)**.** *A Gaussian process regression (GPR) models an unknown function $\phi(x)$, which is a collection of random variables that follow a joint Gaussian distribution. This is specified by two components: the mean function $\mu(x)$, which defines the average value, and the covariance function or kernel $k(x, x')$, which specifies how the variables are related to each other.*

To foster collaboration, robots engage in opportunistic communication to assign tasks and prevent redundant efforts. This collaborative approach mitigates conflicts without the need for extensive long-term planning.

Definition 5 (Communication Network Model)**.** *The communication network at time t can be represented by a directed graph $G(t) = (V, E(t))$ with an edge set $E(t) \subseteq V \times V$. We consider that $(i, j) \in E(t)$ if and only if node i communicates to node j at time t and the distance between $\mathcal{A}^i$ and $\mathcal{A}^j$ is less than equal to a predefined threshold value δ. The matrix $Adj(t) := adj_{ij,t}$, $i, j = 1, \ldots, |V|$, represents the adjacency matrix of $G(t)$ where $adj_{ij,t} \neq 0$ if $(i, j) \in E(t)$, and $|V|$ is the cardinality of V.*

Once the GP model is trained with D_t^i, the i^{th} robot can make predictions about the environmental phenomenon at unobserved locations by sampling from the posterior distribution. Given mean μ_t^i and covariance κ_t^i functions of a GP, the posterior function is a multivariate Gaussian distribution $\mathcal{N}(\nu, \Sigma)$ with a mean vector ν and a covariance matrix Σ. The posterior mean ν is used to predict the measurements at unobserved locations, and the covariance matrix Σ is used to quantify uncertainties associated with these predictions.

Problem. *Given n GP models for a team of n robots in a shared workspace $\mathcal{W}$ with an unknown spatial function ϕ, find controls $u_t^{i\star}$ for $i = 1, \ldots, n$ at time t in such a way that they communicate compressed data with each other in pairs within a threshold distance δ to effectively sample new (informative) locations that minimize the predictive uncertainty as:*

$$u_t^{i\star} = \arg\min_{u_t^i} H\left(\Sigma_{f^i(x_t^i, u_t^i)}\right), \tag{3}$$

where H is the Gaussian posterior entropy calculated for the robot i.

IV. Distributed LCD-RIG System

Figure 1 depicts the architecture of an individual RIG, which employs a behavior tree-based structure. The planner component integrates information-theoretic planning with dynamic quadtree decomposition to facilitate effective exploration in unknown environments. The explorer component enables robots to track informative waypoints assigned by their respective planner components while contributing to a coordinated and safe exploration strategy. Additionally, the learner component trains a Gaussian Process Regression (GPR) model to predict essential sampling locations and rapidly gather valuable data. The individual RIG systems are interconnected through a parallel behavior tree node, allowing them to operate asynchronously for efficient environmental exploration.

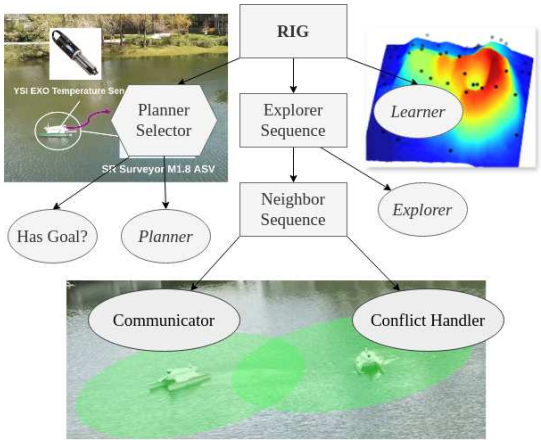


Fig. 1: The proposed LCD-RIG system comprises five components: the Planner Selector, Explorer Sequence, Neighbor Sequence, Communicator, and Conflict Handler. The Planner integrates information-theoretic sampling with dynamic quadtree decomposition to generate a plan, illustrated by the purple line. The Explorer guides robots to informative waypoints provided by the Planner, enabling coordinated strategies. The Learner employs an Attentive Kernel to train a Gaussian Process Regressor (GPR) with few samples (black dots), identifying valuable sampling locations on the colored map.

A. Planner Component

The planner component enables efficient multi-robot exploration of unknown environments through a combination of Gaussian Process Regression (GPR) and quadtree decomposition. Each robot employs an Active Learning –Attentive Kernel (AK) planner [5] that models the environment as a scalar field ϕ using GPR. The AK planner guides the $\mathcal{A}^i$ robot towards informative waypoints w_i by focusing on areas with maximum prediction uncertainty $\Sigma(w_i)$, where w_i is a waypoint in the environment. However, at each planning time t, a waypoint $w_i(t)$ is sampled

from the entire workspace $\mathcal{W}$, introducing redundancy in exploration when multiple robots share the environment.

