Information-Theoretic Exploration with Bayesian Optimization

Shi Bai, Jinkun Wang, Fanfei Chen and Brendan Englot

***Abstract*— We consider an autonomous exploration problem in which a mobile robot is guided by an information-based controller through an *a priori* unknown environment, choosing to collect its next measurement at the location estimated to be most informative within its current field of view. We propose a novel approach to predict mutual information (MI) using Bayesian optimization. Over several iterations, candidate sensing actions are suggested by Bayesian optimization and added to a committee that repeatedly trains a Gaussian process (GP). The GP estimates MI throughout the robot's action space, serving as the basis for an acquisition function used to select the next candidate. The best sensing action in the committee is executed by the robot. This approach is compared over several environments with two batch methods, one which chooses the most informative action from a set of pseudo-random samples whose MI is explicitly evaluated, and one that applies GP regression to this sample set. Our computational results demonstrate that the proposed method provides not only computational efficiency and rapid map entropy reduction, but also robustness in comparison with competing approaches.**

I. INTRODUCTION

We consider a mobile robot that has no prior knowledge of the contents of its environment and must make sequential decisions about where to travel next, comprising an autonomous exploration problem [1]. Specifically, we formulate an information-theoretic exploration problem in which the long-term goal is to reduce entropy throughout the robot's environment map, and the short-term goal is to perform the sensing action in each iteration that will maximize mutual information (MI), along the lines of [2]. We assume the robot is equipped with a range sensor and uses an occupancy grid [3] to represent and reason about the environment. Our solution to this problem is motivated by the recent work of [4] and [5], in which the former gave a rigorous proof that by maximizing mutual information, a robot will be driven to unexplored space, and the latter showed that supervised learning could be used to predict informative actions without evaluating the expected mutual information exhaustively for every possible action. It has also been shown in [6] over several test cases that maximizing information gain over short trajectories will produce efficient global exploration in practice. However, the approach of [5] can suffer when its prediction of information gain is provided insufficient support from its training data, which can be true in complex environments with narrow corridors.

In this work, we propose *actively* selecting the candidate actions whose MI will be explicitly evaluated, using

S. Bai, J. Wang, F. Chen, and B. Englot are with the Department of Mechanical Engineering, Stevens Institute of Technology, Castle Point on Hudson, Hoboken, NJ, 07030. {sbai1, jwang92, fchen7, benglot}@stevens.edu

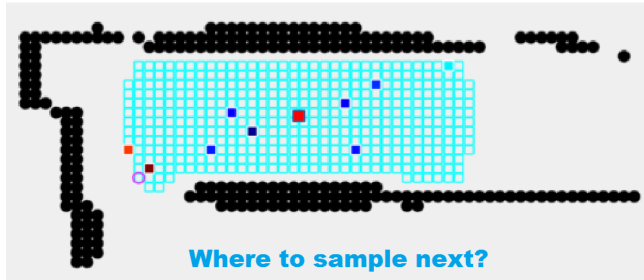


Fig. 1: An illustration of the proposed decision-making process. At each iteration, an existing set of sampled actions in the robot's current field of view (pixels shown in solid colors), whose anticipated MI has been explicitly evaluated by ray-casting, is used to select a new sensing action (purple circle at bottom left). The robot's current location is at the large red square, and the colors of the other solid pixels indicate their anticipated MI (with higher MI in red). Explicit MI evaluations for every candidate sensing action in this action space are shown in Figure 2c for comparison.

Bayesian optimization [7], which is an efficient approach when a cost function is expensive to evaluate [8], [9]. The use of such techniques in robot gait optimization [10], environmental monitoring [11], [12], and rough-terrain navigation [13] have shown the method to be effective in a variety of robotics applications. Specifically, we will estimate a robot's MI objective function using the posterior mean function of a Gaussian process (GP) [14]. An example of the proposed approach actively selecting a candidate for MI evaluation is illustrated in Figure 1. At each iteration of Bayesian optimization, a candidate sensing action is suggested and evaluated, then added to the pool of sensing actions used to approximate the MI objective function. The GP that estimates MI also forms the basis for the *acquisition function* used to select the next candidate sensing action. The most informative action in the pool will be executed.

A. Related Work

Among the earliest information-theoretic exploration strategies are those proposed by Whaite and Ferrie [15] and Elfes [2]. The former work proposes exploring an *a priori* unknown environment with the goal of minimizing entropy, and the latter work specifically proposes exploring to maximize the MI between sensor observations and an occupancy grid map. More recent works in information-theoretic exploration have considered the trade-off between maximizing MI and managing the localization uncertainty in a robot's simultaneous localization and mapping (SLAM) process [16], [17], in addition to the selection of trajectories that maximize map accuracy [18]. Efforts to reduce the computational cost of evaluating MI over many possible future measurements have considered small, carefully selected sets of candidate trajectories, using a skeletonization of the known occupancy map [19] and the evaluation of information gain over a finite number of motion primitives [20], [21], 3D viewpoints [22], or exclusively along the frontiers between known and unknown map regions [23], [24], which is effective in 2D environments.

