Semantic OcTree Mapping and Shannon Mutual Information Computation for Robot Exploration

Arash Asgharivaskasi, *Student Member, IEEE,* Nikolay Atanasov, *Member, IEEE*

***Abstract*—Autonomous robot operation in unstructured and unknown environments requires efficient techniques for mapping and exploration using streaming range and visual observations. Information-based exploration techniques, such as Cauchy-Schwarz quadratic mutual information (CSQMI) and fast Shannon mutual information (FSMI), have successfully achieved active binary occupancy mapping with range measurements. However, as we envision robots performing complex tasks specified with semantically meaningful concepts, it is necessary to capture semantics in the measurements, map representation, and exploration objective. This work presents Semantic octree mapping and Shannon Mutual Information (SSMI) computation for robot exploration. We develop a Bayesian multi-class mapping algorithm based on an octree data structure, where each voxel maintains a categorical distribution over semantic classes. We derive a closed-form efficiently-computable lower bound of the Shannon mutual information between a multi-class octomap and a set of range-category measurements using semantic run-length encoding of the sensor rays. The bound allows rapid evaluation of many potential robot trajectories for autonomous exploration and mapping. We compare our method against state-of-the-art exploration techniques and apply it in a variety of simulated and real-world experiments.**

***Index Terms*—View Planning for SLAM, Reactive and Sensor-Based Planning, Vision-Based Navigation.**

I. Introduction

Accurate modeling, real-time understanding, and efficient storage of a robot's environment are key capabilities for autonomous operation. Occupancy grid mapping [1], [2] is a widely used, simple, yet effective, technique for distinguishing between traversable and occupied space surrounding a mobile robot. OctoMap [3] is an extension of occupancy grid mapping, introducing adaptive resolution to improve the memory usage and computational cost of generating 3-D maps of large environments. Delegating increasingly sophisticated tasks to autonomous robots requires augmenting traditional geometric models with semantic information about the context and object-level structure of the environment [4]–[6]. Robots also are increasingly expected to operate in unknown environments, with little to no prior information, in applications such as disaster response, environmental monitoring, and reconnaissance. This calls for algorithms allowing robots to autonomously explore unknown environments and construct low-uncertainty metric-semantic maps in real-time, while taking collision and visibility constraints into account.

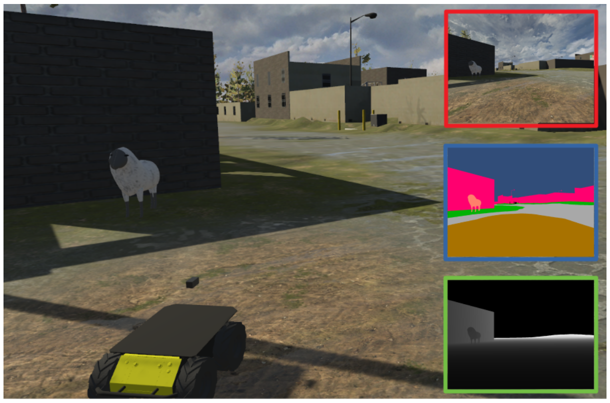


Fig. 1: A robot autonomously explores an unknown environment using an RGBD sensor and a semantic segmentation algorithm.

This paper considers the active metric-semantic mapping problem, requiring a robot to explore and map an unknown environment, relying on streaming distance and object category observations, e.g., generated by semantic segmentation over RGBD images [7]. See Fig. 1 as an illustration of the sensor data available for mapping and exploration. We extend information-theoretic active mapping techniques [8]–[10] from binary to multi-class environment representations. Our approach introduces a Bayesian multi-class mapping procedure which maintains a probability distribution over semantic categories and updates it via a probabilistic range-category perception model. Our main **contributions** are:

1) a Bayesian multi-class octree mapping approach,
2) a closed-form efficiently-computable lower bound for the Shannon mutual information between a multi-class octree map and a set of range-category measurements,
3) efficient C++ implementation achieving real-time high-accuracy performance onboard physical robot systems in 3-D real-world experiments[1].

Unlike a uniform-resolution grid map, our semantic OctoMap enables efficient mutual information evaluation via run-length encoding of the range-category observations. As a result, the informativeness of many potential robot trajectories may be evaluated to (re-)select one that leads to the best trade-off between map uncertainty reduction and motion cost. Unlike traditional class-agnostic exploration methods, our model and information measure capture the uncertainty of different semantic classes, leading to faster and more accurate exploration. The proposed approach relies on general range and class measurements and general pose kinematics, making it suitable for either ground or aerial robots, equipped with either camera or LiDAR sensors, exploring either indoor or outdoor environments.

We gratefully acknowledge support from ARL DCIST CRA W911NF-17-2-0181 and NSF FRR CAREER 2045945. The authors are with the Department of Electrical and Computer Engineering, University of California San Diego, CA 92093, USA {aasghari,natanasov}@eng.ucsd.edu.

[1] Open-source software and videos supplementing this paper are available at https://arashasgharivaskasi-bc.github.io/SSMI_webpage.

We name our method *Semantic octree mapping and Shannon Mutual Information* (SSMI) computation for robot exploration. This paper is an extended version of our previous conference paper [11] enabling its application in large-scale 3-D environments. Direct application of the regular-grid method in the conference version to 3-D environments faces several challenges, including memory inefficiency, because 3-D environments are predominantly made up of free or unknown space, and information computation inefficiency due to naïve summation over all voxels visited by sensor rays. We introduce (a) an octree representation for Bayesian multi-class mapping, performing octree ray-tracing and leaf node merging using semantic category distributions, (b) semantic run-length encoding (SRLE) for sensor rays which enables efficient mutual information computation between a multi-class octree and range-category measurements, and (c) a comprehensive set of experiments and comparisons with state-of-the-art techniques in 2-D and 3-D environments. We demonstrate the effectiveness of SSMI in real-world mapping and autonomous exploration experiments performed in real-time onboard different mobile robot platforms operating in indoor and outdoor environments.

II. Related Work

Frontier-based exploration [12] is a seminal work that highlighted the utility of autonomous exploration and active mapping. It inspired methods [13], [14] that rely on geometric features, such as the boundaries between free and unknown space (frontiers) and the volume that would be revealed by new sensor observations. Due to their intuitive formulation and low computational requirements, geometry-based exploration methods continue to be widely employed in active perception. Recent works include semantics-assisted indoor exploration [15], hex-decomposition-based coverage planning [16], and Laplace potential fields for safe outdoor exploration [17]. More related to our work, receding-horizon "next-best-view" planning [18] presents an active octree occupancy mapping method which executes trajectories built from a random tree whose quality is determined by the amount of unmapped space that can be explored. Similarly, the graph-based exploration methods of [19] and [20] use local random trees in free space to sample candidate viewpoints for exploration, while a global graph maintains the connections among the frontiers in the map. Cao *et al.* [16], [21] introduced hierarchical active 3D coverage and reconstruction which computes a coarse coverage path at the global scale followed by a local planner that ensures collision avoidance via a high-resolution path. As shown by Corah and Michael [22], coverage-based exploration strategies can be formulated as mutual-information maximization policies in the absence of sensor noise. However, in many real-world circumstances sensor measurements are corrupted by non-trivial noise, reducing the effectiveness of geometric exploration methods that do not capture probabilistic uncertainty. For example, due to the domain shift between the training data and the test environment, utilizing a pre-trained semantic segmentation model in the mapping process requires accounting for measurement uncertainty in the exploration policy.

The work by Elfes [23] is among the first to propose an information-based utility function for measuring and minimizing map uncertainty. Information-based exploration strategies have been devised for uncertainty minimization in robot localization or environment mapping [24]–[26]. Information-theoretic objectives, however, require integration over the potential sensor measurements, limiting the use of direct numerical approximations to short planning horizons. Kollar and Roy [27] formulated active mapping using an extended Kalman filter and proposed a local-global optimization, leading to significant gains in efficiency for uncertainty computation and long-horizon planning. Unlike geometry-based approaches, information-theoretic exploration can be directly formulated for active simultaneous localization and mapping (SLAM) [28]–[31], aiming to determine a sensing trajectory that balances robot state uncertainty and visitation of unexplored map regions. Stachniss *et al.* [32] approximate information gain for a Rao-blackwellized particle filter over the joint state of robot pose and map occupancy. Julian *et al.* [8] prove that, for range measurements and known robot position, the Shannon mutual information is maximized over trajectories that visit unexplored areas. However, without imposing further structure over the observation model, computing the mutual information objective requires numerical integration. The need for efficient mutual information computation becomes evident in 3-D environments. Cauchy-Schwarz quadratic mutual information (CSQMI) [9] and fast Shannon mutual information (FSMI) [10] offer efficiently computable closed-form objectives for active occupancy mapping with range measurements. Henderson *et al.* [33] propose an even faster computation based on a recursive expression for Shannon mutual information in continuous maps.

