

# Sampling-based incremental information gathering with applications to robotic exploration and environmental monitoring

The International Journal of  
Robotics Research  
2019, Vol. 38(6) 658–685  
© The Author(s) 2019  
Article reuse guidelines:  
[sagepub.com/journals-permissions](http://sagepub.com/journals-permissions)  
DOI: 10.1177/0278364919844575  
[journals.sagepub.com/home/ijr](http://journals.sagepub.com/home/ijr)



Maani Ghaffari Jadidi<sup>1</sup> , Jaime Valls Miro<sup>2</sup> and Gamini Dissanayake<sup>2</sup>

## Abstract

We propose a sampling-based motion-planning algorithm equipped with an information-theoretic convergence criterion for incremental informative motion planning. The proposed approach allows dense map representations and incorporates the full state uncertainty into the planning process. The problem is formulated as a constrained maximization problem. Our approach is built on rapidly exploring information-gathering algorithms and benefits from the advantages of sampling-based optimal motion-planning algorithms. We propose two information functions and their variants for fast and online computations. We prove an information-theoretic convergence for an entire exploration and information-gathering mission based on the least upper bound of the average map entropy. A natural automatic stopping criterion for information-driven motion control results from the convergence analysis. We demonstrate the performance of the proposed algorithms using three scenarios: comparison of the proposed information functions and sensor configuration selection, robotic exploration in unknown environments, and a wireless signal strength monitoring task in a lake from a publicly available dataset collected using an autonomous surface vehicle.

## Keywords

Motion and path planning, planning under uncertainty, belief space planning, field robotics, robotic information gathering, environmental monitoring, robotic exploration, Gaussian processes, mutual information

## 1. Introduction

Exploration in unknown environments is a major challenge for an autonomous robot and has numerous present and emerging applications ranging from search and rescue operations to space exploration programs. While exploring an unknown environment, the robot is often tasked with monitoring a quantity of interest through a cost or an information quality measure (Dhariwal et al., 2004; Dunbabin and Marques, 2012; Lan and Schwager, 2013; Marchant and Ramos, 2012; Singh et al., 2010; Yu et al., 2015). Robotic exploration algorithms usually rely on geometric frontiers (Ström et al., 2015; Yamauchi, 1997) or visual targets (Kim and Eustice, 2015) as goals to solve the planning problem using geometric/information gain-based *greedy* action selection or planning for a limited horizon. The main drawback of such techniques is that a set of targets constrains the planner search space to the paths that start from the current robot pose to targets. In contrast, a more general class of robotic navigation problem known as robotic information gathering (Binney et al., 2013; Binney and Sukhatme, 2012; Singh et al., 2009), and in particular the rapidly-exploring information gathering (RIG) (Hollinger and

Sukhatme, 2013, 2014) technique, exploits a sampling-based planning strategy to calculate the cost and information gain while searching for traversable paths in the entire space. This approach differs from classic path-planning problems as there is no goal to be found. Therefore, a solely information-driven robot control strategy can be formulated. Another important advantage of RIG methods is their *multi-horizon planning* nature through representing the environment by an incrementally built graph that considers both the information gain and cost.

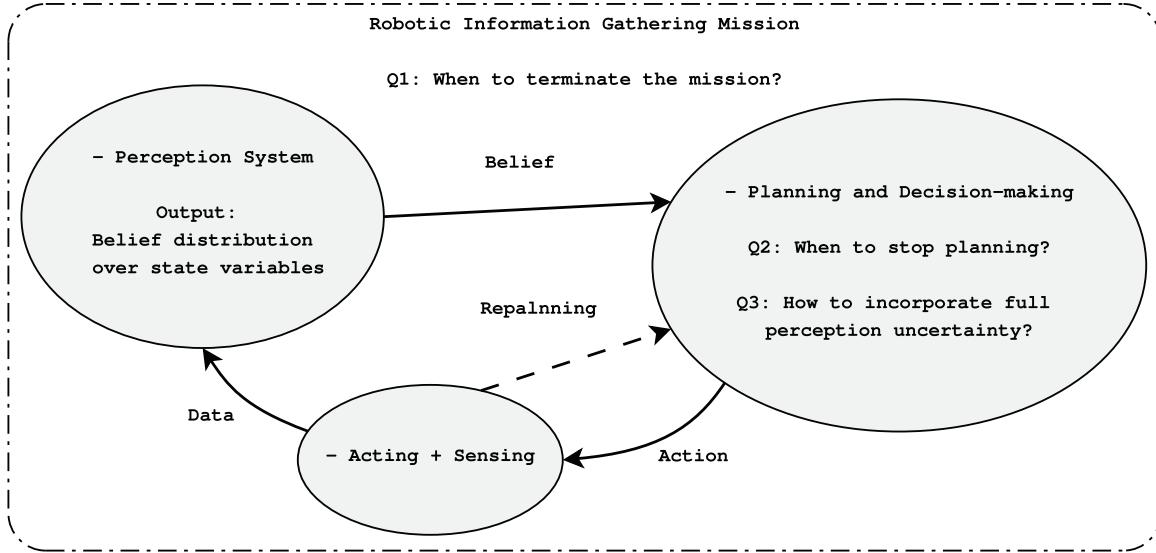
RIG algorithms are suitable for non-myopic robotic exploration techniques. However, the developed methods are not online, the information function calculation is often a bottleneck, and they do not offer an automated

<sup>1</sup>College of Engineering, University of Michigan, Ann Arbor, MI, USA

<sup>2</sup>Centre for Autonomous Systems, University of Technology Sydney, Sydney, Australia

## Corresponding author:

Maani Ghaffari Jadidi, College of Engineering, University of Michigan, Ann Arbor, MI 48109, USA.  
Email: maanig@umich.edu



**Fig. 1.** The abstract system for robotic information gathering. The three important questions that we try to answer in this work are asked (Q1–Q3). Note that the diagram is conceptual and, in practice, different modules can share many attributes and methods such as parameters and sensor models. The replanning problem is not studied in this work, that is the robot remains committed to the planned action until it enters the planning state again.

convergence criterion, i.e. they are *anytime*.<sup>1</sup> To be able to employ RIG for online navigation tasks, we propose an incrementally-exploring information gathering (IIG) algorithm built on the RIG. In particular, the following developments are performed.

- (i) We allow the full state including the robot pose and a dense map to be partially observable.
- (ii) We develop a convergence criterion based on RIC. This convergence criterion is a necessary step for incremental planning to make it possible for the robot to execute planned actions autonomously.
- (iii) We propose two classes of information functions that can approximate the information gain for online applications. The algorithmic implementation of the proposed functions is also provided.
- (iv) We develop a heuristic algorithm to extract the most informative trajectories from a RIG/IIG tree. This algorithm is necessary to extract an executable trajectory (action) from RIG/IIG graphs (trees).
- (v) We prove an information-theoretic automatic stopping criterion for the entire mission based on the least upper bound of the average map (state variables) entropy.
- (vi) We provide results in batch and incremental experiments as well as in publicly available robotic datasets and discuss potential applications of the developed algorithms.

Figure 1 shows a conceptual illustration of the problem studied in this article. The three questions that we try to address are as follows. (Q1) When should we terminate the entire information-gathering mission? (Q2) When

should we stop planning by automatically detecting the convergence of the planner, i.e. planning and acting incrementally? (Q3) How can we incorporate the uncertainty of the perception system into the planner?

This article is organized as follows. In the next section, the related work is discussed. Section 3 includes required preliminaries. In Section 4, we present the problem definition and highlight the differences between RIG and IIG algorithms. We also explain RIG algorithms to establish the required basis. In Section 5, we present the novel IIG algorithm. In Section 6, we propose two information functions that approximate the information quality of nodes and trajectories of the associated motion-planning problem. In Section 7, the problem of path extraction and selection is explained, and a heuristic algorithm is proposed to solve the problem. Section 8 presents the proof for an information-theoretic automatic stopping criterion for exploration and information-gathering missions. In Section 9, we present an extensive evaluation of the proposed strategies including a comparison with relevant techniques in the literature. A lake-monitoring scenario based on a publicly available dataset is also presented together with a discussion on the limitations of this work. Finally, Section 10 concludes the article and discusses possible extensions of the proposed algorithms as future work.

## 2. Related work

Motion-planning algorithms (Latombe, 1991; LaValle, 2006) constitute a broad area of research in the robotic community. In the presence of uncertainty, where the state is not fully observable, measurements and actions are stochastic. The sequential decision-making under uncertainty,

in the most general form, can be formulated as a partially observable Markov decision process (POMDP) or optimal control with imperfect state information (Astrom, 1965; Bertsekas, 1995; Kaelbling et al., 1998; Smallwood and Sondik, 1973). Unfortunately, when the problem is formulated using a dense belief representation, a general-purpose POMDP solver is not a practical choice (Binney et al., 2013).

The sampling-based motion-planning algorithms (Bry and Roy, 2011; Horsch et al., 1994; Karaman and Frazzoli, 2011; Kavraki et al., 1996; Lan and Schwager, 2013; LaValle and Kuffner, 2001) have proven successful in applications in robotics. An interesting case is where the environment is fully observable (known map) and the objective is to make sequential decisions under motion and measurement uncertainty. This problem is also known as *active localization*; for a recent technique to solve such a problem see Aghamohammadi et al. (2014) and references therein. A closely related term that is used in the literature is *belief space planning* (Bry and Roy, 2011; Huynh and Roy, 2009; Kurniawati et al., 2011, 2008; Platt et al., 2010; Prentice and Roy, 2009; Valencia et al., 2013; Van Den Berg et al., 2011). In this series of works, the assumption of a known map can be relaxed and the environment is often represented using a set of features. The objective is to find a trajectory that minimizes the state uncertainty (the total cost) with respect to a fixed or variable (and bounded) planning horizon that can be set according to the *budget* (Indelman et al., 2015).

In the context of feature-based representation of the environment, planning actions for simultaneous localization and mapping (SLAM) using model predictive control (MPC) is studied in Leung et al. (2006a). To enable the robot to explore, a set of “attractors” are defined to resemble informative regions for exploration. In particular, it is concluded as expected that a multi-step (three steps) look-ahead MPC planner outperforms the greedy approach. Atanasov et al. (2014), assuming the sensor dynamics is linear-Gaussian, reduced the stochastic optimal control problem to a deterministic optimal control problem that can be solved offline. The deterministic nature of the problem has been exploited to solve it using forward value iteration (Bertsekas, 1995; Le Ny and Pappas, 2009); furthermore, it is shown that the proposed solution, namely reduced value iteration, has a lower computational complexity than that of forward value iteration and its performance is better than the greedy policy. The work is also extended from a single robot to decentralized *active information acquisition*, and using attractors as dummy exploration landmarks, i.e. frontiers, it has been successfully applied to the *active SLAM* problem (Atanasov et al., 2015).

Another approach to study the problem of robotic navigation under uncertainty is known as informative motion planning or robotic information gathering (Atanasov, 2015; Binney et al., 2013; Binney and Sukhatme, 2012; Hollinger and Sukhatme, 2013, 2014; Leung et al., 2006b; Levine, 2010; Singh et al., 2009). RIG (Hollinger and Sukhatme,

2014) is a technique that solves the problem of informative motion planning using incremental sampling from the workspace and by *partial ordering* of nodes builds a graph that contains informative trajectories.

In the problem we study in this article, the state variables consist of the robot trajectory and a dense map of the environment. We build on the RIG technique by considering both the robot pose and the map being partially observable. We also develop an information-theoretic criterion for the convergence of the search that allows us to perform online robotic exploration in unknown environments. Rapidly-exploring Adaptive Search and Classification (ReASC) (Hollinger, 2015) also improves on the RIG by allowing for real-time optimization of search and classification objective functions. However, ReASC relies on discrete target locations and, similar to Platt et al. (2010), resorts to the maximum likelihood assumption for future observations.

The technique reported by Charrow et al. (2015a,b) is closely related to this work but assumes that the robot poses are known, i.e. fully observable, to solve the problem of information gathering for occupancy mapping with the help of geometric frontiers (Ström et al., 2015; Yamauchi, 1997). The computational performance of the information gain estimation is increased by using Cauchy–Schwarz quadratic mutual information (CSQMI). It is shown that the behavior of CSQMI is similar to that of mutual information (MI) while it can be computed faster (Charrow, 2015: Section 6.2.2 and Figure 6.1). It is argued by Charrow et al. (2015a: Section V.C) that: “*It is interesting that the human operator stopped once they believed they had obtained a low uncertainty map and that all autonomous approaches continue reducing the map’s entropy beyond this point, as they continue until no frontiers are left. However, the final maps are qualitatively hard to differentiate, suggesting a better termination condition is needed.*” We agree with this argument and relax such a constraint by exploiting a sampling-based planning strategy to calculate the cost and information gain while searching for traversable paths in the entire space. We also prove an information-theoretic automatic stopping criterion for the entire mission which can alleviate this issue. In Section 9.3, we conduct robotic exploration experiments to show the effectiveness of the proposed termination condition. We show that the proposed condition enables the robot to produce comparable results while collecting fewer measurements, but sufficient information for the inference.

In particular, the main features of this work that differentiate the present approach from the literature mentioned above can be summarized as follows.

- (i) We allow for dense belief representations. Therefore, we incorporate full state uncertainty, i.e. the robot pose and the map, into the planning process. As a result, the robot behavior has a strong correlation with its perception uncertainty.
- (ii) We take into account all possible future observations and do not resort to maximum likelihood

- assumptions. Therefore, the randomness of future observations is addressed.
- (iii) We take into account both cost and information gain. The cost is included through a measure of distance, and the information gain quantifies the sensing quality and acting uncertainty. Therefore, the planning algorithm runs with respect to available sensing resources and acting limitations.
  - (iv) We propose an information-theoretic notion of planning horizon that leads to an infinite-horizon planning technique and provides a general framework for incremental planning and acting.
  - (v) We offer an automatic stopping criterion for the entire information-gathering mission that can be easily incorporated into many of the available algorithms mentioned previously.

### 3. Preliminaries

In this section, we briefly explain the mathematical notation and preliminaries required throughout this article.

#### 3.1. Mathematical notation

In the present article, probabilities and probability densities are not distinguished in general. Matrices are capitalized in bold, such as in  $\mathbf{X}$ , and vectors are in lowercase bold type, such as in  $\mathbf{x}$ . Vectors are column-wise and  $1 : n$  means integers from 1 to  $n$ . The Euclidean norm is shown by  $\|\cdot\|$ . Here  $|X|$  denotes the determinant of matrix  $X$ . For the sake of compactness, random variables, such as  $X$ , and their realizations,  $x$ , are sometimes denoted interchangeably where it is evident from context. Here  $x^{[i]}$  denotes a reference to the  $i$ th element of the variable. An alphabet such as  $\mathcal{X}$  denotes a set, and the cardinality of the set is denoted by  $|\mathcal{X}|$ . A subscript asterisk, such as in  $\mathbf{x}_*$ , indicates a reference to a test set quantity. The  $n \times n$  identity matrix is denoted by  $\mathbf{I}_n$ . We use  $\text{vec}(x^{[1]}, \dots, x^{[n]})$  to denote a vector such as  $\mathbf{x}$  constructed by stacking  $x^{[i]}$ ,  $\forall i \in \{1 : n\}$ . The function notation is overloaded based on the output type and denoted by  $k(\cdot)$ ,  $\mathbf{k}(\cdot)$ , and  $\mathbf{K}(\cdot)$  where the outputs are scalar, vector, and matrix, respectively. Finally,  $\mathbb{E}[\cdot]$  and  $\mathbb{V}[\cdot]$  denote the expected value and variance of a random variable, respectively.

#### 3.2. Information theory

Entropy is a measure of the uncertainty of a random variable (Cover and Thomas, 1991). The entropy  $H(X)$  of a discrete random variable  $X$  is defined as  $H(X) = \mathbb{E}_{p(x)} [\log \frac{1}{p(x)}] = -\sum_{x \in \mathcal{X}} p(x) \log p(x)$  which implies that  $H(X) \geq 0$ . The joint entropy  $H(X, Y)$  of discrete random variables  $X$  and  $Y$  with a joint distribution  $p(x, y)$  is defined as  $H(X, Y) = -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log p(x, y)$ . The conditional entropy is defined as  $H(Y|X) = -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log p(y|x)$ .

**Theorem 1.** (Chain rule for entropy elements). *Let  $X_1, X_2, \dots, X_n$  be drawn according to  $p(x_1, x_2, \dots, x_n)$ . Then*

$$H(X_1, X_2, \dots, X_n) = \sum_{i=1}^n H(X_i | X_{i-1}, \dots, X_1) \quad (1)$$

The relative entropy or Kullback–Leibler divergence (KLD) is a measure of distance between two distributions  $p(x)$  and  $q(x)$ . It is defined as  $D(p||q) = \mathbb{E}_{p(x)} [\log \frac{p(x)}{q(x)}]$ .

**Theorem 2.** (Information inequality elements). *Let  $X$  be a discrete random variable. Let  $p(x)$  and  $q(x)$  be two probability mass functions. Then*

$$D(p||q) \geq 0 \quad (2)$$

with equality if and only if  $p(x) = q(x) \forall x$ .

The MI,  $I(X; Y) = D(p(x, y)||p(x)p(y)) = H(X) - H(X|Y)$ , is the reduction in the uncertainty of one random variable due to the knowledge of the other.

**Corollary 3.** (Non-negativity of MI). *For any two random variables  $X$  and  $Y$ ,*

$$I(X; Y) \geq 0 \quad (3)$$

with equality if and only if  $X$  and  $Y$  are independent.

*Proof.* We have  $I(X; Y) = D(p(x, y)||p(x)p(y)) \geq 0$ , with equality if and only if  $p(x, y) = p(x)p(y)$ .  $\square$

Some immediate consequences of the provided definitions are as follows.