Bayesian optimization has been applied to robotics problems in domains spanning from policy learning to perception. It has been used to monitor high-concentration areas of unknown spatially and temporally varying scalar fields [11], [12], minimize the vibrations experienced while navigating rough-terrain environments [13], optimize the speed and smoothness of a bipedal gait [10], and learn policies that reduce localization uncertainty in the presence of unknown landmarks [25]. However, to the best of our knowledge, Bayesian optimization has not been applied to the problem of action selection for the exploration of unknown environments modeled by occupancy maps, the goal of which is rapid and complete discovery of the contents of the environment.

B. Paper Organization

We propose and evaluate a methodology to choose sensing actions for exploration using several iterations of Bayesian optimization, with the aim of selecting near-optimal sensing actions consistently throughout the process of exploring an *a priori* unknown environment. A formal definition of the problem is given in Section 2, including brief introductions to Gaussian process (GP) regression and Bayesian optimization. The proposed algorithm is given in Section 3, and the time complexity of the algorithm is analyzed in Section 4. Computational results are presented in Section 5, with conclusions in Section 6.

II. PROBLEM DEFINITION

A. Information Gain

We define the space of mobile robot sensing actions to be the configuration space $\mathscr{C} \subseteq \mathbb{R}^d$, a subset of d-dimensional Euclidean space (assumed 2D in the computational examples to follow). We assume the robot's range sensor provides a 360-degree field of view, and that its occupancy grid map, whose cells are independent, is discretized finely enough to represent the configuration space, in addition to serving as the robot's model of the environment. In the absence of obstacles, the robot is assumed capable of travel from any grid cell in the map to any other cell. A fundamental presumption in this formulation is that the robot's action space is a subset of the spatial configuration space; this, along with our other assumptions, are similar to those made in [4]. The implications of extending the proposed method to robot configuration spaces with more complex topologies will be discussed in Section 6.

We define Shannon's entropy [26] over an occupancy grid map m as follows:

$$H(m) = -\sum_i \sum_j p(m_{i,j}) \log p(m_{i,j}) \tag{1}$$

where index i refers to the individual grid cells of the map and index j refers to the possible outcomes of the Bernoulli random variable that represents each grid cell, which is either free or occupied. Cells whose contents have never been observed are characterized as $p(m_{i,j}) = 0.5$, contributing one unit of entropy per cell. Cells whose contents are perfectly known contribute no entropy to the summation.

We use mutual information $I(m, x_i)$ to evaluate the expected information gain with respect to a specific configuration x_i, defined as follows:

$$I(m, x_i) = H(m) - H(m|x_i) \tag{2}$$

where $H(m)$ is the current entropy of the map, and $H(m|x_i)$ is the expected entropy of the map given a new sensor observation at configuration x_i. Our goal is to choose the optimal configuration x^* whose sensing action maximizes the expected information gain.

$$x^* = \underset{x_i \in \mathscr{C}_{action}}{\operatorname{argmax}} \, I(m, x_i) \tag{3}$$

In (3), $\mathscr{C}_{action}$ represents the subset of the configuration space from which the robot's next sensing action will be selected, typically within a short distance of the robot's current location.

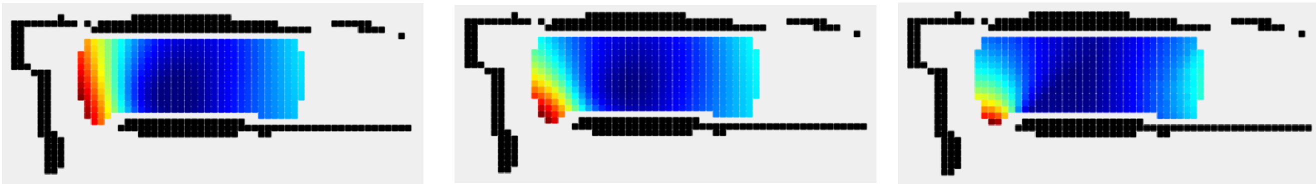
B. Bayesian Optimization

The candidate sensing action suggested by Bayesian optimization is computed using an *acquisition function*, which can take on high values where GP regression predicts high values of the MI objective function, and also in unexplored areas where uncertainty is high. An acquisition function may focus solely on improving the value of the current solution, or it can also adopt the framework of a multi-armed bandit problem, in which a tradeoff is managed among exploration and exploitation of our model of the objective function. We will adopt such an approach, using the acquisition function of the Gaussian process upper confidence bound (GP-UCB) algorithm [27], which is given in Equation 4:

$$x_t = \underset{x \in \mathscr{C}_{action}}{\operatorname{argmax}} \, \mu(x) + \beta \sigma(x) \tag{4}$$

where β is the tradeoff parameter between exploration and exploitation. $\mu(x)$ and $\sigma(x)$ are the predicted mean and variance derived from Gaussian process regression. It has also been proven in [27] that cumulative regret can be bounded using an optimal choice of the β parameter.