To mitigate potential redundancy in exploration, an online quadtree decomposition mechanism is employed. The search space $\mathcal{W} \subseteq \mathbb{R}^2$ is subdivided into cells q using a quadtree structure, with each cell characterized by its visit density $d(q)$. Unexplored areas q_u have low $d(q)$, while densely visited areas q_v have high $d(q)$. Each robot $\mathcal{A}^i$ selects its target region $\mathcal{T}_i$ from its local quadtree Q_i as:

$$\mathcal{T}_i = \arg\max_{q \in Q_i} \text{Area}(q). \tag{4}$$

Note that Eqn. (4) is equivalent to $\mathcal{T}_i = \arg\min_{q \in Q_i} d(q)$, which enhances spatial diversity and minimizes exploration overlap.

The quadtree Q dynamically adapts as exploration progresses, updating target regions $\mathcal{T}_i$. Informative waypoints w_i are sampled from the target regions $\mathcal{T}_i$ for each robot $\mathcal{A}^i$. This ensures that all robots contribute equally to exploring the entire environment. By efficiently selecting informative waypoints, the planner component maximizes team efficiency and minimizes any redundant effort.

B. Explorer Component

To ensure smooth navigation between waypoints during multi-robot exploration, the planner employs Bézier curves to shape trajectories. These parametric curves are defined by control points that influence their shape and direction, allowing for the generation of smooth and efficient paths.

Given waypoints w_i and w_{i+1}, the planner generates an m-degree Bézier curve defined by $m+1$ control points such that $P_0, P_1, \ldots, P_m$. The curve is described by the following parametric equation:

$$B(\tau) = \sum_{i=0}^{m} \binom{m}{i} (1-\tau)^{m-i} \tau^i P_i, \quad \tau \in [0,1]. \tag{5}$$

Here, P_0 and P_m represent the start and end waypoints, respectively.

The explorer dynamically adjusts these control points based on the robot's current position and the distribution of waypoints. This adaptive approach allows the curve to respond to environmental changes and maintain optimal coverage throughout the exploration process. In our decentralized setting, the robot $\mathcal{A}^i$ does not account for the target region T_j of $\mathcal{A}^j$ robot, this can result in interference and uneven waypoint allocation among the team. To address these issues, the communication and conflict handler components enable reliable information exchange and collision avoidance when robots are operating nearby.

1) Communication component: When two robots $\mathcal{A}^i$ and $\mathcal{A}^j$ come within a certain communication radius δ, they exchange their experiences ξ_t^i, ξ_t^j, which are defined as the observed trajectories $\mathcal{X}_t^i, \mathcal{X}_t^j$ along with the field measurements $\mathcal{Y}_t^i$, $\mathcal{Y}_t^j$ obtained on those trajectories, i.e., $\xi_t^i = (\mathcal{X}_t^i, \mathcal{Y}_t^i)$ and $\xi_t^j = (\mathcal{X}_t^j, \mathcal{Y}_t^j)$.

Reliable asynchronous communication is crucial because the robots rely on lazy learning with GPs. Therefore, $\mathcal{A}^i$ robot's GP models improve online as new raw sensor data is progressively added to the training set D_t^i. When communicating with $\mathcal{A}^j$ robot, rather than sharing only the predicted information $\hat{\xi}_t^i$, $\mathcal{A}^i$ robot directly exchanges the raw sensor data ξ_t^i collected along its trajectory.

Given the local training data D_t^j for $\mathcal{A}^j$ robot at time t, communication between robots involves sharing ξ_t^j such that $\xi_t^j \subseteq D_t^j$ is a fixed sparse subset of local experiences sampled from D_t^j to GPR models which improves the uncertainty estimation. The sparse subset ξ_t^j is sampled from D_t^j by selecting K waypoints $\{w_1, ..., w_K\}$ such that:

$$\|w_i - w_{i+1}\| \approx \frac{L}{K-1}, \tag{6}$$

where L is the length of the robot's trajectory, ensuring even coverage. Here, K is the number of points selected to ensure sufficient coverage while limiting data volume. Each robot selects a subset of its raw data points spaced evenly across its trajectory based on the sensing radius. This sparse sampling provides sufficient coverage for modeling while limiting the data volume.

2) Conflict handler component: The conflict handler component addresses the issue of overlapping intermediate goal regions and potential collisions between robots. This is achieved through a velocity obstacle method that refines the robot $\mathcal{A}^i$'s intermediate waypoint location w_i within the workspace $\mathcal{W}$. Let $\text{Neigh}^i(t) = \{\mathcal{A}^j : \|p_t^i - p_t^j\| \leq \delta\}$ be the set of neighboring robots within the communication radius of δ. The avoidance direction is computed based on the states of robots in $\text{Neigh}^i(t)$ in three steps. First, an avoidance search space $\Gamma_i(t)$ is defined, centered at p_t^i, with a safe distance r_s as:

$$\Gamma_i(t) = \{p \in \mathbb{R}^2 : \|p - p_t^i\|_\infty \leq r_s\}. \tag{7}$$