**Theorem 4.** (Conditioning reduces entropy). *For any two random variables  $X$  and  $Y$ ,*

$$H(X|Y) \leq H(X) \quad (4)$$

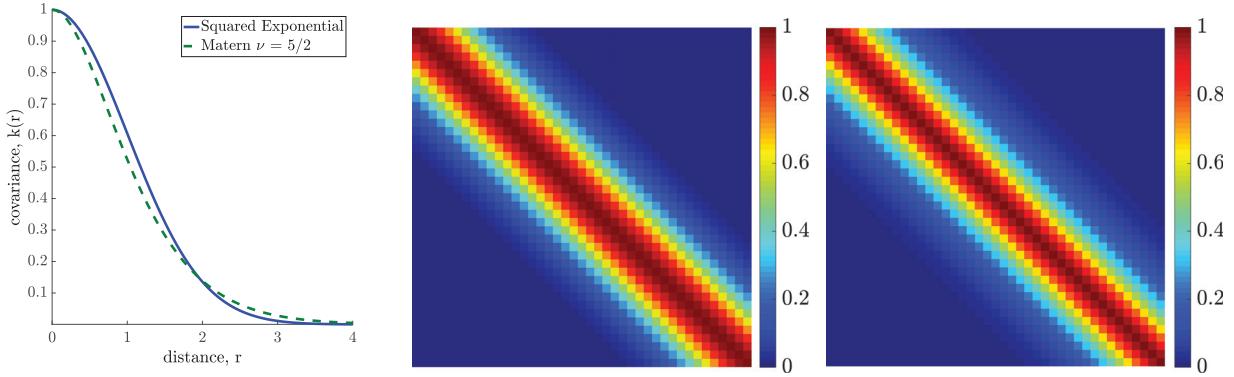
with equality if and only if  $X$  and  $Y$  are independent.

*Proof.* We have  $0 \leq I(X; Y) = H(X) - H(X|Y)$ .  $\square$

We now define the equivalent of the functions mentioned above for probability density functions. Let  $X$  be a continuous random variable whose support set is  $\mathcal{S}$ . Let  $p(x)$  be the probability density function for  $X$ . The differential entropy  $h(X)$  of  $X$  is defined as  $h(X) = -\int_{\mathcal{S}} p(x) \log p(x) dx$ . Let  $X$  and  $Y$  be continuous random variables that have a joint probability density function  $p(x, y)$ . The conditional differential entropy  $h(X|Y)$  is defined as  $h(X|Y) = -\int p(x, y) \log p(x|y) dx dy$ . The relative entropy (KLD) between two probability density functions  $p$  and  $q$  is defined as  $D(p||q) = \int p \log \frac{p}{q}$ . The MI  $I(X; Y)$  between two continuous random variables  $X$  and  $Y$  with joint probability density function  $p(x, y)$  is defined as  $I(X; Y) = \int p(x, y) \log \frac{p(x, y)}{p(x)p(y)} dx dy$ .

#### 3.3. Submodular functions

A set function  $f$  is said to be *submodular* if  $\forall \mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{S}$  and  $\forall s \in \mathcal{S} \setminus \mathcal{B}$ , then  $f(\mathcal{A} \cup s) - f(\mathcal{A}) \geq f(\mathcal{B} \cup s) - f(\mathcal{B})$ .



**Fig. 2.** Illustrative examples of the squared exponential and Matérn ( $\nu = 5/2$ ) covariance function as the distance parameter is increased from 0 to 4 and their corresponding function values in the kernel space, from the left, respectively. The length-scale parameter is set to one.

Intuitively, this can be explained as: by adding observations to a smaller set, we gain more information. The function  $f$  has diminishing return. It is *normalized* if  $f(\emptyset) = 0$  and it is *monotone* if  $f(\mathcal{A}) \leq f(\mathcal{B})$ . The MI is normalized, approximately monotone, and submodular (Krause et al., 2008).

### 3.4. Gaussian processes

Gaussian processes (GPs) are non-parametric Bayesian regression techniques that employ statistical inference to learn dependencies between points in a dataset (Rasmussen and Williams, 2006). The joint distribution of the observed target values,  $\mathbf{y}$ , and the function values (the latent variable),  $\mathbf{f}_*$ , at the query points can be written as

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}(\mathbf{0}, \begin{bmatrix} \mathbf{K}(X, X) + \sigma_n^2 \mathbf{I}_n & \mathbf{K}(X, X_*) \\ \mathbf{K}(X_*, X) & \mathbf{K}(X_*, X_*) \end{bmatrix}) \quad (5)$$

where  $X$  is the  $d \times n$  design matrix of aggregated input vectors  $\mathbf{x}$ ,  $X_*$  is a  $d \times n_*$  query points matrix,  $\mathbf{K}(\cdot, \cdot)$  is the GP covariance matrix, and  $\sigma_n^2$  is the variance of the observation noise, which is assumed to have an independent and identically distributed (i.i.d.) Gaussian distribution. Define a training set  $\mathcal{D} = \{(\mathbf{x}^{[i]}, \mathbf{y}^{[i]}) | i = 1 : n\}$ . The predictive conditional distribution for a single query point  $f_* | \mathcal{D}, \mathbf{x}_* \sim \mathcal{N}(\mathbb{E}[f_*], \mathbb{V}[f_*])$  can be derived as

$$\mu = \mathbb{E}[f_*] = \mathbf{k}(X, \mathbf{x}_*)^T [\mathbf{K}(X, X) + \sigma_n^2 \mathbf{I}_n]^{-1} \mathbf{y} \quad (6)$$

$$\sigma = \mathbb{V}[f_*] = \mathbf{k}(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}(X, \mathbf{x}_*)^T [\mathbf{K}(X, X) + \sigma_n^2 \mathbf{I}_n]^{-1} \mathbf{k}(X, \mathbf{x}_*) \quad (7)$$

The hyperparameters of the covariance and mean function,  $\boldsymbol{\theta}$ , can be computed by minimization of the negative log of the marginal likelihood (NLML) function:

$$\begin{aligned} \log p(\mathbf{y} | X, \boldsymbol{\theta}) = & -\frac{1}{2} \mathbf{y}^T [\mathbf{K}(X, X) + \sigma_n^2 \mathbf{I}_n]^{-1} \mathbf{y} \\ & -\frac{1}{2} \log |\mathbf{K}(X, X) + \sigma_n^2 \mathbf{I}_n| - \frac{n}{2} \log 2\pi \end{aligned} \quad (8)$$

**3.4.1. Covariance function.** Covariance functions are the main part of any GPs. We define a covariance function using the kernel definition as follows.

**Definition 1.** (Covariance function). Let  $\mathbf{x} \in \mathcal{X}$  and  $\mathbf{x}' \in \mathcal{X}$  be a pair of inputs for a function  $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$  known as a kernel. A kernel is called a covariance function, as is the case in GPs, if it is symmetric,  $k(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}', \mathbf{x})$ , and positive semidefinite:

$$\int k(\mathbf{x}, \mathbf{x}') f(\mathbf{x}) f(\mathbf{x}') d\mu(\mathbf{x}) d\mu(\mathbf{x}') \geq 0 \quad (9)$$

for all  $f \in L_2(\mathcal{X}, \mu)$ .

Given a set of input points  $\{\mathbf{x}^{[i]} | i = 1 : n\}$ , a covariance matrix can be constructed using  $\mathbf{K}^{[i,j]} = k(\mathbf{x}^{[i]}, \mathbf{x}^{[j]})$  as its entries.

**3.4.2. Useful kernels.** The *squared exponential* (SE) covariance function has the form  $k(r) = \exp(-\frac{r^2}{2l^2})$  where  $r = \|\mathbf{x} - \mathbf{x}_*\|$  is the distance between two input arguments of the covariance function and  $l$  is the *characteristic length-scale*. This covariance function is the most common kernel used in GPs and is infinitely differentiable. The Matérn family of covariance functions (Stein, 1999) has proven powerful features to model structural correlations (Ghaffari Jadidi et al., 2014; Kim and Kim, 2015). For a single query point  $\mathbf{x}_*$  the function is given by

$$k(r) = \frac{1}{\Gamma(\nu) 2^{\nu-1}} \left[ \frac{\sqrt{2\nu r}}{l} \right]^\nu K_\nu \left( \frac{\sqrt{2\nu r}}{l} \right) \quad (10)$$

where  $\Gamma(\cdot)$  is the Gamma function,  $K_\nu(\cdot)$  is the modified Bessel function of the second kind of order  $\nu$ ,  $l$  is the characteristic length scale, and  $\nu$  is a positive parameter used to control the smoothness of the covariance. In the limit for  $\nu \rightarrow \infty$  this covariance function converges to the SE kernel.

Examples of the SE and Matérn ( $\nu = 5/2$ ) covariance functions as the distance parameter  $r$  increases are shown in Figure 2. The functions are also plotted in kernel space.

## 4. Problem statement

The problem of robotic information gathering is formulated as a maximization problem subject to finite resources, i.e. a budget  $b$ . In Hollinger and Sukhatme (2014), this problem is defined as follows.

**Definition 2.** (Trajectory). Let  $\mathcal{X}_f$  denotes the free workspace. A trajectory,  $\mathcal{P} \in \mathcal{X}_f$ , is a sequence of reachable points in which any two consecutive points are connected using a collision-free path and with respect to the robot motion constraints.

**Problem 1.** (Informative motion planning). Let  $\mathcal{A}$  be the space of all possible trajectories and  $f_I(\mathcal{P})$  be a function that quantifies the information quality along a trajectory  $\mathcal{P}$ . Let  $f_c(\mathcal{P})$  be a function that returns the cost associated with trajectory  $\mathcal{P}$ . Given the available budget  $b$ , the problem can be formulated as follows:

$$\mathcal{P}^* = \arg \max_{\mathcal{P} \in \mathcal{A}} f_I(\mathcal{P}) \quad \text{s.t. } f_c(\mathcal{P}) \leq b \quad (11)$$

Now we express the assumptions in RIG algorithms.

**Assumption 5.** The cost function  $f_c(\mathcal{P})$  is strictly positive, monotonically increasing, bounded, and additive such as distance and energy.

**Remark 1.** The information function  $f_I(\mathcal{P})$  can be modular, time-varying modular, or submodular.

The information function assumption follows from Hollinger and Sukhatme (2014), even though we focus our attention on the submodular class of information functions as the information gathered at any future time during navigation depends on prior robot trajectories. Another reason to consider submodular information functions is to avoid information “double-counting.” This allows us to develop an information-theoretic convergence criterion for RIG/IIG as the amount of available information remains bounded. The following assumptions are directly from Hollinger and Sukhatme (2014), which, in turn, are equivalent to or adapted from Bry and Roy (2011) and Karaman and Frazzoli (2011). The *Steer* function used in Assumption 6 extends nodes towards newly sampled points.

**Assumption 6.** Let  $x_a$ ,  $x_b$ , and  $x_c \in \mathcal{X}_f$  be three points within radius  $\Delta$  of each other. Let the trajectory  $e_1$  be generated by  $\text{Steer}(x_a, x_c, \Delta)$ ,  $e_2$  be generated by  $\text{Steer}(x_a, x_b, \Delta)$ , and  $e_3$  be generated by  $\text{Steer}(x_b, x_c, \Delta)$ . If  $x_b \in e_1$ , then the concatenated trajectory  $e_2 + e_3$  must be equal to  $e_1$  and have equal cost and information.

This assumption is required as in the limit drawn samples are infinitely close together, and the *Steer* function, cost, and information need to be consistent for any intermediate point.

**Assumption 7.** There exists a constant  $r \in \mathbb{R}_{>0}$  such that for any point  $x_a \in \mathcal{X}_f$  there exists an  $x_b \in \mathcal{X}_f$ , such that 1) the ball of radius  $r$  centered at  $x_a$  lies inside  $\mathcal{X}_f$  and 2)  $x_a$  lies inside the ball of radius  $r$  centered at  $x_b$ .

This assumption ensures that there is enough free space near any point for extension of the graph. Violation of this assumption in practice can lead to failure of the algorithm to find a path.

**Assumption 8** (Uniform sampling).<sup>2</sup> Points returned by the sampling function sample are i.i.d. and drawn from a uniform distribution.

### 4.1. Incremental informative motion planning

Now we define the problem of incremental informative motion planning as follows.

**Problem 2.** (Incremental informative motion planning). Let  $s_{0:t_s} \in \mathcal{S}$  be the current estimate of the state up to time  $t_s$ . Let  $\mathcal{A}_t$  be the space of all possible trajectories at time  $t$  and  $f_I(\mathcal{P}_t)$  be a function that quantifies the information quality along a trajectory  $\mathcal{P}_t$ . Let  $f_c(\mathcal{P}_t)$  be a function that returns the cost associated with trajectory  $\mathcal{P}_t$ . Given the available budget  $b_t$ , the problem can be formulated as follows:

$$\begin{aligned} \mathcal{P}_t^* = \arg \max_{\mathcal{P}_t \in \mathcal{A}_t} f_I(\mathcal{P}_t) \quad &\forall t > t_s \\ \text{s.t. } f_c(\mathcal{P}_t) \leq b_t \text{ and } S = s_{0:t_s} \end{aligned} \quad (12)$$

**Remark 2.** The state  $S$  can include the representation of the environment (map), the robot trajectory, and possibly any other variables defined in the state vector. In general, the information function  $f_I(\mathcal{P}_t)$  is responsible for incorporating the state uncertainty in the information gain calculations.

**Remark 3.** In practice, we solve the problem incrementally and use a planning horizon  $T > t_s$ . Here  $T$  is the time when the current planning horizon ends and in the limit goes to  $\infty$ .

The main difference between Problems 1 and 2 is that in the latter the robot does not have the full knowledge of the environment *a priori*. Therefore, Problem 2 is not only the problem of information gathering but also *planning for estimation* as the robot needs to infer the map (and, in general, its pose in the SLAM problem) sequentially. Note that we do not impose any assumptions on the observability of the robot pose and the map; therefore, they can be partially observable as is the case in POMDPs.

The aforementioned problems are both in their offline (non-adaptive) and online (adaptive) forms are NP-hard (Singh et al., 2009). We build our proposed incremental information-gathering algorithm on top of the RIG to solve the interesting problem of autonomous robotic exploration in unknown environments. Furthermore, since the ultimate goal is online applications, we only consider the RIG-tree variant to be extended for sequential planning. This conclusion stems from extensive comparisons of RIG variants provided in Hollinger and Sukhatme (2014). However, we acknowledge that the RIG graph is an interesting case to

**Algorithm 1.** RIG-tree()

---

**Require:** Step size  $\Delta$ , budget  $b$ , free space  $\mathcal{X}_f$ , Environment  $\mathcal{M}$ , start configuration  $\mathbf{x}_{start}$ , near radius  $r$ ;

1. // Initialize cost, information, starting node, node list, edge list, and tree
2.  $I_{init} \leftarrow \text{Information}([], \mathbf{x}_{start}, \mathcal{M})$ ,  $C_{init} \leftarrow 0$ ,  $n \leftarrow \langle \mathbf{x}_{start}, C_{init}, I_{init} \rangle$
3.  $\mathcal{V} \leftarrow \{n\}$ ,  $\mathcal{V}_{closed} \leftarrow \emptyset$ ,  $\mathcal{E} \leftarrow \emptyset$
4. **while** not terminated **do**
5.   // Sample configuration space of vehicle and find nearest node
6.    $\mathbf{x}_{sample} \leftarrow \text{Sample}(\mathcal{X}_f)$
7.    $\mathbf{x}_{nearest} \leftarrow \text{Nearest}(\mathbf{x}_{sample}, \mathcal{V} \setminus \mathcal{V}_{closed})$
8.    $\mathbf{x}_{feasible} \leftarrow \text{Steer}(\mathbf{x}_{nearest}, \mathbf{x}_{sample}, \Delta)$
9.   // Find near points to be extended
10.    $\mathcal{V}_{near} \leftarrow \text{Near}(\mathbf{x}_{feasible}, \mathcal{V} \setminus \mathcal{V}_{closed}, r)$
11.   **for** all  $n_{near} \in \mathcal{V}_{near}$  **do**
12.     // Extend towards new point
13.      $\mathbf{x}_{new} \leftarrow \text{Steer}(\mathbf{x}_{near}, \mathbf{x}_{feasible}, \Delta)$
14.     **if**  $\text{NoCollision}(\mathbf{x}_{near}, \mathbf{x}_{new}, \mathcal{X}_f)$  **then**
15.       // Calculate new information and cost
16.        $I_{new} \leftarrow \text{Information}(I_{near}, \mathbf{x}_{new}, \mathcal{M})$
17.        $c(\mathbf{x}_{new}) \leftarrow \text{Cost}(\mathbf{x}_{near}, \mathbf{x}_{new})$
18.        $C_{new} \leftarrow C_{near} + c(\mathbf{x}_{new})$ ,  $n_{new} \leftarrow \langle \mathbf{x}_{new}, C_{new}, I_{new} \rangle$
19.       **if**  $\text{Prune}(n_{new})$  **then**
20.         **delete**  $n_{new}$
21.       **else**
22.         // Add edges and nodes
23.          $\mathcal{E} \leftarrow \cup\{(n_{near}, n_{new})\}$ ,  $\mathcal{V} \leftarrow \cup\{n_{new}\}$
24.         // Add to closed list if budget exceeded
25.         **if**  $C_{new} > b$  **then**
26.            $\mathcal{V}_{closed} \leftarrow \mathcal{V}_{closed} \cup \{n_{new}\}$
27.         **end if**
28.       **end if**
29.       **end if**
30.     **end for**
31. **end while**
32. **return**  $\mathcal{T} = (\mathcal{V}, \mathcal{E})$

---

consider as under a *partial ordering* assumption it is *asymptotically optimal*.

#### 4.2. RIG algorithms

The sampling-based RIG algorithms find a trajectory that maximizes an information quality metric with respect to a pre-specified budget constraint (Hollinger and Sukhatme, 2014). The RIG is based on RRT\*, RRG, and PRM\* (Karaman and Frazzoli, 2011) and borrow the notion of informative path planning from branch and bound optimization (Binney and Sukhatme, 2012). Algorithm 1 shows the RIG-tree algorithm. The functions that are used in the algorithm are explained as follows.

**Cost:** The cost function assigns a strictly positive cost to a collision-free path between two points from the free space  $\mathcal{X}_f$ .

**Information:** This function quantifies the information quality of a collision-free path between two points from the free space  $\mathcal{X}_f$ .

**Sample:** This function returns i.i.d. samples from  $\mathcal{X}_f$ .

**Nearest:** Given a graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , where  $\mathcal{V} \subset \mathcal{X}_f$ , and a query point  $\mathbf{x} \in \mathcal{X}_f$ , this function returns a vertex  $v \in \mathcal{V}$  that has the “closest” distance to the query point.<sup>3</sup>

**Steer:** This function extends nodes towards newly sampled points and allows for constraints on motion of the robot.<sup>4</sup>

**Near:** Given a graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , where  $\mathcal{V} \subset \mathcal{X}_f$ , a query point  $\mathbf{x} \in \mathcal{X}_f$ , and a positive real number  $r \in \mathbb{R}_{>0}$ , this function returns a set of vertices  $\mathcal{V}_{near} \subseteq \mathcal{V}$  that are contained in a ball of radius  $r$  centered at  $\mathbf{x}$ .