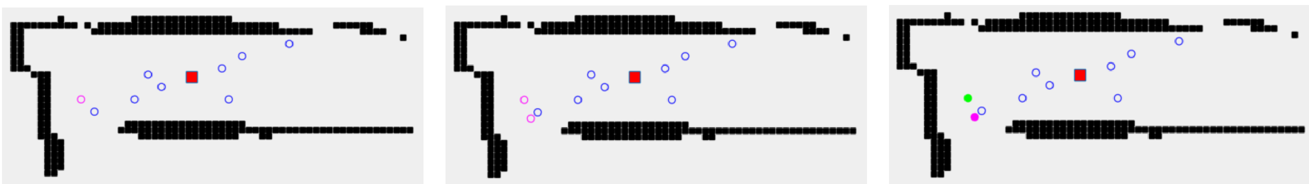
An example of how the acquisition function is used and updated is shown in Figure 2. An initial acquisition function based on 8 pseudo-random samples is shown in Figure 2a, and Figure 2b shows the function after one iteration of Bayesian optimization. Figure 2c shows the "ground truth" prediction of MI, which is exhaustively evaluated over all the possible actions in $\mathscr{C}_{action}$ using ray-casting over the



(a) Acquisition function based on 8 original evaluated sensing actions (in blue below).

(b) Acquisition function after adding the first sample suggested by BayOpt.

(c) Ground Truth: MI evaluated at all grid cells in $\mathscr{C}_{action}$.



(d) The first action selected by BayOpt, shown in purple.

(e) The second action selected by BayOpt, shown alongside the first.

(f) The most informative action from [5] (green), vs. the proposed method (purple)).

Fig. 2: Using Bayesian optimization (BayOpt) to select a highly informative sensing action.

occupancy map. Figures 2d and 2e show the sensing actions suggested by Bayesian optimization in consecutive iterations, using the functions illustrated in the figures above. In Figure 2f, the best sensing action obtained from this approach is shown alongside the best action from the approach of [5], which applies a GP regression to 10 pseudo-random samples, and chooses the action predicted to offer the highest MI.

C. Gaussion Process Regression

We assume a set of training data $\mathbf{x}$ represents the candidate sensing configurations x_i for which $I(m,x_i)$ has been computed. The values of $I(m,x_i)$ for all $x_i \in \mathscr{C}_{action}$ comprise the set of training outputs $\mathbf{y}$. GP regression [28] estimates the output values and corresponding covariance associated with a set of test configurations $\mathbf{x}_*$, according to Equations (4) and (5). The test configurations $\mathbf{x}_*$ will be finely discretized, with the same resolution as the occupancy grid map.

$$\bar{\mathbf{y}}_* = k(\mathbf{x}_*,\mathbf{x})[k(\mathbf{x},\mathbf{x})+{\sigma_n}^2 I]^{-1}\mathbf{y} \tag{5}$$

$$cov(\mathbf{y}_*) = k(\mathbf{x}_*,\mathbf{x}_*) - k(\mathbf{x}_*,\mathbf{x})[k(\mathbf{x},\mathbf{x})+{\sigma_n}^2 I]^{-1}k(\mathbf{x},\mathbf{x}_*) \tag{6}$$

In the above equations, $\bar{\mathbf{y}}_*$ are the estimated values $I(m,x_{i*})$ for the test data $\mathbf{x}_*$, $cov(\mathbf{y}_*)$ is the covariance associated with these outputs, σ_n^2 is a vector of Gaussian noise variances associated with the observed outputs $\mathbf{y}$, and k($\mathbf{x}$,$\mathbf{x}$') is the kernel function, which gives a covariance matrix relating all pairs of inputs. We adopt a Matérn kernel function for this application, given by (7).

$$k(\mathbf{x},\mathbf{x}') = \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\frac{\sqrt{2\nu}|\mathbf{x}-\mathbf{x}'|}{\ell}\right)^{\nu} K_{\nu}\left(\frac{\sqrt{2\nu}|\mathbf{x}-\mathbf{x}'|}{\ell}\right) \tag{7}$$

In (7), ν is a parameter used to vary the smoothness of the covariance, ℓ is a characteristic length, Γ is the gamma function, and K_ν is a modified Bessel function. In contrast with the squared exponential kernel function, which is more commonly used in Gaussian process regression [28], the Matérn kernel can be tuned to capture sharp variations in the estimated outputs. This has met with success in Gaussian process occupancy mapping, in which sharp and sudden transitions in occupancy probability due to obstacles are successfully modeled [23], [29]. Similarly, we anticipate sharp variations in mutual information due to the presence of obstacles, which will obstruct the visibility of some areas and permit the observation of others.

