Incorporating Noise Into Adaptive Sampling

Christopher E. Denniston, Aravind Kumaraguru, David A. Caron and Gaurav S. Sukhatme

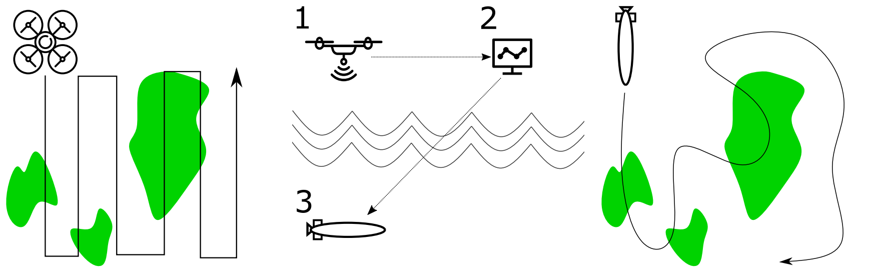
Abstract Adaptive sampling is important in robotic environmental monitoring, allowing a robot to intelligently select sampling locations to build an informative model of a phenomenon of interest. Most adaptive sampling techniques assume the localization noise does not vary with location, or that this variation is negligible, and thus do not model this behavior. In practice, the noise will vary greatly depending on the robot's trajectory and location. Additionally, prior surveys collected by other means, e.g., satellite or drone imagery, may use different state estimators or parameters. If these are used to drive sampling, this dependence may be significant. We provide a unified framework for adaptively collecting and modeling samples when heteroskedastic noise is present. Our framework is agnostic to the distribution of the noise. Our method outperforms others which do not take into account localization noise, validated by simulated trials and noise from a real state estimator.

1 Introduction

Adaptive sampling is the process of using robots to intelligently gather information about an environment. Typically the robot does this by creating an internal model and selecting sampling positions that improve the environment model over time [1].

Using adaptive sampling, a robot can create a high quality survey of an area of interest more efficiently (*e.g.*, in less time or by using less energy) than a pre-planned coverage pattern when certain key properties of the environment, like smoothness or isotropy, can be exploited. This has far reaching impacts in various fields such as the study of algae blooms, oil spill cleanup and search and rescue where the survey area may be expensive to survey or too large for a single vehicle to completely cover.

The authors are with the University of Southern California, Los Angeles, CA, USA {cdennist,akumarag,dcaron,gaurav}@usc.edu. This work was supported by the Southern California Coastal Water Research Project Authority under prime funding from the California State Water Resources Control Board on agreement number 19-003-150.



(a) (b) (c)

Fig. 1: A proposed mission. (a) A UAV performs a lawnmower pilot survey. (b) A base station compiles imagery and sends it to an AUV. (c) The AUV plans the sampling mission from this prior data and newly collected data.

Adaptive sampling systems generally assume that the samples are taken at precise locations. In practice, this assumption is violated because there is always some uncertainty in the robot's pose. This is especially true in the case of autonomous underwater vehicles (AUVs) when the robot does not have GPS and relies integrating inertial measurements. To model this uncertainty, a state estimator is used to derive a belief distribution over the current pose of the robot. Here, we utilize this distribution to more accurately model the phenomenon of interest instead of using the expectation of the robot's location as a point estimate. When the noise is heteroskedastic, the robot should pay more attention to areas of higher positional noise since the model in these regions can be improved with additional noisy samples [2]. We are motivated by the design of aquatic sampling missions (Fig. 1) that exploit a previously generated map of a phenomenon of interest from overhead imagery from an unmanned aerial vehicle (UAV) as a prior for an autonomous underwater vehicle (AUV). The adaptive sampling method must take into account both the uncertainty on the prior data from the UAV as well as the uncertainty of the sampling robot while it is progressing in the mission.

