Active Mapping via Gradient Ascent Optimization of Shannon Mutual Information over Continuous *SE(3)* Trajectories

Arash Asgharivaskasi Shumon Koga Nikolay Atanasov

***Abstract*— The problem of active mapping aims to plan an informative sequence of sensing views given a limited budget such as distance traveled. This paper considers active occupancy grid mapping using a range sensor, such as LiDAR or depth camera. State-of-the-art methods optimize information-theoretic measures relating the occupancy grid probabilities with the range sensor measurements. The non-smooth nature of ray-tracing within a grid representation makes the objective function non-differentiable, forcing existing methods to search over a discrete space of candidate trajectories. This work proposes a differentiable approximation of the Shannon mutual information between a grid map and ray-based observations that enables gradient ascent optimization in the continuous space of *SE(3)* sensor poses. Our gradient-based formulation leads to more informative sensing trajectories, while avoiding occlusions and collisions. The proposed method is demonstrated in simulated and real-world experiments in 2-D and 3-D environments. Materials supplementing this paper are available at: `https://arashasgharivaskasi-bc.github.io/grad_active_mapping/`**

I. Introduction

Mapping an unknown environment using sensor-equipped mobile robots has been widely studied motivated by many real-world applications such as search and rescue operations, planetary exploration, security and surveillance. While the traditional mapping methods have been developed for a given robot trajectory, optimizing the path to increase the accuracy of the constructed map is significant for the operation of autonomous robots in highly unstructured environments. Such an "active mapping" problem renders challenges in the computation of objective function and its reliable optimization method in terms of enhancing the performance and computational feasibility.

One of the most common procedures for incremental estimation of occupancy grid maps is to utilize a narrow-ray range sensor, such as Laser range finder [1, Ch. 9]. However, the resulting observation model occurs to be a piece-wise constant, namely a non-smooth function of measurement poses due to the discrete nature of ray-tracing over a grid map [2, Ch. 5]. Hence, the objective function is piecewise-constant, making the derivatives either 0 or undefined. In addition to the problem of non-smoothness of grid map representations, occlusion, which is caused when a field of view intersects with objects, is a significant challenge in sensing and planning to accomplish several tasks in robotics such as 3-D reconstruction [3], object classification [4], and target tracking [5]. Koga *et al.* [6] has developed dense active mapping using iterative Covariance Regulation (iCR), which utilizes a differentiable field of view that enables deriving an explicit gradient of an information-theoretic objective with respect to a multi-step control sequence. Incorporating occlusion in iCR makes the field of view dependent on the map estimate, which renders a significant challenge for sensing model formulation since it violates the linear-Gaussian property of the mapping by Kalman Filter. Nevertheless, the current work borrows one of the key ideas from iCR: In order to perform gradient-based optimization of the mutual information, one needs to apply a smooth proximity operator among the elements involved in its computation.

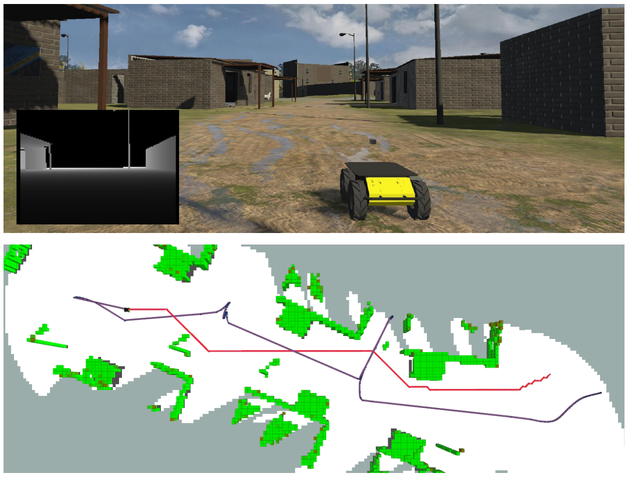


Fig. 1: Occlusion and collision-aware active mapping. Top: Wheeled robot exploring an unknown environment equipped with a depth sensor used for estimating a 3-D map. Bottom: Path planning given the current map. The red trajectory is A^* planning towards a frontier. The purple trajectory is obtained via gradient ascent optimization of the objective function.

The work by Rocha *et al.* [7] was among the first that attempted to derive a gradient-based strategy for active mapping of occupancy grids. The authors proposed a gradient of map entropy with respect to the robot pose at a cell center via finite difference of entropy values at adjacent cells. Julian [2] proposed a divergent beam sensor model, where the width of a beam increases radially as it travels farther through space. While the derived mutual information formula was shown to be differentiable, it suffers from high computational complexity as it requires numerical integration of the objective function. Charrow *et al.* [8] proposed a numerical evaluation for the gradient of Cauchy-Schwarz mutual information (CSQMI) [9] using finite differences of

We gratefully acknowledge support from NSF FRR CAREER 2045945, ONR SAI N00014-18-1-2828, and ARL DCIST CRA W911NF-17-2-0181.

The authors are with the Department of Electrical and Computer Engineering, University of California San Diego, CA 92093, USA {aasghari,skoga,natanasov}@eng.ucsd.edu.

CSQMI evaluated at cell centers. Our work is most similar to [10] and [11], where the authors formulate the information gain as a sum of informative elements weighted by a discount factor. In particular, [10] defines informative elements as frontier cells between free and unexplored areas visible from a candidate pose. However, unlike the mutual information between the map and a sensor observation, using visible frontier size as a proxy for information gain does not take into account the effect of sensor noise which is inevitable in real-world sensing applications [9], [12].

The present work distinguishes itself from the prior methods by proposing an active mapping strategy that allows gradient ascent optimization of the Shannon mutual information (SMI) between the grid map and a sequence of beam-based observations. As opposed to the discrete-space active mapping methods [9], [13]–[15] that aim to plan an informative robot path through evaluating a finite set of candidate trajectories, the current work finds an optimal trajectory over the continuous space of the robot state. Fig. 1 illustrates the proposed gradient-based active mapping using a depth sensor. Our main **contributions** are:

1) a differentiable interpolation of the SMI as well as a closed-form gradient expression,
2) decomposition of the SMI into additive terms over a robot trajectory in *SE(3)*, under sufficient assumptions for the interpolation method.

Unlike the previous works that used map entropy or visible number of frontiers, directly utilizing the SMI allows incorporating the range sensor noise specifications to the objective function; leading to more accurate exploration. Moreover, gradient-based methods allow augmenting the objective function with other differentiable terms (e.g. localization accuracy [16], path cost [17]). With the addition of a differentiable collision penalty to the objective function, we propose an occlusion and collision-aware robot exploration.

II. Problem Statement

Consider a robot with pose $\mathbf{X}_t \in$ *SE(3)* at time t:

$$\mathbf{X}_t := \begin{bmatrix} \mathbf{R}_t & \mathbf{p}_t \\ \mathbf{0}^\top & 1 \end{bmatrix}, \tag{1}$$

where $\mathbf{R}_t \in SO(3)$ is the robot orientation and $\mathbf{p}_t \in \mathbb{R}^3$ is the robot position. The robot is navigating in an environment composed of occupied and free space. A mounted range sensor, e.g. LiDAR or depth camera, provides the robot with a stream of beam-based observations $\mathbf{z}_t \in \mathbb{R}^B$, where B is the number of beams in a laser scan or pixels in a depth image, measuring the distance from the robot's position to the closest obstacle along the beam. We model the map $\mathbf{m}$ as a grid of cells m_i, $i \in \mathcal{I} := \{1, \ldots, N\}$, where each cell can take one of the two states: *free* or *occupied*. To model measurement noise, we consider a probability density function (PDF) $p(\mathbf{z}_t \mid \mathbf{m}, \mathbf{X}_t)$ for each observation. Let $p_t(\mathbf{m}) = p(\mathbf{m} \mid \mathcal{H}_t)$ be the probability mass function (PMF) of the map $\mathbf{m}$ given the history of robot poses and observations $\mathcal{H}_t = \{(\mathbf{X}_\tau, z_\tau)\}_{\tau=1}^t$. A new observation $\mathbf{z}_{t+1}$ made from robot pose $\mathbf{X}_{t+1}$ can then be integrated into the map estimation process using Bayes rule:

$$p_{t+1}(\mathbf{m}) \propto p(\mathbf{z}_{t+1} \mid \mathbf{m}, \mathbf{X}_{t+1}) p_t(\mathbf{m}). \tag{2}$$