III. ALGORITHM DESCRIPTION

Algorithm 1 $AutonomousExploration(x_{init}, m_{init}, InfoThreshold, N_{samples}, N_{iterations})$

$x_k \leftarrow x_{init}$; $m_k \leftarrow m_{init}$; $ActionHistory \leftarrow x_{init}$;
while $ActionHistory \neq \emptyset$ **do**
  $ActionSet \leftarrow \emptyset$;
  $MISet \leftarrow \emptyset$;
  for $x_i \in \mathscr{C}_{action}(x_k, N_{samples})$ **do**
    $MI \leftarrow ObservationPrediction(x_i, m_k)$
    $MISet \leftarrow MISet \cup MI$;
    $ActionSet \leftarrow ActionSet \cup x_i$;
  end for
  for $N_{iterations}$ **do**
    $x_{opt} \leftarrow BayesianOptimization(ActionSet, MISet)$;
    $MI \leftarrow ObservationPrediction(x_{opt}, m_k)$
    $MISet \leftarrow MISet \cup MI$;
    $ActionSet \leftarrow ActionSet \cup x_{opt}$;
  end for
  if $max(ActionSet) > InfoThreshold$ **then**
    $x_{k+1} \leftarrow BestAction(ActionSet)$;
    $ActionHistory \leftarrow ActionHistory \cup x_{k+1}$;
  else
    $x_{k+1} \leftarrow ActionHistory(PreviousAction)$;
    $ActionHistory \leftarrow ActionHistory \setminus x_k$;
  end if
  $m_{k+1} \leftarrow MapUpdate(x_{k+1})$;
end while

The exploration process proceeds according to Algorithm 1. In every iteration, an action set $\mathscr{C}_{action}$ is formulated within the sensor field of view at the robot's current location, and a designated number of sampled actions within the set is evaluated per Equation 2, drawn from a Sobol sequence to ensure a low-discrepancy set of samples [30]. After the evaluation of mutual information over the selected actions, a candidate action suggested by Equation 4 will be evaluated and added to the set of approved candidate actions *ActionSet*. This updated set of actions will be used for choosing the next sample, and so on. After the designated number of iterations of Bayesian optimization, the most informative action will be selected from *ActionSet* whose mutual information has been explicitly computed. If at least one action is identified whose information gain surpasses the designated threshold (which typically has value slightly greater than zero), the robot performs the maximally informative sensing action. However, if none of the actions evaluated surpasses the threshold, the robot takes a step backwards and considers the actions at a previous location along the route traveled, where there may have been informative candidate actions that were not yet performed. The algorithm terminates when the previously taken action set *ActionHistory* is empty. While the emphasis of this work is the efficient operation of a mutual information *controller* that selects informative sensing actions one-by-one, we note that more sophisticated global planning is possible to avoid becoming stuck in the "dead ends" that have already been explored [19], [21].

IV. ALGORITHM COMPLEXITY

The computational complexity of Algorithm 1 at every step of its while-loop is given in (8):

$$O(N_{samples}N_{beams}N_{cells}) + O(N_{iterations}(N_{samples}^3 + N_{samples}^2 N_{actions})) \quad (8)$$

where $N_{samples}$ is the number of designated configurations whose mutual information is explicitly evaluated, $N_{actions}$ is the total number of actions comprising $\mathscr{C}_{action}$ that are estimated using Gaussian process regression, N_{beams} is the number of beams emitted by the robot's range sensor, and N_{cells} is the worst-case number of occupancy grid cells that a beam may intersect. $N_{iterations}$ is the number of GP regressions performed, equivalent to the number of iterations in which Bayesian optimization is applied. The term $N_{samples}N_{beams}N_{cells}$ represents the cost of explicitly evaluating mutual information in all cells intersected by the robot's sensor, for all designated actions $N_{samples}$. The term $N_{samples}^3 + N_{samples}^2 N_{actions}$ represents the cost of performing the subsequent Gaussian process regression, which requires the inversion of a matrix that is square in $N_{samples}$, and its subsequent multiplication with cross-covariance terms that scale with $N_{actions}$, the total number of sensing actions recovered from the "test data" of the Gaussian process regression. In practice, we have worked with $10 \leq N_{samples} \leq 20$, $N_{actions} \sim 300$, $N_{iterations} \sim 5$, $N_{beams} = 360$ (note that N_{beams} can often be much larger, e.g. 307200 for a Kinect sensor), and $N_{cells} \sim 25$, and we have found that in this range, the complexity of the procedure is dominated by the first term of (8), with the cost of the Gaussian process regression relatively minor in comparison to the cost of the mutual information computation. Hence, a much larger number of sensing actions can be evaluated approximately for a small additional cost on top of the initial evaluation of information gain over the original set of samples. Specific examples will be highlighted in the following section.