**NoCollision:** Given two points  $\mathbf{x}_a, \mathbf{x}_b \in \mathcal{X}_f$ , this function returns **true** if the line segment between  $\mathbf{x}_a$  and  $\mathbf{x}_b$  is collision-free and **false** otherwise.

**Prune:** This function implements a pruning strategy to remove nodes that are not “promising.” This can be achieved through defining a *partial ordering* for co-located nodes.

In lines 2–3 the algorithm initializes the starting node of the graph (tree). In lines 6–8, a sample point from workspace  $\mathcal{X}$  is drawn and is converted into a feasible point, from its nearest neighbor in the graph. Line 10 extracts all nodes from the graph that are within radius  $r$  of the feasible point. These nodes are candidates for extending the graph, and each node is converted into a new node using the Steer function in line 13. In lines 14–18, if there exists a collision-free path between the candidate node and the new node, the information gain and cost of the new node are

evaluated. In lines 19–26, if the new node does not satisfy a partial ordering condition, it is pruned; otherwise, it is added to the graph. Furthermore, the algorithm checks for the budget constraint violation. The output is a graph that contains a subset of traversable paths with maximum information gain.

#### 4.3. System dynamics

The equation of motion of the robot is governed by the non-linear partially observable equation as follows:

$$\mathbf{x}_{t+1}^- = f(\mathbf{x}_t, \mathbf{u}_t, \mathbf{w}_t), \quad \mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_t) \quad (13)$$

moreover, with appropriate linearization at the current state estimate, we can predict the state covariance matrix as

$$\Sigma_{t+1}^- = \mathbf{F}_t \Sigma_t \mathbf{F}_t^T + \mathbf{W}_t \mathbf{Q}_t \mathbf{W}_t^T \quad (14)$$

where  $\mathbf{F}_t = \frac{\partial f}{\partial \mathbf{x}}|_{\mathbf{x}_t, \mathbf{u}_t}$  and  $\mathbf{W}_t = \frac{\partial f}{\partial \mathbf{w}}|_{\mathbf{x}_t, \mathbf{u}_t}$  are the Jacobian matrices calculated with respect to  $\mathbf{x}$  and  $\mathbf{w}$ , respectively.

## 5. IIG: incrementally-exploring information gathering

In this section, we present the IIG algorithm, which is essentially RIG with an information-theoretic convergence condition. The algorithmic implementation of IIG is shown in Algorithm 2. We employ IIG to solve the robotic exploration problem with the partially observable state. Both RIG and IIG, through incremental sampling, search the space of possible trajectories to find the maximally informative path; however, owing to the automatic convergence of the IIG, it is possible to run the algorithm online without full knowledge of the state, i.e. the map and robot poses.

We introduce the relative information contribution (RIC) criterion to detect the convergence of the search. The motivation behind this definition is that the number of nodes constantly increases unless an aggressive pruning strategy is used. However, an aggressive pruning strategy leads to potentially pruning nodes that can be part of optimal solutions.<sup>5</sup> Even though the algorithm continues to add nodes, it is possible to evaluate the contribution of each added node in the relative information sense. In other words, adding nodes does not affect the convergence of the algorithm, but does affect the amount of information the algorithm can collect by continuing the search. We define the RIC of a node as follows.

**Definition 3. (RIC).** In Algorithm 2, let  $\mathbf{x}_{new} \in \mathcal{X}_f$  be a reachable point through a neighboring node  $n_{near} \in \mathcal{V}$  returned by the function `Near()`. Let  $I_{new}$  and  $I_{near}$  be the information values of their corresponding nodes returned by the function `Information()`. The RIC of node  $n_{new}$  is defined as

$$RIC \triangleq \frac{I_{new}}{I_{near}} - 1 \quad (15)$$

Equation (15) is conceptually important as it defines the amount of information gain relative to a neighboring point in the IIG graph. In practice, the number of samples it takes before the algorithm finds a new node becomes important. Thus, we define penalized RIC that is computed in line 25.

**Definition 4 (Penalized RIC).** Let the RIC computed using Equation (15). Let  $n_{sample}$  be the number of samples it takes to find the node  $n_{new}$ . The penalized RIC is defined as

$$I_{RIC} \triangleq \frac{RIC}{n_{sample}} \quad (16)$$

An appealing property of  $I_{RIC}$  is that it is non-dimensional, and it does not depend on the actual calculation/approximation of the information values. In practice, as long as the information function satisfies the RIG/IIG requirements, using the following condition, IIG algorithm converges. Let  $\delta_{RIC}$  be a threshold that is used to detect the convergence of the algorithm. Through averaging  $I_{RIC}$  values over a window of size  $n_{RIC}$ , we ensure that continuing the search will not add any significant amount of information to the IIG graph. In Algorithm 2, this condition is shown in line 6 by function `AverageRIC`.

**Remark 4.** In Algorithm 2,  $\delta_{RIC}$  sets the planning horizon from the information-gathering point of view. Through using smaller values of  $\delta_{RIC}$  the planner can reach further points in both spatial and belief space. In other words, if  $\delta_{RIC} \rightarrow 0$ , then  $T \rightarrow \infty$ .

## 6. Information functions algorithms

We propose two classes of algorithms to approximate the information gain at any sampled point from the free workspace. The information function in RIG/IIG algorithms often causes a bottleneck and computationally dominates the other parts. Therefore, even for offline calculations, it is important to have access to functions that, concurrently, are computationally tractable and can capture the essence of information gathering. We emphasize that the information functions are directly related to the employed sensors. However, once the model is provided and incorporated into the estimation/prediction algorithms, the information-theoretic aspects of the provided algorithms remain the same.

The information functions that are proposed are different in nature. First, we discuss MI-based information functions whose calculations explicitly depend on the probabilistic sensor model. We provide a variant of the MI Algorithm in Ghaffari Jadidi et al. (2015, 2016) that is developed for range-finder sensors and based on the beam-based mixture measurement model and the inverse sensor model map prediction (Thrun et al., 2005). We also present an algorithm to approximate the MI upper bound (MIUB) which reveals the maximum achievable information gain.

**Algorithm 2.**  $\text{IIG-tree}()$ 

**Require:** Step size  $\Delta$ , budget  $b$ , free space  $\mathcal{X}_f$ , Environment  $\mathcal{M}$ , start configuration  $\mathbf{x}_{start}$ , near radius  $r$ , relative information contribution threshold  $\delta_{RIC}$ , averaging window size  $n_{RIC}$ ;

```

1. // Initialize cost, information, starting node, node list, edge list, and tree
2.  $I_{init} \leftarrow \text{Information}([], \mathbf{x}_{start}, \mathcal{M})$ ,  $C_{init} \leftarrow 0$ ,  $n \leftarrow \langle \mathbf{x}_{start}, C_{init}, I_{init} \rangle$ 
3.  $\mathcal{V} \leftarrow \{n\}$ ,  $\mathcal{V}_{closed} \leftarrow \emptyset$ ,  $\mathcal{E} \leftarrow \emptyset$ 
4.  $n_{sample} \leftarrow 0$  // Number of samples
5.  $I_{RIC} \leftarrow \emptyset$  // Relative information contribution
6. while  $\text{AverageRIC}(I_{RIC}, n_{RIC}) > \delta_{RIC}$  do
7.   // Sample configuration space of vehicle and find nearest node
8.    $\mathbf{x}_{sample} \leftarrow \text{Sample}(\mathcal{X}_f)$ 
9.    $n_{sample} \leftarrow n_{sample} + 1$  // Increment sample counter
10.   $\mathbf{x}_{nearest} \leftarrow \text{Nearest}(\mathbf{x}_{sample}, \mathcal{V} \setminus \mathcal{V}_{closed})$ 
11.   $\mathbf{x}_{feasible} \leftarrow \text{Steer}(\mathbf{x}_{nearest}, \mathbf{x}_{sample}, \Delta)$ 
12.  // Find near points to be extended
13.   $\mathcal{V}_{near} \leftarrow \text{Near}(\mathbf{x}_{feasible}, \mathcal{V} \setminus \mathcal{V}_{closed}, r)$ 
14.  for all  $n_{near} \in \mathcal{V}_{near}$  do
15.    // Extend towards new point
16.     $\mathbf{x}_{new} \leftarrow \text{Steer}(\mathbf{x}_{near}, \mathbf{x}_{feasible}, \Delta)$ 
17.    if  $\text{NoCollision}(\mathbf{x}_{near}, \mathbf{x}_{new}, \mathcal{X}_f)$  then
18.      // Calculate new information and cost
19.       $I_{new} \leftarrow \text{Information}(I_{near}, \mathbf{x}_{new}, \mathcal{M})$ 
20.       $c(\mathbf{x}_{new}) \leftarrow \text{Cost}(\mathbf{x}_{near}, \mathbf{x}_{new})$ 
21.       $C_{new} \leftarrow C_{near} + c(\mathbf{x}_{new})$ ,  $n_{new} \leftarrow \langle \mathbf{x}_{new}, C_{new}, I_{new} \rangle$ 
22.      if  $\text{Prune}(n_{new})$  then
23.        delete  $n_{new}$ 
24.      else
25.         $I_{RIC} \leftarrow \text{append}(I_{RIC}, (\frac{I_{new}}{I_{near}} - 1) / n_{sample})$  // Equation (16)
26.         $n_{sample} \leftarrow 0$  // Reset sample counter
27.        // Add edges and nodes
28.         $\mathcal{E} \leftarrow \cup\{(n_{near}, n_{new})\}$ ,  $\mathcal{V} \leftarrow \cup\{n_{new}\}$ 
29.        // Add to closed list if budget exceeded
30.        if  $C_{new} > b$  then
31.           $\mathcal{V}_{closed} \leftarrow \mathcal{V}_{closed} \cup \{n_{new}\}$ 
32.        end if
33.      end if
34.    end if
35.  end for
36. end while
37. return  $\mathcal{T} = (\mathcal{V}, \mathcal{E})$ 

```

Then, we exploit the property of GPs to approximate the information gain. In Equation (7), the variance calculation does not explicitly depend on the target vector (measurements) realization. In this case, as long as the underlying process is modeled as GPs, the information gain can be calculated using prior and posterior variances which removes the need to rely on a specific sensor model and calculate the expectation over future measurements. However, note that the hyperparameters of the covariance functions are learned using the training set which contains measurements; therefore, the knowledge of underlying process and measurements is incorporated into the GP through its hyperparameters. Once we established GP variance reduction (GPVR) algorithm, we then use the expected kernel notion (Ghaffari Jadidi et al., 2017) to propagate pose uncertainty into the covariance function resulting in uncertain GP variance reduction (UGPVR) algorithm. In particular, these two information functions are interesting for the following reasons.

- (i) Unlike MI-based (direct information gain calculation), they are non-parametric.
- (ii) GPVR-based information functions provide a systematic way to incorporate input (state) uncertainty into information-gathering frameworks.
- (iii) In the case of incomplete knowledge about the quantity of interest in an unknown environment, they allow for *active learning*.<sup>6</sup>

### 6.1. Mutual information

To calculate MI without information “double-counting” we need to update the map after every measurement prediction. It is possible to perform map prediction using a forward or inverse sensor model (Thrun et al., 2005). Typically using an inverse sensor model results in simpler calculations. We first define two required parameters in the proposed algorithm as follows.

**Definition 5.** (Map saturation probability). The probability that the robot is completely confident about the occupancy status of a point is defined as  $p_{sat}$ .

**Algorithm 3.** InformationMI()

---

**Require:** Robot pose or desired location, current map estimate  $m$ , sensor model, saturation probability  $p_{sat}$ , free point belief  $b_{free}$ , occupied point belief  $b_{occ}$ , numerical integration resolution  $s_z$ , near node information  $I_{near}$ ;

1.  $\bar{m} \leftarrow m$  // Initialize updated map as the current map
2. **if**  $I_{near}$  is not empty **then** // Initialize information gain
3.    $I \leftarrow I_{near}$
4. **else**
5.    $I \leftarrow 0$
6. **end if**
7. // Compute saturation entropy
8.  $h_{sat} \leftarrow -[p_{sat} \log(p_{sat}) + (1 - p_{sat}) \log(1 - p_{sat})]$
9. **for** all  $\alpha$  **do** // Loop over all sensor beams
10.   Compute  $\hat{z}^{[\alpha]}$  and  $\mathcal{I}^{[\alpha]}$  using ray casting in  $m$
11.   // Calculate map conditional entropy along beam  $\alpha$
12.   **for**  $i \in \mathcal{I}^{[\alpha]}$  **do**
13.     // Entropy of point  $i$
14.      $h_i \leftarrow -[\bar{m}^{[i]} \log(\bar{m}^{[i]}) + (1 - \bar{m}^{[i]}) \log(1 - \bar{m}^{[i]})]$
15.     **if**  $h_i < h_{sat}$  **then**
16.       **continue**
17.     **else**
18.        $I \leftarrow I + h_i$  // Add to information gain
19.     **end if**
20.      $\bar{h} \leftarrow 0$  // Initialize map conditional entropy
21.      $z \leftarrow s_z^{-1}$  // Initialize range dummy variable
22.     **while**  $z \leq \hat{z}^{[\alpha]}$  **do**
23.       // Calculate marginal measurement probability  $p_z$
24.        $p_1 \leftarrow p(z|M=0)$
25.        $p_2 \leftarrow 0$
26.       **for**  $j \in \mathcal{I}^{[\alpha]}$  **do**
27.          $p_1 \leftarrow p_1(1 - m^{[j]})$
28.          $p_2 \leftarrow p_2 + p(z|M=m^{[j]})m^{[j]} \prod_{l < j} (1 - m^{[l]})$
29.       **end for**
30.        $p_z \leftarrow p_1 + p_2$
31.       // Map prediction at point  $i$  along beam  $\alpha$  using inverse sensor model
32.       **if**  $\text{isFree}(\bar{m}^{[i]})$  **then**
33.          $\bar{m}^{[i]} \leftarrow \max(p_{sat} - \epsilon, b_{free} * \bar{m}^{[i]})$
34.       **else**
35.          $\bar{m}^{[i]} \leftarrow \min(1 - p_{sat} + \epsilon, b_{occ} * \bar{m}^{[i]})$
36.       **end if**
37.        $\bar{h} \leftarrow \bar{h} + p_z[\bar{m}^{[i]} \log(\bar{m}^{[i]}) + (1 - \bar{m}^{[i]}) \log(1 - \bar{m}^{[i]})]$
38.        $z \leftarrow z + s_z^{-1}$  // Increase range along the beam
39.     **end while**
40.      $I \leftarrow I + \bar{h}s_z^{-1}$
41.   **end for**
42. **end for**
43. **return**  $I$  (total information gain),  $\bar{m}$  (updated map)

---

**Definition 6.** (Map saturation entropy). The entropy of a point from a map whose occupancy probability is  $p_{sat}$ , is defined as  $h_{sat} \triangleq H(p_{sat})$ .

The defined parameters are relevant because they prevent the exhaustive search for information in “less-important” areas. The MI-based information function using an inverse sensor model implementation is shown in Algorithm 3. In line 10, the predicted range measurement for beam  $\alpha$ ,  $\hat{z}^{[\alpha]}$ , is computed using ray casting in the current map estimate where  $\mathcal{I}^{[\alpha]}$  denotes the index set of map points that are in the perception field of the  $\alpha$ th sensor beam. In lines 15–19, the algorithm skips any map point whose entropy surpasses the saturation entropy or adds the map entropy of point  $i$  to the information gain,  $I$ . In lines

20–39, the map conditional entropy by integrating over future measurement is calculated where, in lines 32–36, map prediction is performed using free point belief,  $0 < b_{free} < 1$ , and occupied point belief,  $b_{occ} > 1$ . The predicted probabilities are clamped using  $\epsilon > 0$ , which is a small number relative to  $p_{sat}$  to avoid losing numerical computation accuracy. In line 40, the map conditional entropy is subtracted from initial map entropy ( $\bar{h}$  is negative) using an appropriate numerical integration resolution,  $s_z$ .

It is also interesting to calculate an upper bound for the information gain. Given Algorithm 3, it is trivial to calculate the MIUB using the total amount of map entropy in the current perception field of the robot. It is faster to compute

**Algorithm 4.** InformationMIUB()

**Require:** Robot pose or desired location, current map estimate  $m$ , sensor model, saturation probability  $p_{sat}$ , free point belief  $b_{free}$ , occupied point belief  $b_{occ}$ , numerical integration resolution  $s_z$ , near node information  $I_{near}$ ;

```

1.  $\bar{m} \leftarrow m$  // Initialize updated map as the current map
2. if  $I_{near}$  is not empty then // Initialize information gain
3.    $I_{UB} \leftarrow I_{near}$ 
4. else
5.    $I_{UB} \leftarrow 0$ 
6. end if
7. // Compute saturation entropy
8.  $h_{sat} \leftarrow -[p_{sat} \log(p_{sat}) + (1 - p_{sat}) \log(1 - p_{sat})]$ 
9. for all  $\alpha$  do // Loop over all sensor beams
10.   Compute  $\hat{z}^{[\alpha]}$  and  $\mathcal{I}^{[\alpha]}$  using ray casting in  $m$ 
11.   for  $i \in \mathcal{I}^{[\alpha]}$  do
12.     // Entropy of point  $i$ 
13.      $h_i \leftarrow -[\bar{m}^{[i]} \log(\bar{m}^{[i]}) + (1 - \bar{m}^{[i]}) \log(1 - \bar{m}^{[i]})]$ 
14.     if  $h_i < h_{sat}$  then
15.       continue
16.     else
17.        $I_{UB} \leftarrow I_{UB} + h_i$  // Add to information gain
18.     end if
19.     // Map prediction at point  $i$  along beam  $\alpha$  using inverse sensor model
20.     if isFree( $\bar{m}^{[i]}$ ) then
21.        $\bar{m}^{[i]} \leftarrow \max(p_{sat} - \epsilon, b_{free} * \bar{m}^{[i]})$ 
22.     else
23.        $\bar{m}^{[i]} \leftarrow \min(1 - p_{sat} + \epsilon, b_{occ} * \bar{m}^{[i]})$ 
24.     end if
25.   end for
26. end for
27. return  $I_{UB}$  (total information gain),  $\bar{m}$  (updated map)

```

the upper bound as it only shows the uncertainty from the current map and it does not consider any gain from future measurements. However, in practice, it can be useful for fast and online predictions. More details regarding the difference between maximizing MI and entropy are discussed in Guestrin et al. (2005) and Krause et al. (2008).