Then $\Gamma_i(t)$ is adjusted to ensure it's fully contained within $\mathcal{W}$ as:

$$\Gamma_i'(t) = \{p \in \mathbb{R}^2 : p = w + \mathbf{\Delta},\ w \in \mathcal{W}\}, \tag{8}$$

where $\mathbf{\Delta} = (\Delta_x, \Delta_y)$ is the minimum shift required to ensure $\Gamma_i'(t) \subseteq \mathcal{W}$. Finally, the new intermediate waypoint location w_i' is randomly sampled from the adjusted search space $\Gamma_i'(t)$, subject to the safety constraint:

$$w_i' \in \{p \in \Gamma_i'(t) : \|p - p_t^j\| \geq r_s,\ \forall \mathcal{A}^j \in \text{Neigh}^i(t)\}. \tag{9}$$

C. Learner Component

Each robot utilizes the AKGPR model [1] to learn from the training data using Gaussian process regression with the AK. The key ideas behind AK are: (i) Length scale selection: At each input x, AK computes a weighting vector $\rho(x;\theta)$ that assigns weights to a set of n_K base kernels with different length scales ℓ_m. This allows x to control the size of the surrounding neighborhood which will be relevant for the local prediction. Mathematically, the GP prior to length scale selection is

$$\phi(x) = \sum_{m=1}^{n_K} \rho_m(x;\theta) g_m(x;\ell_m), \tag{10}$$

where $g_m(x;\ell_m) \sim \mathcal{GP}(0, k(x,x'|\ell_m))$. Here $\omega_m(x)$ are normalized weights learned from the data x. Each $g_m(x;\ell_m)$ is drawn from a GP with a kernel $k(x,x'|\ell_m)$ that uses a length scale ℓ_m. (ii) Instance selection: AK also computes a membership vector $\mathbf{z}(x;\phi)$ at each input x. The dot product $\mathbf{z}(x;\phi)^\top \mathbf{z}(x';\phi)$ between two inputs x and x' defines the visibility between them; high dot product means high visibility. The kernel value is masked out for low-visibility inputs. This gives the kernel:

$$k_m([x,\mathbf{z}(x;\phi)],[x',\mathbf{z}(x';\phi)]) = \mathbf{z}(x;\phi)^\top \mathbf{z}(x';\phi) k_{\text{base}}(x,x';\ell_m). \tag{11}$$

Here, k_{base} is a base kernel parametrized by ℓ_m, similar to a radial basis function (RBF) kernel. The membership vectors $\mathbf{z}(x;\phi)$ allow the selection of relevant training instances. Combining the two ideas and defining $\ell = (\ell_1,\ldots,\ell_{n_K})$, $\mathbf{K}(x,x';\phi,\ell) = \text{diag}(k_1(x,x';\phi,\ell),\ldots,k_m(x,x';\phi,\ell))$, $\mathbf{z} = \mathbf{z}(x;\phi)$, $\mathbf{z}' = \mathbf{z}'(x';\phi)$, $\rho = (x;\phi)$, and $\rho' = (x';\phi)$ gives the Attentive Kernel as:

$$\text{ak}(x,x';\theta,\phi,\alpha,\ell) = \alpha(\mathbf{z}^\top\mathbf{z}')(\rho^\top \mathbf{K}\rho'). \tag{12}$$

The training optimizes the parameters θ and ϕ along with the kernel hyperparameters α and ℓ by maximizing the GP marginal likelihood on training data. This enables AKGPR to learn varying length scales and instance relevance functions directly from the data.

To facilitate real-time training when accumulating data from multiple robots, compressing the training set becomes crucial [5]. In our setting, we compressed our training set to a subset of fixed size K. At each timestamp, each robot maintains a training set S_t^i and then selects a subset $S_{t,k}^i$ from its aggregated data. This subset selection ensures that the measurements in $S_{t,k}^i$ are sufficiently novel, thus avoiding redundancy while still being effective in predicting the remaining points in $S_t^i \setminus S_{t,k}^i$.

Achieving this involves minimizing the conditional entropy $H(S_t^i \setminus S_{t,k}^i | S_{t,k})$, which is equivalent to maximizing the entropy $H(S_{t,k}^i)$. However, finding the $S_{t,k}^i$ that maximizes $H(S_{t,k}^i)$ is an NP-hard problem. Therefore, we implement a greedy approximation for rapid compression [6]. This starts with a temporary data set $S_0 = \varnothing$ and iteratively adds points from the unlabeled data S_t^i that results in the largest increase in $H(S_{t,j}^i)$ at iteration j.

The chosen point s_j^i to be added at step j, i.e., $S_{t,j}^i = S_{t,j-1}^i \cup \{s_j^i\}$, minimizes the conditional entropy $H(s_j^i | S_{t,j-1}^i)$. This iterative process selectively retains the most uncertain points in the training set at each step to update the local Gaussian process prediction $\hat{\phi}_t^i$ by maximizing information gain from new data $S_{t,K}^i$. Algorithm 1 summarizes the LCD-RIG execution for each robot.

