Multi-Robot Informative Path Planning from Regression with Sparse Gaussian Processes

Kalvik Jakkala[1] and Srinivas Akella[1]

Abstract— **This paper addresses multi-robot informative path planning (IPP) for environmental monitoring. The problem involves determining informative regions in the environment that should be visited by robots in order to gather the most amount of information about the environment. We propose an efficient sparse Gaussian process-based approach that uses gradient descent to optimize paths in continuous environments. Our approach efficiently scales to both spatially and spatio-temporally correlated environments. Moreover, our approach can simultaneously optimize the informative paths while accounting for routing constraints, such as a distance budget and limits on the robot's velocity and acceleration. Our approach can be used for IPP with both discrete and continuous sensing robots, with point and non-point field-of-view sensing shapes, and for both single and multi-robot IPP. We demonstrate that the proposed approach is fast and accurate on real-world data.**

I. Introduction

Environmental monitoring problems require estimating the current state of phenomena, such as temperature, precipitation, ozone concentration, soil chemistry, ocean salinity, and fugitive gas density ([1], [2], [3], [4]). These problems are closely related to the informative path planning (IPP) problem ([1], [5]) since it is often the case that we have limited resources and, therefore, must strategically determine the regions from which to collect data and the order in which to visit the regions to efficiently and accurately estimate the state of the environment.

The IPP problem has been studied in numerous scenarios: [6] developed IPP for persistent ocean monitoring with underwater gliders, [7] studied IPP for information gathering on three-dimensional mesh surfaces for inspection tasks, [4] presented an IPP approach for localizing gas sources in oil fields, and [8] used IPP for active learning in aerial semantic mapping.

Most IPP approaches implicitly assume that the environment is correlated ([1], [6], [5], [9], [10], [2], [11]). Similarly, we consider IPP problems for environments that are correlated either spatially or spatio-temporally and present an efficient approach that leverages such correlations.

Existing discrete optimization based IPP methods have discretization requirements that limit them to relatively small problems ([1], [5], [2]), making them infeasible for large spatio-temporal environments. Additionally, incorporating routing constraints, such as a distance budget and limits on the robot's velocity and acceleration, significantly increase the problem size when using discrete optimization.

Furthermore, modeling informative paths in continuous domains with potentially continuous sensing robots is a non-trivial problem. The problem is usually addressed using optimization methods such as rapidly-exploring random trees (RRT), genetic algorithms, or Bayesian optimization ([9], [10], [11]). These methods select sensing locations that maximize mutual information (MI) computed using Gaussian processes [12]. But some of these optimization methods are computationally expensive and rely on computing MI, which is also expensive ($\mathcal{O}(n^3)$, where n is the discretized environment size). A few approaches have even considered multi-robot IPP ([13], [10]) but they are also inherently limited by the scalability issues of prior IPP approaches.

Motivated by the above limitations of prior IPP approaches, we present a method that can efficiently generate both discrete and continuous sensing paths, accommodate constraints such as a distance budget and velocity limits, handle point sensors and non-point FoV sensors, and handle both single and multi-robot IPP problems. Our approach leverages gradient descent optimizable sparse Gaussian processes to solve the IPP problem, making it significantly faster compared to prior approaches and scalable to large IPP problems.

II. Multi-Robot Informative Path Planning

We consider a spatially (or spatiotemporally) correlated stochastic process over an environment $\mathcal{V} \subseteq \mathbb{R}^d$ representing a phenomenon such as temperature. We have r robots and must find the set $\mathcal{P}$ of r paths, one for each robot, so that the data from the phenomenon $y_i \in \mathbb{R}$ collected at these locations is sufficient to accurately estimate the phenomenon at every location in the environment. We use the root-mean-square error (RMSE) of the estimates as the measure of accuracy. Since we cannot directly minimize the RMSE, we formulate this problem as one where we want to find the paths $\mathcal{P}$ that maximize the amount of information I. Here I is any function that is a good proxy for accuracy and can be computed without the ground truth labels. Moreover, we also consider constraints $\mathbf{C}$ such as distance budget and velocity limits on the paths:

$$\begin{aligned} \mathcal{P}^* = & \underset{\{\mathcal{P}_i \in \psi, i=1,\ldots,r\}}{\arg\max} \; I(\cup_{i=1}^{r} \text{SAMPLE}(\mathcal{P}_i)), \\ & \text{s.t. Constraint}(\mathcal{P}_{i=1,\ldots,r}) \leq \mathbf{C} \end{aligned} \tag{1}$$

This work was supported in part by NSF under Award Number IIP-1919233. [1]The authors are with the Department of Computer Science, University of North Carolina at Charlotte, Charlotte, NC, USA. Email:{kjakkala, sakella}@charlotte.edu.

Here ψ is the space of paths contained within the environment $\mathcal{V}$, and the SAMPLE function returns the sensing points along a path $\mathcal{P}_i$. When considering discrete sensing robots, each path is constrained to have only s sensing locations. In a continuous sensing model, the SAMPLE function returns all the points along the path, which are used to compute the integral of the information collected along the path. In addition, we also consider point sensors such as temperature probes, and non-point sensors that can have any field-of-view (FoV) shape such as a thermal vision camera with a rectangular FoV.