The recent success of machine learning methods of perception has motivated learning autonomous exploration policies. Chen *et al.* [34] attempt to bridge the sim2sim and sim2real gaps via graph neural networks and deep reinforcement learning. This enables decision-making over graphs containing relevant exploration information which is provided by human experts in order to predict a robot's optimal sensing action in belief space. Lodel *et al.* [35] introduce a deep reinforcement learning policy which recommends next best view that maximizes information gain via defining mutual information as the training reward. Zwecher *et al.* [36] employ deep reinforcement learning to find an exploration policy that plans collision-free coverage paths, while another neural network provides a predicted full map given the partially observed environment. Zhang *et al.* [37] propose a multi-agent reinforcement learning exploration method where regions of interest, free space, and robots are represented as graph nodes, and hierarchical-hops graph neural networks (H2GNN) are used to identify key information in the environment. Related to multi-robot exploration, the authors in [38] utilize an actor-critic strategy to map an unknown environment, where Voronoi partitioning divides the exploration regions among the robots. As this paper demonstrates, incorporating semantic uncertainty in addition to geometric information in the exploration process can be beneficial. Additionally, using Shannon mutual information as an objective function may help train more generalizable exploration policies because it mitigates the need for training sensor-specific models. Hence, the techniques proposed in this paper are complementary to learning approaches and can provide robustness to measurement uncertainty and domain shift caused by sensor and operational condition variations.

Active semantic mapping has recently attracted much attention due to the proliferation of fast object detection and semantic segmentation algorithms implemented on mobile robot platforms. The authors in [39] use a two-layer architecture, where the knowledge representation layer provides a belief over the environment state to the action layer, which subsequently chooses an action to gather information or execute a task. The work in [40] presents a semantic exploration policy which takes an occluded semantic point cloud of an object, finds a match in a database to estimate the full object dimensions, and then generates candidate next observation poses to reconstruct the object. The next best view is computed via a volumetric information gain metric that computes visible entropy from a candidate pose. The semantic map used in this paper is a collection of bounding boxes around objects. Active semantic mapping has also been employed to develop sample-efficient deep learning methods. Blum *et al.* [41] propose an active learning method for training semantic segmentation networks where the novelty (epistemic uncertainty) of the input images is estimated as the distance from the training data in the embedding space, while a path planning method maximizes novelty of future input images along the planned trajectory, assuming novel images are spatially correlated. Georgakis *et al.* [42] actively train a hierarchical semantic map generation model that predicts occupancy and semantics given occluded input. The authors use an ensemble of map generation models in order to predict epistemic uncertainty of the predicted map. The uncertainty is then used to choose trajectories for actively training the model with new images that differ the most with the training data of the current model. SSMI distinguishes itself from the aforementioned works by introducing a dense Bayesian multi-class mapping with a closed-form uncertainty measure, as opposed to sampling-based uncertainty estimation. Moreover, our information-theoretic objective function directly models sensor noise specifications, unlike volumetric information gain.

This paper is most related to CSQMI [9] and FSMI [10] in that it develops a closed-form expression for mutual information. However, instead of a binary map and range-only measurements, our formulation considers a multi-class map with Bayesian updates using range-category measurements. Since the same occupancy map can be derived from many different multi-class maps, the information associated with various object classes will fail to be captured if we solely rely on occupancy information, as the case in CSQMI and FSMI. Therefore, we expect to perform exploration more efficiently by using the multi-class perception model, and consequently, expanding the notion of uncertainty to multiple classes.

III. Problem Statement

Consider a robot with pose $\mathbf{X}_t \in SE(3)$ at time t and deterministic discrete-time kinematics:

$$\mathbf{X}_t := \begin{bmatrix} \mathbf{R}_t & \mathbf{p}_t \\ \mathbf{0}^\top & 1 \end{bmatrix}, \qquad \mathbf{X}_{t+1} = \mathbf{X}_t \exp\left(\tau \hat{\mathbf{u}}_t\right), \tag{1}$$

where $\mathbf{R}_t \in SO(3)$ is the robot orientation, $\mathbf{p}_t \in \mathbb{R}^3$ is the robot position, τ is the time step, and $\mathbf{u}_t := [\mathbf{v}_t^\top, \boldsymbol{\omega}_t^\top]^\top \in \mathcal{U} \subset \mathbb{R}^6$ is the control input, consisting of linear velocity $\mathbf{v}_t \in \mathbb{R}^3$ and angular velocity $\boldsymbol{\omega}_t \in \mathbb{R}^3$. The function $\hat{(\cdot)} : \mathbb{R}^6 \to \mathfrak{se}(3)$ maps vectors in $\mathbb{R}^6$ to the Lie algebra $\mathfrak{se}(3)$. See [43, Chapter 7] for a definition of the Lie groups $SO(3)$ and $SE(3)$ and the corresponding Lie algebras $\mathfrak{so}(3)$ and $\mathfrak{se}(3)$. The robot is navigating in an environment consisting of a collection of disjoint sets $\mathcal{E}_k \subset \mathbb{R}^3$, each associated with a semantic category $k \in \mathcal{K} := \{0, 1, \ldots, K\}$. Let $\mathcal{E}_0$ denote free space, while each $\mathcal{E}_k$ for $k > 0$ represents a different category, such as building, vegetation, terrain (see Fig. 1).

We assume that the robot is equipped with a sensor that provides information about the distance to and semantic categories of surrounding objects along a set of rays $\{\boldsymbol{\eta}_b\}_b$, where b is the ray index, $\boldsymbol{\eta}_b \in \mathbb{R}^3$ with $\|\boldsymbol{\eta}_b\|_2 = r_{max}$, and $r_{max} > 0$ is the maximum sensing range.

Definition 1. A *sensor observation* at time t from robot pose $\mathbf{X}_t$ is a collection $\mathcal{Z}_t := \{\mathbf{z}_{t,b}\}_b$ of range and category measurements $\mathbf{z}_{t,b} := (r_{t,b}, y_{t,b}) \in \mathbb{R}_{\geq 0} \times \mathcal{K}$, acquired along the sensor rays $\mathbf{R}_t \boldsymbol{\eta}_b$ with $\boldsymbol{\eta}_b \in \{\boldsymbol{\eta}_b\}_b$ at robot position $\mathbf{p}_t$.

Such information may be obtained by processing the observations of an RGBD camera or a Lidar with a semantic segmentation algorithm [7]. Fig. 1 shows an example where each pixel in the RGB image corresponds to one sensor ray $\boldsymbol{\eta}_b$, while its corresponding values in the semantic segmentation and the depth images encode category $y_{t,b}$ and range $r_{t,b}$, respectively. The goal is to construct a multi-class map $\mathbf{m}$ of the environment based on the labeled range measurements. We model $\mathbf{m}$ as a grid of cells $i \in \mathcal{I} := \{1, \ldots, N\}$, each labeled with a category $m_i \in \mathcal{K}$. In order to model noisy sensor observations, we consider a probability density function (PDF) $p(\mathcal{Z}_t \mid \mathbf{m}, \mathbf{X}_t)$. This observation model allows integrating the measurements into a probabilistic map representation using Bayesian updates. Let $p_t(\mathbf{m}) := p(\mathbf{m} \mid \mathcal{Z}_{1:t}, \mathbf{X}_{1:t})$ be the probability mass function (PMF) of the map $\mathbf{m}$ given the robot trajectory $\mathbf{X}_{1:t}$ and observations $\mathcal{Z}_{1:t}$ up to time t. Given a new observation $\mathcal{Z}_{t+1}$ obtained from robot pose $\mathbf{X}_{t+1}$, the Bayesian update to the map PMF is:

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

We assume that the robot pose is known and omit the dependence of the map distribution and the observation model on it for brevity. We consider the following problem.

Problem. Given a prior map PMF $p_t(\mathbf{m})$ at time t and a finite planning horizon T, maximize the ratio:

$$\max_{\mathbf{u}_{t:t+T-1}} \frac{I\left(\mathbf{m}; \mathcal{Z}_{t+1:t+T} \mid \mathcal{Z}_{1:t}\right)}{J(\mathbf{X}_{t:t+T-1}, \mathbf{u}_{t:t+T-1})} \;\; \text{subject to} \;\; (1), (2), \tag{3}$$

of the mutual information $I\left(\mathbf{m}; \mathcal{Z}_{t+1:t+T} \mid \mathcal{Z}_{1:t}\right)$ between the map $\mathbf{m}$ and future sensor observations $\mathcal{Z}_{t+1:t+T}$ to the motion cost $J(\mathbf{X}_{t:t+T-1}, \mathbf{u}_{t:t+T-1})$ of the control sequence $\mathbf{u}_{t:t+T-1}$.

The definitions of the mutual information and motion cost terms in (3) are:

$$I\left(\mathbf{m}; \mathcal{Z}_{t+1:t+T} | \mathcal{Z}_{1:t}\right) := \sum_{\mathbf{m} \in \mathcal{K}^N} \int \cdots \int p(\mathbf{m}, \mathcal{Z}_{t+1:t+T} | \mathcal{Z}_{1:t})$$

$$\times \log \frac{p(\mathbf{m}, \mathcal{Z}_{t+1:t+T} \mid \mathcal{Z}_{1:t})}{p(\mathbf{m}|\mathcal{Z}_{1:t}) p(\mathcal{Z}_{t+1:t+T}|\mathcal{Z}_{1:t})} \prod_{\tau=1}^{T} \prod_{b} d\mathbf{z}_{t+\tau,b}, \qquad (4)$$

$$J(\mathbf{X}_{t:t+T-1}, \mathbf{u}_{t:t+T-1}) := q(\mathbf{X}_{t+T}) + \sum_{\tau=0}^{T-1} c(\mathbf{X}_{t+\tau}, \mathbf{u}_{t+\tau}),$$

where the integration in (4) is over all possible values of all sensor beams over all times $\mathbf{z}_{t+\tau,b}$, and the strictly positive terms $q(\mathbf{X})$ and $c(\mathbf{X}, \mathbf{u})$ model terminal and stage motion costs (e.g., distance traveled, elapsed time), respectively.

