

Fully Differentiable Adaptive Informative Path Planning

Kalvik Jakkala and Srinivas Akella

Abstract—Autonomous robots can survey and monitor large environments. However, these robots often have limited computational and power resources, making it crucial to develop an efficient and adaptive informative path planning (IPP) algorithm. Such an algorithm must quickly adapt to environmental data to maximize the information collected while accommodating path constraints, such as distance budgets and boundary limitations.

Current approaches to this problem often rely on maximizing mutual information using methods such as greedy algorithms, Bayesian optimization, and genetic algorithms. These methods can be slow and do not scale well to large or 3D environments. We present an adaptive IPP approach that is fully differentiable, significantly faster than previous methods, and scalable to 3D spaces. Our approach also supports continuous sensing robots, which collect data continuously along the entire path, by leveraging streaming sparse Gaussian processes.

Benchmark results on two real-world datasets demonstrate that our approach yields solutions that are on par with or better than baseline methods while being up to two orders of magnitude faster. Additionally, we showcase our adaptive IPP approach in a 3D space using a system-on-chip embedded computer with minimal computational resources. Our code is available in the SGP-Tools Python library with a companion ROS 2 package for deployment on ArduPilot-based robots.

I. INTRODUCTION

Informative Path Planning (IPP) is a fundamental problem in robotics. It requires finding paths to obtain the maximal amount of novel data about an underlying data field of interest while ensuring that path constraints, such as distance budget limits and boundary constraints, are satisfied. Additionally, the environment is often unexplored, and with no training data available, necessitates adaptive IPP variants. These adaptive variants can plan initial paths and update them by learning from the data collected along the traversed portion of the paths.

The IPP problem is particularly relevant for persistent environmental monitoring, where robots must continuously monitor the environment for tasks such as pollution tracking in lakes and rivers. As such, adaptive approaches that detect and adapt to changes in the data field are crucial [1], [2], [3]. The problem also arises in surface inspection tasks [4], where robots inspect 3D structures like bridges, dams, aircraft wings, and pipeline insulation.

Given the significance of the IPP problem, several authors have addressed it [5], [6], [7], [3]. Many approaches compute the informativeness of locations using mutual information (MI) calculated with Gaussian processes (GPs). However, computing MI requires discretizing the environment

into n candidate locations and has a computational cost of $\mathcal{O}(n^3)$, making it computationally expensive. Additionally, prior approaches often use optimization methods such as Bayesian optimization [8], POMDPs [9], and genetic algorithms [10] to optimize the solutions. Such approaches are computationally expensive and require numerous evaluations of MI. Therefore, the aforementioned approaches do not scale to large environments and 3D spaces.

We present an adaptive IPP approach that is fully differentiable with respect to the sensing locations, significantly faster to optimize, and scalable to large 3D environments. This Adaptive-SGP-IPP approach generalizes our prior sparse Gaussian process (SGP) based non-adaptive SGP-IPP approach [11] to adaptive cases. Our approach can also handle single and multi-robot IPP while accommodating robots with discrete and continuous sensing models and path constraints. Moreover, since our approach is fully differentiable, we can use any gradient-based optimization method [12] such as gradient descent and Newton’s method. Our contributions are:

- We present a natural generalization of our fully differentiable IPP approach to handle adaptive IPP.
- We address adaptive IPP for continuous sensing robots by leveraging streaming sparse Gaussian processes.
- We demonstrate the adaptive IPP approach for a robot in 3D space with a complex sensor model on a resource-limited system-on-chip embedded computer.
- Our code is available in the SGP-Tools Python library¹ with a companion ROS 2 package that can be deployed on ArduPilot-based robots.

II. ADAPTIVE IPP PROBLEM

We are given an environment $\mathcal{V} \subseteq \mathbb{R}^d$ with a phenomenon such as temperature to be monitored. We have r robots and must find the set $\mathcal{P}$ of r paths, one for each robot, so that the data $\mathbf{y} \in \mathbb{R}$ collected along the paths 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 that can be computed without the ground truth labels. Moreover, we also consider constraints $\mathbf{C}$ such as distance budget limits and boundary constraints on the paths:

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

¹<https://github.com/itskalvik/SGP-Tools>

$$\begin{aligned} \mathcal{P}^* = & \arg \max_{\{\mathcal{P}_i \in \psi, i=1, \dots, r\}} I(\cup_{i=1}^r \text{SAMPLE}(\mathcal{P}_i)), \\ \text{s.t. Constraints}(\mathcal{P}_{i=1, \dots, r}) & \leq \mathbf{C} \end{aligned} \quad (1)$$

Here ψ is the space of paths contained within the environment $\mathcal{V}$. The `SAMPLE` function returns the sensing points at the robot path waypoints when modeling a discrete sensing robot, and it returns all the sensed points along the path when modeling a continuous sensing robot. 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. Moreover, we assume no training data from the target environment is available. This *adaptive IPP problem*² generalizes the IPP problem by requiring updates to the solution paths in response to newly gathered data.