The goal is to choose a collision-free pose trajectory to obtain maximally informative measurements for constructing an accurate map. As shown by Julian *et al.* [18], maximizing the Shannon Mutual Information (SMI) between the map $\mathbf{m}$ and a sequence of potential future measurements $\mathbf{z}_{t+1:t+T}$ yields an efficient active mapping strategy. The SMI is defined as:

$$\begin{aligned} &I(\mathbf{m}; \mathbf{z}_{t+1:t+T} \mid \mathbf{X}_{t+1:t+T}, \mathcal{H}_t) := \\ &\sum_{\mathbf{m} \in 2^N} \int \cdots \int p(\mathbf{m}, \mathbf{z}_{t+1:t+T} \mid \mathbf{X}_{t+1:t+T}, \mathcal{H}_t) \times \\ &\log \frac{p(\mathbf{m}, \mathbf{z}_{t+1:t+T} \mid \mathbf{X}_{t+1:t+T}, \mathcal{H}_t)}{p(\mathbf{m} \mid \mathcal{H}_t) p(\mathbf{z}_{t+1:t+T} \mid \mathbf{X}_{t+1:t+T}, \mathcal{H}_t)} \prod_{\tau=1}^{T} d\mathbf{z}_{t+\tau}, \end{aligned} \tag{3}$$

where $\mathcal{H}_t$ represents the realized history of robot poses and observations and, hence, does not appear as an integration variable. Throughout this paper, we assume that the robot pose $\mathbf{X}_t$ is known for all t. The SMI is a function $I(.) :$ *SE(3)*$^T \to \mathbb{R}_{\geq 0}$ of the robot trajectory $\mathbf{X}_{t+1:t+T}$ parameterized by $\mathcal{H}_t$.

Problem. Given a map PMF $p_t(\mathbf{m})$ obtained from prior robot poses and observations $\mathcal{H}_t$ and a finite planning horizon T, find a pose trajectory $\mathbf{X}_{t+1:t+T} \in$ *SE(3)*^{T} that maximizes the SMI between the map $\mathbf{m}$ and the future observations $\mathbf{z}_{t+1:t+T}$ with PDF in (2):

$$\begin{aligned} \max_{\mathbf{X}_{t+1:t+T} \in SE(3)^T} \Big(& I(\mathbf{m}; \mathbf{z}_{t+1:t+T} \,| \mathbf{X}_{t+1:t+T}, \mathcal{H}_t) - \\ & \gamma_c C(\mathbf{X}_{t+1:t+T}) \Big), \end{aligned} \tag{4}$$

where $C(\mathbf{X}_{t+1:t+T})$ is a penalty term capturing the cost of collisions along $\mathbf{X}_{t+1:t+T}$ and $\gamma_c \geq 0$ is the weight of the collision penalty.

In the next section, we propose a differentiable approximation of the SMI function that can be utilized for gradient-based optimization of (4).

III. Proposed Method

A. Notation

We overload $\hat{(.)}$ to denote the mapping from an axis-angle vector $\boldsymbol{\theta} \in \mathbb{R}^3$ to a 3×3 skew-symmetric matrix $\hat{\boldsymbol{\theta}} \in \mathfrak{so}(3)$ as well as from a vector $\boldsymbol{\xi} \in \mathbb{R}^6$ to a 4×4 twist matrix:

$$\boldsymbol{\xi} = \begin{bmatrix} \boldsymbol{\rho} \\ \boldsymbol{\theta} \end{bmatrix} \in \mathbb{R}^6 \qquad \hat{\boldsymbol{\xi}} := \begin{bmatrix} \hat{\boldsymbol{\theta}} & \boldsymbol{\rho} \\ \mathbf{0}^\top & 0 \end{bmatrix} \in \mathfrak{se}(3). \tag{5}$$

We define an infinitesimal change of pose $\mathbf{X} \in SE(3)$ using a right perturbation $\mathbf{X} \exp(\hat{\boldsymbol{\xi}}) \in SE(3)$. The functions $\log(.) :$ *SE(3)* $\to \mathfrak{se}(3)$ and $(.)^\vee : \mathfrak{se}(3) \to \mathbb{R}^6$ denote the inverse mappings associated with $\exp(.)$ and $\hat{(.)}$, respectively. Please refer to [19, Ch.7] for details.

B. One Step Ahead Planning

We first study the case where $T = 1$, i.e., the robot view is optimized only one step into the future. The core idea is to introduce the notion of a viewpoint grid, which is a discrete set of candidate robot poses. Then, the SMI with respect to an arbitrary robot pose is approximated as a linear combination of the SMI with respect to the candidate poses with a differentiable function with respect to an arbitrary robot pose.

Definition 1. A viewpoint grid $\mathcal{G}$ is a set of robot poses $\mathbf{X} \in \mathit{SE(3)}$ with position $\mathbf{p} \in \mathbb{R}^3$ and orientation $\mathbf{R} \in SO(3)$ such that $(\mathbf{p}, \mathbf{R}) \in \mathcal{P} \times \mathcal{R}$, where $\mathcal{P}$ is the set of all map cell centers and $\mathcal{R}$ is a finite set of orientations.

We approximate the SMI at pose $\mathbf{X}$ as a convex combination of the SMI computed over all poses in the view grid $\mathcal{G}$. Namely, $I(\mathbf{m}; \mathbf{z} \mid \mathbf{X}, \mathcal{H}_t) \approx \tilde{I}(\mathbf{m}; \mathbf{z} \mid \mathbf{X}, \mathcal{H}_t)$, and

$$\begin{aligned}\tilde{I}(\mathbf{m}; \mathbf{z} \mid \mathbf{X}, \mathcal{H}_t) &:= \sum_{\mathbf{V} \in \mathcal{G}} \alpha_{\mathbf{V}}(\mathbf{X}) I(\mathbf{m}; \mathbf{z} \mid \mathbf{V}, \mathcal{H}_t), \\ \text{where } \sum_{\mathbf{V} \in \mathcal{G}} \alpha_{\mathbf{V}}(\mathbf{X}) &= 1, \; \forall \mathbf{V} \in \mathcal{G}: \; 0 \leq \alpha_{\mathbf{V}}(\mathbf{X}),\end{aligned} \tag{6}$$

and $\alpha_{\mathbf{V}}(\mathbf{X})$ is a differentiable function with respect to robot pose $\mathbf{X}$. In practice, evaluating the SMI for all $\mathbf{V} \in \mathcal{G}$ during each planning phase would be computationally expensive. Therefore, one needs to design a distance metric for $\alpha_{\mathbf{V}}(\mathbf{X})$ that is only non-zero in close vicinity of the robot pose $\mathbf{X}$. The idea of pulling the robot pose $\mathbf{X}$ out of the SMI function $I(.)$ using a differentiable weighting function $\alpha_{\mathbf{V}}(\mathbf{X})$ makes it possible to obtain non-zero derivatives for the approximate SMI function $\tilde{I}(.)$ with respect to the robot pose. Fig. 2 illustrates the SMI approximated from 3 viewpoints; note how each viewpoint creates a 'field' of information in its vicinity, while the approximate SMI equals the net influence of all 3 viewpoints.