We develop a multi-class extension to the log-odds occupancy mapping algorithm [44, Ch. 9] in Sec. IV and derive an efficient approximation to the mutual information term in Sec. V. In Sec. VI, we present a multi-class extension of the OctoMap [3] algorithm, alongside a fast computation of mutual information over a semantic OctoMap using run-length encoding. This allows autonomous exploration of large 3-D environments by rapidly evaluating potential robot trajectories online and (re-)selecting the one that maximizes the objective in (3). Finally, in Sec. VII, we demonstrate the performance of our approach in simulated and real-world experiments.

IV. Bayesian Multi-class Mapping

This section derives the Bayesian update in (2), using a *multinomial logit model* to represent the map PMF $p_t(\mathbf{m})$ where each cell m_i of the map stores the probability of object classes in $\mathcal{K}$. To ensure that the number of parameters in the model scales linearly with the map size N, we maintain a factorized PMF over the cells:

$$p_t(\mathbf{m}) = \prod_{i=1}^{N} p_t(m_i). \qquad (5)$$

We represent the individual cell PMFs $p_t(m_i)$ over the semantic categories $\mathcal{K}$ using a vector of log odds:

$$\mathbf{h}_{t,i} := \begin{bmatrix} \log \frac{p_t(m_i=0)}{p_t(m_i=0)} & \cdots & \log \frac{p_t(m_i=K)}{p_t(m_i=0)} \end{bmatrix}^\top \in \mathbb{R}^{K+1}, \qquad (6)$$

where the free-class likelihood $p_t(m_i = 0)$ is used as a pivot. Given the log-odds vector $\mathbf{h}_{t,i}$, the PMF of cell m_i may be recovered using the softmax function $\sigma : \mathbb{R}^{K+1} \mapsto \mathbb{R}^{K+1}$:

$$p_t(m_i = k) = \sigma_{k+1}(\mathbf{h}_{t,i}) := \frac{\mathbf{e}_{k+1}^\top \exp(\mathbf{h}_{t,i})}{\mathbf{1}^\top \exp(\mathbf{h}_{t,i})}, \qquad (7)$$

where $\mathbf{e}_k$ is the standard basis vector with k-th element equal to 1 and 0 elsewhere, $\mathbf{1}$ is the vector with all elements equal to 1, and $\exp(\cdot)$ is applied elementwise to the vector $\mathbf{h}_{t,i}$. To derive Bayes rule for the log-odds $\mathbf{h}_{t,i}$, we need to specify an observation model for the range and categry measurements.

Definition 2. The *inverse observation model* of a range-category measurement $\mathbf{z}$ obtained from robot pose $\mathbf{X}$ along sensor ray $\boldsymbol{\eta}$ with respect to map cell m_i is a probability mass function $p(m_i|\mathbf{z}; \mathbf{X}, \boldsymbol{\eta})$.

The Bayesian update in (2) for $\mathbf{h}_{t,i}$ can be obtained in terms of the range-category inverse observation model, evaluated at a new measurement set $\mathcal{Z}_{t+1}$.

Proposition 1. *Let $\mathbf{h}_{t,i}$ be the log odds of cell m_i at time t. Given sensor observation $\mathcal{Z}_{t+1}$, the posterior log-odds are:*

$$\mathbf{h}_{t+1,i} = \mathbf{h}_{t,i} + \sum_{\mathbf{z} \in \mathcal{Z}_{t+1}} (\mathbf{l}_i(\mathbf{z}) - \mathbf{h}_{0,i}) \qquad (8)$$

where $\mathbf{l}_i(\mathbf{z})$ is the inverse observation model log odds:

$$\mathbf{l}_i(\mathbf{z}) := \begin{bmatrix} \log \frac{p(m_i=0|\mathbf{z})}{p(m_i=0|\mathbf{z})} & \cdots & \log \frac{p(m_i=K|\mathbf{z})}{p(m_i=0|\mathbf{z})} \end{bmatrix}^\top. \qquad (9)$$

Proof. See Appendix A. □

To complete the Bayesian multi-class mapping algorithm suggested by (8) we need a particular inverse observation model. When a sensor measurement is generated, the sensor ray continues to travel until it hits an obstacle of category $\mathcal{K} \setminus \{0\}$ or reaches the maximum sensing range r_{max}. The labeled range measurement $\mathbf{z} = (r, y)$ obtained from position $\mathbf{p}$ with orientation $\mathbf{R}$ indicates that map cell m_i is occupied if the measurement end point $\mathbf{p} + \frac{r}{r_{max}} \mathbf{R} \boldsymbol{\eta}$ lies in the cell. If m_i lies along the sensor ray but does not contain the end point, it is observed as free. Finally, if m_i is not intersected by the sensor ray, no information is provided about its occupancy. The map cells along the sensor ray can be determined by a rasterization algorithm, such as Bresenham's line algorithm [45]. We parameterize the inverse observation model log-odds vector in (9) as:

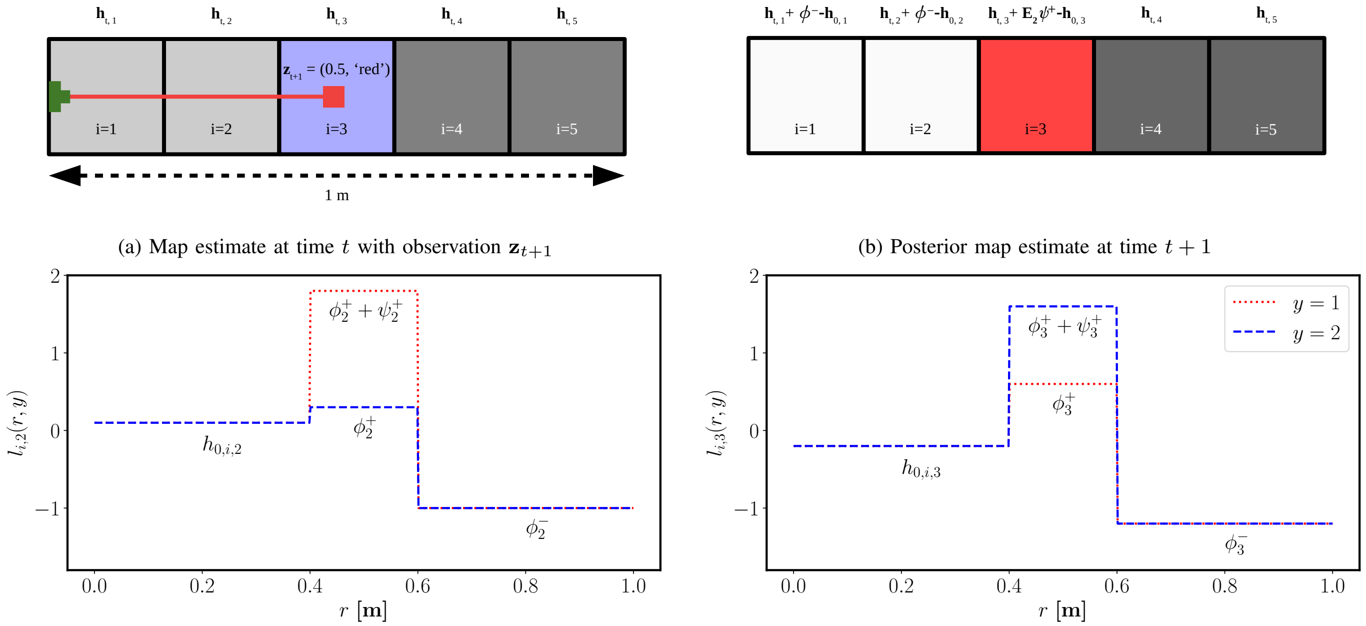
$$\mathbf{l}_i((r, y)) := \begin{cases} \boldsymbol{\phi}^+ + \mathbf{E}_{y+1} \boldsymbol{\psi}^+, & r \text{ indicates } m_i \text{ is occupied}, \\ \boldsymbol{\phi}^-, & r \text{ indicates } m_i \text{ is free}, \\ \mathbf{h}_{0,i}, & \text{otherwise}, \end{cases} \qquad (10)$$

where $\mathbf{E}_k := \mathbf{e}_k \mathbf{e}_k^\top$ and $\boldsymbol{\psi}^+, \boldsymbol{\phi}^-, \boldsymbol{\phi}^+ \in \mathbb{R}^{K+1}$ are parameter vectors, whose first element is 0 to ensure that $\mathbf{l}_i(\mathbf{z})$ is a valid log-odds vector. This parameterization leads to an inverse observation model $p(m_i = k|\mathbf{z}) = \sigma_{k+1}(\mathbf{l}_i(\mathbf{z}))$, which is piecewise constant along the sensor ray. Fig. 2 illustrates our Bayesian multi-class mapping method.