III. RELATED WORK

The Informative Path Planning (IPP) problem is known to be NP-hard [13]. Therefore, only suboptimal solutions can be found for most real-world problems. A dominant approach is to select the sensing locations that maximize information metrics, such as variance and mutual information (MI) [6], [14]. Krause et al. [5] showed that variance-based approaches are fast to compute but result in sensing locations close to the environment's boundaries, with high variance and low information. They instead advocated the use of MI, which resulted in more informative sensing locations albeit with higher computation cost— $\mathcal{O}(|n|^3)$, where n is the number of discretized locations in the environment.

Krause et al. [5] addressed the compute cost issue by leveraging the submodular property of MI and employing greedy algorithms to select a subset of the discretized candidate sensing locations as the solution. Singh et al. [6] introduced a recursive-greedy algorithm that maximized MI for single and multi-robot IPP. Bottarelli et al. [15] developed active learning-based IPP algorithms with a complexity of $\mathcal{O}(|n|^5)$. A key limitation of the above approaches is their environment discretization requirement, which restricts the solution sensing locations of the paths to be a subset of the candidate locations.

Hollinger and Sukhatme [7] enabled IPP in continuous spaces without limiting the sensing locations to a candidate set by presenting IPP algorithms that maximized MI using rapidly-exploring random trees (RRT) and derived asymptotically optimal guarantees. Hitz et al. [10] developed an adaptive multi-robot IPP approach based on a genetic algorithm that could optimize the sensing locations in continuous spaces given a utility function. Ma et al. [1] solved the adaptive IPP problem by maximizing MI using dynamic programming and an online variant of sparse Gaussian processes to learn the model hyperparameters. Francis et al. [8] and Vivaldini et al. [16] leveraged Bayesian optimization for single robot IPP in continuous spaces. A common aspect of the above approaches is their use of MI and the ability to find

paths with sensing locations in continuous spaces. However, they also require numerous evaluations of MI, which is expensive to compute and whose complexity depends on the environment discretization resolution. Even though the methods do not limit the sensing locations to the discretized locations, a more accurate estimate of MI requires a higher discretization resolution. As such, the methods do not scale to large and 3D environments.

Given MI's fundamental compute cost issue, multiple authors have investigated alternative approaches to address the adaptive IPP problem and its variants. Mishra et al. [17] and Berget et al. [18] addressed adaptive IPP by selecting locations with high variance as the solution waypoints. Schmid et al. [19] addressed adaptive IPP for 3D reconstruction. They utilized a quadratic function of the distance from the camera to model the uncertainty in the environment and optimized it using RRTs in an adaptive algorithm. Zhu et al. [4] also addressed IPP for 3D reconstruction and used a probabilistic variance estimate as the uncertainty measure. Moon et al. [20] addressed adaptive IPP using a sampling-based method to reduce entropy. Ott et al. [21] leveraged POMDPs to address the adaptive IPP with multi-modal sensing problem. The above variance and entropy-based approaches can be optimized with minimal computational resources but result in less informative solutions compared to MI-based methods [5].

Miller et al. [22] and Rao et al. [23] addressed continuous-space non-adaptive and adaptive IPP using ergodic control algorithms. These methods assume that we know the information density in the environment, which is not always feasible. Finally, Cao et al. [24] and Rückin et al. [25] leveraged deep reinforcement learning (DRL). However, they require simulating a diverse data set. For a more comprehensive review of adaptive IPP, please refer to [3].

IV. FULLY DIFFERENTIABLE SGP-BASED IPP

Jakkala and Akella [26] presented an approach for fully differentiable sensor placement and generalized their approach to address IPP [11]. They did this by first formulating the sensor placement problem using variational inference. The method leveraged a sparse variational distribution to approximate the data field being monitored. The method was fully differentiable with respect to the sensor placement locations, and the authors also showed that given the hyperparameters, their formulation was equivalent to optimizing sparse Gaussian processes (SGPs) in an unsupervised manner.

The sensor placement approach was generalized to handle IPP by treating the SGP's inducing points as an ordered set, thereby retaining the order in which the waypoints were visited. This also allowed them to add any differentiable path length constraint to ensure that the solution path was within the given distance budget. Moreover, they also showed that the method could handle multi-robot IPP by using a separate set of inducing points and a distance budget term for each robot (Algorithm 1).

The method also allows us to efficiently model continuous sensing robots, i.e., robots that sense continuously along the

²Note that some authors refer to this problem as the online IPP problem.

Algorithm 1: SGP-IPP [11]; θ are the hyperparameters, Φ is a random distribution defined within the boundaries of the environment $\mathcal{V}$, s is the number of specified waypoints, n is the number of random unlabeled locations used to train the SGP, γ is the SGP learning rate, r is the number of robots, λ is the constraint weight factor, $\mathbf{C}$ are the path constraints. VRP is the vehicle routing problem solver.