In this work we aim to extend prior methods for long horizon adaptive sampling and adaptive sampling with uncertain inputs to produce a method which can perform long term planning with localization uncertainty that is agnostic to the belief distribution about the robot state. Adaptive sampling with localization uncertainty and a finite budget can be formulated as follows

$$P^* = \underset{P \in \Phi}{\operatorname{argmax}} \mathbb{E}[F(P)] | c(P) \leq B \tag{1}$$

Where Φ is the space of all robot trajectories, B is some budget, F is some objective function and c is some cost function, P^* is the optimal trajectory [3]. In this work we maximize the expectation the agent has for the objective over the trajectory. This differs from the statement in [3] which does not use the expected objective, but the objective directly. Our work makes the same assumptions about the form of the cost function and objective function (*i.e.* the cost function is monotonically increasing and the objective function is submodular) as those in [3].

2 Related Work

The adaptive sampling survey [1] notes the usage of filters to remove sensor and localization noise, but does not cover methods explicitly incorporating this noise. Prior work [4] using UAV data to determine sampling locations, does not explicitly incorporate either the UAV or ground robot's localization noise. A framework for Bayesian optimization when a robot's pose is uncertain is found in [5]. The authors probabilistically bound the regret and show applications to robots, using a greedy method for objective maximization. A framework for multiple step prediction with uncertainty in input location with a Monte Carlo approximation is given in [6]. An approach [3] incorporating Rapidly Exploring Random Trees (RRT) into the adaptive sampling setting shows that a modified RRT algorithm (RIG-Tree) can maximize submodular functions. This assumes that there is certainty in the pose of the robot and that the pose can be fully specified. In [2] the effectiveness of informative path planning with uncertainty on the input is shown, with the key insight that modeling localization uncertainty can improve the overall mapping quality. This is achieved by augmenting the acquisition function with an explicit term for uncertainty. The modeling assumption in [2] assumes that the mean position is a good description of the state when the sample is taken, which may not be true in multimodal localization scenarios or when localization is particularly bad in some areas. Previous works have investigated using noisy samples from the environment to decrease the uncertainty in the robot pose using techniques similar to adaptive sampling [7]. Unlike our work, [7] does not try to use the characteristics of these noisy samples to approximate the underlying concentration. Our work extends the framework [2, 5] by utilizing the RIG-Tree algorithm to perform longer term planning, using a different modeling technique to incorporate noise, and allowing non-Gaussian sampling distributions.

3 Technical Approach

Gaussian processes are widely used modeling tools for adaptive sampling because they provide a non-parametric, continuous representation with uncertainty quantification. Gaussian processes approximate an unknown function from its known outputs by utilizing the similarity between points from a kernel function ($k(\cdot,\cdot)$) [8]. Function values y_* at any input location $\mathbf{x}_*$ are approximated by a Gaussian distribution:

$$y_*|y \sim \mathcal{N}\left(K_*K^{-1}y,\ K_{**} - K_*K^{-1}K_*^T\right) \tag{2}$$

where y is training output, y_* is test output, $\mathbf{x}$ is training input, and $\mathbf{x}_*$ is test input. K is $k(\mathbf{x},\mathbf{x})$, K_* is $k(\mathbf{x},\mathbf{x}_*)$, K_{**} is $k(\mathbf{x}_*,\mathbf{x}_*)$,

Gaussian Processes With Input Uncertainty: To incorporate localization uncertainty into the environmental model, we extend the framework in [8] for querying a Gaussian process with uncertainty on the input points. The Gaussian process posterior with uncertain inputs [6] is calculated as $p(y|D) = \int p(y|x)p(x)dx$ where y^* is the estimated sample value, x is a sampling location drawn from D (the belief distribution over states at that time from the state estimator). When it is possible to sample from D, Monte Carlo sampling can be used, resulting in $p(y|D) \simeq \frac{1}{N}\Sigma_{n=1}^{N} p(y|x^n)$ where x^n are independent samples from $p(D)$. Approximating $p(y|D)$ as normal by taking the mean and variance of the resulting samples yields $\bar{\mu}(D)$ and $\bar{\sigma}(D)$.

$$\bar{\mu}(D) = \mathbb{E}_x[\mu(x)] \qquad \bar{\sigma}^2(D) = \mathbb{E}_x[\sigma^2(x)] + var_x(\mu(x)) \tag{3}$$

We extend this formulation in two ways. We calculate the likelihood of the samples according to the Gaussian Process distribution and draw the sampled points from their distribution at every step. Critically, our formulation only depends on the ability to sample the robot localizer's belief state at past and future locations and is agnostic to the parameterization of the belief state, which allows this method to support localizers such as pose graphs, Kalman filters and particle filters.