To compute the mutual information between an observation sequence $\mathcal{Z}_{t+1:t+T}$ and the map $\mathbf{m}$ in the next section, we will also need the PDF of a range-category measurement $\mathbf{z}_{\tau,b} \in \mathcal{Z}_{t+1:t+T}$ conditioned on $\mathcal{Z}_{1:t}$. Let $\mathcal{R}_{\tau,b}(r) \subset \mathcal{I}$ denote the set of map cell indices along the ray $\mathbf{R}_\tau \boldsymbol{\eta}_b$ from robot position $\mathbf{p}_\tau$ with length r. Let $\gamma_{\tau,b}(i)$ denote the distance traveled by the ray $\mathbf{R}_\tau \boldsymbol{\eta}_b$ within cell m_i and $i^*_{\tau,b} \in \mathcal{R}_{\tau,b}(r)$ denote the index of the cell hit by $\mathbf{z}_{\tau,b}$. We define the PDF of $\mathbf{z}_{\tau,b} = (r, y)$ conditioned on $\mathcal{Z}_{1:t}$ as:

$$p(\mathbf{z}_{\tau,b}|\mathcal{Z}_{1:t}) = \frac{p_t(m_{i^*_{\tau,b}} = y)}{\gamma_{\tau,b}(i^*_{\tau,b})} \prod_{i \in \mathcal{R}_{\tau,b}(r) \setminus \{i^*_{\tau,b}\}} p_t(m_i = 0). \qquad (11)$$

This definition states that the likelihood of $\mathbf{z}_{\tau,b} = (r, y)$ at time t depends on the likelihood that the cells m_i along the



(a) Map estimate at time t with observation $\mathbf{z}_{t+1}$

(b) Posterior map estimate at time $t+1$

(c) Inverse observation log-odds vector for $i=3$

Fig. 2: Illustration of the Bayesian multi-class mapping given a range-category observation $\mathbf{z}_{t+1} = (r_{t+1}, y_{t+1})$ for an environment with object classes of *white* (free class), *red*, and *blue*, encoded as $y=0$, $y=1$, and $y=2$, respectively: (a) Portion of the map $\mathbf{m}$ along the observation $\mathbf{z}_{t+1}$. Each cell has a multi-class log-odds vector $\mathbf{h}_{t,i} \in \mathbb{R}^3$ for $i \in \{1,\ldots,5\}$ at time t. The cell brightness encodes the occupancy probability, while the cell color represents the most likely category; (b) Map estimate at time $t+1$ after update with $\mathbf{z}_{t+1}$. Note how each multi-class log-odds vector changes based on the inverse observation model $\mathbf{l}_i(\mathbf{z}_{t+1})$; (c) Second and third elements of the inverse observation log-odds vector $\mathbf{l}_i(\mathbf{z}_{t+1})$ for $i=3$ as a function of range r and category y in observation $\mathbf{z}_{t+1}$. Note that the first element of $\mathbf{l}_i(\mathbf{z}_{t+1})$ is always zero.

ray $\mathbf{R}_\tau \boldsymbol{\eta}_b$ of length r are empty and the likelihood that the hit cell $m_{i^*_{\tau,b}}$ has class y. A similar model for binary observations has been used in [8]–[10].

This section described how an observation affects the map PMF $p_t(\mathbf{m})$. Now, we switch our focus to computing the mutual information between a sequence of observations $\mathcal{Z}_{t+1:t+T}$ and the multi-class occupancy map $\mathbf{m}$.

V. Informative Planning

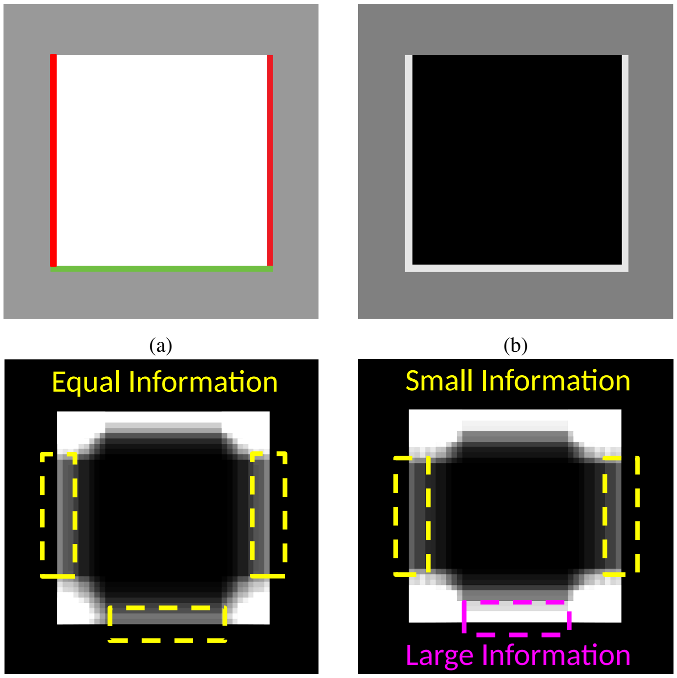
Proposition 1 allows a multi-class formulation of occupancy grid mapping, where the uncertainty of a map cell depends on the probability of each class $p_t(m_i = k)$, instead of only the binary occupancy probability $1 - p_t(m_i = 0)$. Moreover, the inverse observation model in (10) may contain different likelihoods for the different classes which can be used to prioritize the information gathering for specific classes. Fig. 3a shows an example where the estimated map of an environment with 3 classes, *free*, *class*$_1$, *class*$_2$, contains two regions with similar occupancy probability but different semantic uncertainty. In particular, the red and green walls have the same occupancy probability of 0.9, as shown in Fig. 3b, but the red region more certainly belongs to *class*$_1$ and the green region has high uncertainty between the two classes. As can be seen in Fig. 3c, the mutual information associated with a binary occupancy map cannot distinguish between the red and green regions since they both have the same occupancy probability. In contrast, the multi-class map takes into account the semantic uncertainty among different categories, as can be seen in Fig. 3d where the uncertain green region has larger mutual information than the certain red region.

These observations suggest that more accurate uncertainty quantification may be achieved using a multi-class instead of a binary perception model, potentially enabling a more efficient exploration strategy. However, computing the mutual information term in (4) is challenging because it involves integration over all possible values of the observation sequence $\mathcal{Z}_{t+1:t+T}$. Our main result is an efficiently-computable lower bound on $I(\mathbf{m}; \mathcal{Z}_{t+1:t+T} | \mathcal{Z}_{1:t})$ for range-category observations $\mathcal{Z}_{t+1:t+T}$ and a multi-class occupancy map $\mathbf{m}$. The result is obtained by selecting a subset $\underline{\mathcal{Z}}_{t+1:t+T} = \{\mathbf{z}_{\tau,b}\}_{\tau=t+1,b=1}^{t+T,B}$ of the observations $\mathcal{Z}_{t+1:t+T}$ in which the sensor rays are non-overlapping. Precisely, any pair of measurements $\mathbf{z}_{\tau,b}$, $\mathbf{z}_{\tau',b'} \in \underline{\mathcal{Z}}_{t+1:t+T}$ satisfies:

$$\mathcal{R}_{\tau,b}(r_{max}) \cap \mathcal{R}_{\tau',b'}(r_{max}) = \emptyset. \tag{12}$$

In practice, constructing $\underline{\mathcal{Z}}_{t+1:t+T}$ requires removing intersecting rays from $\mathcal{Z}_{t+1:t+T}$ to ensure that the remaining observations are mutually independent. The mutual information between $\mathbf{m}$ and $\underline{\mathcal{Z}}_{t+1:t+T}$ can be obtained as a sum of mutual information terms between single rays $\mathbf{z}_{\tau,b} \in \underline{\mathcal{Z}}_{t+1:t+T}$ and map cells m_i observed by $\mathbf{z}_{\tau,b}$. This idea is inspired by CSQMI [9] but we generalize it to multi-class observations and map.

Proposition 2. *Given a sequence of labeled range observations $\mathcal{Z}_{t+1:t+T}$, let $\underline{\mathcal{Z}}_{t+1:t+T} = \{\mathbf{z}_{\tau,b}\}_{\tau=t+1,b=1}^{t+T,B}$ be a subset*



(a) (b) (c) (d)

Fig. 3: Comparison between the information surfaces of binary and multi-class map representations. (a) Environment with three classes *free*, *class*$_1$, and *class*$_2$ where the white and gray regions represent free and unknown space, respectively, with $\mathbf{p}^{white}(m_i) = [1, 0, 0]$ and $\mathbf{p}^{gray}(m_i) = [0.3, 0.3, 0.3]$. The red and green regions have the same occupancy probability of $p(m_i = \text{occupied}) = 0.9$ but different class uncertainty, i.e., $\mathbf{p}^{red}(m_i) = [0.1, 0.8, 0.1]$ and $\mathbf{p}^{green}(m_i) = [0.1, 0.45, 0.45]$. (b) Binary occupancy map, where the intensity of each pixel is proportional to its occupancy probability, regardless of object class. (c) Occupancy mutual information surface. (d) Semantic mutual information surface. Each pixel in the information surfaces shows the value of mutual information between the map and a set of range-category observations, uniformly sampled from a 360° field of view at each pixel location.