Input: $\theta, \Phi, \mathcal{V}, s, n, \gamma, r, \lambda, \mathbf{C}$
Output: Paths $\mathcal{P} = \{\mathcal{P}_i | \mathcal{P}_i \in \mathcal{V}, i = 1, \dots, r\}$

- 1 $\mathbf{X} \sim \Phi(\mathcal{V})$ // Draw n unlabeled locations
- 2 $\mathbf{X}_m \sim \Phi(\mathcal{V})$ // Draw rs inducing point locations
- 3 $\mathbf{X}_m = \text{VRP}(\mathbf{X}_m)$ // Get r initial paths $\mathcal{P}$
// Add constraints to the SGP's objective function
- 4 $\hat{\mathcal{F}} = \mathcal{F} - \lambda(\text{Constraints}(\mathbf{X}_m) - \mathbf{C})$
// Initialize SGP with the ordered inducing points
- 5 $\varphi = \text{SGP}(\text{mean} = \mathbf{0}, \theta; \mathbf{X}, \mathbf{y} = \mathbf{0}, \mathbf{X}_m, \hat{\mathcal{F}})$
// Optimize the inducing points
- 6 **Loop until convergence:**
- 7 $\mathbf{X}_m \leftarrow \mathbf{X}_m + \gamma \nabla \hat{\mathcal{F}}(\mathbf{X}_m)$
- 8 **return** $\mathcal{P} = \{\{\mathbf{X}_m[i, j] | j = 1, \dots, s\} | i = 1, \dots, r\}$

path and robots with non-point sensors such as cameras. They did this by leveraging the inherent property of their variational formulation and, by extension, SGPs, that the inducing points (i.e., the sensing locations or waypoints) can be transformed using differentiable non-linear operations [27] and still be optimized with respect to the evidence lower bound (ELBO), i.e., the method's objective function:

$$\begin{aligned}
\mathcal{F}(q) &= \int q(\mathbf{f}_m) \log \frac{\mathcal{N}(\mathbf{y} | \boldsymbol{\alpha}, \sigma_{\text{noise}}^2 \mathbf{I}) p(\mathbf{f}_m)}{q(\mathbf{f}_m)} d\mathbf{f}_m \\
&\quad - \frac{1}{2\sigma_{\text{noise}}^2} \text{Tr}(\mathbf{K}_{ff} - \mathbf{Q}) \\
\boldsymbol{\alpha} &= \mathbb{E}[\mathbf{f} | \mathbf{f}_m] = \mathbf{K}_{nm} \mathbf{K}_{mm}^{-1} \mathbf{f}_m \\
\mathbf{Q} &= \mathbf{K}_{nm} \mathbf{K}_{mm}^{-1} \mathbf{K}_{mn}.
\end{aligned} \tag{2}$$

Here, $\mathbf{f}$ are the noise-free latent variables corresponding to the n training inputs $\mathbf{X}$ and their zero labels $\mathbf{y}$. $\mathbf{f}_m$ correspond to the m inducing inputs $\mathbf{X}_m$, and q is the optimal variational distribution. σ_{noise} is the data noise, and the subscripts of a covariance matrix $\mathbf{K}$ indicate the variables to compute it.

This differentiability property allows us to interpolate additional points between pairs of inducing points to approximate the information collected along the whole path or transform an inducing point at a sensing location into multiple points approximating a non-point sensor's field-of-view (FoV) area. Moreover, the method only needs to optimize the original inducing points by leveraging back-propagation, thereby retaining the path segment or FoV shape with minimal additional compute requirements. Please refer to [26] for more details.

However, the key advantage of the approach is its differentiability, which allows us to use gradient-based approaches

such as gradient descent and Newton's method to optimize the sensing locations, while considering any differentiable path constraints in an efficient manner.

V. METHOD: A FULLY DIFFERENTIABLE ADAPTIVE IPP

Adaptive IPP for Discrete Sensing Robots: We now generalize the SGP-based IPP approach to perform adaptive IPP by initializing the SGP's hyperparameters with random values and iteratively alternating between optimizing the solution path(s) with the path planner (Algorithm 1) and updating the SGP's hyperparameters using a separate Gaussian process trained on the data collected from the traversed portion of the path(s) (Algorithm 2). Since the SGP-based IPP approach is significantly faster than earlier discrete optimization-based methods, this adaptive approach will also be significantly faster than previous adaptive counterparts.

Algorithm 2: Adaptive-SGP-IPP

Input: Number of waypoints s
Output: Data $(\mathbf{X}^{\text{path}}, \mathbf{y}^{\text{path}})$ from the traversed path
// SGP hyperparameters from a uniform distribution

- 1 $\theta \sim \mathcal{U}(\cdot)$
- 2 $\mathcal{P} = \text{SGP-IPP}(\theta, s)$ // Get the initial path(s)
- 3 $\text{HParam-GP}(\theta)$ // SSGP for continuous sensing
- 4 $\mathbf{X}^{\text{path}} = \{\}, \mathbf{y}^{\text{path}} = \{\}$
- 5 **for** $j \leftarrow 1$ **to** s **do**
- 6 $\mathbf{X}^{\text{batch}}, \mathbf{y}^{\text{batch}} = \text{SAMPLE}(\mathcal{P}[j])$
- 7 $\theta = \text{HParam-GP.update}(\mathbf{X}^{\text{batch}}, \mathbf{y}^{\text{batch}})$
- 8 $\mathcal{P} = \text{SGP-IPP.update}(\theta, j)$
- 9 $\mathbf{X}^{\text{path}} = \mathbf{X}^{\text{path}} \cup \mathbf{X}^{\text{batch}}, \mathbf{y}^{\text{path}} = \mathbf{y}^{\text{path}} \cup \mathbf{y}^{\text{batch}}$
- 10 **return** $\mathbf{X}^{\text{path}}, \mathbf{y}^{\text{path}}$