III. Related Work

The Informative Path Planning (IPP) Problem is known to be NP-hard [14]. Therefore, only suboptimal solutions can be found for most real-world problems. Numerous IPP methods select utility functions that are submodular ([15], [13], [16], [5]). Submodular functions have a diminishing returns property that can be leveraged to get good approximation guarantees even when optimized using greedy algorithms.

Many IPP approaches use mutual information (MI), an information metric that is submodular [12], as the optimization objective. The methods compute MI using Gaussian processes with known kernel parameters. But MI requires one to discretize the environment, thereby limiting the precision with which the sensing locations can be selected. Also, MI is computationally expensive ($\mathcal{O}(n^3)$, where n is the number of locations in the discretized environment). Singh et al. [17] proposed a recursive-greedy algorithm that maximized MI. The approach addressed both single and multi-robot IPP. Ma et al. [2] solved the IPP problem by maximizing MI using dynamic programming and used an online variant of sparse Gaussian processes for efficiently learning the model parameters. Bottarelli et al. [18] developed active learning-based IPP algorithms with a complexity of $\mathcal{O}(|D|^5)$, where D is the discretized data collection space.

Hollinger and Sukhatme [9] presented IPP algorithms for continuous spaces that maximized MI using rapidly-exploring random trees (RRT) and derived asymptotically optimal guarantees. Miller et al. [19] addressed continuous-space IPP with known utility functions using an ergodic control algorithm. Hitz et al. [10] developed an IPP approach that could simultaneously optimize the sensing locations in continuous spaces by optimizing any utility function. The approach used a B-spline to parametrize a path and maximized the utility function (mutual information) using a genetic algorithm. Francis et al. [11] leveraged Bayesian optimization to find informative paths in continuous spaces. However, similar to discrete optimization and genetic algorithm based approaches, the method was computationally expensive and limited the approach's scalability.

A closely related problem is the correlated orienteering problem (COP), in which one has to plan a path that maximizes the information gain in a correlated environment while restricting the path to a given distance budget. Yu et al. [20] addressed this problem by formulating it as a mixed integer quadratic programming problem. Agarwal and Akella [21] generalized COP to both point locations and 1D features such as roads in environments and incorporated arc routing to efficiently compute a path.

Recently, Rückin et al. [22] leveraged deep reinforcement learning (DRL) to address the IPP problem. However, it requires one to simulate a diverse set of data and utilize significant computational resources to train the RL agent on the data before deployment.

IV. Preliminaries

A. Sparse Gaussian Processes

Gaussian processes (GPs) [23] are one of the most popular Bayesian approaches. The approach is non-parametric, and its computation cost depends on n the size of the training set. The approach's computation cost is dominated by an expensive $\mathcal{O}(n^3)$ matrix inversion operation on the $n \times n$ covariance matrix, which limits the approach to relatively small datasets that have less than $10,000$ samples.

The computational cost issues of GPs have been addressed by multiple authors [24], [25], [26], [27], and the methods are collectively referred to as sparse Gaussian processes (SGPs). The approaches entail finding a sparse set of m samples called *inducing points* ($m \ll n$), which are used to support the Gaussian process. Since there are fewer samples in the new GPs, these approaches reduce the covariance matrix that needs to be inverted to an $m \times m$ matrix, whose inversion is an $\mathcal{O}(m^3)$ operation.

There are multiple SGP approaches; the most well-known approach in the Bayesian community is the variational free energy (VFE) based approach [25], which has had a significant impact on the Gaussian process literature. It is a variational approach that is robust to overfitting and is competitive with other SGP approaches. The approach uses the following to compute the test sample predictions (mean and covariance) with the variational distribution q:

$$\begin{aligned} m^q_{\mathbf{y}}(\mathbf{x}) &= \mathbf{K}_{xm}\mathbf{K}_{mm}^{-1}\boldsymbol{\mu}\,, \\ k^q_{\mathbf{y}}(\mathbf{x},\mathbf{x}') &= k(\mathbf{x},\mathbf{x}') - \mathbf{K}_{xm}\mathbf{K}_{mm}^{-1}\mathbf{K}_{mx'} \\ &\quad + \mathbf{K}_{xm}\mathbf{K}_{mm}^{-1}\mathbf{A}\mathbf{K}_{mm}^{-1}\mathbf{K}_{mx'}\,. \end{aligned} \tag{2}$$

The subscripts of the covariance terms represent the variables used to compute the matrices—m indicates the inducing points $\mathbf{X}_m$. The approach maximizes the following evidence lower bound (ELBO) $\mathcal{F}$ to optimize the variational distribution q:

$$\begin{aligned} \mathcal{F} = &\underbrace{\frac{n}{2}\log(2\pi)}_{\text{constant}} + \underbrace{\frac{1}{2}\mathbf{y}^\top(\mathbf{Q}_{nn}+\sigma^2_{\text{noise}}I)^{-1}\mathbf{y}}_{\text{data fit}} \\ &+ \underbrace{\frac{1}{2}\log|\mathbf{Q}_{nn}+\sigma^2_{\text{noise}}I|}_{\text{complexity term}} - \underbrace{\frac{1}{2\sigma^2_{\text{noise}}}Tr(\mathbf{K}_{nn}-\mathbf{Q}_{nn})}_{\text{trace term}}\,, \end{aligned} \tag{3}$$

where $\mathbf{Q}_{nn} = \mathbf{K}_{nm}\mathbf{K}_{mm}^{-1}\mathbf{K}_{mn}$ and $\mathbf{K}_{mm}$ is the covariance matrix on the inducing points $\mathbf{X}_m$.