The differentiable property of the approximate SMI enables gradient-based optimization of the robot pose in order to generate maximally informative observations. This is done via applying a small perturbation $\boldsymbol{\psi} \in \mathbb{R}^6$ in the robot frame to the pose $\mathbf{X}$ along the direction of the gradient with a step size of l:

$$\begin{aligned}\mathbf{X}^{(k+1)} &= \mathbf{X}^{(k)} \exp(l \hat{\boldsymbol{\psi}}^{(k)}), \\ \boldsymbol{\psi}^{(k)} &= \nabla_{\boldsymbol{\psi}} \tilde{I}(\mathbf{m}; \mathbf{z} \mid \mathbf{X}^{(k)} \exp(\hat{\boldsymbol{\psi}}), \mathcal{H}_t)|_{\boldsymbol{\psi}=0} \\ &= \sum_{\mathbf{V} \in \mathcal{G}} \nabla_{\boldsymbol{\psi}} \alpha_{\mathbf{V}}(\mathbf{X}^{(k)} \exp(\hat{\boldsymbol{\psi}}))|_{\boldsymbol{\psi}=0} I(\mathbf{m}; \mathbf{z} \mid \mathbf{V}, \mathcal{H}_t).\end{aligned} \tag{7}$$

Note that the gradient is a 6-dimensional vector since the robot pose in *SE(3)* has 6 degrees of freedom. In Sec. III-D, we derive a closed form expression of the gradient for a particular selection of the weighting function $\alpha_{\mathbf{V}}(\mathbf{X})$. The gradient ascent rule of (7) concludes our method proposition for finding the (locally) most informative next robot pose when the planning horizon T is set to 1. In the following, we discuss the case where we are interested in optimizing a multi-step sequence of robot poses, given a history of prior observations $\mathcal{H}_t$.

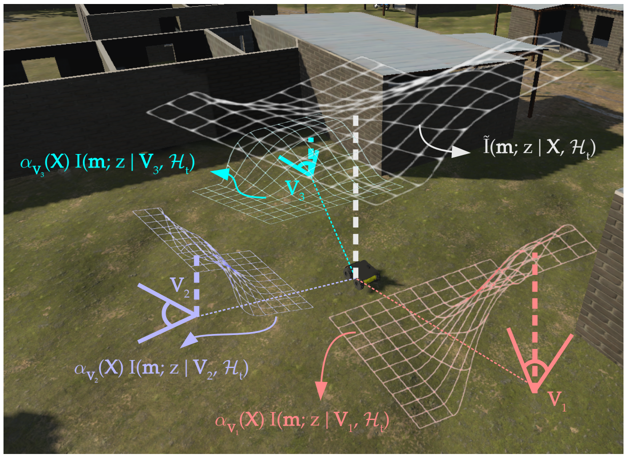

$\alpha_{\mathbf{V}_3}$(X) I(m; z | V$_3$, $\mathcal{H}_t$)
V$_3$
Ĩ(m; z | X, $\mathcal{H}_t$)
V$_2$
$\alpha_{\mathbf{V}_2}$(X) I(m; z | V$_2$, $\mathcal{H}_t$)
$\alpha_{\mathbf{V}_1}$(X) I(m; z | V$_1$, $\mathcal{H}_t$)
V$_1$

Fig. 2: The SMI approximation via 3 viewpoints, colored differently for each $\mathbf{V}_i$, $i \in \{1, 2, 3\}$. The field of view of each viewpoint determines their corresponding SMI $I(\mathbf{m}, \mathbf{z} \mid \mathbf{V}_i, \mathcal{H}_t)$, while the weight $\alpha_{\mathbf{V}_i}(\mathbf{X})$ dictates the contribution of $\mathbf{V}_i$ to the approximated SMI $\tilde{I}(\mathbf{m}, \mathbf{z} \mid \mathbf{X}, \mathcal{H}_t)$, colored white.

C. Trajectory Optimization for Active Mapping

Here, we present the approximation of the SMI for a trajectory, rather than a single robot pose. Analogous to (6), we define the approximated SMI for robot trajectory $\mathbf{X}_{t+1:t+T}$ as a convex combination over all trajectories in the set $\mathcal{G}^T$. Namely, $I(\mathbf{m}; \mathbf{z}_{t+1:t+T} \mid \mathbf{X}_{t+1:t+T}, \mathcal{H}_t) \approx \tilde{I}(\mathbf{m}; \mathbf{z}_{t+1:t+T} \mid \mathbf{X}_{t+1:t+T}, \mathcal{H}_t)$, and

$$\begin{aligned}&\tilde{I}(\mathbf{m}; \mathbf{z}_{t+1:t+T} \mid \mathbf{X}_{t+1:t+T}, \mathcal{H}_t) := \\ &\qquad \sum_{\mathcal{V} \in \mathcal{G}^T} A_{\mathcal{V}}(\mathbf{X}_{t+1:t+T}) I(\mathbf{m}; \mathbf{z}_{t+1:t+T} \mid \mathcal{V}, \mathcal{H}_t),\end{aligned} \tag{8}$$

where the weighting function $A_{\mathcal{V}}(\mathbf{X}_{t+1:t+T})$ for a trajectory $\mathcal{V} = V_{t+1:t+T}$ is defined as:

$$A_{\mathcal{V}}(\mathbf{X}_{t+1:t+T}) := \prod_{\tau=1}^{T} \alpha_{\mathbf{V}_{t+\tau}}(\mathbf{X}_{t+\tau}). \tag{9}$$

The terms $\alpha_{\mathbf{V}_{t+\tau}}(\mathbf{X}_{t+\tau})$ follow the same properties as in (6), making (8) a convex combination of the SMI terms. Computing $I(\mathbf{m}; \mathbf{z}_{t+1:t+T} \mid \mathcal{V}, \mathcal{H}_t)$ requires integration over all instances of combined observations $\mathbf{z}_{t+1:t+T}$ and should be repeated for all $\mathcal{V} \in \mathcal{G}^T$, which is computationally infeasible during the planning time. We aim to impose a structure on the weighting function $\alpha_{\mathbf{V}_{t+\tau}}(\mathbf{X}_{t+\tau})$ that allows breaking down $I(\mathbf{m}; \mathbf{z}_{t+1:t+T} \mid \mathcal{V}, \mathcal{H}_t)$ into independent additive terms. The following conditions enable such a decomposition:

Condition 1. Given a robot pose $\mathbf{X}$, $\alpha_{\mathbf{V}}(\mathbf{X})$ is non-zero only for a subset $\bar{\mathcal{G}}(\mathbf{X}) \subset \mathcal{G}$ of viewpoints within a distance $\xi_{\max}$ from $\mathbf{X}$.

Condition 2. Let $\mathcal{F} \subset \mathbb{R}^3$ be the unobstructed field of view (FOV) of the sensor in robot frame with homogeneous representation $\underline{\mathcal{F}}$. Also, let $\mathbb{U}(\mathbf{X}) := \cup_{\mathbf{V} \in \bar{\mathcal{G}}(\mathbf{X})} \mathbf{V} \underline{\mathcal{F}} \subset \mathbb{R}^3$ be the union of all FOVs belonging to $\bar{\mathcal{G}}(\mathbf{X})$ in the world frame. This condition states that, for any pair of robot poses $\mathbf{X}_i$ and $\mathbf{X}_j$ $(i \neq j)$ in trajectory $\mathbf{X}_{t+1:t+T}$, we have $\mathbb{U}(\mathbf{X}_i) \cap \mathbb{U}(\mathbf{X}_j) = \emptyset$.

The above conditions are sufficient for decomposing the approximate SMI of trajectory $\mathbf{X}_{t+1:t+T}$ to T independent additive terms, resulting in a computationally feasible trajectory optimization formula. The main idea comes from the fact that, given the above conditions, observations $\mathbf{z}_i$ and $\mathbf{z}_j$ ($i \neq j$) made from viewpoints inside $\mathcal{G}(\mathbf{X}_i)$ and $\mathcal{G}(\mathbf{X}_j)$, respectively, are independent random variables. In practice, active sensors such as lasers have bounded FOV which can meet these conditions. For passive sensors (e.g. cameras), it is commonplace to limit the effective range since the estimation accuracy diminishes as we get farther from the sensor; leading to a limited *applicable* FOV.

