

Informative Planning in the Presence of Outliers

Weizhe Chen and Lantao Liu

Abstract—Informative planning seeks a sequence of actions that guide the robot to collect the most informative data to build a large-scale environmental model or learn a dynamical system. Existing work in informative planning mainly focuses on proposing new planners and applying them to various robotic applications such as environmental monitoring, autonomous exploration, and system identification. The informative planners optimize an objective given by a probabilistic model, e.g., Gaussian process regression (GPR). In practice, the ubiquitous sensing outliers can easily affect the model, resulting in a misleading objective. A straightforward solution is to filter out the outliers in the sensing data stream using an off-the-shelf outlier detector. However, informative samples are also scarce by definition so they might be falsely filtered out. In this paper, we propose a method to enable the robot to re-visit the locations where outliers were sampled besides optimizing the informative planning objective. The robot can collect more samples in the vicinity of outliers and update the outlier detector to reduce the number of false alarms. We achieve this by designing a new objective for the Pareto Monte Carlo tree search (MCTS). We demonstrate that the proposed framework performs better than applying an outlier detector naively.

I. INTRODUCTION

Intelligent robots need to sense the environment with onboard sensors and use the collected data to understand their surroundings before performing other subsequent tasks. *Informative planning* [1] seeks a sequence of actions that allow the robot to obtain the most informative data, i.e., the data that contributes the most to learning the environment model while minimizing the cost of traveling and sampling. Typically, the level of informativeness for sampling data from a certain location is quantified by the reduction of predictive uncertainty in a probabilistic model. For example, when dealing with spatial phenomena, a common choice for the probabilistic model is *Gaussian process regression* (GPR), and *mutual information* or *variance/entropy reduction* are commonly used to quantify the information contained in the data [2, 3].

Many informative planners have been proposed during the last two decades [4–9]. The majority of the literature assumes the sensing data is the ground-truth value corrupted by Gaussian noise. In practice, however, sensing data typically contains outliers due to various reasons. For instance, when facing specular reflection, LiDAR returns false values of maximum range even though the beams hit obstacles. When multiple exteroceptive sensors are in use, cross-talks among them also cause “random measurements” [10]. In outdoor environments, unexpected fast transient obstacles, such as

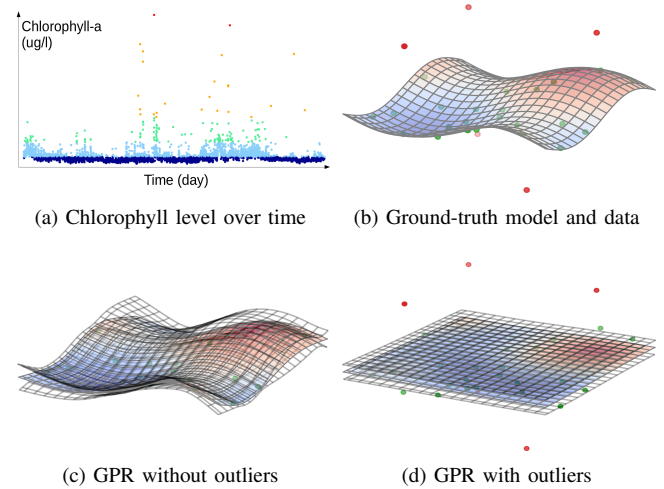


Fig. 1. (a) **Chlorophyll-a sensing data**. The orange and red points are outliers [11]. (b) **Ground-truth environmental model**. Green points are normal data points, and red points are outliers. Red and blue colors on the surface indicate high and low values, respectively. (c) **GPR without outliers** accurately predicts the environment and provides informative confidence bounds (i.e., the upper and lower black grids). (d) **GPR with outliers** leads to inaccurate “flat” prediction and uninformative uncertainty estimates, which further affects the performance of downstream informative planners.

insects, fish, or large dust particles, can also induce inevitable outliers. In addition, sensor wear and tear (e.g., worn probes), among other incomprehensible reasons that cause occasional hardware malfunction, can also introduce outliers. A Gaussian model can be easily distorted by outliers, rendering the “optimal” informative sampling decisions computed from such a model sub-optimal. Fig. 1 illustrates a scenario where outliers mislead the modeling and planning performances.

A straightforward idea to mitigate the influence of outliers is to filter them out from the sensing data stream using an off-the-shelf outlier detector. However, this leads to a high false-alarm rate because informative samples and outliers are statistically similar: *they both change the model significantly and are observed less frequently*. Otherwise, the informative samples would not bring much additional information. To reduce the high false-alarm rate, we propose to optimize two objectives simultaneously. In addition to the original information-seeking objective, the other objective encourages re-visiting the locations where outliers were sampled. In this way, the robot can collect more samples in the region where outliers are detected frequently and update the outlier detector to reduce the number of false alarms. We implement this idea by designing a new objective in the recently proposed Pareto Monte Carlo tree search [12]. We show

The authors are with the Luddy School of Informatics, Computing, and Engineering, Indiana University, Bloomington, IN, 47408, USA. {chenweiz, lantao}@iu.edu.

that the proposed framework performs better than naively applying a state-of-the-art outlier detector.