The lower bound $\mathcal{F}$ has three key terms—the data fit, complexity, and trace terms. The data fit term measures prediction accuracy on the training set data. The complexity and trace terms do not depend on the training set labels; instead, they ensure that the SGP-inducing points are well separated and reduce the SGP's overall uncertainty about the training set. When the trace term becomes zero, the SGP becomes equivalent to a full GP. We refer the reader to Bauer et al. [28] for further analysis of the SVGP's lower bound.

B. SGP-based Sensor Placement

In our concurrent work [29], we laid the foundation for SGP based sensor placement. We proved that any sensor placement problem can be reduced to a regression problem that can be efficiently solved using sparse Gaussian processes. The method also showed that we can train the SGP in an unsupervised manner by setting the labels of the training set and SGP mean to zero, which disables the label-dependent term of the optimization bound used in SVGP (Equation 3). The key advantage of this approach is that it uses the SGP's optimization bound as the utility function, which was shown to behave similar to MI while being significantly cheaper to compute than MI. The sensor placement approach, outlined in Algorithm 1, entailed sampling random unlabeled points in the sensor placement environment and fitting a sparse variational Gaussian process (SVGP, [25]) with known kernel parameters to the sampled points. Once the SGP was trained, the learned inducing points of the SGP are considered the solution sensor placements.

Algorithm 1: Continuous-SGP [29]. k_θ is the kernel with learned parameters, Φ is a random distribution defined over the domain of the environment $\mathcal{V}$, and γ is the SGP learning rate.

Input: $k_\theta, \mathcal{V}, \Phi, s, \gamma$
Output: Sensor placements $\mathcal{A} \subset \mathcal{V}$, $|\mathcal{A}| = s$
$\mathbf{X} \sim \Phi(\mathcal{V})$ // Draw unlabeled locations
$\mathbf{X}_m = \text{RandomSubset}(\mathbf{X}, s)$ // Initialize $\mathbf{X}_m$
// Initialize SVGP with 0 label dataset
$\varphi = \mathcal{SGP}(0, k_\theta; \mathbf{X}, \mathbf{y} = \mathbf{0}, \mathbf{X}_m)$
Loop until *convergence* : $\mathbf{X}_m \leftarrow \mathbf{X}_m + \gamma \nabla \mathcal{F}_\varphi(\mathbf{X}_m)$
return $\mathbf{X}_m$

V. Method

Our SGP based sensor placement approach [29] has two key properties that are relevant to addressing the informative path planning (IPP) problem. First, the SGP approach can generate solution sensor placements for both discrete and continuous environments. Second, the approach is able to obtain sensor placement solutions on par with the ones obtained by maximizing mutual information (MI) but with significantly reduced computational cost.

However, the key limitation of the SGP based sensor placement approach is that it does not consider the order in which the sensing locations are visited. Indeed, in IPP, we need to consider the order in which the sensing locations are visited and potentially also consider other constraints on the path, such as distance budget and velocity limits.

In the following, we first detail our approach to address the visitation order issue of the SGP based sensor placement approach for single-robot IPP. Then we explain how to impose routing constraints, such as a distance budget and velocity limits. After which we generalize our approach to handle multi-robot IPP, and then finally address continuous sensing along the paths and modeling non-point FoV sensors.

A. Single-Robot IPP

We address the visitation order issue in spatially correlated environments by leveraging a travelling salesperson problem (TSP) solver [30]. In the most fundamental version of the single-robot IPP problem, we need not consider any constraints on the travel distance. Therefore, we first obtain s sensor placement locations using the SGP approach and then generate a path that visits all the solution sensing locations by (approximately) solving the TSP, modified to allow for arbitrary start and end nodes:

$$\begin{aligned} \mathbf{X}_m &= \text{Continuous-SGP}(k_\theta, \mathcal{V}, \Phi, s, \gamma) \\ \mathbf{X}_m &= \text{TSP}(\mathbf{X}_m)\,. \end{aligned} \tag{4}$$

In spatio-temporally correlated environments, we do not even have to solve the TSP. This is because the generated solution sensor placements would have an inherent visitation order since they span both space and time. However, such an approach could generate solutions that cannot be traversed by real-world robots that have restrictions on the robot dynamics. The approach can handle a distance constraint if we are allowed to drop a few locations from the selected sensing locations, but it would do that without accounting for the lost information from the dropped sensing locations. Therefore, we must develop a more sophisticated approach to address real-world IPP problems that have constraints such as a distance budget and velocity limits.