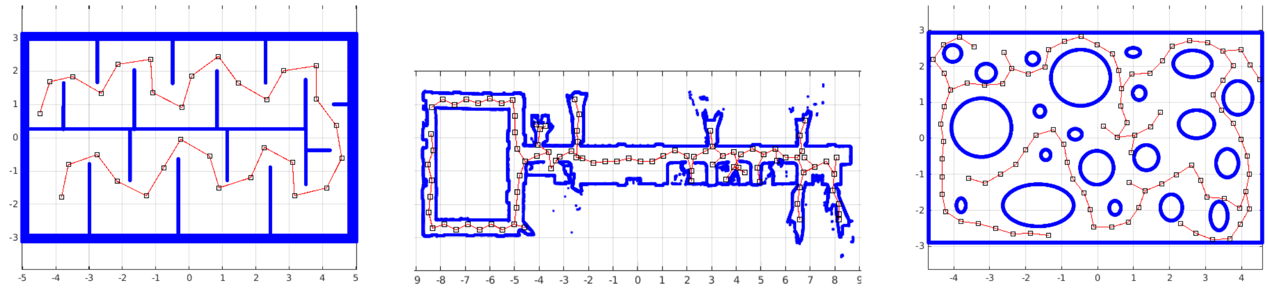
V. COMPUTATIONAL RESULTS

A. Experimental Setup

In our simulation of robot exploration, we assume a mobile robot is equipped with a laser scanner with a 360° field of view and 1° resolution. We assume the scanner is noiseless, and that a grid cell of the occupancy map is determined with certainty to be either free or occupied if it is intersected by any of the sensor's beams. The range of the laser scanner is assumed to be 1 meter, each grid cell is 0.01 meter in dimension, and all sensing actions considered are within 0.5 meters of the robot's current location, ensuring that the next sensing action lies within the current field of view to the extent that its outcome can be reasonably predicted by an MI evaluation over the existing map. We also assume the robot is able to localize accurately, which is often feasible with the aid of laser scan-matching [6], [33]. GP regression computations were performed with the aid of the Gaussian processes for machine learning (GPML) MATLAB library [32]. The hyperparameters selected for the Matérn kernel, tuned to optimize marginal likelihood using a small set of representative training data, were held constant across all maps studied, as was the acquisition function parameter β of Equation (4), which was tuned optimally following the procedure recommended in [27].

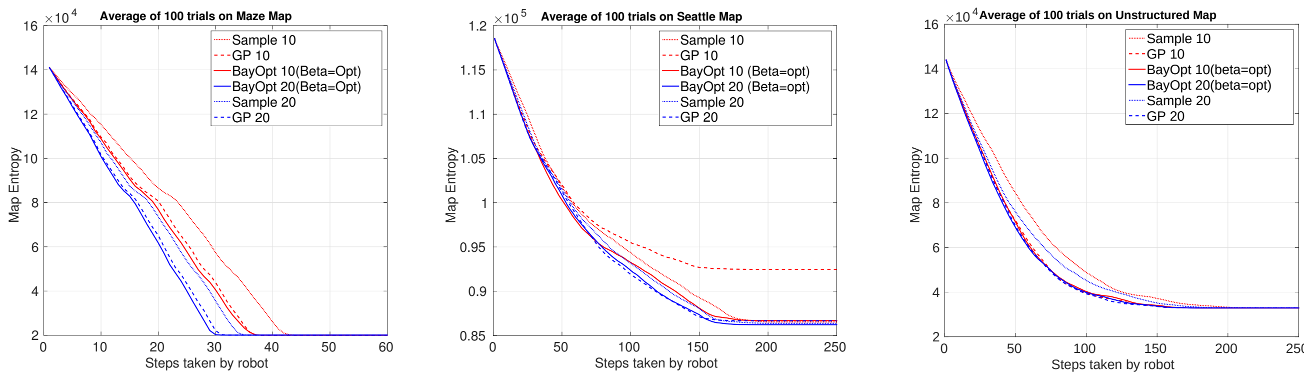
We explored the performance of our algorithm using three different maps: 1) a synthetic "maze" map representing an indoor environment (shown in Figure 3a); 2) the "Seattle map" from the Radish repository [31] (shown in Figure 3b), where any gaps in the building perimeter have been closed manually. and 3) a synthetic "unstructured" map representing a forest-like environment (shown in Figure 3c). The exploration process was simulated using MATLAB.

We initialized the robot randomly within each map and simulated 100 instances of exploration for each of the following cases: a) choosing the best action among 10 or 20 Sobol samples, termed the quasi Monte Carlo (QMC) approach below, b) using 10 or 20 Sobol samples as the basis for Gaussian process regression, and choosing the best action from the approximately continuous action space (as in [5]), and c) using 8 or 16 Sobol samples to bootstrap Bayesian optimization, and choosing the best action after 2 or 4 subsequent iterations of Bayesian optimization, respectively. The robot will terminate its exploration process after taking a designated number of steps (50-250 steps depending on the specific map), unless it terminates automatically beforehand according to Algorithm 1. This designated number of steps is introduced to allow the exploration process to terminate in instances when a robot becomes stuck in a local region of the map without eliminating all of the map's entropy. The computation required for each trial was distributed



(a) Synthetic maze map. (b) Adapted from [31] "Seattle map". (c) Synthetic unstructured map.

Fig. 3: The three different environments used in our computational experiments are visualized as completed occupancy grids. A representative execution trace of the robot exploration process using Bayesian optimization is illustrated in each map, where nodes represent the sensing actions used to construct the map, and edges represent the paths traveled between them.



(a) Results from the maze map. (b) Results from the Seattle map. (c) Results from the unstructured map.

Fig. 4: The results of 100 simulated robot exploration trials, for each of six parameterizations, using the maps of Figure 3. The mean entropy reduction is given over the number of sensing actions performed by the robot for all test cases considered.