of non-overlapping measurements that satisfy (12). *Then, the Shannon mutual information between* $\mathcal{Z}_{t+1:t+T}$ *and a multi-class occupancy map* $\mathbf{m}$ *can be lower bounded as:*

$$I(\mathbf{m};\mathcal{Z}_{t+1:t+T}|\mathcal{Z}_{1:t}) \geq I\left(\mathbf{m};\underline{\mathcal{Z}}_{t+1:t+T}|\mathcal{Z}_{1:t}\right) = \sum_{\tau=t+1}^{t+T}\sum_{b=1}^{B}\sum_{k=1}^{K}\sum_{n=1}^{N_{\tau,b}} p_{\tau,b}(n,k)C_{\tau,b}(n,k), \tag{13}$$

where $N_{\tau,b} := |\mathcal{R}_{\tau,b}(r_{max})|$,

$$p_{\tau,b}(n,k) := p_t(m_{i^*_{\tau,b}} = k) \prod_{i \in \tilde{\mathcal{R}}_{\tau,b}(n)\setminus\{i^*_{\tau,b}\}} p_t(m_i = 0),$$

$$C_{\tau,b}(n,k) := f(\boldsymbol{\phi}^+ + \mathbf{E}_{k+1}\boldsymbol{\psi}^+ - \mathbf{h}_{0,i^*_{\tau,b}}, \mathbf{h}_{t,i^*_{\tau,b}}) + \sum_{i \in \tilde{\mathcal{R}}_{\tau,b}(n)\setminus\{i^*_{\tau,b}\}} f(\boldsymbol{\phi}^- - \mathbf{h}_{0,i}, \mathbf{h}_{t,i}),$$

$$f(\boldsymbol{\phi}, \mathbf{h}) := \log\left(\frac{\mathbf{1}^\top \exp(\mathbf{h})}{\mathbf{1}^\top \exp(\boldsymbol{\phi} + \mathbf{h})}\right) + \boldsymbol{\phi}^\top \sigma(\boldsymbol{\phi} + \mathbf{h}),$$

and $\tilde{\mathcal{R}}_{\tau,b}(n) \subseteq \mathcal{R}_{\tau,b}(r_{max})$ *is the set of the first* n *map cell indices along the ray* $\mathbf{R}_\tau\boldsymbol{\eta}_b$, *i.e.,* $\tilde{\mathcal{R}}_{\tau,b}(n) := \{i \mid i \in \mathcal{R}_{\tau,b}(r), |\mathcal{R}_{\tau,b}(r)| = n, r \leq r_{max}\}$.

Algorithm 1 Information-theoretic Path Planning

```
Input: robot pose $\mathbf{X}_t$, map estimate $p_t(\mathbf{m})$
1: $\mathcal{F} = $ FINDFRONTIERS($p_t(\mathbf{m})$)
2: for $f \in \mathcal{F}$ do
3:     $\mathbf{X}_{t+1:t+T}, \mathbf{u}_{t:t+T-1} = $ PLANPATH($\mathbf{X}_t, p_t(\mathbf{m}), f$)
4:     Compute (3) over $\mathbf{X}_{t:t+T}, \mathbf{u}_{t:t+T-1}$ via (13)
5: return $\mathbf{X}^*_{t:t+T}$, $\mathbf{u}^*_{t:t+T-1}$ with the highest value
```

Proof. See Appendix B. □

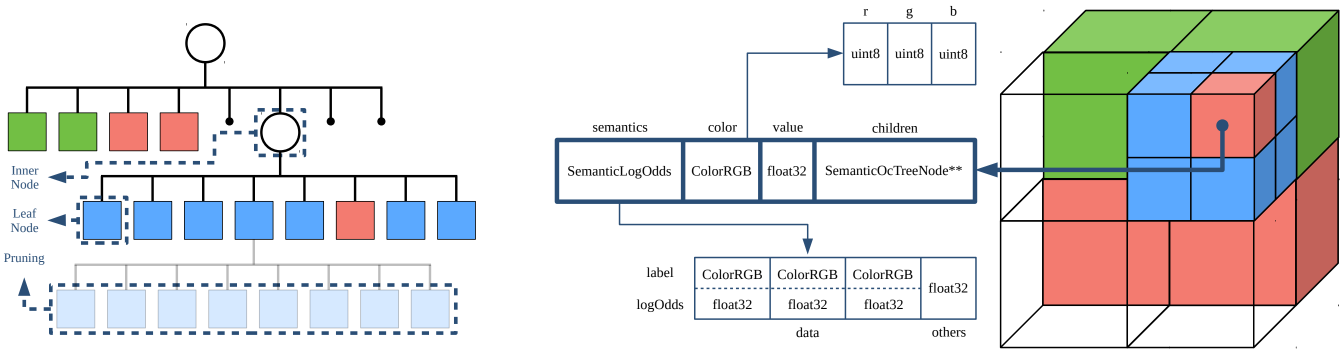
In (13), $p_{\tau,b}(n,k)$ represents the probability that the n-th map cell along the ray $\mathbf{R}_\tau\boldsymbol{\eta}_b$ belongs to object category k while all of the previous cells are free. The function $f(\boldsymbol{\phi}, \mathbf{h})$ denotes the log-ratio of the map PMF $\sigma(\mathbf{h})$ and its posterior $\sigma(\boldsymbol{\phi} + \mathbf{h})$, averaged over object categories in $\mathcal{K}$ (see (26) in Appendix B for more details). As a result, $C_{\tau,b}(n,k)$ is the sum of log-ratios for the first n cells along the ray $\mathbf{R}_\tau\boldsymbol{\eta}_b$ under the same event as the one $p_{\tau,b}(n,k)$ is associated with. Therefore, the lower bound $I\left(\mathbf{m};\underline{\mathcal{Z}}_{t+1:t+T}|\mathcal{Z}_{1:t}\right)$ is equivalent to the expectation of summed log-ratios $C_{\tau,b}(n,k)$ over all possible instantiations of the observations in $\underline{\mathcal{Z}}_{t+1:t+T}$.

Proposition 2 allows evaluating the informativeness according to (3) of any potential robot trajectory $\mathbf{X}_{t:t+T}$, $\mathbf{u}_{t:t+T-1}$. In order to perform informative planning, first, we identify the boundary between the explored and unexplored regions of the map, similar to [12]. This can be done efficiently using edge detection, for example. Then, we cluster the corresponding map cells by detecting the connected components of the boundary. Each cluster is called a *frontier*. A motion planning algorithm is used to obtain a set of pose trajectories to the map frontiers, determined from the current map PMF $p_t(\mathbf{m})$. Alg. 1 summarizes the procedure for determining a trajectory $\mathbf{X}^*_{t:t+T}$, $\mathbf{u}^*_{t:t+T-1}$ that maximizes the objective in (3), where $J(\mathbf{X}_{t:t+T-1}, \mathbf{u}_{t:t+T-1})$ is the length of the corresponding path. This kinematically feasible trajectory can be tracked by a low-level controller that takes the robot dynamics into account.

Evaluation of the mutual information lower bound in Proposition 2 can be accelerated without loss in accuracy for map cells along the observation rays that contain equal PMFs. In the next section, we investigate this property of the proposed lower bound within the context of OcTree-based representations. We begin with proposing a multi-class version of the OctoMap technique, where map cells with equal multi-class probabilities can be compressed into a larger voxel. Next, a fast semantic mutual information formula is presented based on compression of range-category ray-casts over OcTree representations.

VI. Information Computation for Semantic OctoMap Representations

Utilizing a regular-grid discretization to represent a 3-D environment has prohibitive storage and computation requirements. Large continuous portions of many real environments are unoccupied, suggesting that adaptive discretization is significantly more efficient. OctoMap [3] is a probabilistic 3-D mapping technique that utilizes an OcTree data structure to obtain adaptive resolution, e.g., combining many small cells



(a)

(b)

Fig. 4: Example of a semantic OctoMap: (a) A white circle represents an *inner* node such that its children collectively cover the same physical space as the inner node itself. A colored square represents a partition of the 3-D space where all downstream nodes contain identical semantic and occupancy values; therefore, they can be *pruned* into a *leaf* node. Lastly, black dots represent *unexplored* spaces of the environment. (b) geometric representation of the same OcTree with an overview of the *SemanticOcTreeNode* class.

associated with free space into few large cells. In this section, we develop a multi-class version of OctoMap and propose an efficient multi-class mutual information computation which benefits from the OcTree structure.

A. Semantic OctoMap

An OcTree is a hierarchical data structure containing nodes that represent a section of the physical environment. Each node has either 0 or 8 children, where the latter corresponds to the 8 octants of the Euclidean 3-D coordinate system. Thus, the children of a parent node form an eight-way octant partition of the space associated with the parent node. Fig. 4 shows an example of a multi-class OcTree data structure.

We implement a *SemanticOcTreeNode* class as a building block of the multi-class OcTree structure. A *SemanticOcTreeNode* instance stores occupancy, color, and semantic information of its corresponding physical space, as shown in Fig. 4b. The most important data members of the *SemanticOcTreeNode* class are:

- *children*: an array of pointers to *SemanticOcTreeNode* storing the memory addresses of the 8 child nodes,
- *value*: a *float* variable storing the log-odds occupancy probability of the node,
- *color*: a *ColorRGB* object storing the RGB color of the node,
- *semantics*: a *SemanticLogOdds* object maintaining a categorical probability distribution over the semantic labels in the form of a log-odds ratio.

For performance reasons, the *SemanticLogOdds* class only stores the multi-class log-odds for the 3 most likely class labels, with each label represented by a unique RGB color. In this case, the log-odds associated with the rest of the labels lump into a single *others* variable. This relives the multi-class OcTree implementation from dependence on the number of labels that the object classifier can detect. Moreover, it significantly improves the speed of the mapping algorithm in cases with many semantic categories. See Sec. VII-D for an analysis of mapping time versus the number of stored classes.