Although dealing with outliers is essential in practical applications of informative planning algorithms, it has not been well discussed in the literature to the best of our knowledge. The contribution of this paper is to point out the interesting interplay between informative samples and outliers: outliers can also be viewed as “extremely informative samples”. We developed a practical approach to significantly reduce the number of false alarms due to the interplay.

II. RELATED WORK

Informative planning has attracted growing interest, especially in the application of autonomous environmental monitoring [13]. Informative planning extends the optimal sensor placement problem [14] by considering the traversal costs and motion constraints of the mobile robots. Starting from the seminal work [15], various recursive greedy algorithms have been proposed [1, 16, 17]. These methods are based on the *submodularity* of the objective function and provide a performance guarantee. The submodularity requirement has been relaxed to *monotonicity* [18]. Dynamic programming (DP) based methods do not require the objective function to have these special properties. In [2], a sequence of informative waypoints is selected via DP by assuming the underlying map to be in a rectangular and sliced shape. This framework was extended to arbitrary continuous space by connecting the informative waypoints via traveling salesman problem solver [19]. This framework is lifted to online planning by integrating sparse GPs [3]. To develop an efficient planner, Hollinger and Sukhatme [7] extend sampling-based motion planners to robotic information gathering algorithms. Recently, this framework has been extended to online variants [4, 20]. Monte-Carlo tree search (MCTS) based methods are conceptually similar to sampling-based informative planners and have recently garnered great attention [6, 12, 21–23]. Instead of randomly growing the search tree, the MCTS expands tree nodes in a best-first search manner [24]. Since Gaussian processes have been the *de facto* standard for modeling spatiotemporal phenomena in many environmental monitoring applications, Bayesian optimization becomes a natural choice for informative planning [8, 25, 26]. New frameworks for informative planning are constantly emerging with attractive mechanisms such as evolutionary methods [5, 27, 28] and imitation learning [9]. Many of the existing works employ GPR as the probabilistic model. Compared to the large body of work in informative planners, investigations on the probabilistic model in the informative planning framework are relatively sparse. Methods using online sparse GPs [29] have been proposed to tackle the computational bottleneck. The mixture of GPs [30, 31] has also been applied to capture environmental non-stationarity. Different from the existing efforts, we investigate the effects of sensing outliers on model learning and informative planning. To the best of our knowledge, this is the first time that outliers in informative planning have been discussed.

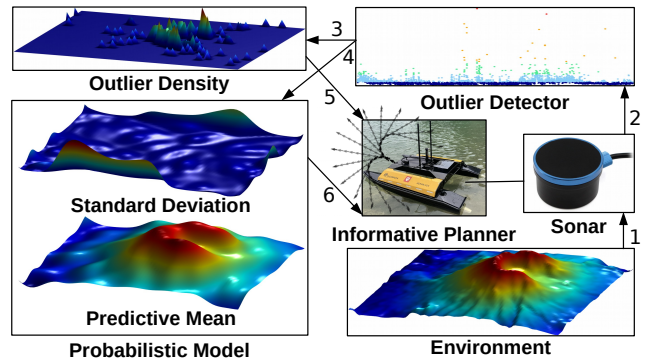


Fig. 2. **System diagram.** (1) The Autonomous Surface Vehicle (ASV) collects depth measurements via the down-facing sonar. (2) An outlier detector keeps track of the outlier density across spatial locations (3) and passes the filtered data to the probabilistic model (4). The outlier density (5) and the predictive standard deviation (6) are fed to the multi-objective informative planner to guide the robot to collect the next batch of data.

III. PROBLEM FORMULATION

Let $\mathcal{T} \triangleq \{1, 2, \dots, T\}$ be the set of decision epochs. At time $t \in \mathcal{T}$, a robot with a fully observed state $\mathbf{s}_t \in \mathcal{S}$ collects scalar measurement y_t via its sensor and takes an action a_t , arriving at the next state \mathbf{s}_{t+1} with transition probability $p(\mathbf{s}_{t+1}|\mathbf{s}_t, a_t)$. In this work, we assume the transition to be known and deterministic, and focus on inferring the noise-free environment state f_{env} from its noisy observation collected so far $\mathbf{y} = [y_1, \dots, y_N]^T$. One example is to estimate an elevation map from range sensor measurements. In this case, the elevation map is a function $f_{\text{env}}(\mathbf{x})$ of the input spatial locations $\mathbf{x} \in \mathbb{R}^D$. The robot maintains an internal model f of the environment learned from the noisy observations \mathbf{y} . The model is updated after taking new observations into account. Ideally, informative planning should use error reduction as the reward signal for the planner. Let $f_e(\mathbf{a})$ be a function that quantifies the modeling error reduction after incorporating the data obtained by taking an action sequence \mathbf{a} , and $f_c(\mathbf{a})$ be a function that returns the cost of executing \mathbf{a} . Informative planning solves the following optimization problem given an available budget B ,

$$\mathbf{a}^* = \arg \max_{\mathbf{a} \in \mathcal{A}} f_e(\mathbf{a}), \text{ s.t. } f_c(\mathbf{a}) \leq B, \quad (1)$$

where \mathcal{A} is the space of all possible action sequences. However, the ground-truth environment state f_{env} is unknown, hindering the computation of error reduction. Informative planning bypasses this problem by utilizing a surrogate reward function $f_i(\mathbf{a})$ given by a probabilistic model instead of $f_e(\mathbf{a})$. Typically, $f_i(\mathbf{a})$ quantifies the informativeness of the data obtained by taking actions \mathbf{a} by measuring the reduction in predictive uncertainty of the probabilistic model.