**Lemma 1** (Information gain upper bound). *For any location in the map, the information gain upper bound is given by the total map entropy calculated using map points in the perception field of the robot at the same location.*

*Proof.* From Theorem 4,  $0 \leq I(M; Z) = H(M) - H(M|Z) \leq H(M)$  which extends to any submap  $M_{sub}$  that is in the perception field of the robot. Note that MI beyond the perception field is zero as  $\forall M \in \mathcal{M} \setminus \mathcal{M}_{sub}, M \perp Z$ .  $\square$

Algorithm 4 shows the MIUB information function in which the integration over predicted measurement is omitted. However, to avoid information double-counting, it is still required to update the map estimate after each function call. Furthermore, the MIUB calculation can be integrated into Algorithm 3 with an insignificant computational load through calculation of  $I_{UB}$  alongside  $I$  in lines 2–6 and 18. As we show later, in the early stage of the search the behavior of  $I_{UB}$  is similar to that of  $I$ , therefore, it is possible to use the upper bound at the beginning of the sampling to speed up the search.

## 6.2. GP variance reduction

Variance reduction is the essence of information gathering. Since predictive variance calculation in Equation (7), does not depend on observations, we can come up with a non-parametric algorithm to estimate variance reduction throughout dense belief representation of the map. For the problem of informative path planning, a similar approach is used in Binney et al. (2013) where the reduction in the trace of the covariance function is considered as the objective function (A-optimality). Here, we are interested in approximating the MI through entropy reduction, i.e. using determinant of the covariance matrix (D-optimality) (Pukelsheim, 2006). This is mainly to keep the proposed IIG framework agnostic about the choice of information functions. We treat each map point as a continuous random variable that is normally distributed. Therefore, we use differential entropy formulation for MI approximation. Differential entropy of a Gaussian random variable,  $X \sim \mathcal{N}(\mu, \sigma^2)$ , can be derived as  $h(X) = \frac{1}{2} \log(2\pi e \sigma^2)$ ; and where  $X \sim \mathcal{N}(2m, \Sigma)$  is a Gaussian random vector of dimension  $n$ , the differential entropy can be derived as  $h(X) = \frac{1}{2} \log((2\pi e)^n |\Sigma|)$ . Now in a Bayesian setup, let  $X \sim \mathcal{N}(\mu_X, \Sigma_X)$  and  $X|Z \sim \mathcal{N}(\mu_{X|Z}, \Sigma_{X|Z})$  be the prior and posterior distribution of the random vector  $X$ . Subsequently, it follows that the MI after receiving observation  $Z$  can be derived as

$$I(X; Z) = \frac{1}{2} [\log(|\Sigma_X|) - \log(|\Sigma_{X|Z}|)] \quad (17)$$

Where possible, the MI should be computed using the full covariance matrix or its block-diagonal approximation. However, typically for large problems with dense belief representations, maintaining and updating the full covariance matrix for thousands of random variables is not tractable. Therefore, to approximate the MI, we suggest a trade-off approach between the tractability and accuracy based on the problem at hand. In the following, we propose an approximation of the MI using the marginalization property of normal distribution. We also discuss the relation of this approximation with the exact MI.

**Lemma 2.** (Marginalization property of a normal distribution mathematical). *Let  $\mathbf{x}$  and  $\mathbf{y}$  be jointly Gaussian random vectors*

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_x \\ \boldsymbol{\mu}_y \end{bmatrix}, \begin{bmatrix} \mathbf{A} & \mathbf{C}^T \\ \mathbf{C} & \mathbf{B} \end{bmatrix}\right) \quad (18)$$

then the marginal distribution of  $\mathbf{x}$  is

$$\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}_x, \mathbf{A}) \quad (19)$$

**Proposition 9.** *Let  $X_1, X_2, \dots, X_n$  have a multivariate normal distribution with covariance matrix  $\mathbf{K}$ . The MI between  $X$  and observations  $Z$  can be approximated as*

$$\hat{I}(X; Z) = \frac{1}{2} \left[ \sum_{i=1}^n \log(\sigma_{X_i}) - \sum_{i=1}^n \log(\sigma_{X_i|Z}) \right] \quad (20)$$

where  $\sigma_{X_i}$  and  $\sigma_{X_i|Z}$  are marginal variances for  $X_i$  before and after incorporating observations  $Z$ , i.e. prior and posterior marginal variances.

*Proof.* Using the marginalization property of a normal distribution, Lemma 2, for every  $X_i$  we have  $\mathbb{V}[X_i] = \mathbf{K}^{[i, i]}$ . Using differential entropy of  $X_i$ , the MI for  $X_i$  can be written as

$$\hat{I}^{[i]}(X_i; Z) = \frac{1}{2} [\log(\sigma_{X_i}) - \log(\sigma_{X_i|Z})] \quad (21)$$

and the total MI can be calculated as  $\hat{I}(X; Z) = \sum_{i=1}^n \hat{I}^{[i]}(X_i; Z)$ . Alternatively, one could build a new covariance matrix by placing marginal variances on its diagonal and use the fact the determinant of a diagonal matrix is the product of its diagonal elements.  $\square$

This approximation makes the information gain calculation for a class of problem with dense belief representation tractable. However, it is interesting to study the effect of this approximation. Intuitively, the determinant of the covariance matrix corresponds to the hypervolume of the subspace spanned by the columns of the covariance matrix. When ignoring the correlation between random variables, Proposition 9, the spanned subspace becomes larger; as a result, the determinant grows which corresponds to higher

entropy. Regarding the information gain, the following statement holds true.

**Proposition 10.** *We have  $\hat{I}(X; Z) \leq I(X; Z)$ .*

*Proof.* See Appendix A.  $\square$

Algorithm 5 shows the details of GPVR information function.<sup>7</sup> Based on the training points generated from predicted measurement  $\mathbf{z}$ , a submap from the current map estimate using nearest-neighbor search is found, lines 7–14. In lines 16–20, GP predictive variances,  $\mathbf{v} = \text{vec}(\mathbf{v}^{[1]}, \dots, \mathbf{v}^{[|\mathcal{M}_D|]})$ , are computed using covariance function  $k(\cdot, \cdot)$  with the same hyperparameters learned for the map inference. In lines 21–24, using Bayesian committee machine (BCM) fusion (Tresp, 2000) the predictive marginal posterior variance is calculated and the information gain is updated consequently.<sup>8</sup> BCM combines estimators which were trained on different datasets and is shown to be suitable for incremental map building (Ghaffari Jadidi et al., 2016). The Cholesky factorization is the most computationally expensive operation of the algorithm. However, it is possible to exploit a sparse covariance matrix such as the kernel in Melkumyan and Ramos (2009) or use a cut-off distance for the covariance function<sup>9</sup> to speed up the algorithm.

We emphasize that to use GPVR algorithm, the underlying process needs to be modeled using GPs, i.e.  $y(\mathbf{x}) \sim \mathcal{GP}(f_m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$  where  $f_m(\mathbf{x})$  is the GP mean function. Therefore, for any map point we have  $m^{[i]} = y(\mathbf{x}_*^{[i]}) \sim \mathcal{N}(\boldsymbol{\mu}^{[i]}, \sigma^{[i]})$ . Furthermore, construction of the training set, in line 8, is part of the GP modeling. For the particular case of occupancy mapping using a range-finder sensor, see (Ghaffari Jadidi, 2017: Chapter 4). For the case where the robot only receives point measurements at any location, such as wireless signal strength, we explain further in Section 9.4.

### 6.3. Uncertain GP variance reduction

Thus far, the developed information functions do not incorporate uncertainties of other state variables that are jointly distributed with the map (such as the robot pose) in information gain calculation. We define the modified kernel  $\tilde{k}$  as follows.

**Definition 7.** (Modified kernel). Let  $k(x, x_*)$  be a kernel and  $X \in \mathcal{X}$  a random variable that is distributed according to a probability distribution function  $p(x)$ . The modified kernel is defined as its expectation with respect to  $p(x)$ , therefore we can write

$$\tilde{k} = \mathbb{E}[k] = \int_{\Omega} k \, dp \quad (22)$$

Through replacing the kernel function in Algorithm 5 with the modified kernel we can propagate the robot pose uncertainty in the information gain calculation. Intuitively, under the presence of uncertainty in other state variables that are correlated with the map, the robot does not take

**Algorithm 5.** InformationGPVR()

---

**Require:** Robot pose or desired location  $p$ , current map/state estimate  $m$ , covariance function  $k(\cdot, \cdot)$ , sensor noise  $\sigma_n^2$ , near node information  $I_{near}$ ;

1.  $\bar{\sigma} \leftarrow \sigma$  // Initialize updated map variance as the current map variance
2. **if**  $I_{near}$  is not empty **then** // Initialize information gain
3.    $I \leftarrow I_{near}$
4. **else**
5.    $I \leftarrow 0$
6. **end if**
7.  $z \leftarrow$  Predict future measurements using  $p$  and  $m$
8.  $\mathcal{D} \leftarrow$  Construct training set using  $z$  and  $p$
9. // Find the corresponding nearest submap
10.  $\mathcal{M}_{\mathcal{D}} \leftarrow \emptyset$
11. **for** all  $x \in \mathcal{D}$  **do**
12.    $x_{nearest} \leftarrow \text{Nearest}(x, \mathcal{M})$
13.    $\mathcal{M}_{\mathcal{D}} \leftarrow \mathcal{M}_{\mathcal{D}} \cup \{x_{nearest}\}$
14. **end for**
15. // Calculate self-covariance and cross-covariance matrices
16.  $\mathbf{C} \leftarrow \mathbf{K}(X, X), \mathbf{C}_* \leftarrow \mathbf{K}(X, X_*)$  //  $X \in \mathcal{D}$  and  $X_* \in \mathcal{M}_{\mathcal{D}}$
17. // Calculate vector of diagonal variances for test points
18.  $\mathbf{c}_{**} \leftarrow \text{diag}(\mathbf{K}(X_*, X_*))$
19.  $\mathbf{L} \leftarrow \text{Cholesky}(\mathbf{C} + \sigma_n^2 \mathbf{I}), \mathbf{V} \leftarrow \mathbf{L} \backslash \mathbf{C}_*$
20.  $\mathbf{v} \leftarrow \mathbf{c}_{**} - \text{dot}(\mathbf{V}, \mathbf{V})^T$  // dot product
21. **for** all  $i \in \mathcal{M}_{\mathcal{D}}$  **do**
22.    $\bar{\sigma}^{[i]} \leftarrow ((\sigma^{[i]})^{-1} + (\mathbf{v}^{[i]})^{-1})^{-1}$  // BCM fusion
23.    $I \leftarrow I + \log(\sigma^{[i]}) - \log(\bar{\sigma}^{[i]})$
24. **end for**
25. **return**  $I$  (total information gain),  $\bar{\sigma}$  (updated map variance)

---

greedy actions as the amount of available information calculated using the modified kernel is less than the original case. Therefore, the chosen actions are relatively more conservative. The integration in Equation (22) can be numerically approximated using Monte Carlo or Gauss–Hermite quadrature techniques (Davis and Rabinowitz, 1984; Press et al., 1996). In the case of a Gaussian assumption for the robot pose, Gauss–Hermite quadrature provides a better accuracy and efficiency trade-off and is preferred.

The algorithmic implementation of UGPVR information function is similar to Algorithm 5, where the difference with GPVR is that the input location is not deterministic, i.e. it is approximated as a normal distribution  $\mathcal{N}(p, \Sigma)$ , and the covariance function is replaced by its modified version. In particular, in line 6.2, we calculate self-covariance and cross-covariance matrices using  $\tilde{k}(\cdot, \cdot)$  with respect to  $\mathcal{N}(p, \Sigma)$ . Given the initial pose belief, the pose uncertainty propagation on the IIG graph can be performed using the robot motion model, i.e. using Equations (13) and (14).

The UGPVR estimates at most the same amount of MI as GPVR. This is because of taking the expectation of the kernel with respect to the robot pose posterior. If we pick the mode of the robot pose posterior, GPVR only uses that input point for MI computation. In contrast, UGPVR averages over all possible values of the robot pose within the support of its distribution. Now it is clear that having an estimate of the robot pose posterior with long tails (yet exponentially bounded) reduces the MI even further owing to averaging. Furthermore, if the robot pose is not known

and we have only access to its estimate, ignoring the distribution can lead to overconfident or inconsistent inference/prediction (Ghaffari Jadidi et al., 2017: Figure 2).

## 7. Path extraction and selection

In the absence of artificial targets such as frontiers, in general, there is no goal to be found by the planner. IIG searches for traversable paths within the map, and the resulting tree shows feasible trajectories from the current robot pose to each leaf node, expanded using the maximum information-gathering policy. Therefore, any path in the tree starting from the robot pose to a leaf node is a feasible action. For robotic exploration scenarios, it is not possible to traverse all the available trajectories since, after execution of one trajectory, new measurements are taken, and the map (and the robot pose) belief is updated; therefore, previous predictions are obsolete, and the robot has to enter the planning state.<sup>10</sup> As such, once the RIG/IIG tree is available, the next step can be seen as decision-making where the robot selects a trajectory as an executable action. One possible solution is finding a trajectory in the IIG tree that maximizes the information gain.

We provide a heuristic algorithm based on a voting method. Algorithm 6 shows the implementation of the proposed method. The algorithm first finds all possible paths using a preorder *depth first search*, function DFSpreorder, and then removes paths that are shorter than a minimum length (using parameter  $0 < \kappa < 1$ ),

**Algorithm 6.** PathSelection()

---

**Require:** RIG/IIG tree  $\mathcal{T}$ , path similarity ratio  $s_{ratio}$ ;

1. // The path length equals the number of nodes in the path.
2. // Find all leaves using depth first search
3.  $\mathcal{V}_{leaves} \leftarrow \text{DFSpreorder}(\mathcal{T})$
4. // Find all paths by starting from each leaf and following parent nodes
5.  $\mathcal{P}_{all} \leftarrow \text{Paths2root}(\mathcal{T}, \mathcal{V}_{leaves})$
6.  $l_{max} \leftarrow \text{Find maximum path length in } \mathcal{P}_{all}$
7.  $l_{min} \leftarrow \text{ceil}(\kappa l_{max})$  // Minimum path length,  $0 < \kappa < 1$
8. **for** all  $\mathcal{P} \in \mathcal{P}_{all}$  **do**
9.   **if**  $\text{length}(\mathcal{P}) \leq l_{min}$  **do**
10.     Delete  $\mathcal{P}$
11.   **end if**
12. **end for**
13.  $n_p \leftarrow |\mathcal{P}_{all}|$  // Number of paths in set  $\mathcal{P}_{all}$
14.  $\text{vote} \leftarrow \text{zeros}(n_p, 1)$
15. // Find longest independent paths
16.  $i \leftarrow 1$
17. **while**  $i \leq n_p - 1$  **do**
18.    $j \leftarrow i + 1$
19.   **while**  $j \leq n_p$  **do**
20.      $l_i \leftarrow \text{length}(\mathcal{P}_i), l_j \leftarrow \text{length}(\mathcal{P}_j)$
21.     // Find number of common nodes between paths  $i$  and  $j$ , and the length ratio they share
22.      $l_{ij} \leftarrow \text{SimilarNodes}(\mathcal{P}_i, \mathcal{P}_j) / \min(l_i, l_j)$
23.     **if**  $l_{ij} > s_{ratio}$  **then**
24.       **if**  $l_i > l_j$  **then** // Path  $i$  is longer
25.          $\text{vote}^{[i]} \leftarrow \text{vote}^{[i]} + 1$
26.          $\text{vote}^{[j]} \leftarrow \text{vote}^{[j]} - 1$
27.       **else** // Path  $j$  is longer
28.          $\text{vote}^{[i]} \leftarrow \text{vote}^{[i]} - 1$
29.          $\text{vote}^{[j]} \leftarrow \text{vote}^{[j]} + 1$
30.       **end if**
31.     **else** // Two independent paths
32.        $\text{vote}^{[i]} \leftarrow \text{vote}^{[i]} + 1$
33.        $\text{vote}^{[j]} \leftarrow \text{vote}^{[j]} + 1$
34.     **end if**
35.      $j \leftarrow j + 1$
36. **end while**
37.  $i \leftarrow i + 1$
38. **end while**
39. // Find paths with maximum vote and select the maximally informative path
40.  $\mathcal{P}_{max} \leftarrow \text{MaxVotePath}(\mathcal{P}_{all}, \text{vote})$
41.  $\mathcal{P}_I \leftarrow \text{MaxInformativePath}(\mathcal{P}_{max})$
42. **return**  $\mathcal{P}_I$

---

lines 3–12. Note that the path length and the length returned by function length are integers and correspond to the number of nodes in the path; consequently, the path length in Algorithm 6 is independent of the actual path scale. Then each path is compared with others using the following strategy. If two paths have more than a specified number nodes in common, then we penalize the shorter path by a negative vote and encourage the longer path by a positive vote. However, if two paths do not have many common nodes, then they are considered as two independent paths, and they receive positive votes, lines 13–38. The function SimilarNodes returns the number of overlapping nodes between two paths. In line 40, the function MaxVotePath returns all paths that have the maximum number of votes. There is usually

more than one path with the maximum vote, therefore, in line 41, the function MaxInformativePath selects the path that overall has the maximum information gain.

Note that path-selection algorithm is independent of the IIG algorithm and is a necessary step to choose an action (trajectory) from the IIG tree. In other words, the IIG tree (graph) can be seen as the space of all feasible informative trajectories whereas the path-selection step finds a trajectory from the graph as an executable action, i.e. the final output of Equation (12). If one decides to include a target (goal) while running IIG, once the target has been reached by the planner the decision has been made and, therefore, there is no need to use Algorithm 6. Moreover, if the target is moving, this can be seen as an instance of the target-tracking problem (Levine, 2010).

## 8. Information-theoretic robotic exploration

In this section, we present the information-theoretic basis for applying the IIG-tree algorithm to solve the autonomous robotic exploration problem. Since the developed algorithm does not rely on geometric features (frontiers) for map exploration, an alternative criterion is required for mission termination. We use the entropy independence bound theorem to leverage such a criterion.