However, a naive implementation of this approach has two key limitations. First, GPs are not designed to handle streaming data; as such, we would need to train the hyperparameter-learning GP from scratch after each data point is collected. Also, the computational cost of training a GP with the collected data will keep increasing until it becomes infeasible after about 10,000 data samples. These issues will not significantly impact the adaptive IPP approach for discrete sensing robots with a point field-of-view (FoV), i.e., robots that only sense the path's vertices. But numerous sensors, such as temperature probes and ocean salinity sensors, continuously collect data along the path and operate well above 10Hz. As such, the GP's compute limit would be quickly reached when modeling continuous sensing robots.

Adaptive IPP for Continuous Sensing Robots: We address the hyperparameter estimation problem by leveraging streaming sparse Gaussian processes (SSGPs) [28] rather than using a full GP to estimate the hyperparameters. SSGPs provide a principled method for handling streaming data, which arrives sequentially in batches. With SSGPs, the hyperparameter updates can be computed using only the current batch of data $\mathbf{y}_{\text{new}}$ and the previously computed variational distribution over all the past data $\mathbf{y}_{\text{old}}$. Additionally, this method leverages a sparse approximation, eliminating the

need to access the entire dataset and significantly reducing training time.

At each update step, to approximate all the data seen up to the current update step, the method computes a new optimal variational posterior distribution $q_{\text{new}}(\hat{\mathbf{f}})$, which replaces the old variational distribution $q_{\text{old}}(\hat{\mathbf{f}})$, using the new batch of data $\mathbf{y}_{\text{new}}$. The optimal distribution is obtained by minimizing the following Kullback–Leibler divergence:

$$\text{KL} \left[q_{\text{new}}(\hat{\mathbf{f}}) || p(\hat{\mathbf{f}} | \mathbf{y}_{\text{old}}, \mathbf{y}_{\text{new}}) \right]. \quad (3)$$

Here, $\hat{\mathbf{f}}$ are all the latent variables corresponding to the data. The above can be analytically minimized to obtain the following optimal posterior variational distribution:

$$\begin{aligned} q_{\text{opt}}(\mathbf{b}) &= p(\mathbf{b}) \mathcal{N}(\hat{\mathbf{y}}; \mathbf{K}_{\hat{\mathbf{f}}\mathbf{b}} \mathbf{K}_{\mathbf{bb}}^{-1} \mathbf{b}, \Sigma), \text{ where} \\ q(\mathbf{a}) &= \mathcal{N}(\mathbf{a}; \mathbf{m}_{\mathbf{a}}, \mathbf{S}_{\mathbf{a}}), \quad p(\mathbf{y}_{\text{new}} | \mathbf{f}) = \mathcal{N}(\mathbf{y}_{\text{new}}; \mathbf{f}, \sigma_{\text{noise}}^2 I), \\ \hat{\mathbf{y}} &= \begin{bmatrix} \mathbf{y}_{\text{new}} \\ \mathbf{D}_{\mathbf{a}} \mathbf{S}_{\mathbf{a}}^{-1} \mathbf{m}_{\mathbf{a}} \end{bmatrix}, \quad \mathbf{K}_{\hat{\mathbf{f}}\mathbf{b}} = \begin{bmatrix} \mathbf{K}_{\mathbf{fb}} \\ \mathbf{K}_{\mathbf{ab}} \end{bmatrix}, \\ \Sigma &= \begin{bmatrix} \sigma_{\text{noise}}^2 I & 0 \\ 0 & \mathbf{D}_{\mathbf{a}} \end{bmatrix}, \text{ and } \mathbf{D}_{\mathbf{a}} = [\mathbf{S}_{\mathbf{a}}^{-1} - \mathbf{K}_{\mathbf{aa}}^{-1}]. \end{aligned} \quad (4)$$

Here, $\mathbf{a}$ and $\mathbf{b}$ represent the latent variables corresponding to the old inducing points (i.e., from the previous update step) and the new inducing points of the posterior variational distributions, respectively. The $\mathbf{f}$ denote the latents corresponding to $\mathbf{y}_{\text{new}}$. The solution optimal posterior distribution $q_{\text{opt}}(\mathbf{b})$ includes a prior term $p(\mathbf{b})$ to regularize the posterior and a likelihood term $\mathcal{N}(\hat{\mathbf{y}})$, which predicts the data labels using the current inducing variables $\mathbf{b}$ and the old variational distribution $q(\mathbf{a})$. This formulation ensures that the information from the old variational distribution $q(\mathbf{a})$ is retained while also incorporating the new information in $\mathbf{y}_{\text{new}}$. For details on the derivation, please refer to [28].

VI. EXPERIMENTS

In this section, we first evaluate our method’s performance using benchmarks on two datasets with three baseline approaches. The benchmarks include single-robot and multi-robot scenarios, as well as discrete sensing and continuous sensing sensor models. We then demonstrate the adaptive approach with a non-point sensing robot operating in a 3D space, emulating a drone equipped with a camera. The experiment demonstrates our method’s real-world applicability.