We do this by leveraging the differentiability of the SGP's optimization objective $\mathcal{F}$ (Equation 3) with respect to the inducing points $\mathbf{X}_m$. The inducing points of the SGP $\mathbf{X}_m$, which we consider as the sensing locations, are used to compute the covariance matrix $\mathbf{K}_{mm}$, which is in turn used to compute the Nyström approximation matrix $\mathbf{Q}_{nn}$ in the objective function $\mathcal{F}$. We can impose constraints on the sensing locations by adding differentiable penalty terms dependent on the inducing points $\mathbf{X}_m$ to the objective function $\mathcal{F}$. Such a method would still be differentiable and can be optimized using gradient descent.

We can use the above method to even impose constraints on the solution *paths*. We do this by first solving the TSP on the SGP's initial inducing points $\mathbf{X}_m$ and treating them as an ordered set, which would give us an initial path that sequentially visits the inducing points. We then augment the SGP's objective function $\mathcal{F}$ with differentiable penalty terms that operate on the ordered inducing points for each path constraint and optimize the SGP to get the solution path. For instance, we can formulate the distance budget constraint as follows:

$$\hat{\mathcal{F}} = \mathcal{F} - \alpha \text{ReLu}(\text{PathLength}(\mathbf{X}_m) - c)\,, \\ \text{where ReLu}(x) = \max(x, 0)\,. \tag{5}$$

Here, PathLength is a function to obtain the total travel distance of the path that sequentially visits each of the inducing points $\mathbf{X}_m$ (treated as an ordered set). α is a weight term used to scale the distance constraint penalty term. The ReLu function ensures that $\mathcal{F}$ remains unchanged if the path length is within the distance budget c and penalizes it only if the length exceeds the distance budget. Note that since we maximize the objective function $\mathcal{F}$, we subtract the distance constraint term.

Similarly, we can accommodate additional constraints on the route, such as limits on the velocity and acceleration. In addition, we can trivially set predefined start and end points for the paths by freezing the gradient updates to the first and last inducing points of the SGP.

B. Multi-Robot IPP

We now address the multi-robot IPP problem. We accomplish this by increasing the number of inducing points in the SGP. If we have r robots and need paths with s sampling locations each, we initialize the SGP with rs inducing points. We can then find the r paths by solving the vehicle routing problem (VRP, [31]). This gives us an ordered set with rs sensing locations that form the r initial paths. We then add the path constraints that operate on each path (every s consecutive inducing points) to the objective function $\mathcal{F}$ and optimize the SGP to get the r solution paths $\mathcal{P}$. The approach is shown in Algorithm 2. We also present a decomposition trick in the Appendix [32] that can be leveraged to further reduce the computational cost of our approach and validate that our approach has optimal waypoint assignments in spatio-temporal environments.

Algorithm 2: Multi-Robot IPP for r paths with s sensing points each. k_θ is the kernel with learned parameters, Φ is a random distribution defined over the domain of the environment $\mathcal{V}$, and γ is the SGP learning rate. $\mathbf{C}$ are the path constraints.

Input: $k_\theta, \mathcal{V}, \Phi,\ s,\ r,\ \mathbf{C},\ \gamma$
Output: $\mathcal{P} = \{\mathcal{P}_i | \mathcal{P}_i \in \psi, |\mathcal{P}_i| = s, i = 1, ..., r\}$

```
X ~ Φ(V)  // Draw unlabeled locations
X_m = RandomSubset(X, rs)  // Initialize X_m
X_m = VRP(X_m)  // Get initial paths
// Add path constraints
F̂ = F − α(Constraint(X_m) − C)
// Initialize SVGP with ordered inducing points
φ = SGP(0, k_θ; X, y = 0, X_m, F̂)
Loop until convergence : X_m ← X_m + γ∇F̂_φ(X_m)
return X_m
```

C. IPP for Continuous and Non-point FoV Sensing Robots

Our approach so far considered only discrete sensing robots with point sensors. However, it is often the case that we want to continuously sense along the robot paths and use sensors such as cameras with non-point field of view (FoV). Although one could generate paths optimized for discrete sensing robots with point sensors and deploy non-point FoV continuous sensing robots, explicitly optimizing the paths for non-point FoV continuous sensing robots will give far more information about the sensing environment. A key advantage of generalizing the SGP based sensor placement approach [29] to IPP is that we can leverage all the properties of GPs and SGPs [23], [33], [24], [25]. We detail two such properties and how they can be used to address IPP for continuous sensing robots.

The first property is that the inducing points of SGPs can be transformed with any non-linear function and still be optimized using gradient descent. We can use such transformations to approximate the data collected along the solution paths. We do this by parameterizing the m inducing points of the SGP as the sensing locations for a discrete sensing robot's path. Then, we apply a transformation—the expansion transformation T_{exp}—to interpolate an additional p points between every consecutive pair of inducing points that form the robot's path $\hat{\mathbf{X}}_m = T_{\text{exp}}(\mathbf{X}_m)$ to get mp inducing points. We can then use the mp inducing points $\mathbf{X}_{mp}$ to compute the SGP's objective function $\hat{\mathcal{F}}$ with path constraints. Note that the interpolation operation is differentiable and allows us to compute the gradients for the original m inducing points $\mathbf{X}_m$ before the transformation.