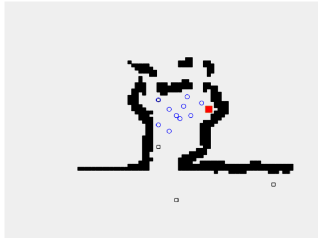
across four cores of an Intel i5 3.0 GHz processor using the MATLAB Parallel Computing Toolbox, on a computer equipped with 4GB RAM running Windows 7.

B. Results of Simulated Robot Exploration

Figure 4 gives results showing the performance of the six problem parameterizations over the maps of Figure 3. In the maze map, Bayesian optimization-based exploration drives down entropy to the maximum extent allowable at a faster rate than both the "batch" GP approach and the QMC approach for each number of sampled actions examined. Additionally, when relying on only 10 sampled actions to predict MI, Bayesian optimization achieves nearly the same performance as evaluating 20 actions from a predetermined QMC sample set (the "Sample 20" case in Fig. 4).

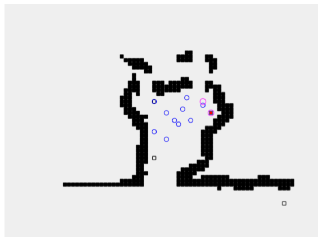
In the Seattle map, batch GP regression has difficulty when applied to only 10 sampled actions, in which case the final map entropy does not reduce to the same level as the other two approaches. However, for both 10 and 20 sampled actions, Bayesian optimization reduces entropy to the maximal extent encountered, at rates comparable to all competing methods. Two representative examples of the batch GP approach encoutering difficulty over this map are shown in Figure 5. In these cases, Bayesian optimization, in contrast, produces good suggestions of informative actions that will lead to further exploration. In the first example (Fig. 5a to 5d), batch GP regression predicts that the robot's current location will yield the largest information gain of any local sensing action (Fig. 5a), causing the robot to stay in the current map region and remain "stuck" until the algorithm terminates. In the second example (Fig. 5e to 5h), the robot's first step is shown as the red square in Figure 5f and the batch GP regression decision (Fig. 5e) is represented by the solid green circle. After the robot moves to the second location shown in Figure 5h, GP regression suggests a return to the previous position per the inference result shown in Figure 5g, thus causing the robot to loop between the two sensing actions until the algorithm terminates. In both of these examples, one iteration of Bayesian optimization (Fig. 5c and 5i) applied to the existing samples will suggest decisions similar to the MI "ground truth" (Fig. 5j).

Finally, in the unstructured map, all parameterizations using GP regression and Bayesian optimization perform better across the board, even when less computational effort is invested in establishing a training data set. In this case, GP regression and Bayesian optimization occasionally select actions from a different homotopy class than the competing QMC method, resulting in fundamentally different



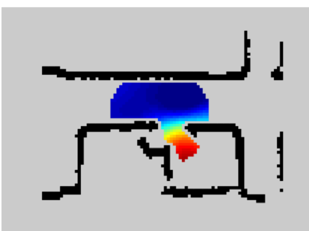
(a) Batch GP inference indicates the robot's current location is the most informative view point.

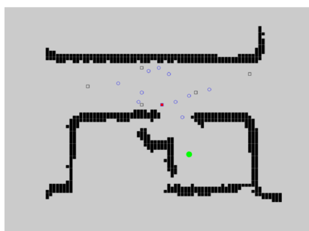
(b) Recommended action from (a) (red square) relative to the sample set used.



(c) Bayesian optimization suggests a more informative action, leading out of the room.

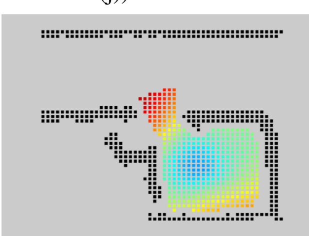
(d) Recommended action from (c) (red circle) relative to the sample set used.





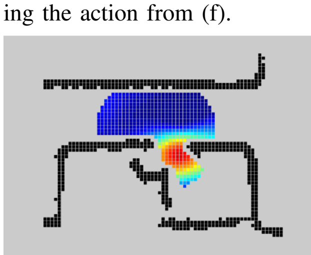
(e) Batch GP inference produces a poor estimate of the MI (truth shown in (j)).

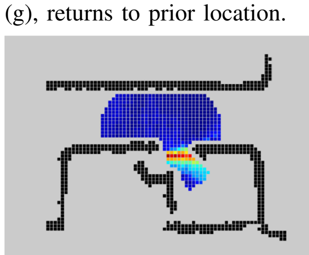
(f) Recommended action from (e) (in green), relative to the sample set used.



(g) Batch GP inference after taking the action from (f).

(h) Recommended action from (g), returns to prior location.





(i) Bayesian optimization suggests a more informative action that ultimately leads to the left.

(j) Ground truth of the MI evaluated using ray-casting over the known portion of the map.

Fig. 5: Representative cases when batch Gaussian Process regression makes poor-quality predictions and causes the robot to become stuck in place, from the Seattle map.