**Theorem 11.** (Independence bound on entropy). *Let  $X_1, X_2, \dots, X_n$  be drawn according to  $p(x_1, x_2, \dots, x_n)$ . Then*

$$H(X_1, X_2, \dots, X_n) \leq \sum_{i=1}^n H(X_i) \quad (23)$$

*with equality if and only if the  $X_i$  are independent.*

*Proof.* The proof follows directly from Theorems 1 and 4.  $\square$

This theorem states that the joint entropy is always smaller than the sum of entropies independently and both sides are equal if and only if the random variables are independent. We start from this inequality and prove that for robotic information gathering or map exploration the least upper bound of the average map entropy that is calculated by assuming independence between map points can be a threshold for mission termination. This is formally stated in the following theorem.

**Theorem 12.** (The least upper bound of the average map entropy). *Let  $n \in \mathbb{N}$  be the number of map points. In the limit, for a completely explored occupancy map, the least upper bound of the average map entropy is given by  $h_{sat} = H(p_{sat})$ .*

*Proof.* From Theorem 11 and through multiplying each side of the inequality by  $\frac{1}{n}$ , we can write the average map entropy as

$$\frac{1}{n} H(M) < \frac{1}{n} \sum_{i=1}^n H(M = m^{[i]}) \quad (24)$$

by taking the limit as  $p(m) \rightarrow p_{sat}$ , then

$$\begin{aligned} \lim_{p(m) \rightarrow p_{sat}} \frac{1}{n} H(M) &< \lim_{p(m) \rightarrow p_{sat}} \frac{1}{n} \sum_{i=1}^n H(M = m^{[i]}) \\ \lim_{p(m) \rightarrow p_{sat}} \frac{1}{n} H(M) &< H(p_{sat}) \\ \sup_{p(m)} \frac{1}{n} H(M) &= H(p_{sat}) \end{aligned} \quad (25)$$

$\square$

The result from Theorem 12 is useful because the calculation of the right-hand side of the inequality (24) is trivial. In contrast, calculation of the left-hand side assuming the map belief is represented by a multi-variate Gaussian, requires maintaining the full map covariance matrix and

computation of its determinant. This is not practical, because the map often has a dense belief representation and can be theoretically expanded unbounded (to a very large extent). In the following, we present some notable remarks and consequences of Theorem 12.

**Remark 5.** The result from Theorem 12 also extends to continuous random variables and differential entropy.

**Remark 6.** Note that we do not assume any distribution for map points. The entropy can be calculated either with the assumption that the map points are normally distributed or treating them as Bernoulli random variables.

**Remark 7.** As  $0 < p_{sat} < 1$  and  $H(p_{sat}) = H(1 - p_{sat})$ , one saturation entropy can be set for the entire map.

**Corollary 13.** (Information gathering termination). *Given a saturation entropy  $h_{sat}$ , the problem of the search for information gathering for desired random variables  $X_1, X_2, \dots, X_n$  whose support is alphabet  $\mathcal{X}$  can be terminated when  $\frac{1}{n} \sum_{i=1}^n H(X_i) \leq h_{sat}$ .*

**Corollary 14.** (Map exploration termination). *The problem of autonomous robotic exploration for mapping can be terminated when  $\frac{1}{n} \sum_{i=1}^n H(M = m^{[i]}) \leq H(p_{sat})$ .*

Corollary 13 generalizes the notion of exploration in the sense of information gathering. Therefore, regardless of the quantity of interest, we can provide a stopping criterion for the exploration mission. Corollary 14 is of great importance for the classic robotic exploration for map completion problem as there is no need to resort to geometric frontiers with a specific cluster size to detect map completion. Another advantage of setting a threshold in the information space is the natural consideration of uncertainty in the estimation process before issuing a mission termination signal.

**Remark 8.** Note that MI-based information functions and map exploration termination condition have different saturation probabilities/entropies that are independent and do not necessarily have similar values. Where it is not clear from the context, we make the distinction explicitly clear.

## 9. Results and discussion

In this section, we examine the proposed algorithms in several scenarios. We use MATLAB implementations of the algorithms that are also made publicly available at [https://github.com/MaaniGhaffari/sampling\\_based\\_planners](https://github.com/MaaniGhaffari/sampling_based_planners).

We first design experiments for comparison of information functions under various sensor parameters such as the number of beams and the sensor range, and their effects on the convergence of IIG. Although the primary objective of this experiment is to evaluate the performance of IIG using each information function, we note that such an experiment can also facilitate sensor selection. In other words, given the map of an environment and a number of sensors with different characteristics, how to select the sensor that has a reasonable balance of performance and cost.

**Table 1.** Parameters for IIG-tree experiments. “Online” parameters are only related to the exploration experiments.

| Parameter                              | Symbol   | Value              |
|--|--|--------------------|
| - General parameters:                  |  |                    |
| Occupied probability                   | $P_{occ}$  | 0.65               |
| Unoccupied probability                 | $P_{free}$   | 0.35               |
| Initial position                       | $x_{init}$   | $[10, 2]$ m        |
| Map resolution                         | $\delta_{map}$   | 0.2 m              |
| $I_{RIC}$ threshold                    | $\delta_{RIC}$   | $5 \times 10^{-4}$ |
| $I_{RIC}$ threshold (Online)           | $\delta_{RIC}$   | $1 \times 10^{-2}$ |
| - MI-based parameters:                 |  |                    |
| Hit std                                | $\sigma_{hit}$   | 0.05 m             |
| Short decay                            | $\lambda_{short}$  | 0.2 m              |
| Hit weight                             | $z_{hit}$  | 0.7                |
| Short weight                           | $z_{short}$  | 0.1                |
| Max weight                             | $z_{max}$  | 0.1                |
| Random weight                          | $z_{rand}$   | 0.1                |
| Numerical integration resolution       | $s_z$  | $2 \text{ m}^{-1}$ |
| Saturation probability                 | $p_{sat}$  | 0.05               |
| Saturation probability (Online)        | $p_{sat}$  | 0.3                |
| Occupied belief                        | $b_{occ}$  | 1.66               |
| Unoccupied belief                      | $b_{free}$   | 0.6                |
| - Covariance function hyperparameters: |  |                    |
| Characteristic length-scale            | $l$  | 3.2623 m           |
| Signal variance                        | $\sigma_f^2$   | 0.1879             |
| - Robot motion model:                  |  |                    |
| Motion noise covariance                | $Q = \text{diag}(0.1\text{m}, 0.1\text{m}, 0.0026\text{rad})^2$        |                    |
| Initial pose uncertainty               | $\sigma_{init} = \text{diag}(0.4\text{m}, 0.1\text{m}, 0\text{rad})^2$ |                    |
| - Path selection:                      |  |                    |
| Minimum path length coefficient        | $\kappa$   | 0.4                |
| Path similarity ratio                  | $s_{ratio}$  | 0.6                |
| - Termination condition (Online):      |  |                    |
| Saturation entropy                     | $h_{sat} = H(p_{sat} = 0.1)$   | 0.3251 nats        |

In the second experiment a robot explores an unknown environment; it needs to solve SLAM incrementally, estimate a dense occupancy map representation suitable for planning and navigation, and automatically detect the completion of an exploration mission. Therefore, the purpose of information gathering is map completion while maintaining accurate pose estimation, i.e. planning for estimation. In these experiments the robot pose is estimated through a pose graph algorithm such as Pose SLAM (Ila et al., 2010), and the map of the unknown environment is computed using the incremental Gaussian processes occupancy mapping (I-GPOM) (Ghaffari Jadidi et al., 2016, 2014). Therefore, the state which includes the robot pose and the map is partially observable.

In the third scenario, we demonstrate another possible application of IIG in a lake-monitoring experiment using experimental data collected by an autonomous surface vehicle (ASV). The dataset used for this experiment is publicly available and also used in the original RIG article (Hollinger and Sukhatme, 2014).<sup>11</sup>

We conclude this section by discussing the limitations of the work including our observations and conjectures.

### 9.1. Experimental setup

We first briefly describe the experiment setup that is used in Sections 9.2 and 9.3. The parameters for experiments

and the information functions are listed in Table 1. The robot is equipped with odometric and laser range-finder sensors. The environment is constructed using a binary map of obstacles. For GPVR-based algorithms, the covariance function is Matérn ( $\nu = 5/2$ )

$$k_{VR} = \sigma_f^2 k_{\nu=5/2}(r) = \sigma_f^2 \left( 1 + \frac{\sqrt{5}r}{l} + \frac{5r^2}{3l^2} \right) \exp\left(-\frac{\sqrt{5}r}{l}\right) \quad (26)$$

with hyperparameters that were learned prior to the experiments and are available in Table 1. The modified kernel in the UGPVR algorithm was calculated using Gauss-Hermite quadrature with 11 sample points.

The path-selection parameters and  $\delta_{RIC}$  are found empirically; however, these quantities are non-dimensional, and we expect that one can tune them easily. In particular,  $\delta_{RIC}$  that sets the planning horizon depends on the desired outcome. Setting  $\delta_{RIC} = 0$  is the ideal case that makes the planner infinite-horizon, but in practice owing to the numerical resolution of computer systems, a reasonably higher value (Table 1) should be chosen. Furthermore, a very small value of  $\delta_{RIC}$ , analogous to numerical optimization algorithms, results in the late convergence of the planner that is usually undesirable.

The termination condition requires a saturation entropy. For the case of occupancy mapping, the problem is

well-studied, and we are aware of desired saturation probability that leads to an occupancy map with sufficient confidence for the status of each point/cell. In addition, when a process posterior is represented using mass probabilities such as occupancy maps, finding a saturation probability and, therefore, the corresponding saturation entropy is possible. Perhaps a hard case is when the posterior is a probability density function. In such situations, further knowledge of the marginal posterior distribution of the state variables is required. For example, if we wish to model state variables as Gaussian random variables with the desired variance, we can use differential entropy of the Gaussian distribution to compute the saturation entropy.

The robot pose covariance was approximated using the uncertainty propagation and local linearization of the robot motion model using Equations (13) and (14). Each run was continued until the algorithm converges without any manual intervention. To obtain the performance of each method in the limit and also examine the convergence of IIG, we used an unlimited budget in all experiments. Moreover, the cost is computed as the Euclidean distance between any two nodes and the budget is the maximum allowable travel distance, which is unlimited. The online parameters in Table 1 refer to exploration experiments and are applied in Section 9.3. The exploration experiments use Corollary 14 as the termination condition for the entire mission.

## 9.2. Comparison of information functions

We now compare the proposed information functions by running IIG-tree using different sensor parameters in the Cave map (Howard and Roy, 2003). As in this experiment the map of the environment is given, the map is initialized as an occupancy grid map by assigning each point the occupied or unoccupied probability according to its ground-truth status. For GPVR-based methods, an initial variance map is set to the value of 1 for all points. The experiments are conducted by increasing the sensor number of beams and range, and the results are collected in Tables 2 and 3, respectively.

In the first experiment, we use 10, 20, and 50 sensor beams with the maximum sensor range fixed at  $r_{max} = 5m$ . The information functions used are MI (Algorithm 3), MIUB (Algorithm 4), GPVR (Algorithm 5), and UGPVR algorithm, and the results are presented in Table 2. Convergence is detected when the average of penalized RIC  $I_{RIC}$  over a window of size 30 drops below the threshold  $\delta_{RIC} = 5 \times 10^{-4}$ . The total information gain/cost is calculated using the sum of all edges information/costs. Therefore, it denotes the total information/cost over the searched space and not a particular path. This makes the results independent of the path-selection algorithm.

From Table 2, MIUB has the lowest runtime as expected, followed by MI, GPVR, and UGPVR, respectively. The calculation of MI can be more expensive than GPVR-based algorithms, but with sparse sensor observations (only 10 beams) and a very coarse numerical integration resolution it

performs faster. In particular, small number of sensor beams, given a sufficiently fine map resolution, satisfies the common assumption of independence between the noise in each individual measurement value (Thrun et al., 2005). The conditional independence assumption of measurements for calculation of MI is also discussed in Charrow et al. (2015b, 2014). Empirically, for the MI function, we can observe in Table 2 that as the number of beams increases, the total information gain decreases. We note that increasing the number of beams leads to measurements becoming more correlated, thereby violating the conditional independence assumption between measurements. From Proposition 10, we know that the estimated MI is always less than the actual MI; therefore, increasing the number of beams, given the map resolution used in the experiments, reduces the MI estimation accuracy. Furthermore, while our proof of Proposition 10 is for the case of Gaussian random variables, *data-processing inequality* (Cover and Thomas, 1991) states that processing of measurements, such as those with simplified correlation models, deterministic or random, cannot increase the information gain. It is important to note that observations presented above stem from the assumptions used and developed models and algorithms. Therefore, these results should not be interpreted to indicate that, in general, using a sparse set of measurements the information gain calculation accuracy increases.

For all the compared algorithms, as the number of beams increases, i.e. taking more observations, the computations take longer, and the number of samples/nodes reduces. However, MI has the fastest convergence speed with, approximately, half of the number of samples taken by GPVR. Incorporating the pose uncertainty in UGPVR leads to a slower convergence. This can be explained as a result of the reduction in the information content of each set of observations by adding uncertainty to the information gain calculation. Another interpretation is that the UGPVR algorithm is, relatively speaking, less greedy or more conservative for information gathering. For GPVR-based information functions, increasing the number of sensor beams, leads to higher total information gain owing to a larger training set  $\mathcal{D}$ . MI in comparison with MIUB has more realistic information gain estimation which is reflected in less total cost on average. In other words, the upper bound of the estimated MI, most likely, overestimate the actual information gain (Lemma 1). The latter results also confirm the advantage of maximizing the MI rather than minimizing the entropy that is discussed in Krause et al. (2008).

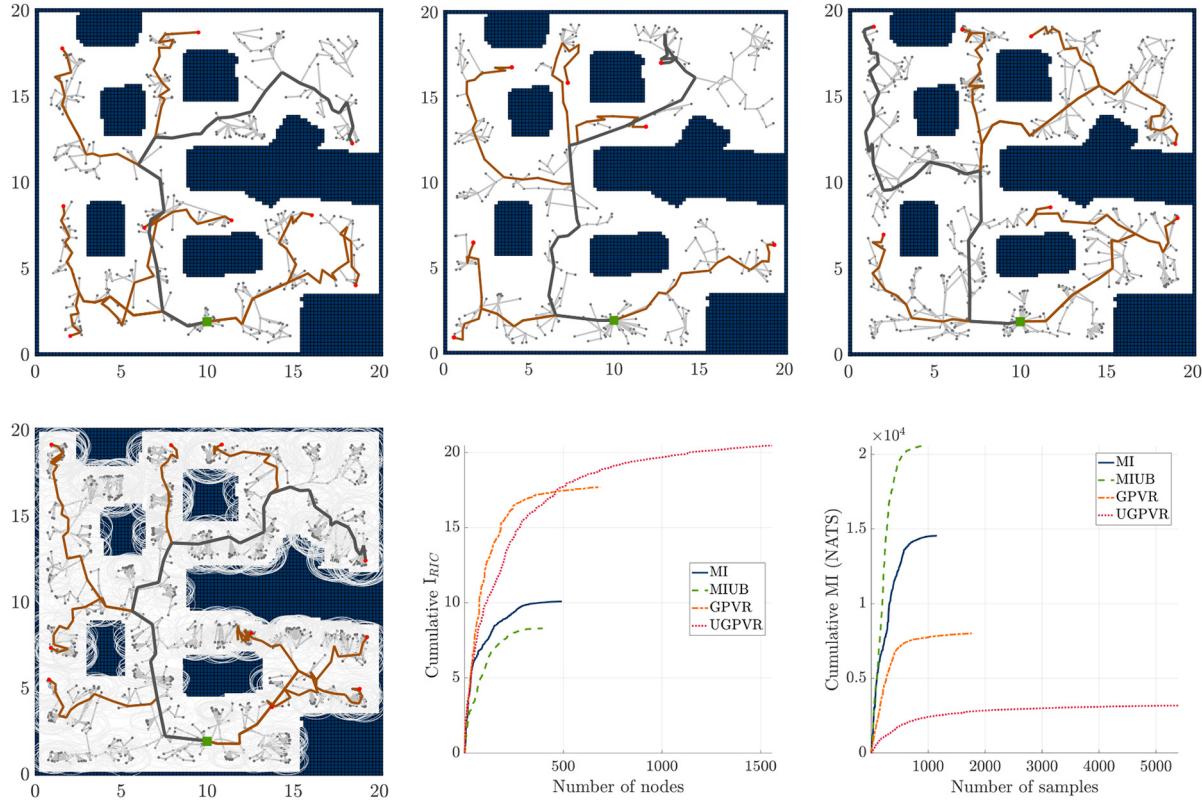
Figure 3 illustrates examples of IIG-tree results using different information functions in the Cave map. The sensor range and the number of beams are set to 5m and 10, respectively. The extracted paths are also shown and the most informative path is separated using a darker color. The convergence of  $I_{RIC}$  for all information is shown in the bottom middle plot, and the bottom right plot shows the cumulative information gain demonstrating its diminishing return over time (as the number of samples grows). The higher value of  $I_{RIC}$  can be correlated with a denser IIG

**Table 2.** Comparison of the information functions in offline IIG-tree experiments by increasing the sensor number of beams from 10 to 50. The figures are averaged over 30 experiments (mean  $\pm$  standard error). For all experiments the following parameters are set in common  $r_{max} = 5\text{m}$ ,  $\delta_{RIC} = 5e - 4$ . The total information gain is reported in nats.