IV. METHOD

Fig. 2 illustrates the system with different modules and their relationship at a high level. The outlier detector takes raw sensing data, filters out the outliers, and outputs an outlier matrix indicating the density of outliers at each

location (Section IV-A). The clean data is fed into the GPR model which produces the predictive mean and standard deviation at each location (Section IV-B). The standard deviation matrix and the outlier matrix together form the multi-objective reward for the Pareto Monte Carlo tree search (Section IV-C). Finally, the robot collects samples along the optimal trajectory given by the planner.

A. Copula-Based Outlier Detection

COPula-based Outlier Detector (COPOD) is one of the top performing outlier detector that is efficient and tuning-free [32]. It is available in the open-source PyOD library [33] and can be readily used in our problem. Firstly, COPOD fits empirical left tail cumulative distribution functions (CDFs) $F_1^l(y) \dots, F_O^l(y)$ using

$$F_o^l(y) = \mathbb{P}((-\infty, y]) = \frac{1}{N} \sum_{n=1}^N \mathbb{I}(y_{n,o} \leq y), \quad (2)$$

and empirical right tail CDFs $F_1^r(y) \dots, F_O^r(y)$ by replacing y with $-y$, where O is the observation dimension. The skewness vector $\mathbf{b} = [b_1, \dots, b_O]$ is also computed using the standard estimation formula

$$b_o = \frac{\frac{1}{N} \sum_{n=1}^N (y_{n,o} - \overline{y_o})^3}{\sqrt{\frac{1}{N-1} \sum_{n=1}^N (y_{n,o} - \overline{y_o})^2}^3}, \quad (3)$$

where the overline $\overline{y_o}$ denotes the mean. Then, we apply $F_o^l(y)$ and $F_o^r(y)$ on each sample to get the empirical copula observations $l_{n,o}$ and $r_{n,o}$, respectively. The skewness corrected empirical copula observations are given by

$$s_{n,o} = \begin{cases} l_{n,o} & \text{if } b_o < 0, \\ r_{n,o} & \text{otherwise.} \end{cases} \quad (4)$$

Lastly, for each dimension, we compute the probability of observing a point at least as extreme as each $y_{n,o}$. The outlier score is the maximum negative log probability given by the left-tail, right-tail, and skewness-corrected empirical copula. Intuitively, outliers should occur less frequently, which means that the tail probabilities should be small, or equivalently, its negative log probabilities should be large. We refer the reader to [32] for more details.

B. Gaussian Process Regression

A Gaussian process (GP) is a collection of random variables, any finite number of which have a joint Gaussian distribution [34]. We place a Gaussian process prior over the function

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) \quad (5)$$

which is specified by a mean function $m(\mathbf{x})$ and a covariance function $k(\mathbf{x}, \mathbf{x}')$ (a.k.a. kernel). Popular choices of kernel functions include Matérn and Gaussian kernels. In this paper, we use a zero mean and the anisotropic Gaussian kernel

$$k(\mathbf{x}, \mathbf{x}') = \alpha^2 \exp \left(-\frac{1}{2} \sum_{d=1}^D \frac{(x_d - x'_d)^2}{\ell_d^2} \right). \quad (6)$$

The amplitude α^2 controls the variance of the random functions and ℓ_d defines the lengthscale, which informally can be thought of as the “radius of a neighborhood” in the d -th dimension. GP Regression (GPR) augments the model by assuming observations are corrupted by an additive Gaussian white noise

$$p(y|\mathbf{x}) = \mathcal{N}(y|f(\mathbf{x}), \sigma^2). \quad (7)$$

We will collect the parameters into $\boldsymbol{\theta} \triangleq \{\alpha, \ell_d, \sigma\}$ which are termed as *hyperparameters*.

According to the definition of GP and the likelihood in Eq. (7), the joint distribution of the observations \mathbf{y} and a set of latent function values \mathbf{f}_\star at arbitrary test input locations \mathbf{X}_\star is a multivariate Gaussian distribution

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_\star \end{bmatrix} \sim \mathcal{N} \left(\mathbf{0}, \begin{bmatrix} \mathbf{K}_y & \mathbf{K}_\star \\ \mathbf{K}_\star^\top & \mathbf{K}_{\star\star} \end{bmatrix} \right), \quad (8)$$

where $\mathbf{K}_y = \mathbf{K}_x + \sigma^2 \mathbf{I}$, \mathbf{K}_x is the covariance matrix given by the covariance function evaluated at each pair of observations, \mathbf{K}_\star is the covariance matrix between observations and the test function values \mathbf{f}_\star , and $\mathbf{K}_{\star\star}$ is the covariance matrix of the test function values. The predictive distribution for GPR is a conditional Gaussian distribution

$$p(\mathbf{f}_\star|\mathbf{y}) = \mathcal{N}(\mathbf{f}_\star|\boldsymbol{\mu}, \boldsymbol{\Sigma}), \text{ where} \quad (9)$$

$$\boldsymbol{\mu} = \mathbf{K}_\star^\top \mathbf{K}_y^{-1} \mathbf{y}, \quad (10)$$

$$\boldsymbol{\Sigma} = \mathbf{K}_{\star\star} - \mathbf{K}_\star^\top \mathbf{K}_y^{-1} \mathbf{K}_\star. \quad (11)$$

Learning in Gaussian process regression refers to determining appropriate values of the hyperparameters, and the common approach is to maximize the log marginal likelihood of the observations

$$\ln p(\mathbf{y}|\boldsymbol{\theta}) = -\frac{1}{2} (\mathbf{y}^\top \mathbf{K}_y^{-1} \mathbf{y} + \ln \det(\mathbf{K}_y) + N \ln(2\pi)), \quad (12)$$

where $\det(\cdot)$ denotes the matrix determinant.