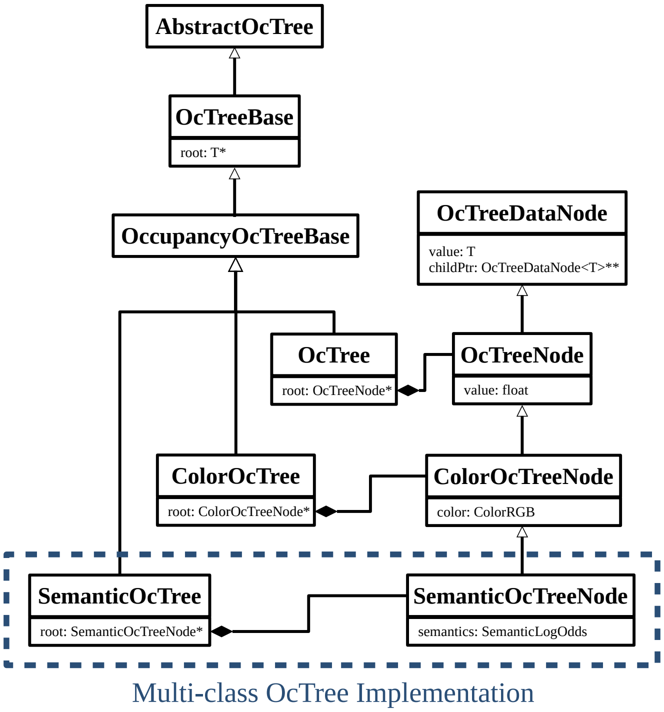


Fig. 5: UML diagram showing the class inheritance used for the implementation of a multi-class OcTree.

The implementation of the multi-class OcTree is completed by defining a *SemanticOcTree* class, which is derived from the *OccupancyOcTreeBase* class of the OctoMap library [3] and uses a *SemanticOcTreeNode* as its node type. Fig. 5 illustrates the derivation of the *SemanticOcTree* and *SemanticOcTreeNode* classes as a UML diagram.

In order to register a new observation to a multi-class OcTree, we follow the standard ray-casting procedure over an OcTree, as in [3], to find the observed leaf nodes. Then, for each observed leaf node, if the observation demands an update, the leaf node is recursively expanded to the smallest resolution and the multi-class log-odds of the downstream

Algorithm 2 Multi-class OcTree Update of Node $\mathbf{n}_i$

```
Input: OcTree node $\mathbf{n}_i$, observation $\mathbf{z} = (r, y)$, mixing coefficient $\alpha$
$s = \mathbf{n}_i.semantics$
$s.d = \mathbf{n}_i.semantics.data$
$s.o = \mathbf{n}_i.semantics.others$
if $\mathbf{z}$ indicates free then
    Update $s$ with $\phi^-$
else if $\mathbf{z}$ indicates class $y$ then
    if class $y$ is among the 3 most likely classes in $s$ then
        Update $s$ with $\boldsymbol{\phi}^+ + \mathbf{E}_{y+1}\boldsymbol{\psi}^+$
    else
        ▷ Derive $h_{aux}$ as a portion $\alpha$ of others class
        $h_{aux} = s.o + \log\alpha$
        $s.o \mathrel{+}= \phi^+_{others} + \log(1-\alpha)$
        $s_c = \text{CONCAT}(s.d, (y, h_{aux}))$
        Update $s_c$ with $\boldsymbol{\phi}^+ + \mathbf{E}_{y+1}\boldsymbol{\psi}^+$
        Perform descending sort on $s_c$ with respect to log-odds values
        ▷ Pick 3 most likely classes
        $s.d = s_c[0:2]$
        ▷ Combine the least likely class with others class
        $s.o = \log\Big(\exp(s_c[3]) + \exp(s.o)\Big)$
▷ Apply thresholds $\underline{s}$ and $\overline{s}$ for log-odds values
$s_f \leftarrow \min\{\max\{s_f, \underline{s}\}, \overline{s}\}$
$\mathbf{n}_i.semantics = s$
return $\mathbf{n}_i$
```

Algorithm 3 Semantic Fusion of Two Child Nodes

```
Input: OcTree nodes $\mathbf{n}_i$ and $\mathbf{n}_j$
$s_i = \mathbf{n}_i.semantics$
$s_j = \mathbf{n}_j.semantics$
▷ Non-repeating list of classes in $s_i$ and $s_j$
$\mathcal{K}_f = \text{UNIQUECLASS}(s_i, s_j)$
▷ Object instantiation for the fused semantics
$s_f = SemanticLogOdds()$
▷ Slice $s_i.o$ into smaller probabilities
$o_i = s_i.o - \log(1 + \mathcal{K}_f.size - s_i.d.size)$
$o_j = s_j.o - \log(1 + \mathcal{K}_f.size - s_j.d.size)$
for $y \in \mathcal{K}_f$ do
    if $y \notin s_i.d.label \land y \in s_j.d.label$ then
        $s_f.d.\text{APPEND}(y, \frac{o_i + s_j.d[y].logOdds}{2})$
    else if $y \in s_i.d.label \land y \notin s_j.d.label$ then
        $s_f.d.\text{APPEND}(y, \frac{s_i.d[y].logOdds + o_j}{2})$
    else
        $s_f.d.\text{APPEND}(y, \frac{s_i.d[y].logOdds + s_j.d[y].logOdds}{2})$
Perform descending sort on $s_f.d$ with respect to log-odds values
$expOthers = \exp(\frac{o_i + o_j}{2})$
for $i > 3$ do
    $expOthers \mathrel{+}= \exp(s_f.d[i].logOdds)$
$s_f.d[3:end].\text{REMOVE}()$
$s_f.o = \log(expOthers)$
$s_f \leftarrow \min\{\max\{s_f, \underline{s}\}, \overline{s}\}$
return $s_f$
```

nodes are updated using (8). At the ray's end point, which indicates an occupied cell, we also update the *color* variable by averaging the observed color with the current stored color of the corresponding node. Alg. 2 details the Bayesian update procedure for the multi-class OcTree.

To obtain a compressed OctoMap, it is necessary to define a rule for information fusion from child nodes towards parent nodes. Depending on the application, different information fusion strategies may be implemented. For example, a conservative strategy would assign the multi-class log-odds of the child node with the highest occupancy probability to the parent node. In this work, we simply assign the average log-odds vector of the child nodes to their parent node as shown in Alg. 3. The benefit of an OctoMap representation is the ability to combine similar cells (leaf nodes) into a large cell (inner node). This is called *pruning* the OcTree. Every time after an observation is integrated to the map, starting from the deepest inner node, we check for each inner node if 1) the node has 8 children, 2) its children do not have any children of their own, and 3) its children all have equal multi-class log-odds. If an inner node satisfies all of these three criteria, its children are pruned and the inner node is converted into a leaf node with the same multi-class log-odds as its children. This helps to compress the majority of the free cells into a few large cells, while the occupied cells usually do not undergo pruning since only their surfaces are observed by the sensor and their inside remains an unexplored region. Due to sensor noise, it is unlikely that cells belonging to the same class (e.g., free or occupied by the same obstacle) attain identical multi-class log-odds. Maximum and minimum limits for the elements of the multi-class log-odds are used so that each cell arrives at a stable state as its multi-class log-odds entries reach the limits. Stable cells are more likely to share the same multi-class probability distribution, consequently increasing the chance of OcTree pruning. However, thresholding causes loss of information near $p_t(m_i = k) = 1$, $k \in \mathcal{K}$ which can be controlled by the maximum and minimum limits.

B. Information Computation

A ray cast through an OcTree representation may visit several large cells within which the class probabilities are homogeneous. We exploit this property to obtain the mutual information between a multi-class OctoMap and a single ray as a summation over a subset of OcTree leaf nodes instead of individual map cells. This simplification provides a significant performance gain with no loss of accuracy. The following formulation can be considered a multi-class generalization of the run-length encoding technique introduced by [10], using the mutual information lower bound in (13) and the multi-class OcTree defined earlier in this section.

Suppose that the map cells along a single beam $\mathbf{R}_\tau\boldsymbol{\eta}_b$ have piecewise-constant multi-class probabilities such that the set $\{m_i \mid i \in \mathcal{R}_{\tau,b}(r_{max})\}$ can be partitioned into $Q_{\tau,b}$ groups of consecutive cells indexed by $\mathcal{R}^q_{\tau,b}(r_{max})$, $q = 1, \ldots, Q_{\tau,b}$,

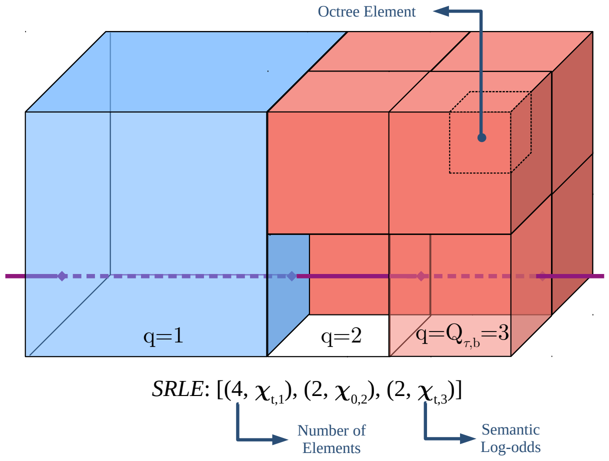


Fig. 6: Ray cast representation as semantic run-length encoding (SRLE). The multi-class log-odds $\boldsymbol{\chi}_{t,q}$ are uniform within each cube. The voxel corresponding to $q = 2$ is unexplored, hence its multi-class log-odds are denoted as $\boldsymbol{\chi}_{0,2}$.

where:

$$p_t(m_i = k) = p_t(m_j = k),\\ \forall i, j \in \mathcal{R}^q_{\tau,b}(r_{max}), \quad \forall k \in \mathcal{K}. \tag{14}$$

In this case, the log-odds probabilities encountered by a ray cast can be compressed using semantic run-length encoding, defined as below.