This approach allows us to account for the information gathered along the whole path. Although the approach is the same as using mp inducing points to parameterize a discrete sensing robot, it is novel when considering non-point FoV sensors. In such cases, we can leverage the expansion transformation to approximate the non-point FoV sensor's whole sensing footprint, which is only possible with the expansion transformation. This is because the transformation allows us to compute the gradients with respect to the position of the sensing locations along the path. Without the transformation, we would have a large number of inducing points without being able to ensure that the points retain the FoV shape of the sensors. However, the issue with this approach is that it would significantly increase the number of inducing points, and since SGPs have cubic computation cost with respect to the number of inducing points, it would severely limit the approach's feasibility.

We address this issue by introducing a property of GPs that GPs are closed under linear transformations [23], [6]. We can leverage this property to model sensors with integrated observations [34], i.e., where the labels are modeled as

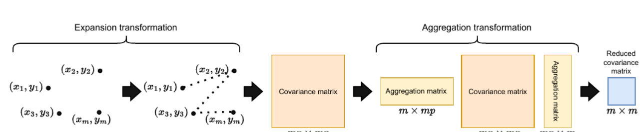


Fig. 1: An illustration of the expansion and aggregation transformations used in IPP for continuous sensing robots.

$y_i = ||\mathbf{w}_i|| \int_0^1 f(\mathbf{w}_i t + \mathbf{z}_i)dt + \epsilon_i$, with $\mathbf{z}_i$ as the start point of a line along which the data is integrated, and $\mathbf{w}_i$ as the direction and length of the line. We do this with the aggregation transformation T_{agg}, which aggregates (with an averaging operation) the covariances corresponding to the p inducing points that approximate each edge of the path (or the FoV of a sensor, or both) and reduces the size of the covariance matrix from $mp \times mp$ back to $m \times m$.

We first use the expansion transformation (T_{exp}) on the m inducing points to map them to a larger set of mp points. Then we use the aggregation transformation T_{agg} on the covariance matrices built using the mp points. The covariances are used to compute $\mathbf{Q}_{nn}$, which is, in turn, used to compute the SVGP's objective function (Equation 3):

$$\mathbf{Q}_{nn} = \mathbf{K}_{n \times mp} T_{\text{agg}} (T_{\text{agg}}^\top \mathbf{K}_{mp \times mp} T_{\text{agg}})^{-1} T_{\text{agg}}^\top \mathbf{K}_{mp \times n} \,. \tag{6}$$

Here $\mathbf{K}_{n \times mp}$ is the covariance between the n training set inputs and the mp inducing points. The aggregation transformation reduces the covariance matrices before inversion. The approach is illustrated in Figure 1, and further details of how to define the transformations are presented in the Appendix. Therefore, the inversion operation cost is reduced to $\mathcal{O}(m^3)$ from $\mathcal{O}(m^3p^3)$. Thus, we reap the benefits of the expansion transformation, which allows us to model continuous sensing and non-point FoV sensing robots, and the reduced computation cost from the aggregation transformation. We found that the aggregation transformation also stabilized the gradients while optimizing the inducing points.

The approach also has the additional advantage of allowing us to efficiently model complex path parametrizations, such as using splines to get smooth paths, being able to account for sensors such as cameras whose FoV varies with the height from the ground, and even model FoVs that account for the shape of the surface, such as stereo vision cameras when used to scan 3D surfaces.

VI. Experiments

We first demonstrate our approach for the unconstrained single robot IPP problem on the ROMS ocean salinity [35] and US soil moisture [36] datasets. The ROMS dataset contains salinity data from the Southern California Bight region, and the US soil moisture dataset contains moisture readings from the continental USA.

We benchmarked our SGP based IPP approach (SGP) that optimizes a path for discretely sensing s locations and our transformation based generalization of our SGP based IPP approach (Arc-SGP) that optimizes the paths while accounting for the information collected along the whole path. We also benchmarked two baselines approaches—Information-Driven Planner (IDP, Ma et al. [2]) and Continuous-Space Informative Path Planner (CIPP, Hitz et al. [10]). IDP leverages discrete optimization to iteratively find discrete sensing locations that maximize mutual information (MI), and CIPP leverages CMA-ES, a genetic algorithm, to find informative sensing locations that maximize MI in continuous spaces.