C. Pareto Monte Carlo Tree Search

We treat the informative planning problem as a sequential decision making problem and approximate the solution using Monte Carlo tree search (MCTS) [24]. A node in the tree contains a state/pose and some statistics. The MCTS iterates in four steps: *node selection*, *expansion*, *simulation*, and *back-propagation*. We now describe the reward function and each step of the MCTS in detail.

Reward Functions. The reward of an action should be the amount of the “informativeness” of the collected data by taking this action. A natural choice for measuring informativeness is the mutual information between the visited locations and the remainder of the space [14]. However, calculating entropy involves the determinant computation of the predictive covariance matrix of the GPR in Eq. (11), which is computationally costly. A cheaper approximation is the reduction in the trace of the predictive covariance matrix, namely, variance reduction. Observing that the predictive uncertainty at a location becomes very small after sampling at that location, we further simplify the maximization of variance reduction to simply maximizing the sum of

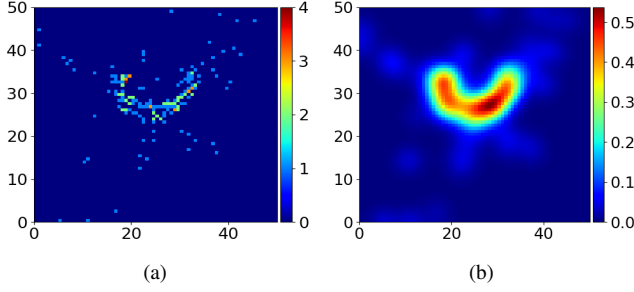


Fig. 3. (a) A matrix storing the outlier count at each spatial location. (b) A smoothed version of (a) using a Gaussian kernel. The smoothed outlier occurrence matrix serves as one of the reward map in Pareto Monte Carlo tree search to revisiting sampling locations with outliers.

predictive standard deviation along the sampling trajectory. To encourage re-visiting the locations where outliers were detected, another reward is the number of outlier occurrence. For better planning efficiency, we pre-compute the predictive standard deviation and the number of outliers on a *query grid* representing all discretized sampling locations before the tree search process. When evaluating the reward function values, we simply access the corresponding matrix elements. To better guide the robot, we also filter the outlier occurrence matrix via a Gaussian kernel. Fig. 3 shows the outlier occurrence matrix and its smoothed version.

Selection. As a best-first search algorithm, MCTS selects nodes that are expected to have a high reward but at the same time try other nodes sufficiently to avoid greedy choices. We trade-off *exploration* and *exploitation* using the tree policy. A well known tree policy is the upper confidence bound (UCB) [35, 36]. We recursively select child nodes with the highest UCB value until a node with unexpanded children is found. For node j , the UCB value adapted to our problem is defined by

$$\text{UCB}_j = \bar{f}_{ij} + C \sqrt{\frac{2 \log N_p}{N_j}}, \quad (13)$$

where N_p is the number of times that the parent of node j has been selected and N_j is the number of times that node j has been selected. The average reward \bar{f}_{ij} encourages the selection of nodes that currently look the most promising and thus stimulates exploitation. The value of the second term is large when N_j is small, which encourages exploring node i if it is not selected enough times. Constant C balances between the exploration and the exploitation which should be set to a similar scale as the rewards.

For vector reward given by the multiple reward functions, we use the Pareto variant of UCB proposed in [12]:

$$\text{PUCB}_j = \bar{\mathbf{f}}_{ij} + C \sqrt{\frac{4 \log N_p + \ln D_r}{2 * N_j}}, \quad (14)$$

where $\bar{\mathbf{f}}_{ij}$ is the average reward vector and D_r is the dimensionality of this vector. When selecting the best child node, we first compute the Pareto optimal set from the Pareto

Algorithm 1 The Proposed Framework

- 1: Collect the initial training data
- 2: Initialize and optimize gpr with the initial data
- 3: Initialize the outlier matrix `outlier` to be zeros
- 4: **while** has sampling budget **do**
- 5: `mean, std=gpr.predict(x_test)`
- 6: `trajectory=PUCT(pose, std, outlier)`
- 7: Follow trajectory and sample new data
- 8: Train the outlier detector on all collected data
- 9: Filter out the outliers in the new data
- 10: gpr appends the new data and optimize for several iterations

UCB vectors of all child nodes. The vectors in the Pareto optimal set cannot be improved for any objective without hurting other objectives. We then randomly select a child node from this set because the Pareto optimal solutions are considered equally optimal if no preference information is given. We refer the reader to [12] for further details of how Pareto MCTS balances multiple objectives. When the UCB tree policy is applied to a MCTS, the algorithm is commonly referred to as UCT. Similarly, we use *PUCT* to represent Pareto UCB applied to MCTS.