Algorithm 1 Gaussian Process With Sample Uncertainty

procedure MONTE CARLO QUERY UNCERTAIN GP(θ, D_q)
  for $i \in iterations$ **do**
    $x_t^i \leftarrow sample(D_t)$
    $l^i \leftarrow P(y_t|GP(x_t^i, y_t, \theta))P(x_t^i|D_t)$
    $x_q^i \leftarrow sample(D_q)$
    $\mu_q^i, \sigma_q^i = GP(x_t^i, y_t, \theta).query(x_q^i)$
  $\mu \leftarrow \mathbb{E}_q[\mu_q]$
  $\sigma^2 \leftarrow \mathbb{E}_q[\sigma_q^2] + var_q(\mu_q)$

D_q is the belief distribution over the query points, D_t is the belief distribution over the previously collected points, l^i is the ith likelihood which is used for weighting and sub-sampling and θ are pretrained hyper-parameters.

Path Planning for Adaptive Sampling with Input Uncertainty: RIG-Tree [3] is an algorithm for long-horizon adaptive sampling planning. Each node in the tree keeps track of the total objective and cost along a path to that node. The objective functions for adaptive sampling generally use μ and σ from the Gaussian process model which can be extended to the stochastic case using Algorithm 1. The objective used for adaptive sampling is conditional entropy (which has also been used in prior work as the objective for sensor placement [9]).

$$f(x) = \frac{1}{2}\log(2\pi e \bar{\sigma}^2(D)) \tag{4}$$

Given the ability to calculate the stochastic objective function, we present Uncertain Input RIG-Tree, an extension of RIG-Tree to the stochastic case using Algorithm 1 to compute the expected information. Uncertain Input RIG-Tree must also extend the formulation of RIG nodes to keep track of the state of the filter at each proposed location along a trajectory. This is done in order to calculate the objective at proposed sampling location.

4 Experiments

4.1 Simulation Experiments

To test the performance of Algorithm 1 when applied to adaptive sampling, we test both the ability of the Monte-Carlo approximation to approximate the underlying distribution and the ability to approximate the objective function. Testing was done in environments designed to test interpolation methods (Eqs. (5) and (6)). x, y and z are the robot's global coordinates.

$$f(x,y,z) = 4(x-2+8y-8z^2)^2 + (3-4y)^2 + 16\sqrt{z+1}(2z-1)^2 \tag{5}$$

$$f(x,y,z) = 100(e^{-2/x^{1.75}} + e^{-2/y^{1.5}} e^{-2/z^{1.25}})^2 \tag{6}$$

Comparison of Monte-Carlo Methods In order to test the predictive power of Algorithm 1, we sample the environment at 1000 evenly spaced samples and add Gaussian noise. Permuted subsets of the samples are used as training data for the different methods to be tested. This training data are used to predict the ground truth and compare the error between the two. The data span a 3D cube with all dimensions in $[0,1]$. The noisy data were generated by adding Gaussian noise with $\sigma = 0.01$. The following are tested: A noiseless GP representing the ground truth, a traditional GP representing a model on noisy data without any corrections, subsampling the data based on likelihood, and weighing the data based on likelihood. Fig. 2 shows the outcome of this experiment, illustrating that likelihood weighting and keeping all samples produces the best estimate. When subsampling, *All* keeps all iterations, *100* keeps only the 100 most likely iterations and *Single* keeps only the most likely iteration.