Time Cons. Per Step (Secs)	**10 GP**	**10 BayOpt**	**10 QMC**	**20 GP**	**20 BayOpt**	**20 QMC**
μ	4.13	3.71	4.41	7.44	8.51	8.79
σ	0.04	0.15	0.08	0.15	0.87	0.12

Steps Taken Per Trial	**10 GP**	**10 BayOpt**	**10 QMC**	**20 GP**	**20 BayOpt**	**20 QMC**
μ	75	72	89	63	61	71
σ	4.87	2.83	5.65	5.11	1.43	5.58

TABLE I: The results shown here are the average of 100 trials over the maze map shown in Figure 3. For the six test cases examined (in which "QMC" refers to the quasi Monte Carlo approach, derived from Sobol samples; "GP" refers to the batch GP regression approach and "BayOpt" refers to Bayesian optimization approach), at top we give a comparison of the mean and standard deviation of computation time required per sensing action, and at bottom we show a comparison of the mean and standard deviation of the total number of steps taken by the robot in the course of driving its entropy to the minimum designated value.

paths among the different parameterizations. The use of GP regression or Bayesian optimization to select moves from the continuous space of sensing actions accumulates a more significant advantage, such that regression over 10 samples performs better than explicitly evaluating the MI at 20 samples. Hence, more informative outcomes are selected with substantially less computational effort. Representative trajectories of the robot when using Bayesian optimization are given in Figure 3, for each map. These trajectories represent full exploration of their respective environments, reaching the lower limit of map entropy sufficient for termination of Algorithm 1.

Finally, Table 1 gives the computation time required, and the number of steps taken by the robot in the exploration process, for all examples implemented over the maze map of Figure 3a. The computational cost of Bayesian optimization and GP regression are slightly better than the QMC approach, because the former two will drive the robot to more open areas containing fewer obstructions, which reduces the computational cost of the first term in (8). At 20 sampled actions, the Bayesian optimization approach begins to incur a heavier computational cost than the batch GP approach, due the additional number of GP regressions required.

VI. CONCLUSIONS

We have proposed a novel approach to predict mutual information using Bayesian optimization, for the purpose of exploring *a priori* unknown environments and producing a comprehensive occupancy map. In the examples considered, Bayesian optimization facilitates the selection of competitive informative sensing actions compared to batch GP regression, always reducing map entropy to an equivalent extent and at an equivalent rate, if not superior. The benefits of actively selecting additional sensing actions to include in an MI prediction are most evident in complex environments with narrow corridors, where significant gains can be made with marginal additional computational effort.

However, if the number of active samples selected with Bayesian optimization were to increase substantially, the number of explicit MI evaluations $N_{samples}$ would grow, and the procedure's computational complexity may no longer be real-time viable. An appropriate tradeoff must be established between the marginal improvements in information gain with additional Bayesian optimization samples, and the bandwidth of the robot's decision-making process.

A key area for future work is the adaptation of this approach to ground robots with 3D range-sensing capability, and a reduced angular field of view such that a robot's angular orientation defines a unique sensing action. It is anticipated that the proposed Bayesian optimization approach will offer improved scalability of mutual information controllers to higher-dimensional action spaces, extracting the maximal benefit from every candidate sensing action for which MI is evaluated, and leveraging the power of Gaussian processes to interpolate among a sparse set of sampled actions. However, the suitability of the Matérn kernel function for capturing mutual information over the SE2 configuration space must first be validated, to ensure the results obtained in the Euclidean action spaces explored in this work are extensible to action spaces of different topologies.

ACKNOWLEDGMENTS

The authors would like to thank Kiril Manchevski from the Mechanical Engineering Department of Stevens Institute of Technology for help with computational resources. This research has been supported in part by the National Science Foundation, grant number IIS-1551391.

REFERENCES

[1] S. Thrun, W. Burgard, and D. Fox, "Exploration," In *Probabilistic Robotics*, pp. 569-605, MIT Press, 2005.

[2] A. Elfes, "Robot Navigation: Integrating Perception, Environmental Constraints and Task Execution within a Probabilistic Framework," *Proceedings of the International Workshop on Reasoning with Uncertainty in Robotics*, pp. 93-129, 1995.

[3] A. Elfes, "Using Occupancy Grids for Mobile Robot Perception and Navigation," *Computer*, vol. 22(6), pp. 46-57, 1989.

[4] B.J. Julian, S. Karaman, and D. Rus. "On Mutual Information-Based Control of Range Sensing Robots for Mapping Applications," *The International Journal of Robotics Research*, vol. 33(10), pp. 1375-1392, 2014.

[5] S. Bai, J. Wang, K. Doherty, and B. Englot, "Inference-Enabled Information-Theoretic Exploration of Continuous Action Spaces," *Proceedings of the 17th International Symposium on Robotics Research*, 2015.

[6] B. Charrow, S. Liu, V. Kumar, and N. Michael, "Information-theoretic Mapping Using Cauchy-Schwarz Quadratic Mutual Information." *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 4791-4798, 2015.