Expansion. We randomly select an action from the selected node and delete this action from the list of available actions. We then take this action and calculate the corresponding reward. A new node is created based on the new state and reward. We append this new node to the children list of the selected node.

Simulation. The goal of this step is to estimate the expected reward of the action of the newly expanded node by executing a default/rollout policy. Here we define the rollout policy to be “moving forward” for some steps. This policy is designed to estimate the expected reward along the direction of executing an action.

Back-propagation. We add the average reward received in simulation to each node along the selection path and increase the number of visits of these nodes. These four steps are repeated until the computational budget for the robot has reached. We recursively select the child node with the highest number of visits to get the final informative action sequence.

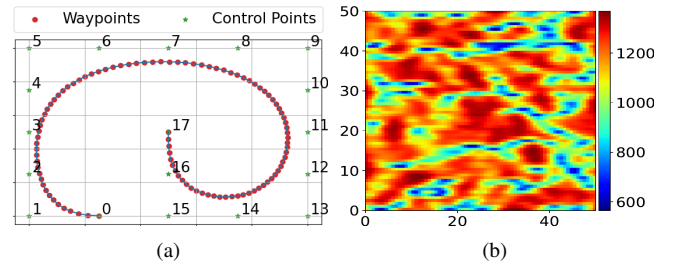


Fig. 4. (a) The Bezier curve to generate *a priori* sampling path for collecting initial training data for the initial optimization of the GPR and computing data pre/post-processing statistics. (b) Final predictive standard deviation of UCT-NONE.

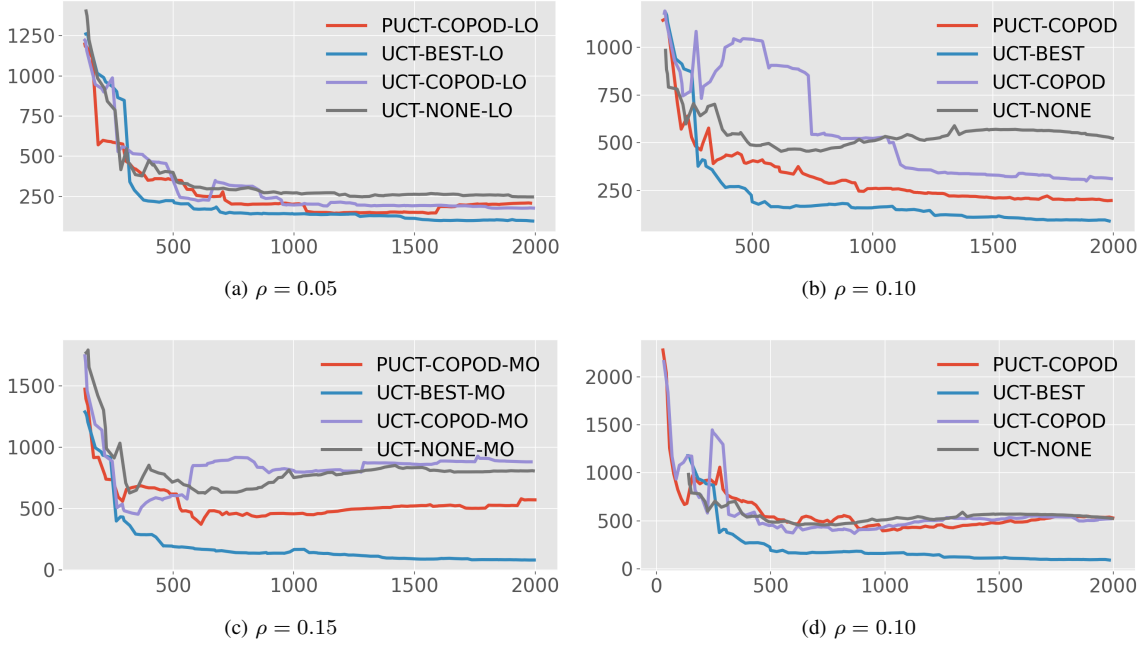


Fig. 5. **Root mean squared errors** of the proposed PUCT-COPOD and other three baselines. UCT-BEST filters outliers with the ground-truth labels, so it serves as a best-case baseline; UCT-NONE, which is the worst case baseline, does not filter outliers at all. UCT-COPOD directly applies the outlier detector to the sensing data stream. Parameter ρ controls the number of outliers: higher value indicates more outliers (MO) while lower value means less outliers (LO). In (a)-(c), the outlier detector only filters out the outliers in the newly acquired batch without changing the historical data. In (d), the outlier detector re-examine *all* the historical data at each decision epoch. (a) Performance difference of all the methods is insignificant when the number of outliers is very small. (b) PUCT-COPOD and UCT-COPOD are both in between the best-case baseline and the worst-case baseline after collecting 1000 samples while PUCT-COPOD performs better. At the early stage, UCT-COPOD is even worse than UCT-NONE because it filters out some informative samples. (c) When the number of outliers increases, UCT-COPOD loses advantage over UCT-NONE. The performance of PUCT-COPOD is also affected but its still better than UCT-COPOD and UCT-NONE. (d) Both PUCT-COPOD and UCT-COPOD are ineffective if the detector filters outliers in all historical data.