Definition 3. A *semantic run-length encoding* (SRLE) of a ray $\mathbf{R}_\tau \boldsymbol{\eta}_b$ cast through a multi-class OcTree is an ordered list of tuples of the form $[(\omega_{\tau,b,q}, \boldsymbol{\chi}_{t,q})]_{q=1}^{Q_{\tau,b}}$, where $\omega_{\tau,b,q}$ and $\boldsymbol{\chi}_{t,q}$ respectively represent the width and the log-odds vector of the intersection between the ray and the cells in $\mathcal{R}^q_{\tau,b}(r_{max})$. The width $\omega_{\tau,b,q}$ is the number of OcTree *elements* along the ray intersection, where an OcTree element is a cell with the smallest physical dimensions.

Fig. 6 shows an example of SRLE over a semantic OctoMap. While SRLE can be used in a uniform-resolution grid map, it is particularly effective of a multi-class OcTree, which inherently contains large regions with homogeneous multi-class log-odds. Additionally, the OcTree data structure allows faster ray casting since it can be done over OcTree leaf nodes [46], [47], instead of a uniform-resolution grid as in [45].

SRLE ray casting delivers substantial gains in efficiency for mutual information computation since the contribution of each group $\{m_i \mid i \in \mathcal{R}^q_{\tau,b}(r_{max})\}$ in the innermost summation of (13) can be obtained in closed form.

Proposition 3. *The Shannon mutual information between a single range-category measurement* $\mathbf{z}_{\tau,b}$ *and a semantic OctoMap* $\mathbf{m}$ *can be computed as:*

$$I(\mathbf{m}; \mathbf{z}_{\tau,b} | \mathcal{Z}_{1:t}) = \sum_{k=1}^{K} \sum_{q=1}^{Q_{\tau,b}} \rho_{\tau,b}(q,k) \Theta_{\tau,b}(q,k) \tag{15}$$

where $Q_{\tau,b}$ *is the number of partitions along the ray* $\mathbf{R}_\tau \boldsymbol{\eta}_b$ *that have identical multi-class log-odds and the multi-class probabilities for each partition are denoted as:*

$$\begin{cases} \pi_t(q,k) = p_t(m_i = k) \\ \boldsymbol{\chi}_{t,q} = \mathbf{h}_{t,i} \end{cases} \qquad \forall i \in \mathcal{R}^q_{\tau,b}(r_{max}).$$

Furthermore, defining $\omega_{\tau,b,q} = |\mathcal{R}^q_{\tau,b}(r_{max})|$ *as the width of q-th partition, we have:*

$$\rho_{\tau,b}(q,k) := \pi_t(q,k) \prod_{j=1}^{q-1} \pi_t^{\omega_{\tau,b,j}}(j,0),$$

$$\Theta_{\tau,b}(q,k) := \beta_{\tau,b}(q,k) \frac{1 - \pi_t^{\omega_{\tau,b,q}}(q,0)}{1 - \pi_t(q,0)} + \frac{f(\boldsymbol{\phi}^- - \boldsymbol{\chi}_{0,q}, \boldsymbol{\chi}_{t,q})}{(1 - \pi_t(q,0))^2} \Big[(\omega_{\tau,b,q} - 1) \pi_t^{\omega_{\tau,b,q}+1}(q,0) - \omega_{\tau,b,q} \pi_t^{\omega_{\tau,b,q}}(q,0) + \pi_t(q,0) \Big],$$

$$\beta_{\tau,b}(q,k) := f(\boldsymbol{\phi}^+ + \mathbf{E}_{k+1}\boldsymbol{\psi}^+ - \boldsymbol{\chi}_{0,q}, \boldsymbol{\chi}_{t,q}) + \sum_{j=1}^{q-1} \omega_{\tau,b,j} f(\boldsymbol{\phi}^- - \boldsymbol{\chi}_{0,j}, \boldsymbol{\chi}_{t,j}).$$

Proof. See Appendix C. □

In (15), $\rho_{\tau,b}(q,k)$ relates to the event that the partition $\mathcal{R}^q_{\tau,b}(r_{max})$ belongs to category k while all of the previous partitions along the ray $\mathbf{R}_\tau \boldsymbol{\eta}_b$ are free. Analogous to the definition of $C_{\tau,b}(n,k)$ in Proposition 2, $\beta_{\tau,b}(q,k)$ is the weighted sum of log-ratios $f(\boldsymbol{\phi}, \boldsymbol{\chi})$ for the first q partitions along the ray $\mathbf{R}_\tau \boldsymbol{\eta}_b$ under the same event as the one $\rho_{\tau,b}(q,k)$ is associated with. Accumulating the multi-class probabilities within the partition $\mathcal{R}^q_{\tau,b}(r_{max})$ yields $\Theta_{\tau,b}(q,k)$, see (33) for more details. Therefore, the mutual information in (15) is equivalent to the expectation of accumulated log-ratios $\Theta_{\tau,b}(q,k)$ over all possible instantiations of $\mathbf{z}_{\tau,b}$.

Proposition 3 allows an extension of the mutual-information lower bound in Proposition 2 to semantic OctoMap representations, summarized in the corollary below. The proof follows directly from the additive property of mutual information between a semantic OctoMap and a sequence of independent observations.

Corollary 1. *Given a sequence of range-category observations* $\mathcal{Z}_{t+1:t+T}$*, the Shannon mutual information between* $\mathcal{Z}_{t+1:t+T}$ *and a semantic OctoMap* $\mathbf{m}$ *can be lower bounded as:*

$$I(\mathbf{m}; \mathcal{Z}_{t+1:t+T} | \mathcal{Z}_{1:t}) \geq I\left(\mathbf{m}; \underline{\mathcal{Z}}_{t+1:t+T} | \mathcal{Z}_{1:t}\right) = \sum_{\tau=t+1}^{t+T} \sum_{b=1}^{B} \sum_{k=1}^{K} \sum_{q=1}^{Q_{\tau,b}} \rho_{\tau,b}(q,k) \Theta_{\tau,b}(q,k), \tag{16}$$

where $\underline{\mathcal{Z}}_{t+1:t+T}$ *is a subset of non-overlapping measurements that satisfy* (12)*, and* $\rho_{\tau,b}(q,k)$ *and* $\Theta_{\tau,b}(q,k)$ *are defined in Proposition 3.*

The same approach as in Alg. 1 is used for autonomous exploration over a semantic OctoMap. However, we employ the information computation formula of (16) to quantify the informativeness of candidate robot trajectories. The active

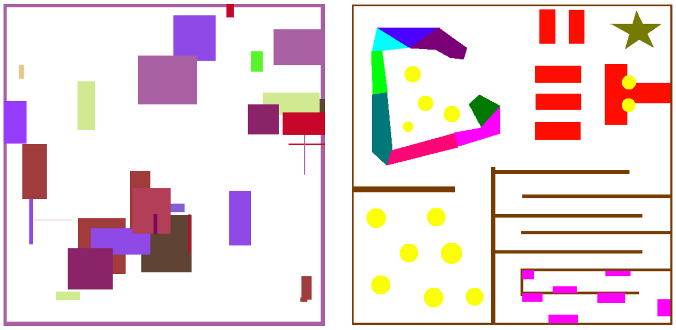


Fig. 7: Synthetic environments used for comparisons among frontier-based exploration [12], FSMI [10], and SSMI. Different semantic categories are represented by distinct colors. Left: An instance of procedurally generated random environment with 10 object classes. Right: Hand-designed environment with corridor and block structures with 12 object classes.

mapping method in Alg. 1 provides a greedy exploration strategy, which does not change subsequent control inputs based on the updated map distribution. Greedy exploration may be sub-optimal and manifests itself as back and forth travel between map frontiers. We alleviate this behavior by (a) computing the information along the whole trajectory as opposed only at the frontiers or next best view, and (b) re-plan frequently to account for the updated map distribution. Discounted by distance traveled as the cost of a trajectory, this leads to a more accurate calculation of information gain along a candidate path which rules out most of the back and forth visiting behaviour. It is also important to mention that the main scope of this work is introduction of a novel multi-class semantic OcTree representation and the mutual information between such model and range-category observations. Our method enables fast and accurate evaluation of information for any set of candidate trajectories, likes of which can be generated by random tree methods [10], [48] or hierarchical planning strategies [21] or, in the simplest form, a greedy approach that computes paths to each frontier. We believe utilizing our proposed information measure to score candidate viewpoints would be complementary, rather than an alternative, to the state-of-the-art exploration methods that use sophisticated optimization strategies [19], [21], [49].