Noisy Observations We compare our approach with RIG-Tree without doing noise modeling. Both methods use Eq. (4) as the objective, but the baseline does not use Algorithm 1 for computing the objective. The environments are the same as Fig. 2. The robot runs for a fixed time budget of 3000m in a 30x30x30m cube taking samples every meter. The internal model is compared to a ground truth model over 30^3 evenly spaced points. Prior data are given to the robot in the form of 10 by 10 samples (modeled after pictures from a UAV) taken as noisy samples from the function, with $\sigma = 5$. For the noise characteristics, we use a simplified noise model. The noise characteristics for the underwater vehicle are varied linearly from $\sigma = 1$ at minimum depth to $\sigma = 3$ at maximum depth in all spatial directions. Fig. 3 illustrates the results of this experiment. Using Monte Carlo sampling reduces the variance between

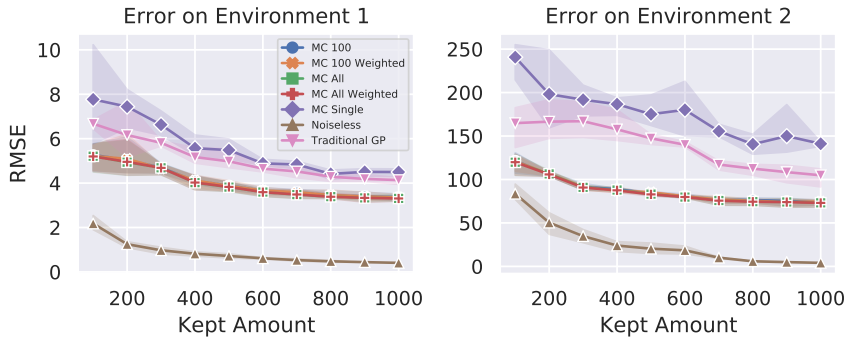


Fig. 2: Error when predicting from noisy data using Algorithm 1. Environment 1 is sampling Eq. (5), Environment 2 is Eq. (6). All MC methods were run for 1000 iterations. Six seeds each.

seeds and, in the first environment, lowers the average error. In environment two, the average error is comparable, but the variance between runs is smaller.

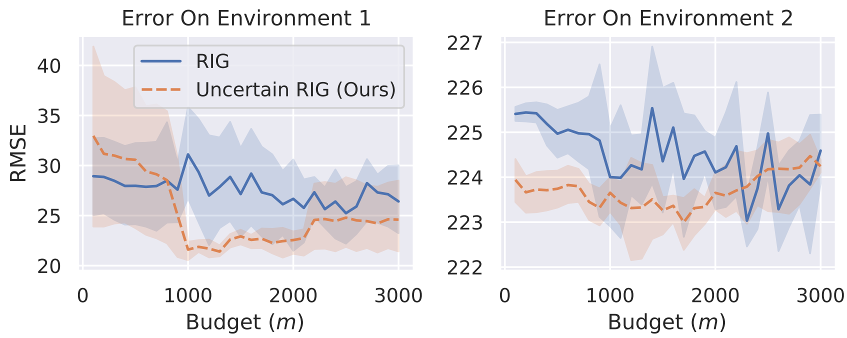
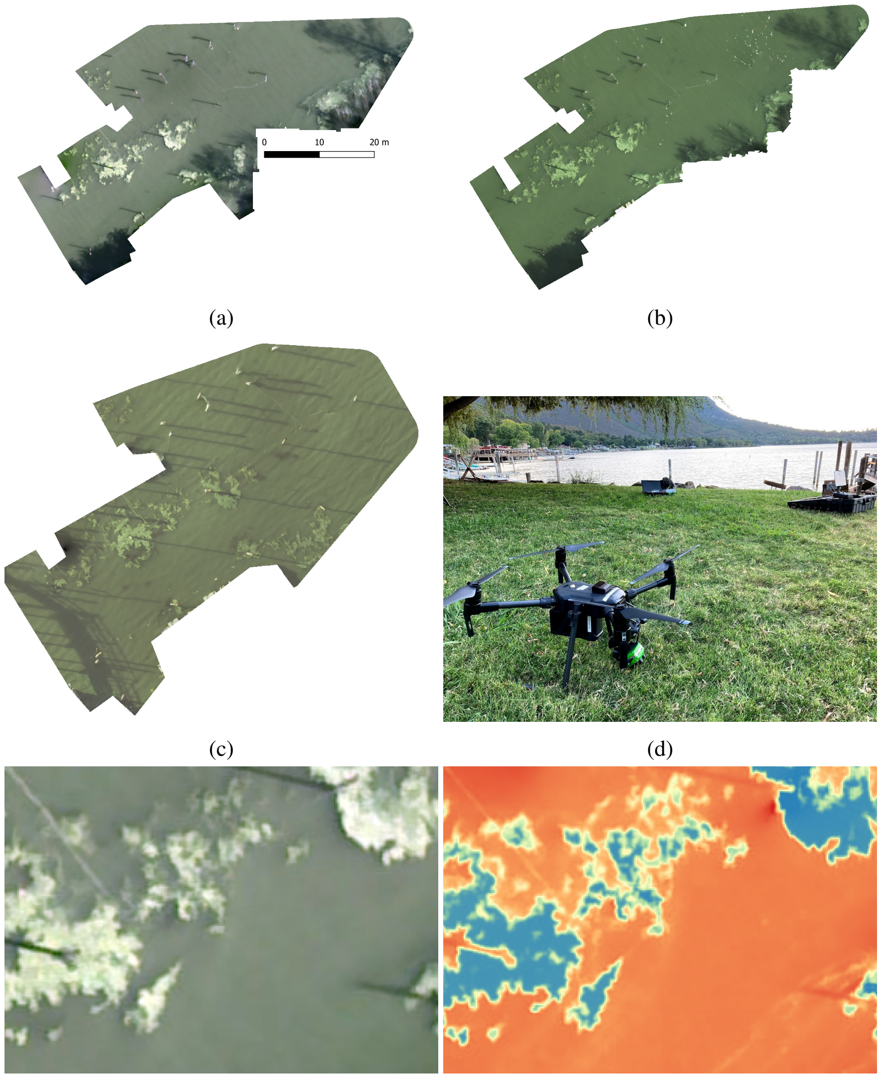