D. Overall Framework

Algorithm 1 summarized main steps of the proposed framework. First, we collect the *pilot* data following a path that does not depend on the informative planning framework. For example, we will use Bezier curve in Fig. 4(a). The pilot data is used for an initial optimization of the GPR hyperparameters through Eq. (12) and computing the statistics for pre/post-processing of the data. In a decision epoch, the robot searches for an optimal trajectory using PUCT based on the predictive standard deviation given by the GPR and the outlier matrix given by the outlier detector. The robot then collects samples along the trajectory. Finally, we detect outliers in this batch of new samples, get rid of them, feed the clean data to the GPR, and optimize the hyperparameters for several iterations.

V. EXPERIMENTAL RESULTS

We would like to answer the following questions.

- How do outliers affect an informative planning system if they are not filtered out? To answer this question, we use the standard GPR and UCT without any outlier detector as the worst-case baseline (UCT-NONE).
- What is the ideal performance if we have a perfect outlier detector? We use the ground-truth outlier labels as the prediction of a perfect outlier detector. We refer to this best-case competitor as UCT-BEST.

- What if we simply apply an off-the-shelf detector to deal with the outliers? In this method, we use COPOD to get rid of outliers (UCT-COPOD).
- Is the proposed method able to mitigate the negative effect of outliers? We denote our proposed method as PUCT-COPOD.

Considering that the second question requires the ground-truth outlier labels and to better understand the results, we inject some “spike” outliers to the sensing data which are similar to those in Fig. 1(a). The robot collects a batch of data along the planned trajectory in each decision epoch. We randomly select ρ portion of the batch for outlier injection. Specifically, we first sample a random amplitude uniformly from the range $[1, 2]$ with a random sign. Then, the amplitude is multiplied with the data range of the batch computed by the 0.05 and 0.95 quantiles. We use the elevation map of the Mount Saint Helens, shown in Fig. 6a, as the environment. The planar robot follows the Dubins car dynamics: $[\dot{x}_1, \dot{x}_2, \dot{\theta}] = [v \cos \theta, v \sin \theta, u]$, $u \in \mathcal{U}$, where $\mathbf{x} = [x_1, x_2]^T \in \mathbb{R}^2$ and $\theta \in [0, 2\pi)$ are the position and orientation of the vehicle, u is the control input that represents the robot’s steering angle. We set the linear velocity v to be a constant and only control the steering angle u . The possible steering angles $[-0.15, 0.15]$ (in radians) are discretized into 5 choices.

The GPR is initialized using 100 training samples gen-

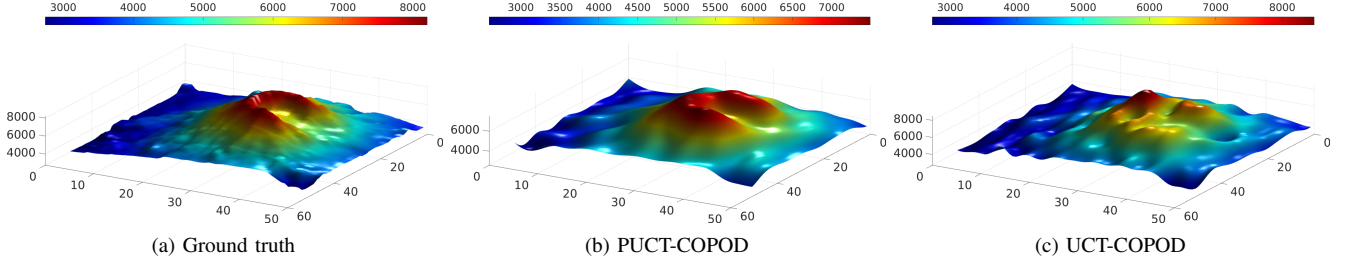


Fig. 6. Final predictive means of PUCT-COPOD and UCT-COPOD.

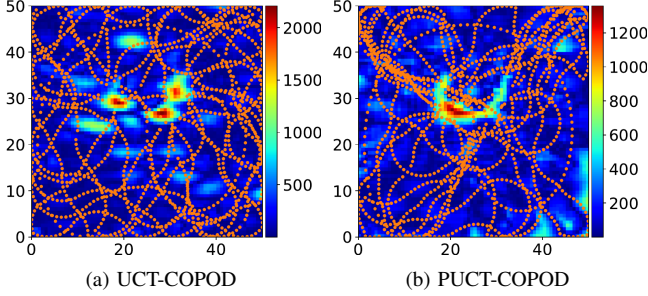


Fig. 7. The absolute error maps and training samples after cleaning the outliers of (a) UCT-COPOD and (b) PUCT-COPOD, respectively. Many informative samples at the high-variability region are falsely filtered out in (a), leading to some high-error “holes”. As a comparison, (b) shows that the proposed method can mitigate this effect to some extent.

erated by a Bezier curve shown in Fig. 4(a) and optimize the hyperparameters for 500 iterations. This *a priori* Bezier sampling path can be applied to any environment for collecting the *pilot* data for the initial GPR optimization and computing data pre/post-processing statistics. The path circles around the whole environment to help us get a sense of the workspace extent and a rough estimate of the observation range. Using the statistics computed from the initial training data, all inputs \mathbf{X} will be scaled to the range of $[-1, 1]$ and all observations \mathbf{y} will be standardized to have zero mean and unit variance. We use 500 tree search iterations and 5 rollout iterations. The sampling budget is set to 2000 samples.