We used two datasets for our benchmarks—bathymetry and elevation data. The bathymetry data [29] was collected by the National Oceanic and Atmospheric Administration (NOAA) using an echosounder in the Mississippi Sound region, Mississippi. The elevation data [30] was collected by NOAA using an aerial lidar on Wrangell Island, Alaska.

Our benchmarks include three baselines—the adaptive and non-adaptive variants of the Continuous-Space Informative Path Planner (CIPP) [10], and SGP-IPP [31]. CIPP leverages CMA-ES, a genetic algorithm, to find informative paths that maximize mutual information (MI) in continuous spaces. We selected this method as a baseline because it is closely

related and capable of handling adaptive multi-robot IPP in continuous spaces. CIPP does not assume access to ideal hyperparameters; instead, the hyperparameters are updated after visiting each waypoint using the collected data. Non-Adaptive-CIPP is the non-adaptive variant of CIPP; it assumes access to ideal hyperparameters modeling the data field in the environment and uses these hyperparameters in a GP to measure MI. SGP-IPP is the non-adaptive variant of our SGP-based approach. Similar to Non-Adaptive-CIPP, it assumes access to the ideal hyperparameters and uses them in the SGP to find the solutions. Please refer to [31] and [11] for additional baselines that establish the performance of the SGP-based approach.

An RBF kernel [27] was used to model the correlations of the datasets in all our experiments. We evaluated the solution paths by estimating the data field in the environment using a GP. The GP was initialized with the kernel function hyperparameters learned from 1000 randomly sampled labeled points from the dataset. The data collected from each solution path was then used as the GP’s training set. GPs are non-parametric; as such, they only use the hyperparameters and data from the path to estimate the remaining data field. The root-mean-square error (RMSE) between the ground truth data and the GP’s estimate was used to quantify the solution paths. The benchmarks were executed five times with random initial hyperparameters on a node with 12 cores (Intel(R) Xeon(R) Gold 6154 3.00GHz CPU) and 64 GB of RAM.

Single Robot with a Discrete Sensing Model: In this benchmark, we configured all the IPP approaches to model a single robot with a discrete sensing model, i.e., the robot senses only at the vertices of the solution path. Note that we do not enforce a distance budget in this benchmark, making it a sequential adaptive sensor placement problem. Both the non-adaptive approaches used a full GP to learn the hyperparameters from the collected data. The GPs and the SGPs in this and the following benchmarks were optimized using a conjugate-gradient descent optimizer [32]. We repeated the experiment with different numbers of waypoints, ranging from 5 to 100, in increments of 5. The mean and standard deviation of the RMSE for each method on both datasets are shown in Figure 1(a) and Figure 1(b). In Figure 1(c) and Figure 1(d), we show the mean and standard deviation of the total IPP runtime for the non-adaptive approaches—SGP-IPP and Non-Adaptive-CIPP. For the adaptive approaches—Adaptive-SGP-IPP and CIPP—we report the mean and standard deviation of the average IPP update runtime, i.e., the time taken to update the future waypoints using the data collected up to the current waypoint.

The Adaptive-SGP-IPP approach consistently finds solutions with RMSE scores that are on par or better than those of the baseline approaches, including the non-adaptive approaches, which have access to significantly more prior information about the environment since they are given the optimal hyperparameters. Moreover, the Adaptive-SGP-IPP approach is up to 125 times faster than CIPP, making it more suitable for real-world robots with limited compute.

Single Robot with a Discrete Sensing Model and

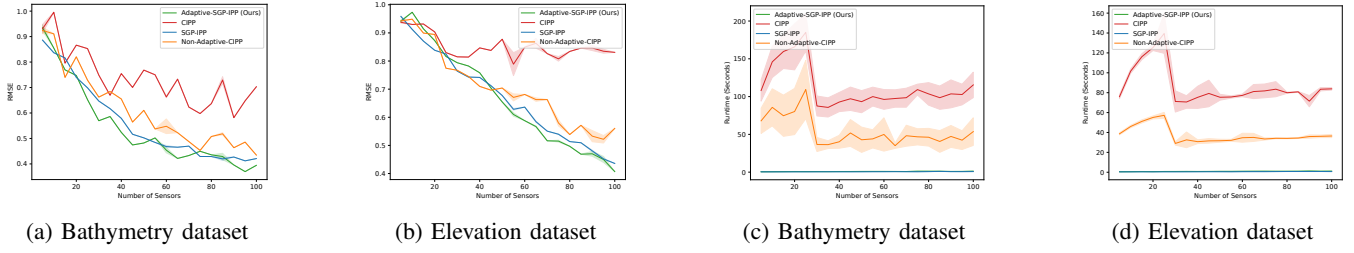


Fig. 1: Single robot adaptive IPP with a discrete sensing robot. Mean and standard deviation of the RMSE ((a) and (b)) and IPP update runtime ((c) and (d)); lower is better.