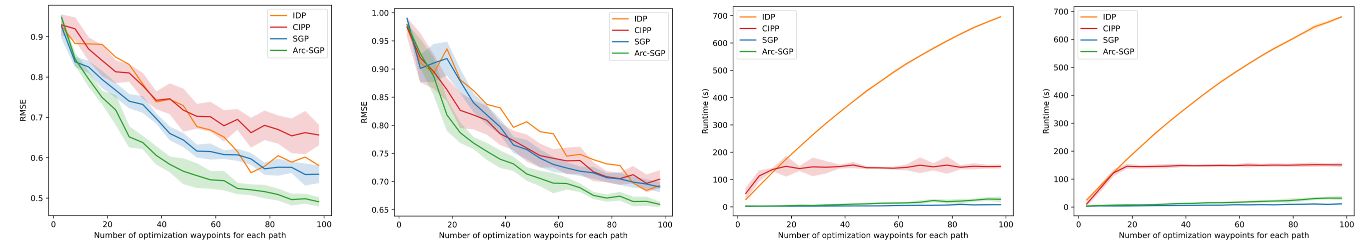
An RBF kernel [23] was used to model the spatial correlations of the datasets (the baselines use it to measure MI). We evaluated the paths by gathering the ground truth data along the generated solution paths (i.e., by continuous sensing robots) and estimating the state of the whole environment from the collected data. The root-mean-squared error (RMSE) between the ground truth data and our estimates was used to quantify the solution paths. We generated solution paths for both the datasets with the number of path sensing locations ranging from 3 to 100 in increments of 5. The experiment was repeated 10 times. The mean and standard deviation of the RMSE and runtime results on the ROMS and US soil moisture datasets are shown in Figure 2.

As we can see, our SGP approach is consistently on par or better than the two baselines in terms of RMSE, and our Arc-SGP approach has a considerably lower RMSE than the other approaches in all cases. Also, both our approaches substantially outperform the baselines in computation time (up to 35 times faster). In both the baselines, a significant amount of computation time is spent on computing MI, while our SGP approach's objective approximates the same in a computationally efficient manner (detailed in our foundational work [29]). Indeed, the MI computation cost is the key reason why both IDP and CIPP cannot scale to spatio-temporally correlated environments, since even with a coarse discretization, it would be far too computationally expensive. Also, since our approaches rely on gradient information, they are significantly faster to converge compared to the discrete and genetic algorithm based baseline approaches.

We now demonstrate our approach for multi-robot IPP. We used the same kernel parameters as we did in the previous experiments. The solution paths were generated for four robots with the number of optimization waypoints ranging from 3 to 25 in increments of 5 for each robot's path. We evaluated the SGP, Arc-SGP, and CIPP methods, which support multi-robot IPP. The RMSE and runtime results on the ROMS and US soil moisture datasets are shown in Figure 3.

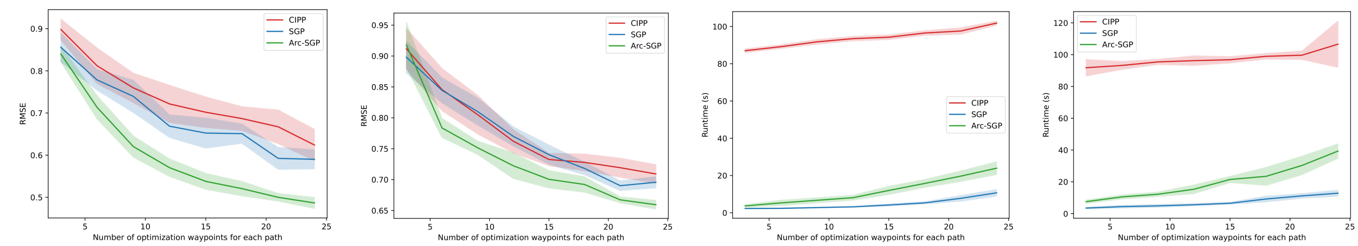
Our SGP approach is again consistently on par or better than the CIPP approach in terms of RMSE, and our Arc-SGP's RMSE is notably lower than both SGP and CIPP approaches. Moreover, both our approaches—SGP and Arc-SGP—significantly outperform CIPP (up to 26 times faster) in terms of compute time.

In the next demonstration, we show our approach for IPP with a distance constraint. A Gaussian process was used to sample dense spatio-temporal temperature data. We used an RBF kernel with a length scale of 7.70 m, 19.46 m, and 50.63 mins along the x, y, and temporal dimensions, respectively. We generated paths by optimizing the inducing points in our SGP approach with distance budgets of 10 m, 20 m, and 40 m; the results are shown in Figure 4. Our approach consistently saturates the distance budget without exceeding it to get the maximum amount of new data, evident from the paths' RMSE scores. For reference, we also show the paths generated for three robots in the same environment (Figure 5). We do not show the reconstructions since the



(a) RMSE (ROMS) (b) RMSE (Soil) (c) Runtime (ROMS) (d) Runtime (Soil)

Fig. 2: RMSE and runtime results for single robot IPP with the IDP, CIPP, SGP, and Arc-SGP approaches on the ROMS and US soil datasets.



(a) RMSE (ROMS) (b) RMSE (Soil) (c) Runtime (ROMS) (d) Runtime (Soil)

Fig. 3: RMSE and runtime results for four robot IPP with the CIPP, SGP, and Arc-SGP approaches on the ROMS and US soil datasets.

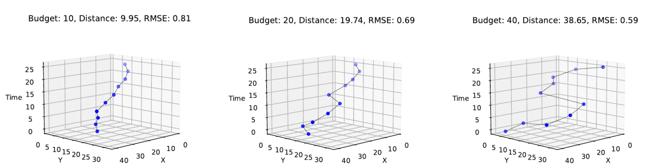


Fig. 4: Data collection paths generated using a spatio-temporal kernel function for different distance budgets.