Fig. 5 shows the root mean squared error (RMSE) of the four methods versus the number of training samples. When the number of outliers is very small ($\rho = 0.05$ in Fig. 5(a)), the difference among all the methods is negligible. After increasing ρ to 0.1, as expected, the error of UCT-BEST drops drastically and reaches the lowest error. After the decrease at the beginning, the error of UCT-None is actually increasing. The inclusion of outliers not only makes the prediction deteriorated but also messes up the standard deviation (see Fig. 4(b)), which further affects the downstream informative planning. The result of UCT-COPOD is interesting — although the final error is lower than that of UCT-NONE, it is the worst at the beginning. The reason for this can be seen from Fig. 7(a). Many informative samples at the high-variability region are falsely filtered out, leading to some high-error “holes”. As a comparison, Fig. 7(b) shows

that the proposed method can mitigate this effect to some extent. This brings better RMSE curve to PUCT-COPOD in Fig. 5(b). When there are more outliers ($\rho = 0.15$), UCT-COPOD loses the advantage over UCT-NONE. The performance of PUCT-COPOD is also affected but it is still better than UCT-COPOD and UCT-NONE. Finally, we also study the case where the detector filters outliers in all the historical data instead of the newly acquired batch. Both PUCT-COPOD and UCT-COPOD are ineffective in this case. One way to understand the result is that, when the robot has not sampled the volcanic area, the detector can easily detect extremely high or low values because the detector is trained on the historical data sampled from the flatland. However, after collecting samples from the volcanic area and retaining the detector, it is difficult for the detector to distinguish the aforementioned extreme values and the normal elevation measurements around the volcanic area during re-examination. In addition to the quantitative difference in the error reduction, the final predictions of the proposed method and the baseline are also qualitatively different. Fig. 6 shows that the prediction of PUCT-COPOD is closer to the ground-truth environment than that of UCT-COPOD.

VI. CONCLUSION

We present a framework to enable the robot to re-visit the locations where outliers were sampled besides optimizing the conventional informative planning objective. This is very different from existing informative planning approaches which mainly focus on planners that optimize over uncertainty or confidence based informativeness, which can be easily deteriorated by various sensing outliers, resulting in a misleading objective and sub-optimal sampling behaviors. We propose an approach to filter out the outliers from the sensing data stream using an off-the-shelf outlier detector. By designing a new planning objective with a Pareto variant of Monte Carlo tree search, our new framework allows the robot to collect more samples in the vicinity of the outliers and update the outlier detector to reduce the number of false alarms. Results show that the proposed framework performs much better than only applying the outlier detector.

VII. ACKNOWLEDGEMENT

We gratefully acknowledge the support of NSF grants with grant numbers 1906694, 4848921, and 4848922. We also thank the valuable comments from anonymous reviewers.