Fig. 3: Error when predicting from noisy data using Algorithm 1. Environment 1 is sampling Eq. (5), Environment 2 is Eq. (6). The robot is simulated for a fixed budget and compared against a noiseless evenly spaced ground truth. Each method runs with three seeds.

4.2 Experiments with Data Collected from Real Environment

In order to evaluate the usability of our algorithm on real world data we evaluate the error of a robot taking samples from orthomosaics created by flying a drone over water. These orthomosaics were created from data taken in Clear lake, California and created using structure from motion software. The UAV used to collect these images carries a hyperspectral imaging camera with a 710nm band. We use the reflectivity of this band as the desired concentration to be modeled.



(a) (b) (c) (d) (e) (f)

Fig. 4: Environments created from data gathered in Clear lake, California from data collected by a UAV with a hyperspectral camera (Fig. 4d). Fig. 4e shows an algal mass in true color while Fig. 4f shows the hyperspectral reflectivity at the target band.

Pose Graph-Based Localization To model the covariance of the agent at different locations, we take the GPS/IMU readings from the UAV and construct and solve a pose graph using GTSAM [10]. This is to model the error the robot would have if it were to plan using the raw imagery provided by the UAV, in a system like the one described in Fig. 1. The robot is run for 2000m, re-planning every 20m.

We compare with a baseline method which simply integrates the noisy samples and attempts to predict from those. This uses the RIG-Tree planner method similar to the last experiment. This is compared against two other methods. The first method does not use Algorithm 1 in calculating the variation for the objective equation, Eq. (4), but does use Monte Carlo estimation for computing the underlying concentration to compare for computing the error. The final method uses Algorithm 1 for calculating both the objective function and the estimate of the underlying concentration. The