Algorithm 1 LCD-RIG Algorithm for Robot i

1: Randomly initiate $\mathcal{A}^i,\ldots,\mathcal{A}^n$ robots with pilot data $D_0^i = \varnothing$ and $\xi_0^i = \varnothing$
2: **for** $t = 1,2,\ldots,T$ **do**
3: **/* Sensing and Marking Visited Locations */**
4: Sample independent training point (x_t^i, y_t^i)
5: $\xi_t^i = \xi_{t-1}^i \cup \{(x_t^i, y_t^i)\}$,
6: Update local quadtree $Q^i(t)$ with each x_t^i in ξ_t^i
7: **/* Communication and Conflict Handling */**
8: **if** Distance$(\mathcal{A}^i, \mathcal{A}^j) \le \delta$ **then**
9: Receive ξ_t^j from neighbors j
10: **if** Distance$(\mathcal{A}^i, \mathcal{A}^j) \le r_s$ **then**
11: Recompute intermediate waypoint w_i'
12: **end if**
13: **end if**
14: **/* Distributed Data Compression and Update */**
15: Update local quadtree $Q^i(t)$ with each ξ_t^j in Neigh$^i(t)$.
16: Aggregate data points $S_t^i \leftarrow D_{t-1}^i \cup \xi_t^j$
17: **if** $|S_t^i| > K$ **then**
18: $S_{t,0}^i = \varnothing$
19: **for** $k = 1$ to K **do**
20: **for** $s \in S_t^i \setminus S_{t,k-1}^i$ **do**
21: $H(s|S_{t,k-1}^i) \leftarrow \frac{1}{2}\ln\left(2\pi e \sigma^2_{s|S_{t,j}^i}\right)$ See [6]
22: **end for**
23: $s_{t,k}^i \leftarrow \arg\max_{s\in S_t^i} H(s|S_{t,k-1}^i)$
24: $S_{t,k}^i \leftarrow S_{t,k-1}^i \cup \{s_{t,k}^i\}$
25: **end for**
26: **end if**
27: Update training set $D_t^i \leftarrow D_{t-1}^i \cup S_{t,K}^i$
28: Update local Gaussian process predictions:

$$\hat{\phi}_t^i = \mathbb{P}(\phi(x^*)|D_t^i, x^*) \sim \mathcal{N}(\mu_t^i, \Sigma_t^i)$$

29: **end for**

V. Algorithm Analysis

This section demonstrates that the LCD-RIG algorithm guarantees the coverage of the entire space with the finest sensing resolution via a team of decentralized robots as the largest unexplored rectangle in the quadtree decreases. Furthermore, by bounding the cumulative regret, the LCD-RIG algorithm guarantees that the robots' exploration strategies are nearly optimal, minimizing the inefficiencies in their paths and actions.

Theorem 1. *An LCD-RIG system that selects the largest undivided rectangle for recursive quadtree decomposition (Algorithm 1, line 6) of a target space $\mathcal{W}$ will eventually terminate after covering $\mathcal{W}$ with the finest resolution. Specifically, after t steps, the area of the biggest unexplored rectangle in a quadtree is bounded by*

$$\max_{q\in Q(t)} \text{Area}(q) \le \frac{\text{Area}(\mathcal{W})}{4^{\lfloor \log_4(3t+1)\rfloor}}. \tag{13}$$

Proof. We start by proving that there are $3t+1$ rectangles in each quadtree $Q(t)$ i.e. $|Q(t)| = 3t+1$. In this case, we proceed by induction; for the base case $t = 0$, $Q(0) = \{\mathcal{W}\}$, then $|Q(0)| = 1$. For the induction step $t = k$, we have $|Q(k)|| = 3k+1$. When computing the quadtree $Q(t+1)$, the largest rectangle $q' \in Q(t)$ is replaced with four child nodes $q_{q'}^1,\ldots,q_{q'}^4$, obtained by dividing q' into four rectangles. Thus, $Q(t+1) = (Q(k)\setminus\{q'\}) \cup \{q_{q'}^1,\ldots,q_{q'}^4\}$ and $|Q(t+1)| = |Q(k)| + 4 - 1 = 3(k+1)+1$, thereby completing this section of the proof.

The algorithm always selects the largest rectangle to be divided into four equal-area subrectangles, and the area of each rectangle has the form $\frac{\text{Area}(\mathcal{W})}{4^a}$ for a certain integer a. Here, we distinguish two cases (i) The number

of rectangles is a power of 4, i.e., $3t+1 = 4^a$. In this case, all rectangles have the same area $\frac{\text{Area}(\mathcal{W})}{4^a} = \frac{\text{Area}(\mathcal{W})}{4^{\log_4(3t+1)}} = \frac{\text{Area}(\mathcal{W})}{4^{\lfloor \log_4(3t+1) \rfloor}}$. (ii) The number of rectangles is not a power of 4; therefore, $3t+1 = 4^{a-1}+b$ where 4^{a-1} is the largest power of 4 less than $3t+1$ and $b > 0$. In this case, there are rectangles whose area is $\frac{\text{Area}(\mathcal{W})}{4^a}$ and there is at least one larger rectangle whose area is $\frac{\text{Area}(\mathcal{W})}{4^{a-1}}$, bounding the area of the remaining ones. Furthermore, it can be shown that $\lfloor \log_4(3t+1) \rfloor = a-1$ for this case. Therefore, in both cases, we have bounded the area of the largest rectangle by (13). □

In our earlier work [25], we have shown that under the assumptions of a discrete observation set D and a known field ϕ, the regret between the optimal locational utility and the utility efficiently obtained through the proposed algorithm is sub-linear in T.