REFERENCES

- [1] Amarjeet Singh, Andreas Krause, Carlos Guestrin, William J Kaiser, and Maxim A Batalin. Efficient planning of informative paths for multiple robots. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, volume 7, pages 2204–2211, 2007.
- [2] Nannan Cao, Kian Hsiang Low, and John M Dolan. Multi-robot informative path planning for active sensing of environmental phenomena: A tale of two algorithms. In *Proceedings of the International Conference on Autonomous Agents and Multi-agent Systems (AAMAS)*, pages 7–14, 2013.
- [3] Kai-Chieh Ma, Lantao Liu, Hordur K Heidarrsson, and Gaurav S Sukhatme. Data-driven learning and planning for environmental sampling. *Journal of Field Robotics (JFR)*, 35(5):643–661, 2018.
- [4] Lukas Maximilian Schmid, Michael Pantic, Raghav Khanna, Lionel Ott, Roland Siegwart, and Juan Nieto. An efficient sampling-based method for online informative path planning in unknown environments. *IEEE Robotics and Automation Letters (RA-L)*, 2020.
- [5] Marija Popović, Gregory Hitz, Juan Nieto, Inkyu Sa, Roland Siegwart, and Enric Galceran. Online informative path planning for active classification using UAVs. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5753–5758, 2017.
- [6] Graeme Best, Oliver M Cliff, Timothy Patten, Ramgopal R Mettu, and Robert Fitch. Dec-MCTS: Decentralized planning for multi-robot active perception. *The International Journal of Robotics Research*, 38(2-3):316–337, 2019.
- [7] Geoffrey A Hollinger and Gaurav S Sukhatme. Sampling-based robotic information gathering algorithms. *The International Journal of Robotics Research*, 33(9):1271–1287, 2014.
- [8] Roman Marchant and Fabio Ramos. Bayesian optimisation for intelligent environmental monitoring. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2242–2249, 2012.
- [9] Sanjiban Choudhury, Mohak Bhardwaj, Sankalp Arora, Ashish Kapoor, Gireeja Ranade, Sebastian Scherer, and Debadeepta Dey. Data-driven planning via imitation learning. *The International Journal of Robotics Research*, 37(13-14):1632–1672, 2018.
- [10] Sebastian Thrun. Probabilistic robotics. *Communications of the ACM*, 45(3):52–57, 2002.
- [11] Benoit Espinola. *Statistical analysis of water data from an online EXO2 monitoring sonde: Modelling Chlorophyll-a using an EXO2 sonde*. PhD thesis, Metropolia University of Applied Sciences, Helsinki, Finland, 2017.
- [12] Weizhe Chen and Lantao Liu. Pareto Monte Carlo tree search for multi-objective informative planning. In *Proceedings of Robotics: Science and Systems (RSS)*, June 2019.
- [13] Matthew Dunbabin and Lino Marques. Robots for environmental monitoring: Significant advancements and applications. *IEEE Robotics & Automation Magazine*, 19(1):24–39, 2012.
- [14] Andreas Krause, Ajit Singh, and Carlos Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9(Feb):235–284, 2008.
- [15] Chandra Chekuri and Martin Pal. A recursive greedy algorithm for walks in directed graphs. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS’05)*, pages 245–253, 2005.
- [16] Alexandra Meliou, Andreas Krause, Carlos Guestrin, and Joseph M Hellerstein. Nonmyopic informative path planning in spatio-temporal models. In *Proceedings of the 22nd International Conference on Artificial Intelligence (AAAI)*, volume 10, pages 16–7, 2007.
- [17] Geoffrey Hollinger, Sanjiv Singh, Joseph Djugash, and Athanasios Kehagias. Efficient multi-robot search for a moving target. *The International Journal of Robotics Research*, 28(2):201–219, 2009.
- [18] Jonathan Binney, Andreas Krause, and Gaurav S Sukhatme. Optimizing waypoints for monitoring spatiotemporal phenomena. *The International Journal of Robotics Research*, 32(8):873–888, 2013.
- [19] Kai-Chieh Ma, Lantao Liu, and Gaurav S Sukhatme. An information-driven and disturbance-aware planning method for long-term ocean monitoring. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2102–2108, 2016.
- [20] Maani Ghaffari Jadidi, Jaime Valls Miro, and Gamini Dissanayake. Sampling-based incremental information gathering with applications to robotic exploration and environmental monitoring. *The International Journal of Robotics Research*, 38(6):658–685, 2019.
- [21] Akash Arora, P Michael Furlong, Robert Fitch, Salah Sukkarieh, and Terrence Fong. Multi-modal active perception for information gathering in science missions. *Autonomous Robots*, pages 1–27, 2018.
- [22] Philippe Morere, Roman Marchant, and Fabio Ramos. Sequential bayesian optimization as a POMDP for environment monitoring with UAVs. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 6381–6388, 2017.
- [23] Genevieve Flaspohler, Victoria Preston, Anna PM Michel, Yogesh Girdhar, and Nicholas Roy. Information-guided robotic maximum seek-and-sample in partially observable continuous environments. *IEEE Robotics and Automation Letters*, 4(4):3782–3789, 2019.
- [24] Cameron B Browne, Edward Powley, Daniel Whitehouse, Simon M Lucas, Peter I Cowling, Philipp Bohnlshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games*, 4(1):1–43, 2012.

- [25] Shi Bai, Jinkun Wang, Fanfei Chen, and Brendan Englot. Information-theoretic exploration with bayesian optimization. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1816–1822, 2016.
- [26] Chun Kai Ling, Kian Hsiang Low, and Patrick Jaillet. Gaussian process planning with lipschitz continuous reward functions: Towards unifying bayesian optimization, active learning, and beyond. In *Proceedings of the 30th International Conference on Artificial Intelligence (AAAI)*, 2016.
- [27] Marija Popović, Teresa Vidal-Calleja, Gregory Hitz, Jen Jen Chung, Inkyu Sa, Roland Siegwart, and Juan Nieto. An informative path planning framework for UAV-based terrain monitoring. *Autonomous Robots*, 44 (6):889–911, 2020.
- [28] Gregory Hitz, Enric Galceran, Marie-Ève Garneau, François Pomerleau, and Roland Siegwart. Adaptive continuous-space informative path planning for online environmental monitoring. *Journal of Field Robotics*, 34(8):1427–1449, 2017.
- [29] Kai-Chieh Ma, Lantao Liu, and Gaurav S Sukhatme. Informative planning and online learning with sparse gaussian processes. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4292–4298, 2017.
- [30] Wenhao Luo and Katia Sycara. Adaptive sampling and online learning in multi-robot sensor coverage with mixture of gaussian processes. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 6359–6364, 2018.
- [31] Ruofei Ouyang, Kian Hsiang Low, Jie Chen, and Patrick Jaillet. Multi-robot active sensing of non-stationary gaussian process-based environmental phenomena. In *Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 573–580, 2014.
- [32] Zheng Li, Yue Zhao, Nicola Botta, Cezar Ionescu, and Xiyang Hu. Copod: copula-based outlier detection. In *IEEE International Conference on Data Mining (ICDM)*, pages 1118–1123, 2020.
- [33] Yue Zhao, Zain Nasrullah, and Zheng Li. Pyod: A python toolbox for scalable outlier detection. *Journal of Machine Learning Research*, 20:1–7, 2019.
- [34] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005. ISBN 026218253X.
- [35] Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov):397–422, 2002.
- [36] Levente Kocsis and Csaba Szepesvári. Bandit based monte-carlo planning. In *European Conference on Machine Learning (ECML)*, pages 282–293. Springer, 2006.