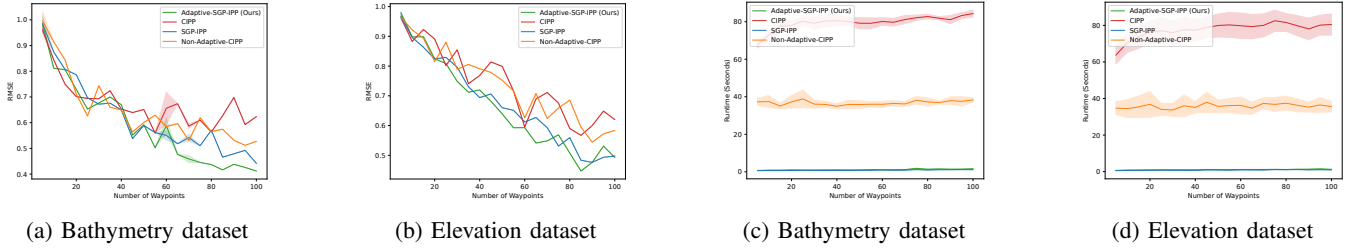


Fig. 2: Single robot adaptive IPP with a discrete sensing robot and a distance budget. Mean and standard deviation of the RMSE ((a) and (b)) and IPP update runtime ((c) and (d)); lower is better.

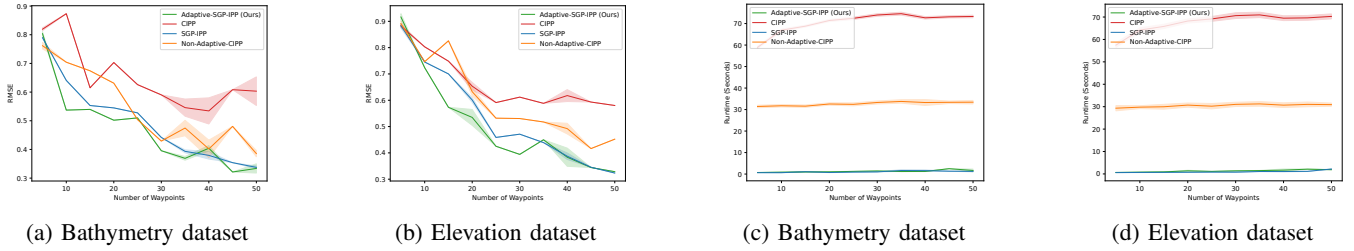


Fig. 3: Four robot adaptive IPP with discrete sensing robots and distance budgets. Mean and standard deviation of the RMSE ((a) and (b)) and IPP update runtime ((c) and (d)); lower is better.

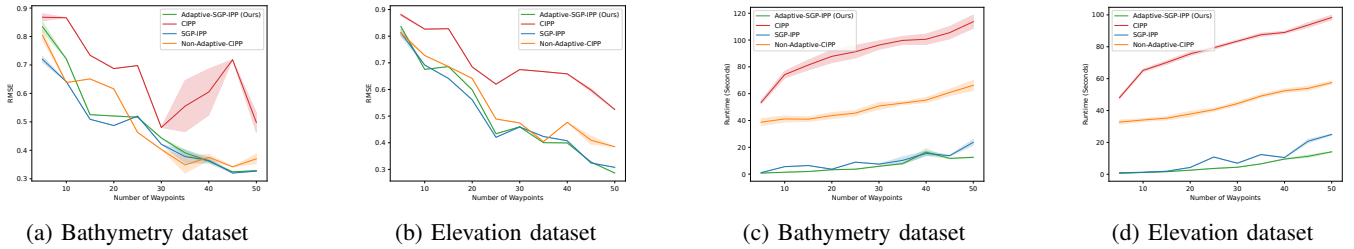


Fig. 4: Four robot adaptive IPP with continuous sensing robots and a distance budgets. Mean and standard deviation of the RMSE ((a) and (b)) and IPP update runtime ((c) and (d)); lower is better.

Distance Budget: We repeated the previous experiment with an added distance budget constraint. We set the distance budget to be proportional to the number of sensing locations; please refer to our code for the details. Figure 2 shows the results; the Adaptive-SGP-IPP approach finds solutions with RMSE scores on par or better than the baselines while being up to 70 times faster than CIPP.

Four Robots with a Discrete Sensing Model and Distance Budget: Next, we benchmark the centralized multi-robot IPP with a distance budget constraint. Since we collect four times the data, we limit the maximum number of waypoints to 50. We repeated the experiment ten times and

report the RMSE scores in Figure 3(a) and Figure 3(b) and show the algorithm runtimes in Figure 3(c) and Figure 3(d). Similar to the single-robot experiments, the Adaptive-SGP-IPP approach consistently achieves good RMSE scores with considerably lower runtimes.