Lemma 1 ([25]). *Let r_t be the regret at stage $t \in [1, T]$. If the field ϕ is a Lipschitz realization of a GP with mean $\mu_i(t)$ and variance $\sigma_i^2(t)$, an asynchronous distributed algorithm ensures the cumulative regret of the i-th robot R_T such that $R_T := \int_1^T r_t \, dt$ is bounded as:*

$$R_T \leq \int_1^T c\sqrt{\beta(t)}\sigma_i(t-1)x_i(t) \exp\left(\frac{2\epsilon^2 L^2}{\sigma_i(t-1)^2}\right) dt, \quad (14)$$

where $\beta(t) = 2\log(|D_t^i|\pi_t/\partial)$, $c, \delta, \epsilon > 0$, $\sum_{t\geq 1} \pi_t^{-1} = 1, \pi_t > 0$ and $x_i(t)$ is the continuous realization of x_t^i.

Instead of circular penalization regions, the LCD-RIG algorithm utilizes rectangular regions utilizing the quadtree data structure to reduce regrets by avoiding redundant sampling.

Theorem 2. *The cumulative regret for the i-th robot exploring a workspace $\mathcal{W}$ using the proposed LCD-RIG algorithm scales as $O(\text{Area}(\mathcal{W})R_T)$ in the worst case with a uniform scalar field distribution. However, for typical non-uniform scalar fields exhibiting spatial locality, the cumulative regret scales as $O(\sqrt{\text{Area}(\mathcal{W})}R_T)$ on average.*

Proof Sketch. The worst-case analysis considers a scenario where the ϕ is uniformly distributed across the task extent. In such a scenario, a perfect recursive z-order curve is identified as the optimal decomposition. If the robot employs a row-by-row mapping strategy, it tends to select increasingly large rectangles, deviating significantly from the optimal path. This deviation leads to a regret that scales linearly with the grid size, resulting in $O(\text{Area}(\mathcal{W})R_T)$ regret.

However, typical scalar fields exhibit non-constant features identifying local regions, with informative regions in unexplored areas. This is justified by the lazy learning nature of GPR used here. The GPR learner does not have an explicit predictive model but rather uses all training data directly for predictions. Thus, its prediction uncertainty increases for samples far from the training set. Therefore, selecting the largest rectangle during the planning phase tends to capture significant portions of the remaining area, reducing suboptimal choices. This planning behavior results in a cumulative regret that grows at a slower rate, approximately $O(\sqrt{\text{Area}(\mathcal{W})}R_T)$, as the grid size increases. □

In terms of the algorithm's complexity, the update of local mapping for each robot yields $\mathcal{O}(|D|^2)$ time [1] for a 2D workspace, where $|D|$ training points are utilized. In this method, the training points belong to the quadtree rectangle in which the robot is located. If K samples are taken at a given timestamp t, there will be $\mathcal{O}(tK)$ samples divided among the $3K+1$ rectangles forming the quadtree. Assuming uniform distribution for sufficiently large t, the number of samples in a given rectangle is proportional to its area, i.e., $\mathcal{O}(\frac{tK}{4^{\lfloor \log_4(3t+1) \rfloor}}) = \mathcal{O}(\frac{tK}{3t+1}) = \mathcal{O}(K)$. The last equality arises from the fact that $\lim_{t\to\infty} \mathcal{O}(\frac{tK}{3t+1}) = \mathcal{O}(K)$. Therefore, this results in $\mathcal{O}(K^2)$ computation complexity and $\mathcal{O}(|D|)$ memory complexity associated with quadtree storage.

Regarding communication, the proposed algorithm differs from the one presented in [5] by pairwise communicating only their experiences ξ when two robots are within a fixed distance, rather than at every time step. This leads to a constant-time communication complexity per time step per robot. Consequently, the communication structure is simplified, mitigating congestion issues, especially in scenarios involving many agents or limited communication channels.

The proposed methods demonstrate improved communication efficiency compared to previous methods [16, 6, 24, 5]. In our method, robots communicate pairwise to exchange compressed experiences of size K, resulting in a communication complexity of $\mathcal{O}(K)$. This is more efficient than earlier methods that had a communication complexity of $\mathcal{O}(|E|K)$, where $n-1 \leq |E| \leq n(n-1)/2$. It's important to note that the number of robots in the proposed ad-hoc network is considered constant.

VI. Experimental Results

We have developed our implementation based on the existing AK planner code[1] provided by [1] and shared our LCD-RIG code on GitHub[2].

We evaluate the LCD-RIG system's performance in the Florida International University (FIU) MMC Lake and three digital elevation maps from the NASA SRTM [1] simulated environments. A team of ASVs, following Dubins' car kinematic models (max velocity 1 m/s, control frequency 10 Hz), collects data within a 20×20 meter workspace to model unknown environments. Each ASV has a single-beam range sensor (3 Hz, unit Gaussian white noise) and aims to minimize elevation prediction error given 3500 samples.