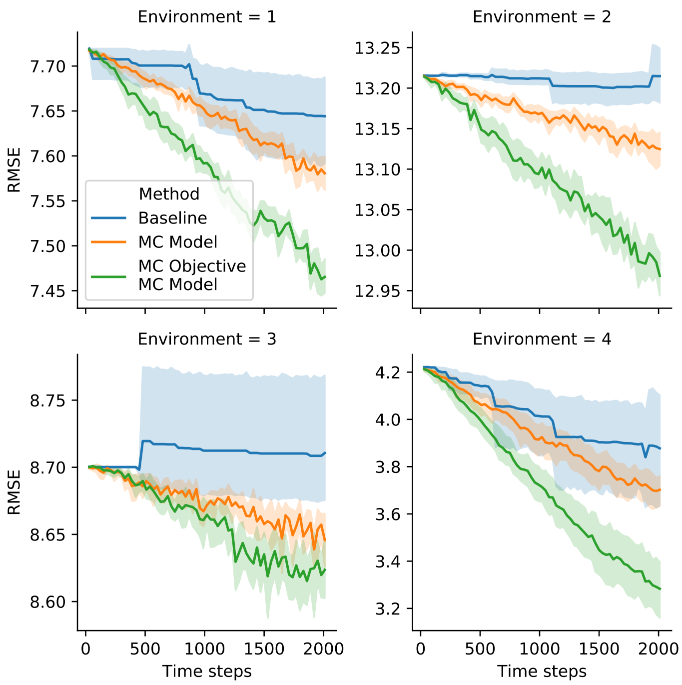


Fig. 5: Error between noisy samples and ground truth on four environments represented in Section 4.2. Noise is generated from a pose-graph based localization using the UAV's GPS/IMU. 10 seeds used.

results of the experiment are seen in Fig. 5. The full method outperforms both the baseline method and the method which does not plan with the added variance from the uncertainty in the robots pose. Both improvements work together to provide a boost in accuracy from both a better estimate of the underlying value and better planning to take more samples from higher noise areas. In environment two the baseline method on average performs better but has a very high variance in the final error, while the Monte Carlo methods tend to have a low variance in the end error. In environment four the robot cannot make a better model by taking noisy samples because there is too much noise on the samples to accurately reconstruct without doing a Monte Carlo estimation.

Particle Filter-Based Localization To test the ability of the proposed method to work with non-parameteric posterior distributions from state estimators, we simulate a surface vehicle which takes noisy ranges measurements from 10 landmarks and localizes itself using a particle filter. The robot is run for 2000m, re-planning every 20m. To update the localizer, the robot receives noisy measurements to 10 randomly placed landmarks with $\sigma = 3$. The filter is initialized with 100 particles normally distributed around the robot's true starting location. The baseline method uses the weighted mean of the particle filter as the sampling location while the Monte-Carlo methods sample the distribution of particles according to their weight to sample from D. To simulate the filter when planning, the particles and weights are added to each node. When extending samples, the particle filter is simulated forward with the parent's particles and weights in order to calculate the objective for the sampling location. As seen in Fig. 6 the Monte-Carlo based methods outperform the baseline method with a non-Gaussian posterior.

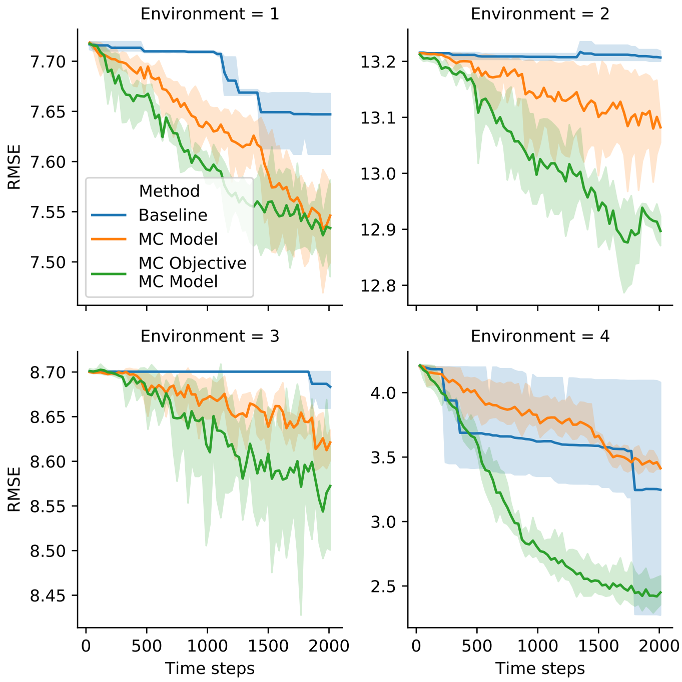


Fig. 6: Error between noisy samples and ground truth on four environments collected from the environments represented in Section 4.2. A particle filter with noisy measurement to 10 landmarks is used. 3 seeds run.