Four Robots with a Continuous Sensing Model and Distance Budget: We repeated the above experiment with a continuous sensing model, i.e., the robots continuously sense along the entire solution path. Depending on the distance between each pair of waypoints, the number of data samples collected can exceed 1000, making it computationally expensive to update the hyperparameters using a GP. There-

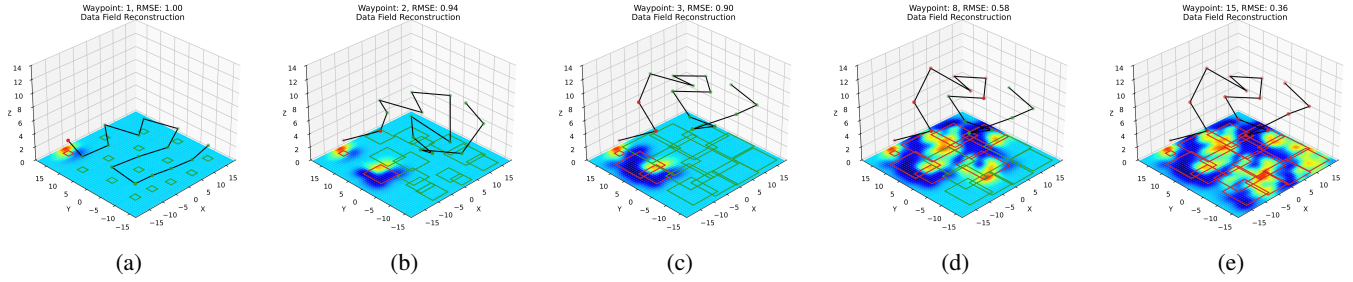


Fig. 5: Adaptive-SGP-IPP solution for a discrete sensing robot with a square height-dependent FoV area sensor; the heat map represents the data field reconstruction (elevation data). The waypoints in red are the visited locations, and the waypoints in green are the future waypoints. The squares represent the FoV of the sensor at each waypoint. We observe that the optimized waypoints balance the ground sampling resolution and the FoV area.

fore, we employ an SSGP optimized with the conjugate-gradient descent algorithm to learn the hyperparameters in both the adaptive approaches—Adaptive-SGP-IPP and CIPP as detailed in Section V. Additionally, all the approaches approximate continuous sensing robots by interpolating five additional waypoints between each pair of original waypoints, all of which are then used to compute the objective function: MI for CIPP-based approaches and the evidence lower bound (ELBO) for the SGP-based approaches. All the data collected along the traversed path is used to estimate the data field in the environment during evaluation. We report the results in Figure 4. The Adaptive-SGP-IPP approach shows strong RMSE performance and lower computation times, making it well-suited for adaptive IPP on real robots with minimal computational power.

Adaptive IPP with a Non-Point FoV in a 3D Space:

Next, we demonstrate the Adaptive-SGP-IPP approach with a single robot operating in a 3D space with a non-point field-of-view (FoV) discrete sensing sensor model. The robot is modeled to emulate an aerial drone equipped with a stereo-vision depth camera for mapping the elevation in a given area. This experiment used the elevation data from the NOAA elevation dataset [30]. Additionally, the non-point FoV was modeled to scale the 2D square area covered by the camera on the ground to be quadratically proportional to the height of the waypoint from the ground. Note that in our non-point sensor model used in the IPP approach, the uncertainty in the collected data is quadratically proportional to the height from the ground. We configured the Adaptive-SGP-IPP approach to visit 15 waypoints and used an SSGP for the hyperparameters, as the camera model captures large amounts of data at each waypoint. For additional technical details, please refer to our code. To demonstrate the method’s efficiency, we executed it on a Raspberry Pi 4 with 4 GB of RAM, a system-on-chip (SoC) commonly used in robots.

Figure 5(a) shows the initial path of the robot, which starts with uniformly distributed sensing locations close to the ground. The SSGP was initialized with an RBF kernel with a length scale of 1.0 m in both X and Y. Next, the method used the data collected at the first waypoint to update the hyperparameters using an SSGP and the path using the SGP approach. The SSGP updated the X and Y length scales to 4.53 m and 1.16 m respectively. Figure 5(b) shows the

updated path. We see that the waypoints have been shifted to be higher from the ground to ensure good coverage. Also, since the X-axis length scale is much larger than the Y-axis length scale, the method distributed the waypoints farther apart along the X-axis compared to the Y-axis.

Figure 5(d) shows the path at waypoint 8. The SSGP converged to length scales of 3.73 m and 3.42 m along the X and Y axes, respectively. Therefore, the waypoints are evenly distributed along both axes while still ensuring that the waypoints are high enough from the ground to ensure full coverage of the environment. Figure 5(e) shows the final traversed path, which captures all the critical regions of the environment using only the given number of waypoints. On average, the SSGP hyperparameter and path updates took 5.07 seconds and 8.17 seconds, respectively, demonstrating our approach’s real-world feasibility. Moreover, we implemented our code in Python; one can get even faster updates by implementing the method in Jax or C++.

VII. CONCLUSION

This paper addressed the adaptive informative path planning (IPP) problem. Most earlier approaches relied on computationally expensive discrete optimization methods, which do not scale well to large or 3D environments. We introduced a straightforward generalization of our fully differentiable IPP algorithm to handle adaptive IPP with significant performance benefits. The method accommodates both single and multi-robot IPP, as well as discrete and continuous sensing robots with path constraints. We provided benchmarks on two real-world datasets to demonstrate our approach’s significant computational efficiency and the informativeness of its solutions. Additionally, we showcased our approach for adaptive IPP in 3D space with a non-point sensing robot on a Raspberry Pi 4 system-on-chip computing platform with minimal computational resources. Our code is available in the SGP-Tools Python library, along with a companion ROS 2 package that can be deployed on ArduPilot-based robots. Future work will involve demonstrating our approach with real robots and exploring related problems, such as adaptive IPP for 3D surface inspection, decentralized IPP, and IPP with deep neural network-based path constraints to address complex scenarios, such as ensuring tethered underwater vehicles do not become entangled in the environment.