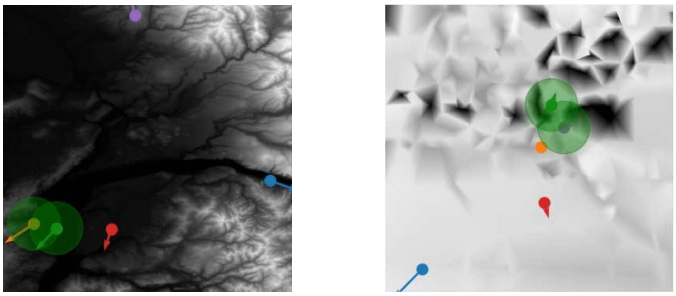
A high-performing team should achieve a lower prediction error by the end of the task and demonstrate a faster reduction in the prediction error curve as well as a higher area coverage rate. This is accomplished by prioritizing

[1]https://github.com/Weizhe-Chen/attentive_kernels
[2]https://github.com/RedwanNewaz/LCD_RIG

TABLE I: Algorithm Complexity Comparison

Complexity	Full GPR [31]	RGPR [16]	DGPR [21, 22]	DRGPR [23]	Local SGPR [6, 24]	CBDGPR [5]	This Letter
Computation	$\mathcal{O}(n^4\|D\|^4)$	$\mathcal{O}(n^3\|D\|^3)$	$\mathcal{O}(\|D\|^4)$	$\mathcal{O}(\|D\|^3)$	$\mathcal{O}(n^3\|D\|\|S\|^2)$	$\mathcal{O}(n\|D\|(\|S\|^2+\|S\|^3))$	$\mathcal{O}(n(K^2+\|D\|^2))$
Memory	$\mathcal{O}(n^2\|D\|^2)$	$\mathcal{O}(n^2\|D\|^2)$	$\mathcal{O}(n\|D\|^2)$	$\mathcal{O}(n\|D\|^2)$	$\mathcal{O}(n^3\|S\|^2)$	$\mathcal{O}(n\|S\|^2)$	$\mathcal{O}(n\|D\|)$
Communication	$\mathcal{O}(n^2\|D\|)$	$\mathcal{O}(n^2\|D\|)$	$\mathcal{O}(\|E\|K)$	$\mathcal{O}(\|E\|K)$	$\mathcal{O}(n^2\|S\|)$	$\mathcal{O}(\|E\|K)$	$\mathcal{O}(K)$

visits to the most informative locations that have not been explored by the team. However, global communication does not exist, and robots can only communicate within a fixed communication radius, as illustrated in Fig. 2 with green circles.



(a) N45W123 Dataset [1]. (b) FIU-MMC Campus Lake.

Fig. 2: Two of the environments used for simulation purposes. The green balls around the robots indicate potential collation and the opportunity to exchange information.

We conducted 15 simulation trials for each environment, varying the team size between 3, 4, and 5 robots. The results in Fig. 3 show that a higher number of robots leads to the broader spatial coverage. However, Fig. 4, shows that in certain simulated environments, a smaller team can achieve a lower overall root mean square error (RMSE) for distributed predictions at test points, primarily due to congestion effects from higher robot density. As the number of robots increases, a trade-off arises between maximizing area coverage and maintaining inter-robot safety. While aiming to maximize high-entropy coverage, robots may diverge in their decisions or states, leading to compromised coverage. Additionally, asynchronous communication in larger teams impacts the RMSE.

Table I contrasts our method with previous methods. Our asynchronous communication method simplifies communication complexity, depending solely on the number of deployed robots. Moreover, the informative sampling (Section IV-C) significantly reduces the computational burden for GP training, while memory usage remains comparable to other methods. This method relies on periodic communication among robots. Two extreme cases illustrate its operation: (*i*) When robots frequently target similar locations, asynchronous communication occurs almost continuously, resembling synchronous communication. (*ii*) Rare communication results in robots resorting to individual GPs, emphasizing the importance of eventual communication. Strategies to address this include adjusting robot count and communication radius δ.

We conducted field experiments using three SeaRobotics

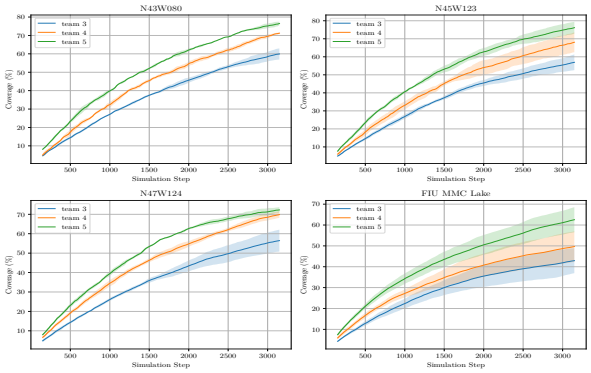


Fig. 3: Coverage trends across simulation stages for varying team sizes.

Surveyor ASVs to sample and map water temperature within a 28 m × 28 m workspace at FIU's MMC Campus lake. Each ASV, shown in Fig. 5a, was equipped with a YSI EXO2 sonde to measure various water quality parameters, but our focus was on mapping temperature using the LCD-RIG algorithm. The ASVs were localized by combining GPS and IMU data, and water temperature was measured at a sample rate of 1 Hz. Throughout the experiment, the LCD-RIG algorithm effectively coordinated the motion of the ASVs, enabling them to efficiently sample and map the temperature field while actively avoiding inter-vehicle collisions. The algorithm's performance is validated by the successful generation of an online temperature map, as depicted in Fig. 5c.