| Number of beams $n_z$                          | 10   | 20   | 50   |
|--|--|--|--|
| <b>Mutual information (MI)</b>                 |  |  |  |
| Planning time (min)                            | <b>6.88 <math>\pm</math> 0.16</b>            | 8.44 $\pm$ 0.27                              | 14.64 $\pm$ 0.81                             |
| Number of samples                              | 911 $\pm$ 19                                 | 588 $\pm$ 13                                 | <b>435 <math>\pm</math> 20</b>               |
| Number of nodes                                | 402 $\pm$ 5                                  | 280 $\pm$ 5                                  | <b>181 <math>\pm</math> 8</b>                |
| Total info. gain                               | <b><math>1.40 \times 10^4 \pm 148</math></b> | <b><math>1.36 \times 10^4 \pm 173</math></b> | $1.25 \times 10^4 \pm 313$                   |
| Total cost (m)                                 | 336.3 $\pm$ 3.8                              | 258.5 $\pm$ 3.7                              | <b>187.2 <math>\pm</math> 7.7</b>            |
| <b>Mutual information upper bound (MIUB)</b>   |  |  |  |
| Planning time (min)                            | <b>4.53 <math>\pm</math> 0.16</b>            | 6.95 $\pm$ 0.28                              | 12.88 $\pm$ 0.51                             |
| Number of samples                              | 1,014 $\pm$ 24                               | 629 $\pm$ 17                                 | <b>486 <math>\pm</math> 15</b>               |
| Number of nodes                                | 444 $\pm$ 8                                  | 303 $\pm$ 7                                  | <b>212 <math>\pm</math> 5</b>                |
| Total info. gain                               | <b><math>2.04 \times 10^4 \pm 169</math></b> | $1.96 \times 10^4 \pm 207$                   | $1.98 \times 10^4 \pm 403$                   |
| Total cost (m)                                 | 358.3 $\pm$ 5.6                              | 272.0 $\pm$ 5.0                              | <b>211.5 <math>\pm</math> 4.4</b>            |
| <b>GP variance reduction (GPVR)</b>            |  |  |  |
| Planning time (min)                            | <b>9.31 <math>\pm</math> 0.39</b>            | 13.10 $\pm$ 0.41                             | 15.07 $\pm$ 0.61                             |
| Number of samples                              | 1,677 $\pm$ 53                               | 1,135 $\pm$ 36                               | <b>874 <math>\pm</math> 36</b>               |
| Number of nodes                                | 616 $\pm$ 15                                 | 477 $\pm$ 13                                 | <b>382 <math>\pm</math> 11</b>               |
| Total info. gain                               | $7.84 \times 10^3 \pm 62$                    | $9.123 \times 10^3 \pm 80$                   | <b><math>1.31 \times 10^4 \pm 176</math></b> |
| Total cost (m)                                 | 490.5 $\pm$ 9.5                              | 392.4 $\pm$ 7.9                              | <b>320.6 <math>\pm</math> 7.2</b>            |
| <b>Uncertain GP variance reduction (UGPVR)</b> |  |  |  |
| Planning time (min)                            | <b>19.75 <math>\pm</math> 0.48</b>           | 27.89 $\pm$ 0.94                             | 47.79 $\pm$ 2.16                             |
| Number of samples                              | 4,746 $\pm$ 136                              | 2,633 $\pm$ 102                              | <b>1,828 <math>\pm</math> 81</b>             |
| Number of nodes                                | 1,344 $\pm$ 28                               | 913 $\pm$ 25                                 | <b>704 <math>\pm</math> 22</b>               |
| Total info. gain                               | $3.19 \times 10^3 \pm 16$                    | $3.73 \times 10^3 \pm 22$                    | <b><math>5.55 \times 10^3 \pm 52</math></b>  |
| Total cost (m)                                 | 894.0 $\pm$ 16.3                             | 621.4 $\pm$ 15.6                             | <b>497.8 <math>\pm</math> 12.5</b>           |

**Table 3.** Comparison of the information functions in offline IIG-tree experiments by increasing the sensor range,  $r_{max}$ , from 5 to 20m (averaged over 30 experiments, mean  $\pm$  standard error). For all experiments the following parameters are set in common  $n_z = 10$ ;  $\delta_{RIC} = 5e - 4$ . The total information gain is reported in nats.

| Range $r_{max}$ (m)                            | 5  | 10                                | 20   |
|--|--|-----------------------------------|--|
| <b>Mutual information (MI)</b>                 |  |                                   |  |
| Planning time (min)                            | <b>5.00 <math>\pm</math> 0.13</b>            | 6.60 $\pm$ 0.10                   | 7.88 $\pm$ 0.11                              |
| Number of samples                              | 979 $\pm$ 22                                 | 782 $\pm$ 21                      | <b>780 <math>\pm</math> 20</b>               |
| Number of nodes                                | 423 $\pm$ 7                                  | <b>352 <math>\pm</math> 8</b>     | 360 $\pm$ 6                                  |
| Total info. gain                               | $1.40 \times 10^4 \pm 146$                   | $1.45 \times 10^4 \pm 177$        | <b><math>1.57 \times 10^4 \pm 220</math></b> |
| Total cost (m)                                 | 351.7 $\pm$ 4.8                              | <b>307.8 <math>\pm</math> 5.4</b> | 309.2 $\pm$ 4.0                              |
| <b>Mutual information upper bound (MIUB)</b>   |  |                                   |  |
| Planning time (min)                            | 3.67 $\pm$ 0.10                              | <b>3.56 <math>\pm</math> 0.08</b> | 3.87 $\pm$ 0.11                              |
| Number of samples                              | 1015 $\pm$ 23                                | 813 $\pm$ 15                      | <b>811 <math>\pm</math> 20</b>               |
| Number of nodes                                | 444 $\pm$ 7                                  | <b>368 <math>\pm</math> 6</b>     | 369 $\pm$ 7                                  |
| Total info. gain                               | <b><math>2.04 \times 10^4 \pm 161</math></b> | $1.98 \times 10^4 \pm 228$        | $1.97 \times 10^4 \pm 272$                   |
| Total cost (m)                                 | 360.1 $\pm$ 5.2                              | <b>309.9 <math>\pm</math> 4.2</b> | 311.6 $\pm$ 4.5                              |
| <b>GP variance reduction (GPVR)</b>            |  |                                   |  |
| Planning time (min)                            | <b>6.20 <math>\pm</math> 0.22</b>            | 7.03 $\pm$ 0.24                   | 6.60 $\pm$ 0.22                              |
| Number of samples                              | 1,745 $\pm$ 71                               | 1,393 $\pm$ 41                    | <b>1,248 <math>\pm</math> 48</b>             |
| Number of nodes                                | 636 $\pm$ 18                                 | 559 $\pm$ 13                      | <b>523 <math>\pm</math> 16</b>               |
| Total info. gain                               | <b><math>7.97 \times 10^3 \pm 56</math></b>  | $7.80 \times 10^3 \pm 71$         | $7.83 \times 10^3 \pm 62$                    |
| Total cost (m)                                 | 501.7 $\pm$ 11.2                             | 445.0 $\pm$ 7.5                   | <b>422.7 <math>\pm</math> 9.8</b>            |
| <b>Uncertain GP variance reduction (UGPVR)</b> |  |                                   |  |
| Planning time (min)                            | <b>20.52 <math>\pm</math> 0.65</b>           | 26.3 $\pm$ 0.68                   | 26.68 $\pm$ 0.54                             |
| Number of samples                              | 4,994 $\pm$ 167                              | 3,661 $\pm$ 109                   | <b>3,385 <math>\pm</math> 87</b>             |
| Number of nodes                                | 1,404 $\pm$ 35                               | 1,139 $\pm$ 25                    | <b>1,078 <math>\pm</math> 19</b>             |
| Total info. gain                               | $3.17 \times 10^3 \pm 19$                    | $3.25 \times 10^3 \pm 21$         | <b><math>3.27 \times 10^3 \pm 21</math></b>  |
| Total cost (m)                                 | 934.4 $\pm$ 22.3                             | 765.2 $\pm$ 15.5                  | <b>724.1 <math>\pm</math> 12.3</b>           |



**Fig. 3.** An example of results from running the IIG-tree algorithm in the Cave map. From the top left, the plots show the IIG-tree graph for MI, MIUB, GPVR, and UGPVR MI information functions. The selected paths computed using Algorithm 6 are also shown where the most informative path is shown using a darker color. MI and UGPVR show similar behaviors despite different quantitative outcomes. The last two plots in the bottom row, from left, show the convergence graph of the cumulative penalized RIC  $I_{RIC}$  and the evolution of the cumulative information gain for all information functions, respectively. The sensor range and the number of beams are set to 5 m and 10, respectively.

graph. In addition, the MIUB essentially minimizes the entropy which tends to explore the boundary of the space, unlike MI that takes into account the perception field of the sensor. This behavior is also observed and discussed by Krause et al. (2008: Figure 4). The GPVR and UGPVR curves have similar trends; however, owing to the incorporated pose uncertainty the amount of information gain is remarkably lower, which explains the longer tail of information gain evolution before the convergence of the algorithm and the higher RIC from each node. In other words, farther nodes have higher pose uncertainties. Therefore, the discrimination between farther and nearer nodes from RIC is less than the case where the pose uncertainty is ignored.

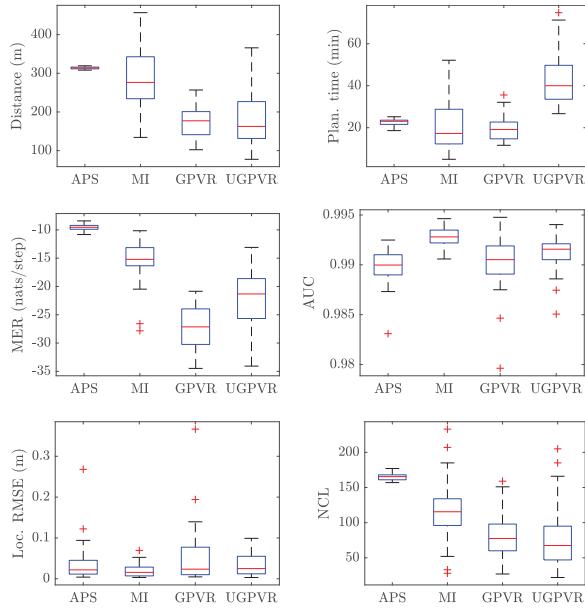
In the second experiment, the number of beams is kept fixed at  $n_z = 10$ , but the sensor range is increased from 5 to 20m. The results shown in Table 3 are consistent with the previous test. The variations are smaller as the growth in the size of the observation set (training set for GPVR-based functions), owing to increased range, is much smaller, mostly as a result of the geometry of the environment. Note that the first column of Table 3 and first column of Table 2 are for two different runs with same parameters. The differences in planning time are due to the fact that the experiments were carried out in a

high-performance computing facility where the reported time is affected by other users activities. Furthermore, the difference between  $r_{max} = 10$  and  $r_{max} = 20$  cases is marginal which shows increasing the sensor range more than 10m does not improve the information-gathering process in this particular environment.

Finally, a practical conclusion can be that increasing the sensor range and the number of beams (observations) increases the computational time but does not necessarily make the corresponding information function superior. As long as the approximation can capture the essence of information gain estimation consistently, the search algorithm performs well. Clearly the shape of the environment affects the result, e.g. the maximum and minimum perception field at different areas of the map. For the experiments in the following section, we select  $n_z = 10$  and  $r_{max} = 5$ m, which lead to the fastest computational time with reasonable total information gain and cost values.

### 9.3. Robotic exploration in unknown environments

In this section, we examine use of the proposed algorithms for autonomous robotic exploration in an unknown



**Fig. 4.** The box plots show the statistical summary of the comparison of different exploration strategies in the Cave dataset. The compared methods are APS, IIG-MI, IIG-GPVR, and IIG-UGPVR. The results are from 30 independent exploration rounds for each technique. The time comparison is based on the planning time. In general, APS is faster because the OGM is faster than I-GPOM; however, IIG-MI can also be implemented using OGMs. NCL stands for the number of closed loops.

environment (Figure 4). This requires solving the SLAM problem for localization possibly using observations to features in the environment. It is also necessary to build a dense map representation that shows occupied and unoccupied regions for planning and navigation. We solve the localization problem using Pose SLAM, the occupancy mapping using I-GPOM, and the planning using IIG-tree with different information functions. The I-GPOM models the occupancy map of the environment as GPs, which can be used by IIG to implement the proposed GPVR-based information functions.

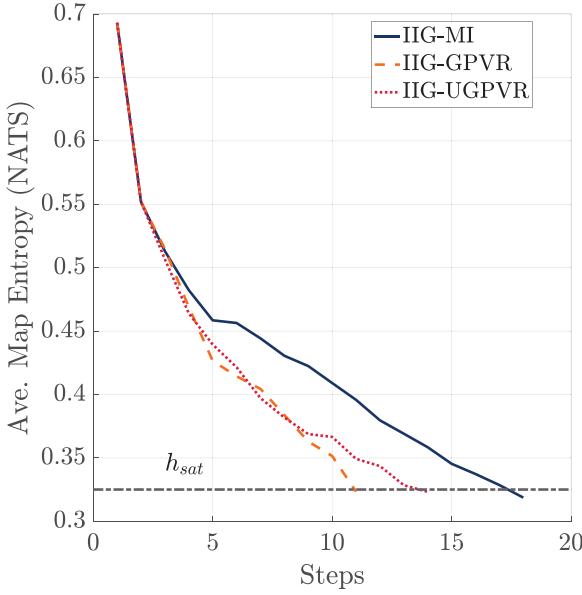
We also compare our results with Active Pose SLAM (APS) (Valencia et al., 2012), which is an information-gain-based technique that considers explicit loop closures by searching through nearby poses in the pose graph. As APS uses an occupancy grid map (OGM) (Elfes, 1987; Moravec and Elfes, 1985) for mapping and frontiers (Yamauchi, 1997) as exploration target candidates, we generate the equivalent I-GPOM using all poses and observations at the end of an experiment in order to compare mapping performance. Pose SLAM parameters were set and fixed regardless of the exploration method. The localization root mean square error (RMSE) was computed at the end of each experiment by the difference in the robot estimated and ground-truth poses. The occupancy maps are compared using the area under the receiver operating characteristic curve (AUC) as AUC is known to be more suitable for domains with skewed class distribution and unequal

classification error costs (Fawcett, 2006). The probability that the classifier ranks a randomly chosen positive instance higher than a randomly chosen negative instance can be understood using the AUC of the classifier. The AUC values of 0.5 and 1 correspond to random and ideal performance, respectively.

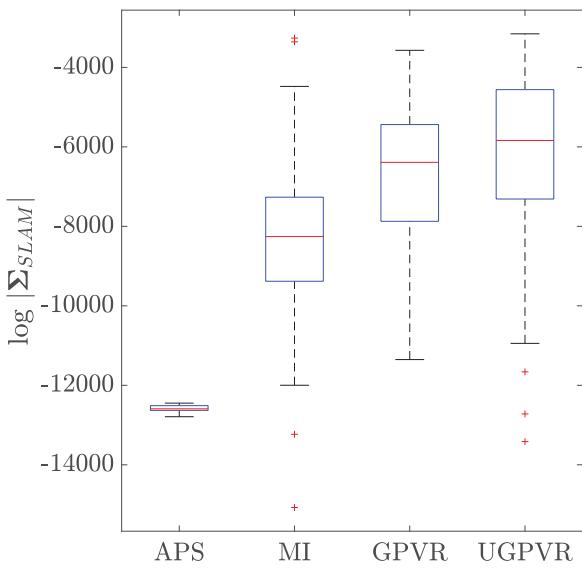
Figure 4 shows the statistical summary of the results from exploration experiments in the Cave environment. The results are from 30 exploration rounds in which each mission was terminated automatically using Corollary 14 and saturation probability  $p_{sat} = 0.1$  ( $h_{sat} = 0.3251$ nats). For APS, the mission was terminated when there was no cluster of geometric frontiers with the size larger than 12 cells; in addition, any frontier closer than 2m to the robot was ignored to speed up the map exploration. The IIG-GPVR has the lowest travel distance, highest map entropy reduction rate, but larger localization error. IIG-UGPVR demonstrates a more conservative version of IIG-GPVR where the pose uncertainty propagation provides better localization and mapping performance at the expense of more planning time. This behavior is expected and shows the consistency between our problem definition and algorithmic development. We note that our time comparison is focused on the planning time with mapping implemented using I-GPOM. Mapping with OGM will speed up both APS and IIG-MI.

IIG-MI has the lowest localization error with a similar planning time on average. IIG-MI makes direct use of sensor model for information gain estimation. The fact that the sensor model for range finders is an accurate model combined with the correlation between the map and robot pose leads to implicit pose uncertainty reduction. This result reveals the fundamental difference between MI approximation using a direct method (taking the expectation over future measurements) and GPs. However, it is important to note the ability of IIG using the GPVR-based functions to aggressively reduce the state estimation (map inference) uncertainty with fewer measurements (fewer loop closures) without a significant undesirable effect on the robot localization.

Figure 5 shows an example of the evolution of the average map entropy during the exploration mission. The robot terminated the exploration when the average map entropy dropped below the saturation entropy (Corollary 14). It is obvious that a technique with a faster map entropy reduction rate has achieved a faster convergence for the entire map exploration task. APS continues to explore until no frontier with “significant size” is left, which results in exhaustively searching many areas of the map multiple times. This results in a higher travel distance, more loop closures, and lower map entropy reduction rate (higher number of steps). In contrast, IIG stops collecting measurements (information) when there is sufficient information available for inference, i.e. task completion. Therefore, IIG leads to more efficient management of robotic exploration and information-gathering missions. Figure 6 further demonstrates this effect by plotting the statistical summary



**Fig. 5.** The evolution of the termination condition of different IIG exploration strategies in the Cave dataset. The curves are strongly related to the map entropy rate, i.e. a higher map entropy reduction rate results in faster termination of the mission. Unlike IIG, APS uses a geometric condition as the stopping criterion, and the mission was terminated when there was no cluster of geometric frontiers with the size larger than 12 cells.



**Fig. 6.** The statistical summary of the log of the determinant of the Pose SLAM covariance matrix at the end of each exploration run in the Cave dataset. The compared methods are, from the left, APS, IIG-MI, IIG-GPVR, and IIG-UGPVR. IIG algorithms can terminate the information-gathering process when there is sufficient information available for inference according to the requested saturation entropy (termination condition). Such an ability is not available in the APS framework because the robot has to complete the map based on its geometrical shape.