REFERENCES

- [1] K.-C. Ma, L. Liu, H. K. Heidarrsson, and G. S. Sukhatme, "Data-driven learning and planning for environmental sampling," *Journal of Field Robotics*, vol. 35, no. 5, pp. 643–661, 2018.
- [2] 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.
- [3] M. Popović, J. Ott, J. Rückin, and M. J. Kochenderfer, "Learning-based methods for adaptive informative path planning," *Robotics and Autonomous Systems*, vol. 179, p. 104727, 2024.
- [4] 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)*, pp. 1488–1494, 2021.
- [5] 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.
- [6] A. Singh, A. Krause, C. Guestrin, and W. J. Kaiser, "Efficient informative sensing using multiple robots," *J. Artif. Int. Res.*, vol. 34, p. 707–755, Apr. 2009.
- [7] 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.
- [8] 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.
- [9] F. Chen, S. Bai, T. Shan, and B. Englot, "Self-learning exploration and mapping for mobile robots via deep reinforcement learning," in *AIAA SciTech Forum*, p. 0396, 2019.
- [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] K. Jakkala and S. Akella, "Multi-robot informative path planning from regression with sparse Gaussian processes," in *IEEE International Conference on Robotics and Automation, ICRA*, (Yokohama, Japan), May 2024. <https://arxiv.org/pdf/2309.07050.pdf>.
- [12] J. Nocedal and S. Wright, *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering, Springer New York, 2006.
- [13] G. Hollinger and S. Singh, "Proofs and experiments in scalable, near-optimal search by multiple robots," in *Robotics: Science and Systems IV*, pp. 206–213, 2009.
- [14] J. Binney, A. Krause, and G. S. Sukhatme, "Optimizing waypoints for monitoring spatiotemporal phenomena," *The International Journal of Robotics Research*, vol. 32, no. 8, pp. 873–888, 2013.
- [15] 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.
- [16] K. C. T. Vivaldini, T. H. Martinelli, V. C. Guizilini, J. R. Souza, M. D. Oliveira, F. T. Ramos, and D. F. Wolf, "UAV route planning for active disease classification," *Autonomous Robots*, vol. 43, pp. 1137–1153, Jun 2019.
- [17] R. Mishra, M. Chitre, and S. Swarup, "Online Informative Path Planning Using Sparse Gaussian Processes," in *2018 OCEANS - MTS/IEEE Kobe Techno-Oceans (OTO)*, pp. 1–5, 2018.
- [18] G. E. Berget, J. Eidsvik, M. O. Alver, and T. A. Johansen, "Dynamic stochastic modeling for adaptive sampling of environmental variables using an AUV," *Autonomous Robots*, vol. 47, pp. 483–502, Apr 2023.
- [19] L. Schmid, M. Pantic, R. Khanna, L. Ott, R. Siegwart, and J. Nieto, "An efficient sampling-based method for online informative path planning in unknown environments," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1500–1507, 2020.
- [20] B. Moon, S. Chatterjee, and S. Scherer, "TIGRIS: An Informed Sampling-based Algorithm for Informative Path Planning," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5760–5766, 2022.
- [21] J. Ott, E. Balaban, and M. J. Kochenderfer, "Sequential Bayesian Optimization for Adaptive Informative Path Planning with Multimodal Sensing," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7894–7901, 2023.
- [22] 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.
- [23] A. Rao, I. Abraham, G. Sartoretti, and H. Choset, "Sparse Sensing in Ergodic Optimization," in *Distributed Autonomous Robotic Systems*, pp. 113–126, Springer Nature Switzerland, 2024.
- [24] Y. Cao, Y. Wang, A. Vashisth, H. Fan, and G. A. Sartoretti, "CAT-NIPP: Context-Aware Attention-based Network for Informative Path Planning," in *6th Annual Conference on Robot Learning*, 2022.
- [25] 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)*, pp. 4473–4479, 2022.
- [26] K. Jakkala and S. Akella, "Efficient Sensor Placement from Regression with Sparse Gaussian Processes in Continuous and Discrete Spaces," 2023. Preprint. <https://arxiv.org/pdf/2303.00028.pdf>.
- [27] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, 2005.
- [28] T. D. Bui, C. Nguyen, and R. E. Turner, "Streaming sparse Gaussian process approximations," in *Advances in Neural Information Processing Systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017.
- [29] NOAA, "H11617, NOS Hydrographic Survey in the vicinity of Mississippi Sound, Mississippi (2007)," 2007.
- [30] NOAA, "2023 AK DGGs Lidar DEM: Wrangell Island, AK," 2023.
- [31] 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.
- [32] C. Bishop, *Pattern Recognition and Machine Learning*. Springer, New York, 2006.