Proposition 1. *Under Cond. 1 and 2, the approximated SMI can be expressed as the sum of individual SMI approximations for each pose in the trajectory $\mathbf{X}_{t+1:t+T}$:*

$$\tilde{I}(\mathbf{m}; \mathbf{z}_{t+1:t+T} \mid \mathbf{X}_{t+1:t+T}, \mathcal{H}_t) = \sum_{\tau=1}^{T} \tilde{I}(\mathbf{m}; \mathbf{z}_{t+\tau} \mid \mathbf{X}_{t+\tau}, \mathcal{H}_t). \tag{10}$$

Proof. See Appendix A. □

The result of Prop. 1 enables computationally feasible trajectory optimization for robot exploration in an unknown environment. Since each term in (10) is only dependent upon a single pose in the trajectory, the gradient ascent rule in (7) can be directly employed to update each robot pose $\mathbf{X}_{t+\tau}$, $\tau \in \{1, \dots, T\}$. In the following part, we introduce a practical gradient-based solution to the problem of active mapping stated in (4).

D. Active Mapping via Gradient Ascent

A key advantage of gradient-based optimization is the possibility of adding various reward or penalty terms to the objective function, enabling achievement of a more complex optimization goal. We begin by defining the collision penalty term $C(\mathbf{X}_{t+1:t+T})$ in the objective function of (4), which is responsible for driving the optimized robot pose away from obstacles within the environment. However, since we do not know the map a priori, we resort to the estimation of the map to extract the obstacles.

Definition 2. Let $\hat{\mathcal{E}}_f(p_t(\mathbf{m}))$ be the maximum-likelihood estimation of the free space at time t. For a position $\mathbf{p} \in \mathbb{R}^3$, we define *free distance* as follows:

$$d(\mathbf{p}, p_t(\mathbf{m})) = \min_{\mathbf{b} \in \partial \hat{\mathcal{E}}_f(p_t(\mathbf{m}))} \|\mathbf{p} - \mathbf{b}\|_2. \tag{11}$$

It is important to consider that large mutual information occurs near the boundary between the free space and the unknown parts of the map [18]. Therefore, one should seek a balance between large clearance from obstacles and informativeness of observation made from the resulting robot pose. We define the collision cost $C(\mathbf{X}_{t+1:t+T})$ as sum of the log-values of inverse free distance for each pose $\mathbf{X}_{t+\tau}$ with position $\mathbf{p}_{t+\tau}$ in the trajectory:

$$C(\mathbf{X}_{t+1:t+T}) = -\sum_{\tau=1}^{T} \log\left(d(\mathbf{p}_{t+\tau}, p_t(\mathbf{m}))\right). \tag{12}$$

Using a logarithmic scale causes a large penalty for poses close to obstacles, while it does not discourage approaching the unknown region from a safe distance due to its suppressed gradient over large inputs.

In addition to the collision cost, we add a penalty term to the objective function to enforce Cond. (2) during each planning phase, minimizing the overlap among the sensor FOVs $\mathbf{X}_{t+\tau}\mathcal{F}$ in the candidate trajectory. We consider a pairwise penalty term for poses within the trajectory as follows:

$$\begin{aligned} q(\mathbf{p}_i, \mathbf{p}_j) &= \max\{0, 2\delta_q - \|\mathbf{p}_i - \mathbf{p}_j\|_2\}^2 \\ \delta_q &= |\mathcal{F}| + \xi_{\max}, \end{aligned} \tag{13}$$

where $\mathbf{p}_i$ and $\mathbf{p}_j$ are robot positions for poses $\mathbf{X}_i$ and $\mathbf{X}_j$, respectively, $|\mathcal{F}|$ is the diameter of $\mathcal{F}$, and $\xi_{\max}$ denotes the maximum distance from robot pose $\mathbf{X}$ to a viewpoint in $\bar{\mathcal{G}}(\mathbf{X})$ (cf. Cond. 1). The penalty term (13) effectively discourages the case where the 2-norm ball $\mathcal{B}(\mathbf{p}_i, \delta_q)$ centered around $\mathbf{p}_i$ with radius δ_q coincides with $\mathcal{B}(\mathbf{p}_j, \delta_q)$. Since $\mathcal{B}(\mathbf{p}_i, \delta_q)$ contains $\mathbb{U}(\mathbf{X}_i)$, $q(\mathbf{p}_i, \mathbf{p}_j) = 0$ is sufficient to ensure Cond. (2) is not violated for a pair of poses $\mathbf{X}_i$ and $\mathbf{X}_j$. Note that Cond. (1) is an inherent property of the weighting function $\alpha_{\mathbf{V}}(\mathbf{X})$ and can be evaluated offline. Putting all the components together, the differentiable objective function for gradient-based active mapping is expressed as follows:

$$\begin{aligned} &f(\mathbf{X}_{t+1:t+T}) = \\ &\sum_{\tau=1}^{T} \Bigg(\sum_{\mathbf{V} \in \bar{\mathcal{G}}(\mathbf{X}_{t+\tau})} \alpha_{\mathbf{V}}(\mathbf{X}_{t+\tau}) [I(\mathbf{m}; \mathbf{z} \mid \mathbf{V}, \mathcal{H}_t) + \\ &\gamma_c \log(d(\mathbf{v}, p_t(\mathbf{m})))] - \frac{\gamma_q}{2} \sum_{\substack{\tau'=1 \\ \tau' \neq \tau}}^{T} q(\mathbf{p}_{t+\tau}, \mathbf{p}_{t+\tau'}) \Bigg), \end{aligned} \tag{14}$$

where $\mathbf{v}$, $\mathbf{p}_{t+\tau}$, and $\mathbf{p}_{t+\tau'}$ are the corresponding positions of poses $\mathbf{V}$, $\mathbf{X}_{t+\tau}$, and $\mathbf{X}_{t+\tau'}$.

So far we assumed a general definition for the differentiable weighting function $\alpha_{\mathbf{V}}(\mathbf{X})$ that satisfies Cond. 1 and 2. We use the following definition for $\alpha_{\mathbf{V}}(\mathbf{X})$:

$$\alpha_{\mathbf{V}}(\mathbf{X}) = \frac{\upsilon(\delta(\boldsymbol{\xi}_{\mathbf{X},\mathbf{V}}))(1 + \cos\delta(\boldsymbol{\xi}_{\mathbf{X},\mathbf{V}}))}{\sum_{\mathbf{U} \in \mathcal{G}} \upsilon(\delta(\boldsymbol{\xi}_{\mathbf{X},\mathbf{U}}))(1 + \cos\delta(\boldsymbol{\xi}_{\mathbf{X},\mathbf{U}}))}, \tag{15}$$

where $\delta(\boldsymbol{\xi}_{\mathbf{X},\mathbf{V}})$ is the distance between poses $\mathbf{V}, \mathbf{X} \in SE(3)$:

$$\delta(\boldsymbol{\xi}_{\mathbf{X},\mathbf{V}}) = \frac{\pi}{\xi_{\max}} \sqrt{\boldsymbol{\xi}_{\mathbf{X},\mathbf{V}}^{\top} \Gamma \boldsymbol{\xi}_{\mathbf{X},\mathbf{V}}}, \quad \boldsymbol{\xi}_{\mathbf{X},\mathbf{V}} = \log(\mathbf{X}^{-1}\mathbf{V})^{\vee}. \tag{16}$$

Here, Γ is a diagonal matrix containing positive coefficients and $\boldsymbol{\xi}_{\mathbf{X},\mathbf{V}}$ is the difference of two *SE(3)* poses in the local frame of $\mathbf{X}$. The indicator function $\upsilon(\delta(\boldsymbol{\xi}_{\mathbf{X},\mathbf{V}}))$ in (15) is equal to one only when $0 \leq \delta(\boldsymbol{\xi}_{\mathbf{X},\mathbf{V}}) \leq \pi$ and zero otherwise, which in effect limits the SMI approximation to the viewpoints within the radius $\xi_{\max}$ from $\mathbf{X}$, satisfying Cond. 1. Note that the discontinuity of $\upsilon(\delta(\boldsymbol{\xi}_{\mathbf{X},\mathbf{V}})$ occurs at the same point where $1 + \cos\delta(\boldsymbol{\xi}_{\mathbf{X},\mathbf{V}}) = 0$; hence $\alpha_{\mathbf{v}}(\mathbf{X})$ is differentiable with respect to all poses $\mathbf{X} \in$ *SE(3)*. Fig. 3 shows an example of the accurate SMI evaluation at different robot positions compared to the approximate SMI of (6) using the weighting function of (15). Note that, while being