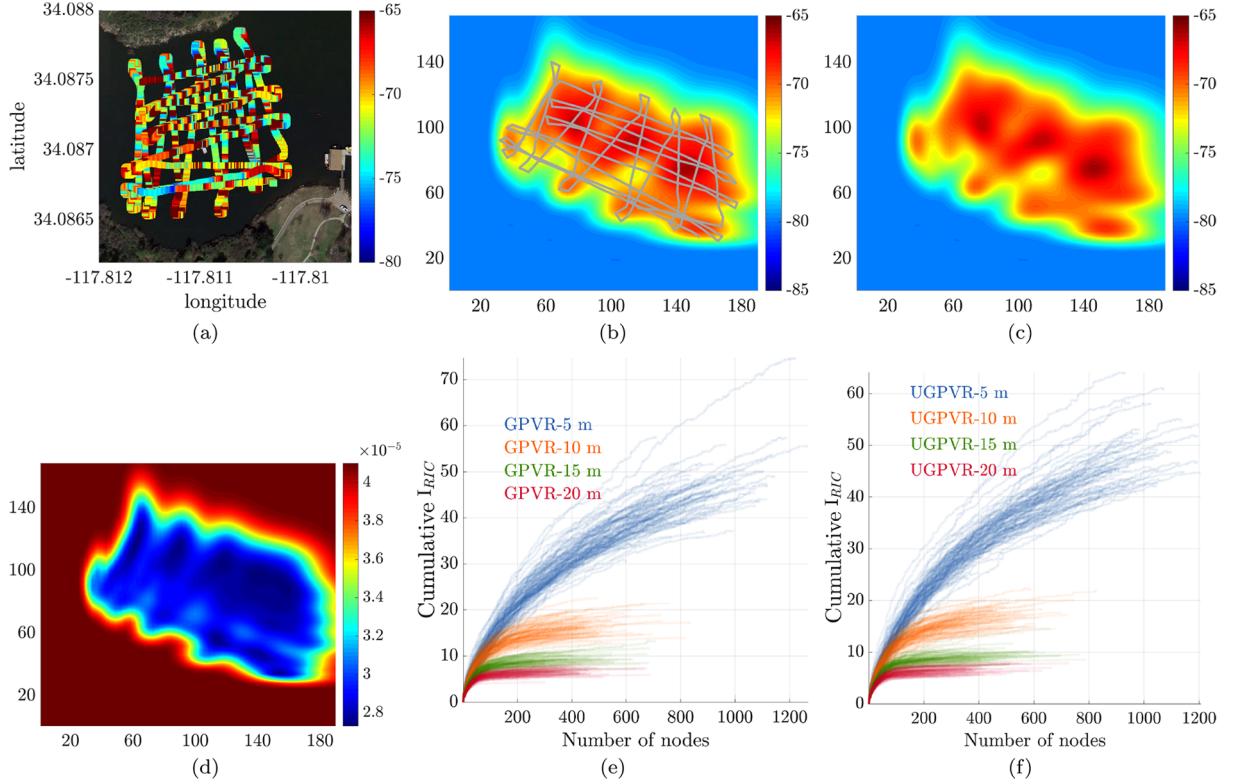
of the log of the determinant of the Pose SLAM covariance matrix for the entire robot trajectory at the end of each exploration run.

All the proposed techniques achieve a lower number of loop closures, while at the same time, on average, better localization and mapping performance as well as travel distance. APS explicitly searches for loop closures by performing data association using the current measurement (laser scan) and previous scans with nearby poses available in the pose graph. This is a particularly expensive process and often limits the search area to small regions, especially, when the SLAM graph becomes less sparse owing to the large number of loop closures. Finally, we would like to emphasize that IIG-UGPVR is a fully integrated information-theoretic planner that takes full state estimate (the robot pose and map) uncertainty into account, it offers an automatic termination condition for the entire mission, has an adjustable planning horizon,  $\delta_{RIC}$ , and treats future measurements as random by taking expectations over them. In contrast, APS simplifies the information gain computation by resorting to a conditional independence assumption between the robot pose and the map. The true strength of APS for achieving comparable outcomes is through shifting the computational load towards the data association for predicting informative loop closures along a planned trajectory. Therefore, the work presented in this paper is an alternative that is fundamentally different. Furthermore, the termination condition proposed in this article is general and can be easily incorporated into APS or other exploration techniques. For examples of qualitative results including Pose SLAM, I-GPOM, and OGM plots, see Ghaffari Jadidi et al. (2016) and Ghaffari Jadidi (2017).

#### 9.4. Lake-monitoring experiment

In this experiment, we demonstrate the performance of IIGs in a lake-monitoring scenario. The ASV can localize using a GPS unit and a Doppler velocity log. The communication with the ground station is through a 802.11 wireless connection and at any location the wireless signal strength (WSS) can be measured in dBm. The Puddingstone Lake dataset is a publicly available dataset that includes about 2,700 observations. The dataset is collected through a full survey of the lake area located at Puddingstone Lake in San Dimas, CA (latitude 34.088854, longitude -117.810667) (Hollinger and Sukhatme, 2014). The objective is to find a trajectory for the robot to maintain a strong connectivity with the base station while taking physical samples in the lake.

Figure 7 shows the satellite view of the lake area together with the survey trajectories and regressed maps that are used as a proxy for ground truth. Figure 7b shows the survey trajectories on the WSS surface where the longitudes and latitudes are converted into their corresponding distances using the haversine formula. Figures 7c and 7d



**Fig. 7.** Lake-monitoring scenario: (a) satellite view of the lake and the survey trajectories using ASV; (b) the robot trajectories in metric scale on WSS map; (c) GP WSS mean surface (in dBm); and (d) GP WSS covariance surface. The convergence results of IIG-GPVR and IIG-UGPVR using 5, 10, 15, and 20m sensing ranges are shown in (e) and (f), respectively. As the sensing range increases, the planner converges faster. The results are generated using  $\delta_{RIC} = 5 \times 10^{-4}$  and an unlimited budget (travel distance).

illustrate the GP WSS mean and covariance surfaces, respectively. The ground-truth map of WSS is built using GP regression with a constant mean function, SE covariance function with automatic relevance determination (Neal, 1996), and a Gaussian likelihood function which makes the exact inference possible. Furthermore, it is well-known that, in line-of-sight scenarios, radio signals propagation can be characterized based on the Friis free-space model (Goldsmith, 2005; Rappaport, 1996). In this model, the signal attenuation is proportional to the logarithm of the distance. Therefore, to improve the regression accuracy we use logarithmic scales for input points during GP training and inference phases. The number of training points was down-sampled to 267 observations and the surface was inferred using 3,648 query points.

We store the GP output using a *kd*-tree data structure to be able to perform fast online nearest-neighbor inquiries within a sensing range at any location in the map. In order to simulate the WSS monitoring experiment, at any location the robot can take measurements within a sensing range from the ground-truth GP maps using *Near* function. This step is the measurement prediction in line 7 of Algorithms 5 and UGPVR and the training set,  $\mathcal{D}$  is the output of *Near* function. As training points are already part of the map, the queried submap coincides with the training set, i.e.  $\mathcal{D} = \mathcal{M}_{\mathcal{D}}$ . For

information functions, we used Algorithms 5 and UGPVR as they are natural choices for scenarios involving spatial phenomena and environmental monitoring. The modified kernel in the UGPVR algorithm was calculated using Gauss-Hermite quadrature with 11 sample points. The robot pose covariance was approximated using the uncertainty propagation and local linearization of the robot motion model.<sup>12</sup>

Each technique uses unlimited budget which is the total travel distance and continues to search the environment until convergence. Then the WSS field is reconstructed with the same resolution as the ground-truth map using collected measurements along the most informative path, which is extracted from the IIG-tree using Algorithm 6. The convergence results with 5, 10, 15, and 20m sensing ranges are shown in Figures 7e and 7f for 100 independent runs. Table 4 shows the comparison results between IIG-GPVR and IIG-UGPVR using several criteria including RMSE. The IIG-GPVR algorithm is faster and can compute the most informative trajectory in only a few seconds. This is promising for online applications where the robot needs to replan along the trajectory. We note that, as expected from results in Subsection 9.2, by increasing the sensing radius, the information gain rises and the total cost and RMSE decrease. However, in practice, a large sensing radius can be infeasible.

**Table 4.** Lake-monitoring experiments using IIG with GPVR and UGPVR information functions. For the comparison, the sensing range is varied from 5 to 20 m. The results are averaged over 100 runs (mean  $\pm$  standard error). For all experiments  $\delta_{RIC} = 5e - 4$ .

| Range (m)         | 5                          | 10                         | 15                         | 20                         |
|-------------------|----------------------------|----------------------------|----------------------------|----------------------------|
| <b>IIG-GPVR</b>   |                            |                            |                            |                            |
| Time (s)          | $5.26 \pm 0.16$            | $2.92 \pm 0.08$            | $2.38 \pm 0.09$            | $2.32 \pm 0.07$            |
| RMSE (dBm)        | $4.54 \pm 0.05$            | $4.28 \pm 0.06$            | $3.97 \pm 0.06$            | $3.61 \pm 0.07$            |
| Number of samples | $830.6 \pm 18.7$           | $507.9 \pm 13.0$           | $437.0 \pm 15.6$           | $428.1 \pm 12.9$           |
| Number of nodes   | $826.1 \pm 18.7$           | $454.5 \pm 12.8$           | $359.9 \pm 14.5$           | $331.6 \pm 12.7$           |
| Total info. gain  | $2.08 \times 10^4 \pm 160$ | $2.42 \times 10^4 \pm 111$ | $2.62 \times 10^4 \pm 133$ | $2.77 \times 10^4 \pm 130$ |
| Total cost (km)   | $6.77 \pm 0.11$            | $4.50 \pm 0.09$            | $3.67 \pm 0.11$            | $3.42 \pm 0.10$            |
| <b>IIG-UGPVR</b>  |                            |                            |                            |                            |
| Time (s)          | $33.98 \pm 1.30$           | $25.93 \pm 0.67$           | $29.26 \pm 0.98$           | $53.69 \pm 1.61$           |
| RMSE (dBm)        | $4.55 \pm 0.05$            | $4.26 \pm 0.05$            | $4.00 \pm 0.06$            | $3.56 \pm 0.07$            |
| Number of samples | $844.8 \pm 19.8$           | $496.0 \pm 12.3$           | $467.4 \pm 15.4$           | $413.0 \pm 13.3$           |
| Number of nodes   | $839.2 \pm 19.7$           | $447.9 \pm 12.4$           | $389.1 \pm 14.9$           | $325.8 \pm 12.8$           |
| Total info. gain  | $2.12 \times 10^4 \pm 176$ | $2.49 \times 10^4 \pm 107$ | $2.69 \times 10^4 \pm 120$ | $2.80 \times 10^4 \pm 113$ |
| Total cost (km)   | $6.81 \pm 0.13$            | $4.46 \pm 0.09$            | $3.91 \pm 0.11$            | $3.39 \pm 0.10$            |

Figure 8 shows the illustrative examples of IIG-GPVR with different sensing ranges. As the sensing range increases, the graph becomes gradually sparser. This can be understood from the fact that the information from a larger neighborhood is integrated into each node. Note that the objective is to explore the entire area while maintaining a strong wireless connectivity with the based station, therefore, the robot does not need to always travel in the area with high signal strengths. However, it travels most of the time in the areas with strong connectivity. Figure 9 shows the same scenario using IIG-UGPVR. The main difference is that the robot behaves more conservatively and tends to return to the regions with strong connectivity. The growth in pose uncertainty by traveling farther, reduces the information gain of those areas. In both IIG-GPVR and IIG-UGPVR results, we can see that by increasing the sensing range the robot tends to explore farther distances which is a natural behavior, in Figures 8d and 9d.

### 9.5. Limitations and observations

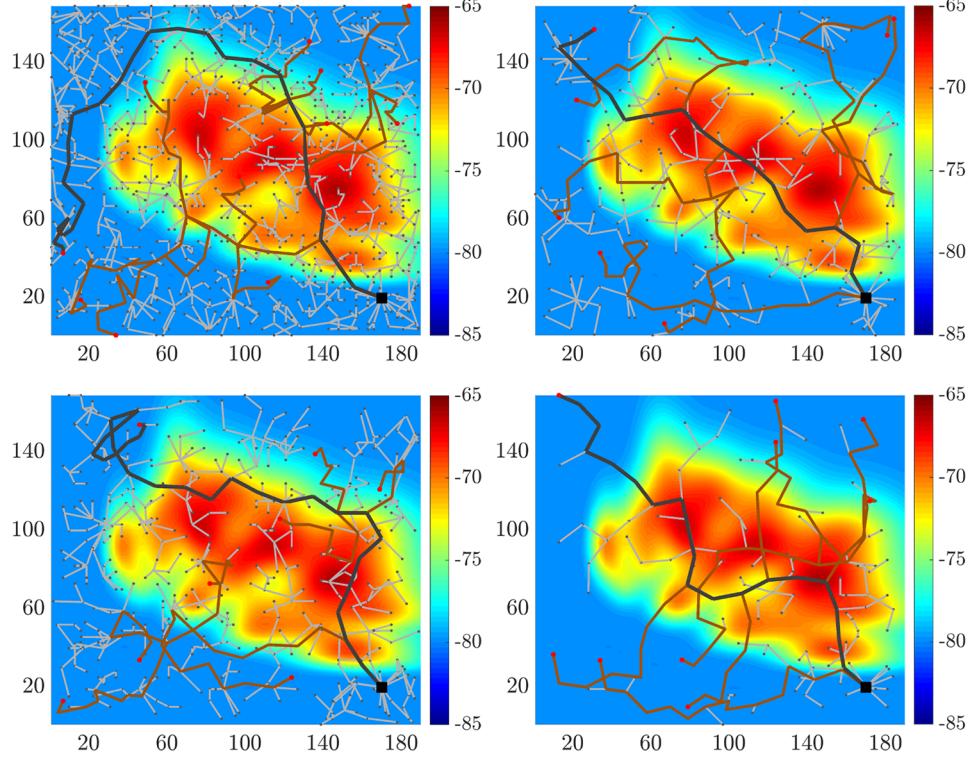
The proposed algorithms can provide an approximate solution to the robotic exploration problem in unknown environments as a basic navigation task as well as environmental monitoring tasks. Although IIG can be implemented for online tasks, it has the limitations of its ancestors. Therefore, violating the main assumptions, such as availability of free area near any point for graph expansion, can result in failure of the task. More conceptually, most of the present robotic navigation algorithms are not truly adaptive as they commit to a decision once it is planned and replanning occurs only if the amount of computation is manageable, and often not with a high frequency. So there will be a window of time when the robot acts based on the previously made decision. Although humans act in a similar fashion, we can make decisions more spontaneously (fully adaptive). This problem can be severe in environments that are highly dynamic as the robot is not as responsive as it is required.

To our experience, using a coarse approximation, i.e. fewer beams than the actual range-finder sensor, results in not only faster search in the workspace and reducing the computational time, but giving the chance to more samples to be candidates as part of the potential most informative path. This is a promising feature that the algorithm works reasonably well without near exact estimation of information quality. Moreover, uniform sampling produces reasonable results in the early stage, however as the graph grows, this sampling strategy becomes less efficient. Biasing the sampling towards directions that are less explored may lead to a faster convergence.

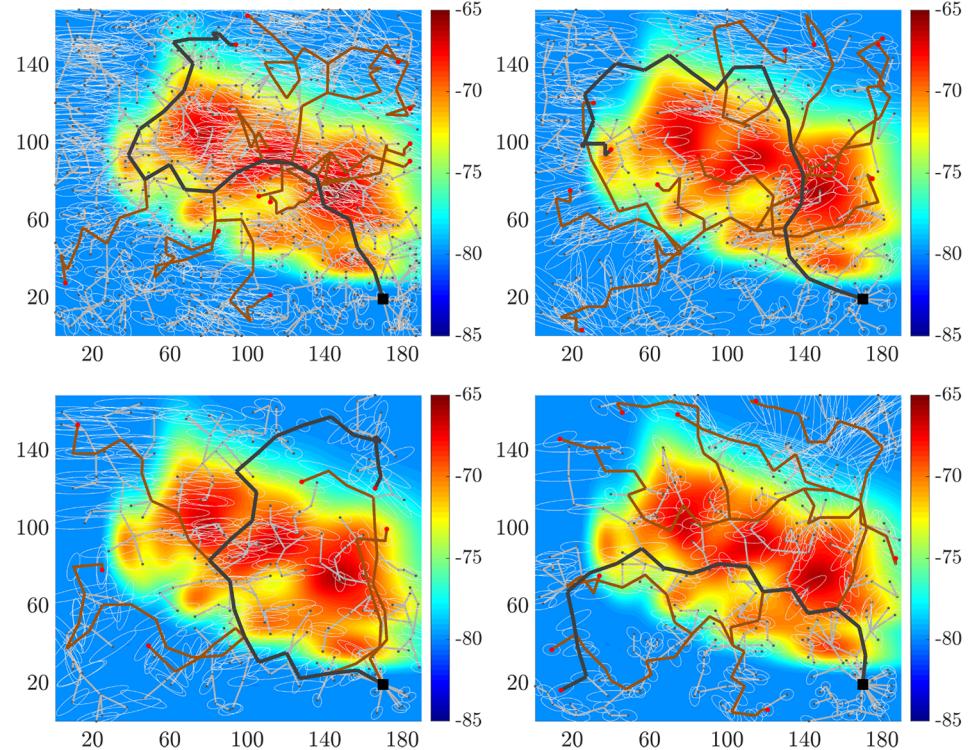
## 10. Conclusion and future work

In this work, we have developed a sampling-based planning algorithm for incremental robotic information gathering. The proposed algorithm is based on RIG and offers an information-theoretic convergence criterion using the notion of penalized RIC. We formulated the problem as an information maximization problem subject to budget and state estimate constraints. The unique feature of this approach is that it can handle dense belief representations as well as the robot pose uncertainty. We proposed MI-based and GPVR-based information functions that can be used in both RIG and IIG frameworks. The proposed IIG-tree algorithm using a UGPVR information function considers pose uncertainties in planning and provides an approximate solution for finding maximally informative paths under partially observable state variables.

We proved an information-theoretic termination condition for the entire information-gathering mission based on the least upper bound of the average state variables entropy. The proposed algorithms can directly be applied to environmental monitoring and robotic exploration tasks in unknown environments. The proposed algorithms also have potential applications in solving the sensor configuration



**Fig. 8.** Informative motion planning for WSS monitoring in the lake area using IIG-GPVR with sensing range: (a) 5 m; (b) 10 m; (c) 15 m; and (d) 20 m.



**Fig. 9.** Informative motion planning for WSS monitoring in the lake area using IIG-UGPVR with sensing range: (a) 5 m; (b) 10 m; (c) 15 m; and (d) 20 m. By incorporating pose uncertainties in planning, the informative trajectories are those in which not only the wireless connectivity is strong, but the pose uncertainty along the trajectories is minimized.

selection problem that we did not fully investigate as it is beyond the scope of this work. Furthermore, the planning horizon does not need to be set as a number of steps. Instead, the information-theoretic representation of the planning horizon used here brings more flexibility for managing the convergence of the algorithm.

We would like to share some ideas that are natural extensions of this work. For the sake of clarity, we itemize them as follows.