In environment four, the Monte Carlo run is outperformed by the baseline method. This environment has the lowest data range of all the environments and may not benefit from using a Monte Carlo estimation as much. Both Monte Carlo methods do suffer from high variance which could potentially be reduced with more iterations of the model estimator.

5 Conclusion

We propose a novel way to incorporate localization noise into adaptive sampling. When there is sufficient and varying noise on the state observations the robot's state estimator will not be accurate. This is common in field scenarios where the pose observations contain noise. We find that frameworks which use the state estimator's maximum likelihood estimate of the robot's pose may fail to properly model the true underlying field being sampled. By doing Monte-Carlo estimation from the belief distribution over robot states generated by the localizer, we are able to better estimate the concentration of an external field from sparse samples compared to previous methods. We show that the field estimation may be poor if the uncertainty in the robot's state estimator is not taken into account. Our framework is a novel approach for adaptive sampling under different types of robot pose uncertainty, such as those generated by a state of the art pose estimation framework. We show that doing a Monte-Carlo estimate of both the objective function during planning and the field estimation improves the planning and model used in adaptive sampling. This modification of an existing adaptive sampling algorithm improves performance when modeling from noisy pose observations and reduces the variance in the error between the robot's model and ground truth. We validate our findings on synthetic functions showing that the plan and model can be improved by our method. We also validate our findings on four real world data-sets from aerial imagery over a lake using complex state estimators from real world dynamics.

References

1. J. Hwang, N. Bose, and S. Fan, "AUV Adaptive Sampling Methods: A Review," *Applied Sciences*, vol. 9, p. 3145, Jan. 2019.
2. M. Popovic, T. A. Vidal-Calleja, J. J. Chung, J. Nieto, and R. Siegwart, "Informative Path Planning for Active Mapping under Localization Uncertainty.," in *Robotics & Automation Letters*, 2019.
3. G. A. Hollinger and G. S. Sukhatme, "Sampling-based robotic information gathering algorithms," *IJRR*, vol. 33, no. 9, pp. 1271–1287, 2014.
4. T. Manderson, S. Manjanna, and G. Dudek, "Heterogeneous Robot Teams for Informative Sampling," *Workshop on Informative Path Planning and Adaptive Sampling at Robotics Science and Systems*, June 2019.
5. R. Oliveira, L. Ott, and F. Ramos, "Bayesian Optimisation Under Uncertain Inputs," *2nd International Conference on Artificial Intelligence and Statistics (AISTATS 2019)]*, Feb. 2019.
6. A. Girard, C. E. Rasmussen, J. Q. Candela, and R. Murray-Smith, "Gaussian Process Priors with Uncertain Inputs Application to Multiple-Step Ahead Time Series Forecasting," in *Advances in Neural Information Processing Systems 15*, pp. 545–552, MIT Press, 2003.
7. N. Xu, K. H. Low, J. Chen, K. K. Lim, and E. B. Ozgul, "GP-Localize: Persistent Mobile Robot Localization using Online Sparse Gaussian Process Observation Model," *arXiv:1404.5165 [cs, stat]*, Apr. 2014. arXiv: 1404.5165.
8. C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning.* MIT press, 2006. ISSN: 0129-0657.
9. C. Guestrin, A. Krause, and A. P. Singh, "Near-optimal sensor placements in gaussian processes," in *Proceedings of the 22Nd International Conference on Machine Learning*, ICML '05, (New York, NY, USA), pp. 265–272, ACM, 2005.
10. C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "IMU Preintegration on Manifold for Efficient Visual-Inertial Maximum-a-Posteriori Estimation," in *Robotics: Science and Systems*, 2015.