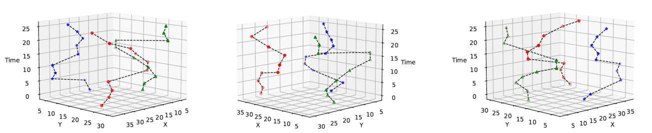


Fig. 5: Three different views of our multi-robot IPP solution paths, with path lengths of 47.29 m, 47.44 m, and 47.20 m. The data from all 3 paths gave us an RMSE of 0.34.

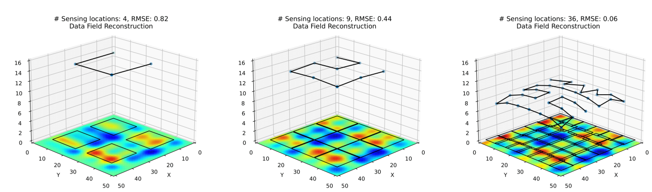


Fig. 6: Solution paths for a discrete sensing robot with a square height-dependent FoV area (black squares) sensor. The solution paths adjust the sensor height to ensure a good balance between the ground sampling resolution and the coverage area.

data is spatio-temporal, which is difficult to show in 2D.

Figure 6 shows our SGP approach for a discrete sensing robot, i.e., it senses only at the path's vertices (blue points). We considered a 3D environment with densely sampled elevation data and parametrized the path so that we account for the robot's sensing FoV area to be dependent on the robot's height from the ground. We used an RBF kernel with a length scale of 3 m.

Also, note that our approach can solve for paths in more complex environments and use non-stationary kernels [23] to capture intricate correlation patterns in environments. In addition, please refer to the Appendix for additional experiments and more algorithm details. We also detail how to efficiently infer the state of the environment and estimate the kernel parameters in the Appendix [32].

VII. Conclusion

We presented an efficient continuous space approach to informative path planning using sparse Gaussian processes that can address various challenges related to monitoring in spatially and spatio-temporally correlated environments. Our approach can model routing constraints, and handle discrete and continuous sensing robots with arbitrary FoV shapes. Furthermore, our method generalizes to multi-robot IPP problems as well. We demonstrated that our approach is fast and accurate for IPP on real-world data. We also presented our IPP solutions for different distance budgets, multi-robot scenarios, and with non-point FoV sensing robots. Our future work will build upon this approach to extend its applicability to online and decentralized IPP problems.

References

[1] J. Binney, A. Krause, and G. S. Sukhatme, "Informative path planning for an autonomous underwater vehicle," in *2010 IEEE International Conference on Robotics and Automation (ICRA)*, 2010, pp. 4791–4796.

[2] K.-C. Ma, L. Liu, H. K. Heidarsson, and G. S. Sukhatme, "Data-driven learning and planning for environmental sampling," *Journal of Field Robotics*, vol. 35, no. 5, pp. 643–661, 2018.

[3] 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.

[4] K. Jakkala and S. Akella, "Probabilistic Gas Leak Rate Estimation Using Submodular Function Maximization With Routing Constraints," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5230–5237, 2022.

[5] J. Binney, A. Krause, and G. S. Sukhatme, "Optimizing waypoints for monitoring spatiotemporal phenomena," *The International Journal of Robotics Research*, vol. 32, no. 8, pp. 873–888, 2013.

[6] R. N. Smith, M. Schwager, S. L. Smith, B. H. Jones, D. Rus, and G. S. Sukhatme, "Persistent Ocean Monitoring with Underwater Gliders: Adapting Sampling Resolution," *Journal of Field Robotics*, vol. 28, no. 5, pp. 714–741, 2011.

[7] H. Zhu, J. J. Chung, N. R. Lawrance, R. Siegwart, and J. Alonso-Mora, "Online Informative Path Planning for Active Information Gathering of a 3D Surface," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 1488–1494.

[8] J. Rückin, L. Jin, F. Magistri, C. Stachniss, and M. Popović, "Informative Path Planning for Active Learning in Aerial Semantic Mapping," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 11 932–11 939.

[9] 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.

[10] G. Hitz, E. Galceran, M.-E. Garneau, F. Pomerleau, and R. Siegwart, "Adaptive Continuous-Space Informative Path Planning for Online Environmental Monitoring," *Journal of Field Robotics*, vol. 34, no. 8, pp. 1427–1449, 2017.

[11] G. Francis, L. Ott, R. Marchant, and F. Ramos, "Occupancy map building through Bayesian exploration," *The International Journal of Robotics Research*, vol. 38, no. 7, pp. 769–792, 2019.

[12] 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. 8, pp. 235–284, 2008.

[13] A. Singh, A. Krause, C. Guestrin, and W. J. Kaiser, "Efficient informative sensing using multiple robots," *J. Artif. Int. Res.*, vol. 34, no. 1, p. 707–755, Apr. 2009.

[14] G. Hollinger and S. Singh, "Proofs and experiments in scalable, near-optimal search by multiple robots," in *Robotics: Science and Systems IV*, 2009, pp. 206–213.