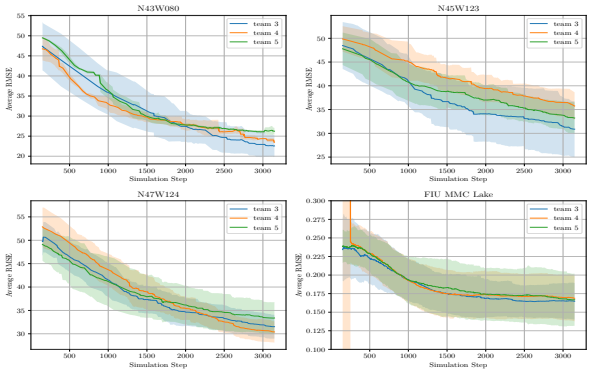
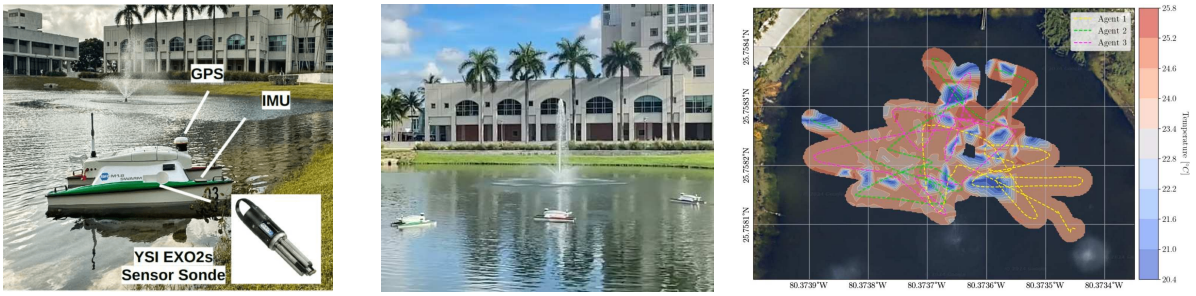


Fig. 4: Average RMSE trends across simulation stages for varying team sizes.

VII. Conclusion

We proposed the Limited Communication Decentralized Robot Information Gathering (LCD-RIG) system, where multiple robots collect information in a distributed manner to map online scalar fields under the constraints of asynchronous and limited communication. Comprehensive



(a) ASV and sensor sonde (b) Robots within the workspace (c) Final snapshot: temperature map

Fig. 5: Field Experiment at Florida International University's MMC Campus.

simulations across various non-stationary environments demonstrated the scalability of the LCD-RIG system, with distributed area coverage increasing and RMSE decreasing over time as more data was gathered. Theoretical analysis proved the convergence and bounded regret of the proposed algorithm, along with low computational, memory, and communication costs compared to existing methods. Our physical experiments in a lake environment validated the real-world applicability of the system. Our future endeavors will concentrate on developing low-cost predictive models to incrementally learn online scalar fields.

References

[1] W. Chen, R. Khardon, and L. Liu, "AK: Attentive kernel for information gathering," in *Proc. Robot.: Sci. Syst.*, 2022.

[2] S. Zhu, X. Chen, X. Liu, G. Zhang, and P. Tian, "Recent progress in and perspectives of underwater wireless optical communication," *Prog. Quant. Elect.*, vol. 73, no. 100274, 2020.

[3] A. Zolich, D. Palma, K. Kansanen, K. Fjørtoft, J. Sousa, K. H. Johansson, Y. Jiang, H. Dong, and T. A. Johansen, "Survey on communication and networks for autonomous marine systems," *J. Intell. Robot. Syst.*, vol. 95, no. 3, pp. 789–813, 2019.

[4] M. A. A. Ali, "Investigation of multiple input–single output technique for wireless optical communication system under coastal water," *Opt. Quant. Elect.*, vol. 52, no. 9, pp. 1–17, 2020.

[5] T. Ding, R. Zheng, S. Zhang, and M. Liu, "Resource-efficient cooperative online scalar field mapping via distributed sparse Gaussian process regression," *IEEE Robot. Autom. Lett.*, vol. 9, no. 3, pp. 2295 – 2302, 2024.

[6] M. E. Kepler and D. J. Stilwell, "An approach to reduce communication for multi-agent mapping applications," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pp. 4814–4820, 2020.

[7] A. Benevento, M. Santos, G. Notarstefano, K. Paynabar, M. Bloch, and M. Egerstedt, "Multi-robot coordination for estimation and coverage of unknown spatial fields," in *Proc. IEEE Int. Conf. Robot. Autom.*, pp. 7740–7746, 2020.

[8] M. Santos, U. Madhushani, A. Benevento, and N. E. Leonard, "Multi-robot learning and coverage of unknown spatial fields," in *Proc. IEEE Int. Symp. Multi-Robot Multi-Agent Syst.*, pp. 137–145, 2021.