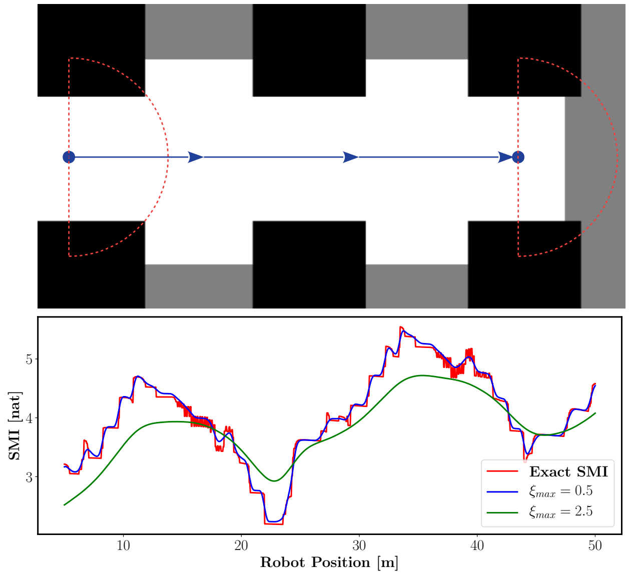


Fig. 3: Example of the SMI approximation using (15). Top: Partially known environment where black, white, and gray regions depict occupied, free, and unknown areas. The robot equipped with a 180° range sensor moves along a straight line shown as the blue arrowed segment. Bottom: Exact value of the SMI evaluated at each robot position alongside approximate SMI values for various radii ξ_{max}.

differentiable, the approximations follow the occlusion-aware behavior of the exact SMI, i.e. they peak at the positions where more of the unexplored region is visible. With $\alpha_{\mathbf{V}}(\mathbf{X})$ specified as (15), we can compute a closed-form expression for the gradient of the approximate SMI as follows.

Proposition 2. *Using the weighting function of* (15)*, the gradient of the approximate SMI with respect to the robot pose can be obtained as follows:*

$$\begin{aligned}\nabla_{\boldsymbol{\psi}} \tilde{I}(\mathbf{m};\mathbf{z} \mid \mathbf{X}\exp(\hat{\boldsymbol{\psi}}), \mathcal{H}_t)|_{\boldsymbol{\psi}=0} &= \left(\frac{\pi}{\xi_{\max}\eta(\mathbf{X})}\right)^2 \\ &\times \mathcal{J}_R^\top(\boldsymbol{\xi}_{\mathbf{X}}) \sum_{\mathbf{V}\in\bar{\mathcal{G}}(\mathbf{X})} \Lambda_{\mathbf{V}}(\mathbf{X})\boldsymbol{\xi}_{\mathbf{X},\mathbf{V}}, \\ \Lambda_{\mathbf{V}}(\mathbf{X}) = [\eta(\mathbf{X}) I(\mathbf{m};\mathbf{z} \mid \mathbf{V}, \mathcal{H}_t) - \beta(\mathbf{X})] & \\ &\times \frac{\sin\delta(\boldsymbol{\xi}_{\mathbf{X},\mathbf{V}})}{\delta(\boldsymbol{\xi}_{\mathbf{X},\mathbf{V}})} \mathcal{J}_R^\top(-\boldsymbol{\xi}_{\mathbf{X},\mathbf{V}})\Gamma, \\ \eta(\mathbf{X}) = |\bar{\mathcal{G}}(\mathbf{X})| + \sum_{\mathbf{U}\in\bar{\mathcal{G}}(\mathbf{X})} \cos\delta(\boldsymbol{\xi}_{\mathbf{X},\mathbf{U}}), & \\ \beta(\mathbf{X}) = \sum_{\mathbf{U}\in\bar{\mathcal{G}}(\mathbf{X})} I(\mathbf{m};\mathbf{z} \mid \mathbf{U}, \mathcal{H}_t)(1+\cos\delta(\boldsymbol{\xi}_{\mathbf{X},\mathbf{U}})), & \\ \boldsymbol{\xi}_{\mathbf{X}} = \log(\mathbf{X})^\vee, &\end{aligned} \tag{17}$$

where $\bar{\mathcal{G}}(\mathbf{X})$ *is the subset of viewpoints* $\mathbf{V} \in \mathcal{G}$ *that* $\upsilon(\delta(\boldsymbol{\xi}_{\mathbf{X},\mathbf{V}})) = 1$, *and* $\mathcal{J}_R(.)$ *is the right Jacobian of SE(3).*

Proof. See Appendix B. □

The objective function (14) can used with any Bayesian sensing model (2). In the context of occupancy grid mapping via a range sensor, the log-odds technique using a narrow beam [1, Ch. 9] is a natural choice for the sensing model which provides a computationally simple yet accurate representation of the environment. Using such a model, the SMI between the grid map and a beam z can be written as a weighted sum of occlusion probabilities [14], [15], [18]:

$$I(\mathbf{m}; z \mid \mathbf{X}, \mathcal{H}_t) = \sum_{i=1}^{N_{\mathbf{X}}} w_i(\mathcal{H}_t) p(i \mid \mathcal{H}_t) \tag{18}$$

where $N_{\mathbf{X}}$ is the maximum number of map cells that can be visited by a sensor beam emitted from pose $\mathbf{X}$ and $p(i|\mathcal{H}_t)$ is the probability that the beam is occluded at i-th cell along its path. Hence, the corresponding SMI expression is occlusion-aware, namely the value of the mutual information is directly controlled by whether or not the observations are occluded.

The benefit of using the occlusion-aware SMI formulation of (18) comes with the computational burden of ray-tracing during each evaluation of the SMI, since (18) requires identifying the map cells along the observation beam. Going back to the objective function (14), it is required to perform $B\sum_{\tau=1}^{T} |\hat{\mathcal{G}}(\mathbf{X}_{t+\tau})|$ ray-tracings for a range sensor with B beams during every evaluation of $f(\mathbf{X}_{t+1:t+T})$. Since all viewpoints in $\mathcal{G}$ are located at map cell centers, the ray-tracing for a viewpoint $\mathbf{V}$ with position $\mathbf{v}$ and orientation $\mathbf{R}$ can be obtained from a ray-tracing from the map origin with the same orientation and simply translated by $\mathbf{v}$. Considering that all viewpoints in $\mathcal{G}$ have a fixed set of orientations $\mathcal{R}$, we skip online ray-tracing by computing $|\mathcal{R}|$ ray-tracings from the origin only once and query ray-tracings by applying translation $\mathbf{v}$ for each viewpoint $\mathbf{V}$. To further accelerate the optimization, we avoid repeated evaluations of $I(\mathbf{m};\mathbf{z} \mid \mathbf{V}, \mathcal{H}_t)$ by storing the values for each viewpoint $\mathbf{V}$ until the map is updated. For computing the free distance $d(\mathbf{v}, p_t(\mathbf{m}))$, we once again use the fact that all viewpoints are located at map cell centers; therefore we can obtain $d(\mathbf{v}, p_t(\mathbf{m}))$ from the distance transform of $\hat{\mathcal{E}}_f(p_t(\mathbf{m}))$ in Def. 2 scaled by the map resolution. This needs to be computed only once for each planning step, since the distance transform provides values of $d(\mathbf{v}, p_t(\mathbf{m}))$ for all cell centers $\mathbf{v}$.