[15] C. Chekuri and M. Pal, "A recursive greedy algorithm for walks in directed graphs," in *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*, 2005, pp. 245–253.

[16] A. Krause and C. Guestrin, "Submodularity and its applications in optimized information gathering," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 4, Jul. 2011.

[17] A. Singh, A. Krause, C. Guestrin, and W. J. Kaiser, "Efficient informative sensing using multiple robots," *J. Artif. Int. Res.*, vol. 34, no. 1, p. 707–755, Apr. 2009.

[18] L. Bottarelli, M. Bicego, J. Blum, and A. Farinelli, "Orienteering-based informative path planning for environmental monitoring," *Engineering Applications of Artificial Intelligence*, vol. 77, pp. 46 – 58, 2019.

[19] L. M. Miller, Y. Silverman, M. A. MacIver, and T. D. Murphey, "Ergodic Exploration of Distributed Information," *IEEE Transactions on Robotics*, vol. 32, no. 1, pp. 36–52, 2016.

[20] J. Yu, M. Schwager, and D. Rus, "Correlated orienteering problem and its application to informative path planning for persistent monitoring tasks," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 342–349.

[21] S. Agarwal and S. Akella, "The Correlated Arc Orienteering Problem," in *Algorithmic Foundations of Robotics XV*, S. M. LaValle, J. M. O'Kane, M. Otte, D. Sadigh, and P. Tokekar, Eds. Cambridge, Massachusetts, USA: Springer International Publishing, 2023, pp. 402–418.

[22] J. Rückin, L. Jin, and M. Popović, "Adaptive Informative Path Planning Using Deep Reinforcement Learning for UAV-based Active Sensing," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 4473–4479.

[23] C. E. Rasmussen and C. K. Williams, *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, 2005.

[24] E. Snelson and Z. Ghahramani, "Sparse Gaussian Processes using Pseudo-inputs," in *Advances in Neural Information Processing Systems*, Y. Weiss, B. Schölkopf, and J. Platt, Eds., vol. 18. MIT Press, Cambridge, 2006.

[25] M. Titsias, "Variational learning of inducing variables in sparse Gaussian processes," in *Proceedings of Machine Learning Research*, 2009, pp. 567–574.

[26] T. N. Hoang, Q. M. Hoang, and B. K. H. Low, "A Unifying Framework of Anytime Sparse Gaussian Process Regression Models with Stochastic Variational Inference for Big Data," in *Proceedings of the 32nd International Conference on Machine Learning*, F. Bach and D. Blei, Eds., vol. 37. Lille, France: PMLR, 07–09 Jul 2015, pp. 569–578.

[27] T. D. Bui, J. Yan, and R. E. Turner, "A Unifying Framework for Gaussian Process Pseudo-Point Approximations Using Power Expectation Propagation," *The Journal of Machine Learning Research*, vol. 18, no. 1, p. 3649–3720, Jan 2017.

[28] M. Bauer, M. van der Wilk, and C. E. Rasmussen, "Understanding Probabilistic Sparse Gaussian Process Approximations," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, ser. NIPS'16. Curran Associates Inc., New York, 2016, p. 1533–1541.

[29] K. Jakkala and S. Akella, "Efficient Sensor Placement from Regression with Sparse Gaussian Processes in Continuous and Discrete Spaces," 2023, manuscript submitted for publication. [Online]. Available: https://arxiv.org/abs/2303.00028

[30] L. Perron and V. Furnon, "OR-Tools," Google. [Online]. Available: https://developers.google.com/optimization/

[31] P. Toth and D. Vigo, *Vehicle Routing: Problems, Methods, and Applications, Second Edition*. Philadelphia, USA: Society for Industrial and Applied Mathematics, 2014.

[32] K. Jakkala and S. Akella, "Multi-Robot Informative Path Planning from Regression with Sparse Gaussian Processes," 2023, manuscript submitted for publication. [Online]. Available: https://itskalvik.github.io/publications/IPP

[33] R. Murray-Smith and B. A. Pearlmutter, "Transformations of Gaussian Process Priors," in *Deterministic and Statistical Methods in Machine Learning*, J. Winkler, M. Niranjan, and N. Lawrence, Eds. Berlin, Heidelberg: Springer, 2005, pp. 110–123.

[34] K. Longi, C. Rajani, T. Sillanpää, J. Mäkinen, T. Rauhala, A. Salmi, E. Haeggström, and A. Klami, "Sensor Placement for Spatial Gaussian Processes with Integral Observations," in *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, ser. Proceedings of Machine Learning Research, J. Peters and D. Sontag, Eds., vol. 124. PMLR, 03–06 Aug 2020, pp. 1009–1018.

[35] A. F. Shchepetkin and J. C. McWilliams, "The regional oceanic modeling system (ROMS): a split-explicit, free-surface, topography-following-coordinate oceanic model," *Ocean Modelling*, vol. 9, no. 4, pp. 347–404, 2005.

[36] [Online]. Available: https://www.drought.gov/data-maps-tools/nasa-sport-lis-soil-moisture-products