[9] I. Abraham and T. D. Murphey, "Decentralized ergodic control: distribution-driven sensing and exploration for multiagent systems," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 2987–2994, 2018.

[10] M. Corah, C. O'Meadhra, K. Goel, and N. Michael, "Communication-efficient planning and mapping for multi-robot exploration in large environments," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1715–1721, 2019.

[11] A. Asgharivaskasi and N. Atanasov, "Distributed optimization with consensus constraint for multi-robot semantic octree mapping," in *Work. Coll. Percept. and Learn.*, 2023.

[12] K. Cesare, R. Skeele, S.-H. Yoo, Y. Zhang, and G. Hollinger, "Multi-UAV exploration with limited communication and battery," in *Proc. IEEE Int. Conf. Robot. Autom.*, pp. 2230–2235, 2015.

[13] Y. Tian, K. Liu, K. Ok, L. Tran, D. Allen, N. Roy, and J. P. How, "Search and rescue under the forest canopy using multiple UAVs," *Int. J. Robot. Res.*, vol. 39, no. 10-11, pp. 1201–1221, 2020.

[14] K. McGuire, C. De Wagter, K. Tuyls, H. Kappen, and G. C. de Croon, "Minimal navigation solution for a swarm of tiny flying robots to explore an unknown environment," *Sci. Robot.*, vol. 4, no. 35, p. eaaw9710, 2019.

[15] D. Burt, C. E. Rasmussen, and M. Van Der Wilk, "Rates of convergence for sparse variational Gaussian process regression," in *Proc. Int. Conf. Mach. Learn.*, pp. 862–871, 2019.

[16] L. Csato and M. Opper, "Sparse on-line Gaussian processes," *Neural Comp.*, vol. 14, no. 3, pp. 641–668, 2002.

[17] M. Deisenroth and J. W. Ng, "Distributed Gaussian processes," in *Proc. Int. Conf. Mach. Learn.*, pp. 1481–1490, 2015.

[18] H. Liu, J. Cai, Y. Wang, and Y. S. Ong, "Generalized robust bayesian committee machine for large-scale Gaussian process regression," in *Proc. Int. Conf. Mach. Learn.*, pp. 3131–3140, 2018.

[19] G. P. Kontoudis and D. J. Stilwell, "Decentralized nested Gaussian processes for multi-robot systems," in *Proc. IEEE Int. Conf. Robot. Autom.*, pp. 8881–8887, 2021.

[20] A. Lederer, Z. Yang, J. Jiao, and S. Hirche, "Cooperative control of uncertain multi-agent systems via distributed Gaussian processes," *IEEE Trans. Autom. Control*, 2022.

[21] Z. Yuan and M. Zhu, "Communication-aware distributed Gaussian process regression algorithms for real-time machine learning," in *Proc. America Control Conf.*, pp. 2197–2202, 2020.

[22] A. Lederer, Z. Yang, J. Jiao, and S. Hirche, "Cooperative control of uncertain multiagent systems via distributed Gaussian processes," *IEEE Trans. Autom. Control*, vol. 68, pp. 3091–3098, May 2023.

[23] 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 Robot. Autom. Lett.*, vol. 5, no. 4, pp. 5905–5912, 2020.

[24] E. Zobeidi, A. Koppel, and N. Atanasov, "Dense incremental metric-semantic mapping for multiagent systems via sparse Gaussian process regression," *IEEE Trans. Robot.*, vol. 38, pp. 3133–3153, October 2022.

[25] A. A. R. Newaz, M. Alsayegh, T. Alam, and L. Bobadilla, "Decentralized multi-robot information gathering from unknown spatial fields," *IEEE Robot. Autom. Lett.*, vol. 8, no. 5, pp. 3070–3077, 2023.

[26] B. Ru, A. Alvi, V. Nguyen, M. A. Osborne, and S. Roberts, "Bayesian optimisation over multiple continuous and categorical inputs," in *Proc. Int. Conf. Mach. Learn.*, pp. 8276–8285, 2020.

[27] W. Luo and K. Sycara, "Adaptive sampling and online learning in multi-robot sensor coverage with mixture of Gaussian processes," in *Proc. IEEE Int. Conf. Robot. Autom.*, pp. 6359–6364, 2018.

[28] G. Notomista, C. Pacchierotti, and P. R. Giordano, "Multi-robot persistent environmental monitoring based on constraint-driven execution of learned robot tasks," in *Proc. IEEE Int. Conf. Robot. Autom.*, pp. 6853–6859, 2022.

[29] M. Lauri, J. Pajarinen, and J. Peters, "Information gathering in decentralized POMDPs by policy graph improvement," in *Proc. Int. Conf. Auton. Agents Multi-Agent Syst.*, pp. 1143–1151, 2019.

[30] A. A. R. Newaz, T. Alam, G. M. Reis, L. Bobadilla, and R. N. Smith, "Long-term autonomy for AUVs operating under uncertainties in dynamic marine environments," *IEEE Robot.*

Autom. Lett., vol. 6, no. 4, pp. 6313–6320, 2021.
[31] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2005.