[7] Z. Ghahramani, "Probabilistic Machine Learning and Artificial Intelligence," *Nature*, vol. 521(7553), pp. 452-459, 2015.

[8] D.R. Jones, M. Schonlau, and W.J. Welch "Efficient Global Optimization of Expensive Black-Box Functions." *Journal of Global Optimization* vol. 13, pp. 455-492, 1998.

[9] M.J. Sasena, "Flexibility and Efficiency Enhancements for Constrained Global Design Optimization with Kriging Approximations," PhD Thesis, University of Michigan, 2002.

[10] D. Lizotte, T. Wang, M. Bowling, and D. Schuurmans, "Automatic Gait Optimization with Gaussian Process Regression." *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 944-949, 2007.

[11] R. Marchant and F. Ramos, "Bayesian Optimization for Intelligent Environmental Monitoring," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2242-2249, 2012.

[12] R. Marchant and F. Ramos, "Bayesian Optimisation for Informative Continuous Path Planning," *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 6136-6143, 2014.

[13] J.R. Souza, R. Marchant, L. Ott, D.F. Wolf, and F. Ramos, "Bayesian Optimisation for Active Perception and Smooth Navigation," *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 4081-4087, 2014.

[14] E. Brochu, V.M. Cora, and N. De Freitas, "A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning," Technical Report TR-2009-23, Department of Computer Science, University of British Columbia, 2009.

[15] P. Whaite and F.P. Ferrie, "Autonomous Exploration: Driven by Uncertainty," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19(3), pp. 193-205, 1997.

[16] F. Bourgault, A.A. Makarenko, S.B. Williams, B. Grocholsky, and H.F. Durrant-Whyte, "Information Based Adaptive Robotic Exploration," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 540-545, 2002.

[17] C. Stachniss, G. Grisetti, and W. Burgard, "Information Gain-Based Exploration Using Rao-Blackwellized Particle Filters," *Proceedings of Robotics: Science and Systems*, 2005.

[18] T. Kollar and N. Roy, "Trajectory Optimization using Reinforcement Learning for Map Exploration," *The International Journal of Robotics Research*, vol. 27(2), pp. 175-196, 2008.

[19] T. Kollar and N. Roy, "Efficient Optimization of Information-Theoretic Exploration in SLAM," *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 1369-1375, 2008.

[20] K. Yang, S.K. Gan, and S. Sukkarieh, "A Gaussian Process-Based RRT Planner for the Exploration of Unknown and Cluttered Environment with a UAV," *Advanced Robotics*, vol. 27(6), pp. 431-443, 2013.

[21] B. Charrow, G. Kahn, S. Patil, S. Liu, K. Goldberg, P. Abbeel, N. Michael, and V. Kumar, "Information-Theoretic Planning with Trajectory Optimization for Dense 3D Mapping," *Proceedings of Robotics: Science and Systems*, 2015.

[22] K.M. Wurm, D. Hennes, D. Holz, R.B. Rusu, C. Stachniss, K. Konolige, and W. Burgard, "Hierarchies of Octrees for Efficient 3D Mapping," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4249-4255, 2011.

[23] M.G. Jadidi, J.V. Miro, R. Valencia, and J. Andrade-Cetto, "Exploration on Continuous Gaussian Process Frontier Maps," *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 6077-6082, 2014.

[24] M.G. Jadidi, J. V. Miro, and G. Dissanayake, "Mutual Information-based Exploration on Continuous Occupancy Maps," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 6086-6092, 2015.

[25] R. Martinez-Cantin, N. de Freitas, A. Doucet, and J.A. Castellanos, "Active Policy Learning for Robot Planning and Exploration Under Uncertainty," *Proceedings of Robotics: Science and Systems*, 2007.

[26] C.E. Shannon and W. Weaver, *The Mathematical Theory of Communication*, University of Illinois Press, 1949.

[27] N. Srinivas, A. Krause, S.M. Kakade, and M. Seeger, "Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design," *Proceedings of the 27th International Conference on Machine Learning*, 2010.

[28] C.E. Rasmussen and C.K.I. Williams, *Gaussian Processes for Machine Learning*, The MIT Press, 2006.

[29] S. Kim and J. Kim, "Continuous Occupancy Maps using Overlapping Local Gaussian Processes," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4709-4714, 2013.

[30] I.M. Sobol, "On the Distribution of Points in a Cube and the Approximate Evaluation of Integrals," *USSR Computational Mathematics and Mathematical Physics*, vol. 7(4), pp. 86-112, 1967.

[31] A. Howard and N. Roy, "The Robotics Data Set Repository (Radish)," 2003 [Online]. Available: http://radish.sourceforge.net/

[32] C.E. Rasmussen and H. Nickisch, "Gaussian Processes for Machine Learning (GPML) Toolbox," *The Journal of Machine Learning Research*, vol. 11, pp. 3011-3015, 2010.

[33] J. Zhang and S. Singh, "Low-drift and Real-time Lidar Odometry and Mapping," *Autonomous Robots*, 10.1007, 2016.