Gradient-based optimization of the objective function (14) allows local maximization of trajectory informativeness as well as the distance from obstacles. Given an initial trajectory which can be provided by frontier-based exploration [13], we perform gradient ascent for $n_{\max}$ steps or until the improvement in $f(\mathbf{X}_{t+1:t+T})$ is less than 0.1%. The output of this optimization will be a kinematically feasible trajectory $\mathbf{X}^*_{t+1:t+T} \in SE(3)^T$ which can be tracked by a low-level controller specified by the robot dynamics. Alg. 1 summarizes our procedure for occlusion and collision-aware active mapping. We evaluate the performance of the proposed method in the next section.

IV. Experiments

In this section we evaluate the performance of our proposed active mapping method in several simulated and real-world experiments in comparison to baseline exploration strategies: frontier-based exploration (Frontier) [13], FSMI [14], SSMI [15], and optimized next best view using

Algorithm 1 Active Mapping via Gradient Ascent

```
Input: $\mathbf{X}_t$, $p_t(\mathbf{m})$, $T$, $n_{\max}$, $pcrt$
if $pcrt$ is None then
    $pcrt = \text{PRECOMPUTERAYTRACING}()$
$\mathbf{d} = \text{DISTANCETRANSFORM}(p_t(\mathbf{m}))$
$k \leftarrow 0$
$\mathbf{X}^{(k)}_{t+1:t+T} = \text{INITPATH}(\mathbf{X}_t, p_t(\mathbf{m}), T)$
while not $\text{TERMINATE}(f(\mathbf{X}^{(k)}_{t+1:t+T}), n_{\max})$ do
    $\mathbf{X}^{(k+1)}_{t+1:t+T} = \text{GRADASCENT}(f(\mathbf{X}^{(k)}_{t+1:t+T}), pcrt, \mathbf{d})$
    $k \leftarrow k + 1$
return $\mathbf{X}^{(k)}_{t+1:t+T}$
```

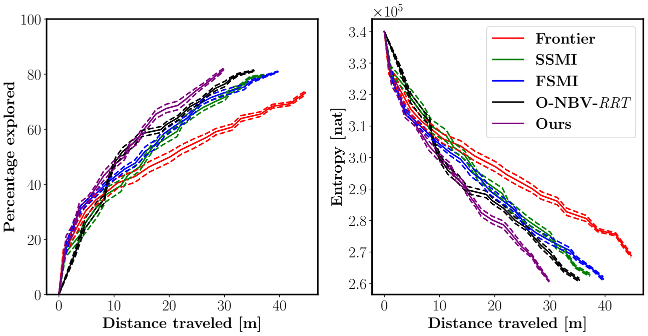


Fig. 4: 2-D simulated active mapping performance compared among various methods. The results are averaged over 10 randomly generated environments with 3 random starting positions. The dashed lines represent ±0.25σ deviation from the mean.

RRT (O-NBV-*RRT*) [10]. In Sec. IV-A, we perform 2-D active mapping using the proposed method alongside the baselines in a set of 10 randomly generated 2-D environments. Sec. IV-B contains large-scale active mapping using OctoMap representation [20] in a simulated 3-D Unity environment. Lastly, in Sec. IV-C we demonstrate the performance of our method in a real-world environment using a wheeled ground robot.

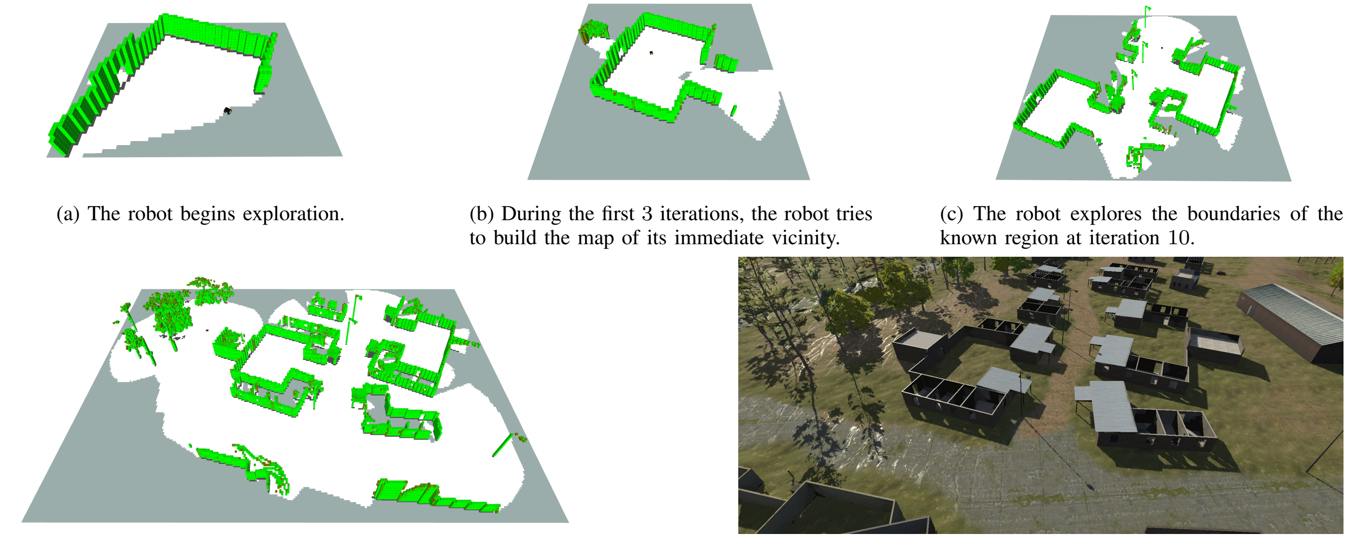
Across all experiments, each method uses the log-odds mapping from range measurements [1, Ch. 9] but selects robot trajectories based on their own criteria. Moreover, during each planning step for 2-D exploration, we identify the frontiers using edge detection over the most likely map at time t, i.e. the mode of $p_t(\mathbf{m})$. Then, we cluster the frontier map cells by detecting the connected components. We plan a path from robot pose $\mathbf{X}_t$ to the center of each frontier using A^* graph search to produce candidate paths for FSMI and SSMI. For Frontier, we pick the path that maximizes the ratio between the frontier size and path length. For O-NBV-*RRT*, we pick the goal of the same path used by Frontier, but instead plan an *RRT* path towards the goal. Our method also uses the same path used by Frontier as the initial trajectory. For exploration in 3-D environments, we first project the most likely OctoMap at time t onto the ground level in order to derive a 2-D occupancy map and proceed with similar steps as in 2-D exploration. We selected maximum size of neighboring viewpoints $\xi_{\max} = 2$, maximum number of gradient ascent iterations $n_{\max} = 50$, step size $l = 10$, collision penalty weight of $\gamma_c = 5 \times 10^{-4}$, and $\gamma_q = 1$. For Γ in (16), we use a diagonal matrix with $[1, 1, 0.1]$ as the elements on the diagonal.

A. 2-D Active Mapping in Simulation

Here we compare our proposed method to Frontier, FSMI, SSMI, and O-NBV-*RRT*. The experiments are performed in 10 randomly generated 2-D environments with dimensions $60m \times 60m$ and 3 random starting positions for each instance. We consider a robot equipped with a LiDAR sensor of range $10m$ and 90° field of view where each measurement beam is added with Gaussian noise of $\mathcal{N}(0, 0.1)$. Fig. 4 shows the 2-D simulation results among all methods. We witness superior exploration performance of our proposed method, which can be attributed to the fact that our method applies occlusion-aware optimization for each pose in a candidate trajectory; resulting in high utilization of information gathering opportunities. The subpar performance of O-NBV-*RRT* can be caused by the fact that O-NBV-*RRT* does not account for sensor noise during pose optimization. Robot trajectories obtained by Frontier, FSMI, and SSMI are computed using A^* search, and no further optimization is performed over single robot poses along the trajectory; resulting in a very restricted set of candidate solutions.