- (i) The algorithms in this article could be extended for multi-robot planning; exploiting the information gain available through wireless communication channels to manage the coordination between robots while a feedback controller satisfied kinodynamic constraints.
- (ii) We developed IIG-tree based on the RIG-tree algorithm. IIG could also be used to extend RIG-roadmap and RIG-graph. RIG-graph is asymptotically optimal and can asymptotically explore all possible budget-constraint trajectories, it produces a fully connected graph and is an interesting case to study.
- (iii) Whereas we provide a procedural implementation of the proposed algorithms, and it suffices for research and comparison purposes, we believe the integration of the algorithms in open-source libraries such as the Open Motion Planning Library (OMPL) (Sucan et al., 2012) has advantages for possible applications of this work.
- (iv) The GPVR-based algorithms could be used for active learning to model the quantity of interest online using GPs.
- (v) IIG/RIG frameworks could be used for maximizing a multi-objective information function to perform multiple tasks concurrently. In a naive approach, the objective function can be defined as the sum of information functions; however, if random variables from different information functions are correlated, e.g. through the robot pose, the total information gain calculation will not be accurate. Hence, developing a more systematic approach is an interesting avenue to follow.
- (vi) A sparse covariance matrix could be constructed using the Sparse covariance function. Exploiting the sparse structure of the covariance matrix for large-scale problems could be an interesting extension of this work for online implementations.
- (vii) Finally, we solved the decision-making step using a heuristic algorithm based on a voting method (Algorithm 6). The study of this path-selection algorithm and its effect on the overall performance is also an interesting future work.

## Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

## ORCID iD

Maani Ghaffari Jadidi  <https://orcid.org/0000-0002-4734-4295>

## Notes

1. Being anytime is typically a good feature, but we are interested in knowing when the algorithm converges.
2. Results extend naturally to any absolutely continuous distribution with density bounded away from zero on workspace  $\mathcal{X}$  (Karaman and Frazzoli, 2011).
3. Here we use Euclidean distance.
4. Through this function, it is possible to make the planner kinodynamic.
5. Note that more than one optimal trajectory at each time can exist, e.g. when the robot needs to explore two equally important directions.
6. Although this is one of the most interesting aspects of GPVR-based information functions, it is beyond the scope of this article, and we leave it as a possible extension of this work.
7. The algorithm uses MATLAB-style operations for matrix inversion, Cholesky factorization, and dot product with matrix inputs.
8. Note that the constant factor  $\frac{1}{2}$  is removed because it does not have any effect in this context.
9. The positive semidefinite property of the covariance matrix needs to be preserved.
10. Here we assume the robot remains committed to the selected action, i.e. there is no replanning while executing an action.
11. The dataset is available at <http://research.engr.oregonstate.edu/rdml/software>
12. The pose uncertainties are not part of the original dataset. Hence, we calculate them by simulation.

## References

- Agha-mohammadi AA, Chakravorty S and Amato NM (2014) FIRM: Sampling-based feedback motion-planning under motion uncertainty and imperfect measurements. *The International Journal of Robotics Research* 33(2): 268–304.
- Astrom K (1965) Optimal control of Markov processes with incomplete state information. *Journal of Mathematical Analysis and Applications* 10(1): 174–205.
- Atanasov N, Le Ny J, Daniilidis K and Pappas GJ (2014) Information acquisition with sensing robots: Algorithms and error bounds. In: *Proceedings IEEE International Conference on Robotics and Automation*. IEEE, pp. 6447–6454.
- Atanasov N, Le Ny J, Daniilidis K and Pappas GJ (2015) Decentralized active information acquisition: Theory and application to multi-robot SLAM. In: *Proceedings IEEE International Conference on Robotics and Automation*. IEEE, pp. 4775–4782.
- Atanasov NA (2015) Active information acquisition with mobile robots. PhD Thesis, University of Pennsylvania.
- Bertsekas DP (1995) *Dynamic programming and optimal control*, volume 1. Belmont, MA: Athena Scientific.
- Binney J, Krause A and Sukhatme GS (2013) Optimizing waypoints for monitoring spatiotemporal phenomena. *The International Journal of Robotics Research* 32(8): 873–888.
- Binney J and Sukhatme GS (2012) Branch and bound for informative path planning. In: *Proceedings IEEE International Conference on Robotics and Automation*. IEEE, pp. 2147–2154.

- Bry A and Roy N (2011) Rapidly-exploring random belief trees for motion planning under uncertainty. In: *Proceedings IEEE International Conference on Robotics and Automation*. IEEE, pp. 723–730.
- Charrow B (2015) Information-theoretic active perception for multi-robot teams. PhD Thesis, University of Pennsylvania.
- Charrow B, Kahn G, Patil S, et al. (2015a) Information-theoretic planning with trajectory optimization for dense 3D mapping. In: *Robotics: Science and Systems*.
- Charrow B, Liu S, Kumar V and Michael N (2015b) Information-theoretic mapping using Cauchy–Schwarz quadratic mutual information. In: *Proceedings IEEE International Conference on Robotics and Automation*. IEEE, pp. 4791–4798.
- Charrow B, Michael N and Kumar V (2014) Cooperative multi-robot estimation and control for radio source localization. *The International Journal of Robotics Research* 33(4): 569–580.
- Cover TM and Thomas JA (1991) *Elements of Information Theory*. New York: John Wiley & Sons.
- Davis PJ and Rabinowitz P (1984) *Methods of Numerical Integration*. New York: Academic Press.
- Dhariwal A, Sukhatme GS and Requicha AA (2004) Bacterium-inspired robots for environmental monitoring. In: *Proceedings IEEE International Conference on Robotics and Automation*, Vol. 2. IEEE, pp. 1436–1443.
- Dunbabin M and Marques L (2012) Robots for environmental monitoring: Significant advancements and applications. *IEEE Robotics and Automation Magazine* 19(1): 24–39.
- Elfes A (1987) Sonar-based real-world mapping and navigation. *IEEE Journal of Robotics and Automation* 3(3): 249–265.
- Fawcett T (2006) An introduction to ROC analysis. *Pattern Recognition Letters* 27(8): 861–874.
- Ghaffari Jadidi M (2017) Gaussian processes for information-theoretic robotic mapping and exploration. PhD Thesis, University of Technology, Sydney.
- Ghaffari Jadidi M, Valls Miro J and Dissanayake G (2015) Mutual information-based exploration on continuous occupancy maps. In: *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 6086–6092.
- Ghaffari Jadidi M, Valls Miro J and Dissanayake G (2016) Gaussian process autonomous mapping and exploration for range sensing mobile robots. *Autonomous Robots* 42(2): 273–290.
- Ghaffari Jadidi M, Valls Miro J and Dissanayake G (2017) Warped Gaussian processes occupancy mapping with uncertain inputs. *IEEE Robotics and Automation Letters* 2(2): 680–687.
- Ghaffari Jadidi M, Valls Miro J, Valencia R and Andrade-Cetto J (2014) Exploration on continuous Gaussian process frontier maps. In: *Proceedings IEEE International Conference on Robotics and Automation*, pp. 6077–6082.
- Goldsmith A (2005) *Wireless Communications*. Cambridge: Cambridge University Press.
- Guestrin C, Krause A and Singh AP (2005) Near-optimal sensor placements in Gaussian processes. In: *Proceedings of the 22nd International Conference on Machine Learning*. New York: ACM Press, pp. 265–272.
- Hollinger GA (2015) Long-horizon robotic search and classification using sampling-based motion planning. In: *Robotics: Science and Systems*.
- Hollinger GA and Sukhatme GS (2013) Sampling-based motion planning for robotic information gathering. In: *Robotics: Science and Systems*.
- Hollinger GA and Sukhatme GS (2014) Sampling-based robotic information gathering algorithms. *The International Journal of Robotics Research* 33(9): 1271–1287.
- Horsch T, Schwarz F and Tolle H (1994) Motion planning with many degrees of freedom-random reflections at c-space obstacles. In: *Proceedings IEEE International Conference on Robotics and Automation*. IEEE, pp. 3318–3323.
- Howard A and Roy N (2003) The robotics data set repository (Radish). URL <http://radish.sourceforge.net>.
- Huynh VA and Roy N (2009) icLQG: combining local and global optimization for control in information space. In: *Proceedings IEEE International Conference on Robotics and Automation*. IEEE, pp. 2851–2858.
- Ila V, Porta J and Andrade-Cetto J (2010) Information-based compact Pose SLAM. *IEEE Transactions on Robotics* 26(1): 78–93.
- Indelman V, Carlone L and Dellaert F (2015) Planning in the continuous domain: A generalized belief space approach for autonomous navigation in unknown environments. *The International Journal of Robotics Research* 34(7): 849–882.
- Kaelbling LP, Littman ML and Cassandra AR (1998) Planning and acting in partially observable stochastic domains. *Artificial Intelligence* 101(1): 99–134.
- Karaman S and Frazzoli E (2011) Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research* 30(7): 846–894.
- Kavraki LE, Švestka P, Latombe JC and Overmars MH (1996) Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *Robotics and Automation, IEEE Transactions on* 12(4): 566–580.
- Kim A and Eustice RM (2015) Active visual SLAM for robotic area coverage: Theory and experiment. *The International Journal of Robotics Research* 34(4–5): 457–475.
- Kim S and Kim J (2015) GPmap: A unified framework for robotic mapping based on sparse Gaussian processes. In: *Field and Service Robotics*. New York: Springer, pp. 319–332.
- Krause A, Singh A and Guestrin C (2008) Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *The Journal of Machine Learning Research* 9: 235–284.
- Kurniawati H, Du Y, Hsu D and Lee WS (2011) Motion planning under uncertainty for robotic tasks with long time horizons. *The International Journal of Robotics Research* 30(3): 308–323.
- Kurniawati H, Hsu D and Lee WS (2008) SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces. In: *Robotics: Science and Systems*.
- Lan X and Schwager M (2013) Planning periodic persistent monitoring trajectories for sensing robots in Gaussian random fields. In: *Proceedings IEEE International Conference on Robotics and Automation*. IEEE, pp. 2415–2420.
- Latombe JC (1991) *Robot Motion Planning (The Springer International Series in Engineering and Computer Science*, Vol. 124). New York: Springer.
- LaValle SM (2006) *Planning algorithms*. Cambridge: Cambridge University Press.
- LaValle SM and Kuffner JJ (2001) Randomized kinodynamic planning. *The International Journal of Robotics Research* 20(5): 378–400.
- Le Ny J and Pappas GJ (2009) On trajectory optimization for active sensing in Gaussian process models. In: *Proceedings*

- IEEE Conference on Decision and Control.* IEEE, pp. 6286–6292.
- Leung C, Huang S and Dissanayake G (2006a) Active SLAM using model predictive control and attractor based exploration. In: *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems.* IEEE, pp. 5026–5031.
- Leung C, Huang S, Kwok N and Dissanayake G (2006b) Planning under uncertainty using model predictive control for information gathering. *Robotics and Autonomous Systems* 54(11): 898–910.
- Levine DS (2010) Information-rich path planning under general constraints using rapidly-exploring random trees. Master's Thesis, Massachusetts Institute of Technology.
- Marchant R and Ramos F (2012) Bayesian optimisation for intelligent environmental monitoring. In: *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems.* IEEE, pp. 2242–2249.
- Melkumyan A and Ramos F (2009) A sparse covariance function for exact Gaussian process inference in large datasets. In: *Proceedings of IJCAI*, Vol. 9, pp. 1936–1942.
- Moravec HP and Elfes A (1985) High resolution maps from wide angle sonar. In: *Proceedings IEEE International Conference on Robotics and Automation*, Vol. 2. IEEE, pp. 116–121.
- Neal RM (1996) *Bayesian learning for neural networks*, volume 118. Springer New York.
- Platt R Jr, Tedrake R, Kaelbling L and Lozano-Perez T (2010) Belief space planning assuming maximum likelihood observations. In: *Robotics: Science and Systems*.
- Prentice S and Roy N (2009) The belief roadmap: Efficient planning in belief space by factoring the covariance. *The International Journal of Robotics Research* 28(11–12): 1448–1465.
- Press WH, Teukolsky SA, Vetterling WT and Flannery BP (1996) *Numerical recipes in C*, Vol. 2. Cambridge: Cambridge University Press.
- Pukelsheim F (2006) *Optimal Design of Experiments*. Philadelphia, PA: Society for Industrial and Applied Mathematics.
- Rappaport TS (1996) *Wireless Communications: Principles and Practice*, Vol. 2. Englewood Cliffs, NJ: Prentice Hall.
- Rasmussen C and Williams C (2006) *Gaussian Processes for Machine Learning*, Vol. 1. Cambridge, MA: MIT Press.
- Singh A, Krause A, Guestrin C and Kaiser WJ (2009) Efficient informative sensing using multiple robots. *Journal of Artificial Intelligence Research* 34: 707–755.
- Singh A, Ramos F, Whyte HD and Kaiser WJ (2010) Modeling and decision making in spatio-temporal processes for environmental surveillance. In: *Proceedings IEEE International Conference on Robotics and Automation.* IEEE, pp. 5490–5497.
- Smallwood RD and Sondik EJ (1973) The optimal control of partially observable Markov processes over a finite horizon. *Operations Research* 21(5): 1071–1088.
- Stein M (1999) *Interpolation of Spatial Data: Some Theory for Kriging*. Berlin: Springer Verlag.
- Ström DP, Nencini F and Stachniss C (2015) Predictive exploration considering previously mapped environments. In: *Proceedings IEEE International Conference on Robotics and Automation.* IEEE, pp. 2761–2766.
- Sucan IA, Moll M and Kavraki LE (2012) The open motion planning library. *IEEE Robotics and Automation Magazine* 19(4): 72–82.
- Thrun S, Burgard W and Fox D (2005) *Probabilistic Robotics*, Vol. 1. Cambridge, MA: MIT Press.
- Tresp V (2000) A Bayesian committee machine. *Neural Computation* 12(11): 2719–2741.
- Valencia R, Morta M, Andrade-Cetto J and Porta JM (2013) Planning reliable paths with Pose SLAM. *IEEE Transactions on Robotics* 29(4): 1050–1059.
- Valencia R, Valls Miro J, Dissanayake G and Andrade-Cetto J (2012) Active Pose SLAM. In: *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1885–1891.
- Van Den Berg J, Abbeel P and Goldberg K (2011) LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information. *The International Journal of Robotics Research* 30(7): 895–913.
- Von Mises R (1964) *Mathematical Theory of Probability and Statistics*. New York: Academic Press.
- Yamauchi B (1997) A frontier-based approach for autonomous exploration. In: *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation* pp. 146–151.
- Yu J, Karaman S and Rus D (2015) Persistent monitoring of events with stochastic arrivals at multiple stations. *IEEE Transactions on Robotics* 31(3): 521–535.

## Appendix A. Proof of MI approximation inequality

In this appendix, we provide the proof of Proposition 10. First, we present the required preliminaries and then our proof by contradiction.

**Lemma 3** (Hadamard's inequality). Let  $A = [a_1 \ a_2 \ \dots \ a_n]$  be an  $n \times n$  matrix such that  $a_i \in \mathbb{R}^n$  for all  $i \in \{1 : n\}$ . Hadamard's inequality asserts that

$$|A| \leq \prod_{i=1}^n \|a_i\| \quad (27)$$

**Corollary 15.** Let  $S \in \mathbb{R}^{n \times n}$  be a positive-definite matrix whose diagonal entries are  $s_{11}, \dots, s_{nn}$ . From Lemma 3, it follows that

$$|S| \leq \prod_{i=1}^n s_{ii} \quad (28)$$

*Proof.* Since  $S$  is positive definite, using Cholesky factorization, we have  $S = LL^T$  and  $|S| = |LL^T| = |L||L^T| = |L|^2$ . For any diagonal elements of  $S$ , we have  $s_{ii} = l_i l_i^T = \|l_i\|^2$  where  $l_i$  is the  $i$ th column of  $L$ . Using the Hadamard's inequality, it follows that

$$|S| = |L|^2 \leq \prod_{i=1}^n \|l_i\|^2 = \prod_{i=1}^n s_{ii} \quad (29)$$

□

We now derive a few useful inequalities before the final proof. Let  $X \sim \mathcal{N}(\mu_X, \Sigma_X)$  and  $X|Z \sim \mathcal{N}(\mu_{X|Z}, \Sigma_{X|Z})$  be the prior and posterior distribution of the  $n$ -dimensional random vector  $X$ , respectively. Let  $h(X) = \frac{1}{2}\log((2\pi e)^n |\Sigma_X|)$

and  $h(X|Z) = \frac{1}{2} \log((2\pi e)^n |\Sigma_{X|Z}|)$  be the differential entropy and conditional entropy, respectively; and  $\hat{I}(X; Z) = \hat{h}(X) - \hat{h}(X|Z)$  the corresponding entropies and MI by applying Proposition 9, where  $\hat{h}(X) = \frac{1}{2} \log((2\pi e)^n \prod_{i=1}^n \sigma_{X_i})$  and  $\hat{h}(X|Z) = \frac{1}{2} \log((2\pi e)^n \prod_{i=1}^n \sigma_{X_i|Z})$ . Using Corollary 15, the following inequalities are immediate:

$$h(X) \leq \hat{h}(X) \quad (30)$$

$$h(X|Z) \leq \hat{h}(X|Z) \quad (31)$$

$$I(X; Z) \leq \hat{h}(X) - h(X|Z) \quad (32)$$

$$h(X) - \hat{h}(X|Z) \leq \hat{I}(X; Z) \quad (33)$$

$$h(X) - \hat{h}(X|Z) \leq \hat{h}(X) - h(X|Z) \quad (34)$$

and summing (32) and (33) leads to

$$I(X; Z) \leq \hat{I}(X; Z) + \xi(X; Z) \quad (35)$$

where  $\xi(X; Z) \triangleq \hat{h}(X) + \hat{h}(X|Z) - h(X) - h(X|Z)$  and  $\xi(X; Z) \geq 0$ , with equality if and only if  $X$  contains uncorrelated random variables, i.e.  $\hat{h}(X) = h(X)$  and  $\hat{h}(X|Z) = h(X|Z)$ .

*Proof of Proposition 10.* Suppose  $\hat{I}(X; Z) \geq I(X; Z)$ . By subtracting this inequality from (35) we have

$$\underbrace{I(X; Z) - \hat{I}(X; Z)}_{-a^2} \leq \underbrace{\hat{I}(X; Z) - I(X; Z)}_{a^2} + \xi(X; Z)$$

$$\xi(X; Z) \geq -2a^2$$

because  $\xi(X; Z) \geq 0$ ,  $a = 0$  and  $\hat{I}(X; Z) = I(X; Z)$  are the only possibilities. Now suppose  $\hat{I}(X; Z) \leq I(X; Z)$ ; by summing this inequality and (35) we have

$$I(X; Z) + \hat{I}(X; Z) \leq \hat{I}(X; Z) + I(X; Z) + \xi(X; Z)$$

$$\xi(X; Z) \geq 0$$

which shows that information can only be lost and not gained.  $\square$