B. Exploration in 3-D Unity Simulation

In this part, we test our proposed method in a photo-realistic 3-D Unity simulated environment, shown in Fig. 5e. We use a Husky robot equipped with a depth camera that provides a 3-D point cloud used for building an OctoMap. OctoMap [20] is occupancy grid mapping method based on OcTree data structure and the log-odds technique that provides a scalable way to store the 3-D map for large environments through compressing the map cells with similar occupancy probability. For experiments in 3-D environments, we operate our proposed method in two different modes. In the first mode (Exp-2D), we project both the OctoMap and 3-D point clouds onto the $z = 0$ surface, obtaining a 2-D representation of the map and the observations. Then we compute informative trajectories similar to the 2-D active mapping in Sec. IV-A. By executing the optimal trajectory, we receive new 3-D point cloud observations that incrementally update the OctoMap. The first mode of active mapping only requires ray-tracing in a 2-D grid map, saving computation time while trading for accuracy in evaluating mutual information. In the second mode (Exp-3D), we directly use the 3-D sensor model for ray-tracing within the OctoMap. This mode of operation is expected to evaluate mutual information more accurately since the exact sensor model and 3-D occlusion have been taken into account. Fig. 5 shows several exploration iterations of active mapping process using Exp-3D. Furthermore, we compare our method with FSMI, where run length encoding is utilized for accelerated evaluation of the SMI over an OcTree. Fig. 6. (top) shows simulation results for experiments in the 3-D Unity environment for 20 exploration iterations. We observe Exp-2 finds a good balance between exploration efficiency and computation time; while Exp-3D



(a) The robot begins exploration.

(b) During the first 3 iterations, the robot tries to build the map of its immediate vicinity.

(c) The robot explores the boundaries of the known region at iteration 10.

(d) OctoMap after 20 exploration iterations

(e) Photo-realistic Unity simulation environment

Fig. 5: Time lapse of 3-D active mapping using the proposed method in simulated Unity environment. The robot receives depth measurements using an RGBD camera, and incrementally builds an OctoMap as it explores the unknown environment.

TABLE I: Clearance from obstacles compared between Frontier and our proposed method averaged over 20 exploration iterations.

Algorithm	3-D Unity Simulation	Real World
Frontier	1.2 m	0.48 m
Exp-2D	***3.9** m*	1.7 m
Exp-3D	3.4 m	***1.84** m*

is slightly more efficient in terms of distance traveled, it takes longer computation time to perform exploration compared to Exp-2D. This comes from the fact that the objects in the simulation have a uniform profile in the z direction, hence the information in the 2-D and 3-D maps are almost the same. TABLE. I (middle column) compares the average clearance from the obstacles among different methods. Compared to Frontier, it is clear that the proposed method chooses informative trajectories that avoid approaching obstacles.

C. Real-world Experiments

We deployed Exp-2D and Exp-3D on a ground wheeled robot to autonomously explore an indoor environment. The robot was equipped with an NVIDIA Xavier NX GPU, a Hokuyo UST-10LX LiDAR, and an Intel RealSense D435i RGBD camera. We implemented our software stack using *Robot Operating System* (ROS) [21]. Robot localization was carried out using ICP scan matching of LiDAR measurements [22]. 3-D point clouds from the depth images were used to build an OctoMap. The complete implementation was able to update the OctoMap every $0.11s$ on average. Fig. 6. (bottom) and TABLE. I (right column) show the performance of active mapping for 20 exploration iterations and the average clearance from obstacles, respectively. The real-world experiments confirm the findings in the simulations in terms of the efficiency of the proposed method.

V. Conclusion

This paper developed a differentiable approximation of the Shannon mutual information between a probabilistic occupancy grid map and range sensor measurements. Our formulation enables gradient-based optimization of informative occlusion-aware sensing trajectories in 3-D and allows the inclusion of additional differentiable penalty terms, such as collision cost. We demonstrated in simulated and real-world experiments that our method outperforms the state-of-the-art techniques due to its ability to optimize the sensing views in continuous space. In a future work, we will investigate

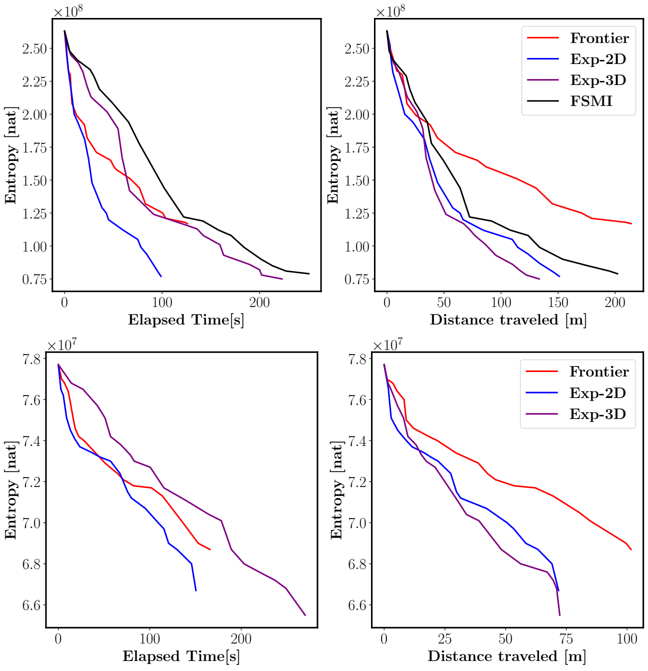


Fig. 6: 3-D active mapping performance compared between Frontier, FSMI, and our method for 20 exploration iterations. Exp-2D uses a projected 2-D perception model for planning, while Exp-3D performs ray-tracing directly on the OctoMap. Top row: Unity simulation environment. Bottom row: Real-world experiments.

extending the proposed method to multiple agents, where we expect gradient-based optimization to also be significantly more efficient than discrete space search.

Appendix A Proof of Prop. 1

Cond. (1) and (2) effectively state that for any pair of viewpoints $\mathbf{V} \in \hat{\mathcal{G}}(\mathbf{X}_i)$ and $\mathbf{U} \in \hat{\mathcal{G}}(\mathbf{X}_j)$, the two sets of map cells inside the FOVs of $\mathbf{V}$ and $\mathbf{U}$ do not intersect. This is true since the spaces inside $\mathbf{V}\mathcal{F}$ and $\mathbf{U}\mathcal{F}$ are always subsets of $\mathbb{U}(\mathbf{X}_i)$ and $\mathbb{U}(\mathbf{X}_j)$, respectively, while Cond. (2) states that $\mathbb{U}(\mathbf{X}_i) \cap \mathbb{U}(\mathbf{X}_j) = \emptyset$. Consequently, the observations made from $\mathbf{V}$ and $\mathbf{U}$ are independent random variables, resulting in the following decomposition of the SMI:

$$\begin{aligned} &I(\mathbf{m}; \mathbf{z}_v, \mathbf{z}_u \mid \mathbf{V}, \mathbf{U}, \mathcal{H}_t) = \\ &\qquad I(\mathbf{m}; \mathbf{z}_v \mid \mathbf{V}, \mathcal{H}_t) + I(\mathbf{m}; \mathbf{z}_u \mid \mathbf{U}, \mathcal{H}_t). \end{aligned}$$

Following Cond. (2), the above decomposition can be applied for any set of viewpoints $\mathcal{V} := \mathbf{V}_{t+1:t+T}$ where $\mathbf{V}_{t+\tau} \in \hat{\mathcal{G}}(\mathbf{X}_{t+\tau})$, $\tau \in \{1, \ldots, T\}$. Hence we have:

$$\begin{aligned} &\sum_{\tau=1}^{T} \sum_{\mathcal{V} \in \mathcal{G}^T} A_{\mathcal{V}}(\mathbf{X}_{t+1:t+T}) I(\mathbf{m}; \mathbf{z}_{t+\tau} \mid \mathbf{V}_{t+\tau}, \mathcal{H}_t) = \\ &\sum_{\tau=1}^{T} \tilde{I}(\mathbf{m}; \mathbf{z}_{t+\tau} \mid \mathbf{X}_{t+\tau}, \mathcal{H}_t) \sum_{\mathcal{V} - \mathbf{V}_{t+\tau} \in \mathcal{G}^{T-1}} \frac{A_{\mathcal{V}}(\mathbf{X}_{t+1:t+T})}{\alpha_{\mathbf{V}_{t+\tau}}(\mathbf{X}_{t+\tau})}. \end{aligned}$$

Based on the definition of $A_{\mathcal{V}}(\mathbf{X}_{t+1:t+T})$, the inner sum is equal to 1, which yields the expression in (8). □

Appendix B Proof of Prop. 2

Since the approximate SMI in (6) is linear with respect to $\alpha_{\mathbf{V}}(\mathbf{X})$ terms, the overall gradient computation can be reduced to a weighted sum of individual gradients of $\alpha_{\mathbf{V}}(\mathbf{X})$ with respect to robot pose $\mathbf{X}$. Also, it is only needed to compute gradients for viewpoints where $\upsilon(\delta(\boldsymbol{\xi}_{\mathbf{X},\mathbf{V}})) = 1$ since the rest of the viewpoints do not affect the derivations:

$$\begin{aligned} &[\nabla_{\boldsymbol{\psi}} \alpha_{\mathbf{V}}(\mathbf{X} \exp{(\hat{\boldsymbol{\psi}})})]|_{\boldsymbol{\psi}=0}^{\top} = \\ &\frac{\frac{\partial \cos \delta(\boldsymbol{\xi}_{\mathbf{X} \exp{(\hat{\boldsymbol{\psi}})}, \mathbf{V}})}{\partial \boldsymbol{\psi}} \eta(\mathbf{X}) - (1 + \cos \delta(\boldsymbol{\xi}_{\mathbf{X},\mathbf{V}})) \frac{\partial \eta(\mathbf{X} \exp{(\hat{\boldsymbol{\psi}})})}{\partial \boldsymbol{\psi}}}{\eta^2(\mathbf{X})}, \end{aligned}$$

where $\eta(\mathbf{X})$ is defined in (17). Both partial derivations in the numerator require computing $\frac{\partial \cos \delta(\boldsymbol{\xi}_{\mathbf{X} \exp{(\hat{\boldsymbol{\psi}})}, \mathbf{V}})}{\partial \boldsymbol{\psi}}$. Applying the chain rule, we have:

$$\begin{aligned} \frac{\partial \cos \delta(\boldsymbol{\xi}_{\mathbf{X} \exp{(\hat{\boldsymbol{\psi}})}, \mathbf{V}})}{\partial \boldsymbol{\psi}} &= \left.\frac{\partial \cos \delta}{\partial \delta}\right|_{\delta=\delta(\boldsymbol{\xi}_{\mathbf{X},\mathbf{V}})} \\ \times \left.\frac{\partial \delta(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}}\right|_{\boldsymbol{\xi}=\boldsymbol{\xi}_{\mathbf{X},\mathbf{V}}} &\left.\frac{\partial \log(\exp(-\hat{\boldsymbol{\psi}})\mathbf{X}^{-1}\mathbf{V})^{\vee}}{\partial \boldsymbol{\psi}}\right|_{\boldsymbol{\psi}=0}. \end{aligned}$$

The first two partial derivatives can be obtained via differentiation in $\mathbb{R}$ and $\mathbb{R}^6$, respectively. The last partial derivative can be obtained via applying small perturbation $\boldsymbol{\psi}$ in the robot frame:

$$\left.\frac{\partial \log(\exp(-\hat{\boldsymbol{\psi}})\mathbf{X}^{-1}\mathbf{V})^{\vee}}{\partial \boldsymbol{\psi}}\right|_{\boldsymbol{\psi}=0} = \mathcal{J}_R(-\boldsymbol{\xi}_{\mathbf{X},\mathbf{V}})\mathcal{J}_R(\boldsymbol{\xi}_{\mathbf{X}}).$$

Summing over all $\mathbf{V} \in \hat{\mathcal{G}}$ leads to the expression in (17). □

References

[1] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press Cambridge, 2005.

[2] B. J. Julian, "Mutual information-based gradient-ascent control for distributed robotics," Ph.D. dissertation, Massachusetts Institute of Technology, 2013.

[3] J. Maver and R. Bajcsy, "Occlusions as a guide for planning the next view," *IEEE transactions on pattern analysis and machine intelligence*, vol. 15, no. 5, pp. 417–433, 1993.

[4] N. Atanasov, B. Sankaran, J. Le Ny, G. J. Pappas, and K. Daniilidis, "Non-myopic view planning for active object classification and pose estimation," *IEEE Transactions on Robotics*, vol. 30, no. 5, pp. 1078–1090, 2014.

[5] H. Ma, W. Zou, S. Sun, Z. Zhu, and Z. Kang, "FOV constraint region analysis and path planning for mobile robot with observability to multiple feature points," *International Journal of Control, Automation and Systems*, vol. 19, no. 11, pp. 3785–3800, 2021.

[6] S. Koga, A. Asgharivaskasi, and N. Atanasov, "Active Exploration and Mapping via Iterative Covariance Regulation over Continuous SE(3) Trajectories," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 2735–2741.

[7] R. Rocha, J. Dias, and A. Carvalho, "Cooperative multi-robot systems: A study of vision-based 3-d mapping using information theory," *Robotics and Autonomous Systems*, vol. 53, no. 3-4, pp. 282–311, 2005.

[8] B. Charrow, G. Kahn, S. Patil, S. Liu, K. Goldberg, P. Abbeel, N. Michael, and V. Kumar, "Information-theoretic planning with trajectory optimization for dense 3d mapping." in *Robotics: Science and Systems*, vol. 11, 2015, pp. 3–12.

[9] B. Charrow, S. Liu, V. Kumar, and N. Michael, "Information-theoretic mapping using Cauchy-Schwarz quadratic mutual information," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2015.

[10] D. Deng, R. Duan, J. Liu, K. Sheng, and K. Shimada, "Robotic exploration of unknown 2d environment using a frontier-based automatic-differentiable information gain measure," in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, 2020, pp. 1497–1503.

[11] D. Deng, Z. Xu, W. Zhao, and K. Shimada, "Frontier-based automatic-differentiable information gain measure for robotic exploration of unknown 3d environments," *arXiv preprint arXiv:2011.05288*, 2020.

[12] S. Shen, N. Michael, and V. Kumar, "Stochastic differential equation-based exploration algorithm for autonomous indoor 3d exploration with a micro-aerial vehicle," *The International Journal of Robotics Research (IJRR)*, vol. 31, no. 12, pp. 1431–1444, 2012.

[13] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, 1997, pp. 146–151.

[14] Z. Zhang, T. Henderson, V. Sze, and S. Karaman, "Fsmi: Fast computation of Shannon mutual information for information-theoretic mapping," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2019, pp. 6912–6918.

[15] A. Asgharivaskasi and N. Atanasov, "Active Bayesian Multi-class Mapping from Range and Semantic Segmentation Observations," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2021.

[16] J. Knuth and P. Barooah, "Collaborative 3d localization of robots from relative pose measurements using gradient descent on manifolds," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2012.

[17] N. Mizuno, K. Ohno, R. Hamada, H. Kojima, J. Fujita, H. Amano, T. Westfechtel, T. Suzuki, and S. Tadokoro, "Enhanced path smoothing based on conjugate gradient descent for firefighting robots in petrochemical complexes," *Advanced Robotics*, vol. 33, no. 14, pp. 687–698, 2019.

[18] B. J. Julian, S. Karaman, and D. Rus, "On mutual information-based control of range sensing robots for mapping applications," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2013.

[19] T. D. Barfoot, *State Estimation for Robotics*. Cambridge University Press, 2017.

[20] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, 2013, software available at https://octomap.github.io.

[21] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng, *et al.*, "ROS: an open-source Robot Operating System," in *ICRA Workshop on Open-source Software*, 2009.

[22] A. Censi, "An ICP variant using a point-to-line metric," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2008, pp. 19–25.