C. Computational Complexity

Note that the mutual information computations in both (13) and (16) can be performed recursively. For (13), we have:

$$
\begin{aligned}
p_{\tau,b}(n+1,k) &= p_{\tau,b}(n,k)\frac{p_t(m_{j^*_{\tau,b}}=k)\,p_t(m_{i^*_{\tau,b}}=0)}{p_t(m_{i^*_{\tau,b}}=k)},\\
C_{\tau,b}(n+1,k) &= C_{\tau,b}(n,k)\\
&\quad - f(\boldsymbol{\phi}^+ + \mathbf{E}_{k+1}\boldsymbol{\psi}^+ - \mathbf{h}_{0,i^*_{\tau,b}}, \mathbf{h}_{t,i^*_{\tau,b}})\\
&\quad + f(\boldsymbol{\phi}^+ + \mathbf{E}_{k+1}\boldsymbol{\psi}^+ - \mathbf{h}_{0,j^*_{\tau,b}}, \mathbf{h}_{t,j^*_{\tau,b}})\\
&\quad + f(\boldsymbol{\phi}^- - \mathbf{h}_{0,i^*_{\tau,b}}, \mathbf{h}_{t,i^*_{\tau,b}}),
\end{aligned}
\tag{17}
$$

where $j^*_{\tau,b}$ and $i^*_{\tau,b}$ correspond to the index of farthest map cell in $\tilde{\mathcal{R}}_{\tau,b}(n+1)$ and $\tilde{\mathcal{R}}_{\tau,b}(n)$, respectively. A similar recursive pattern can be found in (16):

$$
\begin{aligned}
\rho_{\tau,b}(q+1,k) &= \rho_{\tau,b}(q,k)\frac{\pi_t(q+1,k)\,\pi_t^{\omega_{\tau,b,q}}(q,0)}{\pi_t(q,k)},\\
\beta_{\tau,b}(q+1,k) &= \beta_{\tau,b}(q,k)\\
&\quad - f(\boldsymbol{\phi}^+ + \mathbf{E}_{k+1}\boldsymbol{\psi}^+ - \boldsymbol{\chi}_{0,q}, \boldsymbol{\chi}_{t,q})\\
&\quad + f(\boldsymbol{\phi}^+ + \mathbf{E}_{k+1}\boldsymbol{\psi}^+ - \boldsymbol{\chi}_{0,q+1}, \boldsymbol{\chi}_{t,q+1})\\
&\quad + \omega_{\tau,b,q} f(\boldsymbol{\phi}^- - \boldsymbol{\chi}_{0,q}, \boldsymbol{\chi}_{t,q}).
\end{aligned}
\tag{18}
$$

This implies that the innermost summations of (13) and (16) can be obtained in $O(N_{\tau,b})$ and $O(Q_{\tau,b})$, respectively, where $N_{\tau,b}$ is the number of map cells along a single ray $\mathbf{R}_\tau\boldsymbol{\eta}_b$ up to its maximum range, and $Q_{\tau,b}$ is the number of groups of consecutive cells that possess the same multi-class probabilities. In an environment containing K object classes, evaluating the informativeness of a trajectory composed of T observations, where each observation contains B beams, has a complexity of $O(TBKN_{\tau,b})$ for a regular-grid multi-class representation and a complexity of $O(TBKQ_{\tau,b})$ for a multi-class OcTree representation.

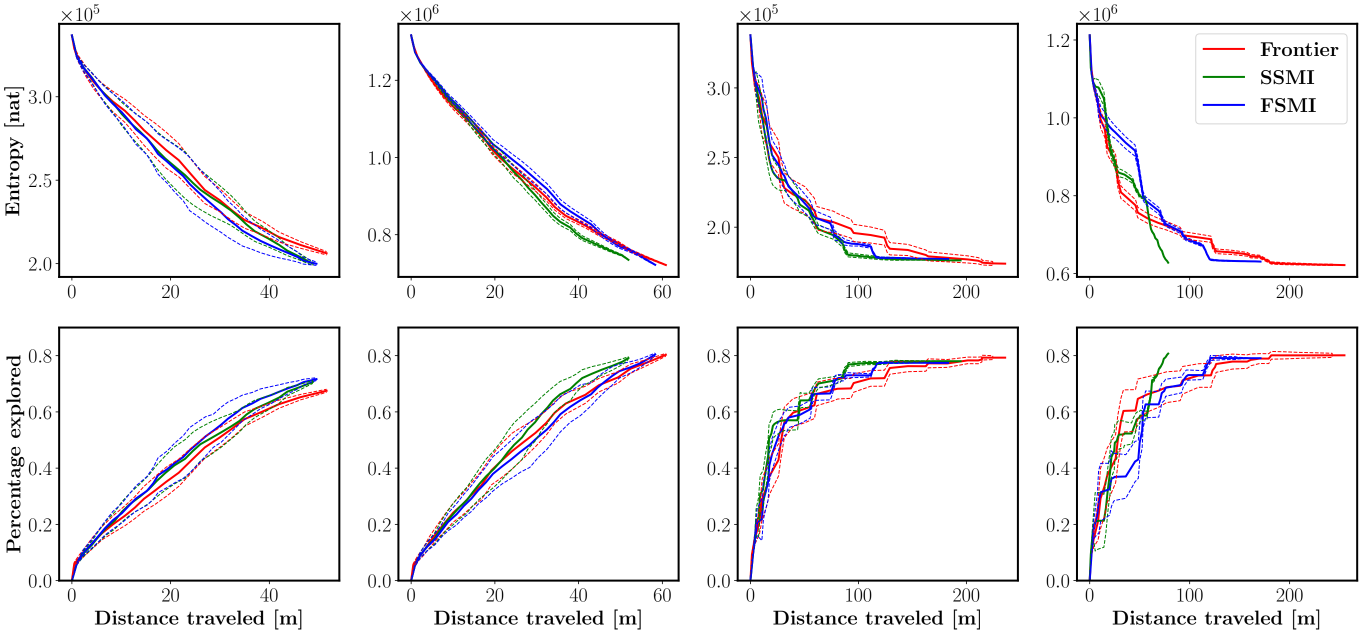
As we demonstrate in Sec. VII-C, for a ray $\mathbf{R}_\tau\boldsymbol{\eta}_b$ we often observe that $Q_{\tau,b}$ is significantly smaller than $N_{\tau,b}$ thanks to the OcTree pruning mechanism. Since $N_{\tau,b}$ scales linearly with the map resolution, the complexity of information computation over a semantic OctoMap grows sub-linearly with respect to the inverse of the OcTree element dimensions, which is a parameter analogous to the map resolution.

VII. Experiments

In this section, we evaluate the performance of SSMI in simulated and real-world experiments. We compare SSMI with two baseline exploration strategies, i.e., frontier-based exploration [12] and FSMI [10], in a 2-D active binary mapping scenario in Sec. VII-A and a 2-D active multi-class mapping scenario in Sec. VII-B. All three methods use our range-category sensor model in (10) and our Bayesian multi-class mapping in (8) but select informative robot trajectories $\mathbf{X}_{t+1:t+T}(\mathbf{u}_{t:t+T-1})$ based on their own criteria. In Sec. VII-C, we evaluate the improvement in ray tracing resulting from SRLE through an experiment in a 3-D simulated Unity environment. In Sec. VII-D, we investigate the influence of the number of stored semantic classes on mapping performance. Additionally, in Sec. VII-E, we use a similar 3-D simulation environment to apply SSMI alongside Frontier, FSMI, and hierarchical coverage maximization method TARE [21]. In this section we use our OcTree-based multi-class information computation introduced in Sec. VI in order to demonstrate large-scale realistic active multi-class mapping. Finally, in Sec. VII-F and Sec. VII-G, we test SSMI mapping and exploration in real environments using ground wheeled robots. An open-source implementation of SSMI is available on GitHub[2].

In each planning step of 2-D exploration, we identify frontiers by applying edge detection on the most likely map at time t (the mode of $p_t(\mathbf{m})$). Then, we cluster the edge cells by detecting the connected components of the boundaries

[2] https://github.com/ExistentialRobotics/SSMI.



(a) Random map / Binary exploration

(b) Random map / Semantic exploration

(c) Structured map / Binary exploration

(d) Structured map / Semantic exploration

Fig. 8: Simulation results for active mapping on the environments in Fig. 7, for 20 exploration iterations. Solid and dotted lines represent mean and 1 standard deviation from the mean, respectively. (a), (b): Exploration performance averaged over 10 random environments with 3 random starting positions for each instance. Exploration in the random environments sometimes did not terminate before the maximum number of iterations, and therefore the corresponding curves do not flatten. This can be attributed to the fact that random maps, with the same size as the structured map, contain more frontiers that need to be explored in each iteration since each scan has a higher probability of being occluded in multiple angles due to the lack of certain patterns such as corridors. (c), (d): Exploration performance on the structured environment averaged over 3 random starting positions. For the structured map, the exploration terminates before reaching the maximum number of iterations, which explains the flat curves at the end of the corresponding plots.

between explored and unexplored space. We plan a path from the robot pose $\mathbf{X}_t$ to the center of each frontier using A^* graph search and provide the path to a low-level controller to generate $\mathbf{u}_{t:t+T-1}$. For 3-D exploration, we first derive a 2-D occupancy map by projecting the most likely semantic OctoMap at time t onto the $z=0$ surface and proceed with similar steps as in 2-D path planning.

A. 2-D Binary Exploration

We consider active binary occupancy mapping first. We compare SSMI against Frontier and FSMI in 1 structured and 10 procedurally generated 2-D environments, shown in Fig. 7. A 2-D LiDAR sensor is simulated with additive Gaussian noise $\mathcal{N}(0, 0.1)$. Fig. 8a and Fig. 8c compare the exploration performance in terms of map entropy reduction and percentage of the map explored per distance traveled among the three methods. SSMI performs similarly to FSMI in that both achieve low map entropy by traversing significantly less distance compared to Frontier.

B. 2-D Multi-class Exploration

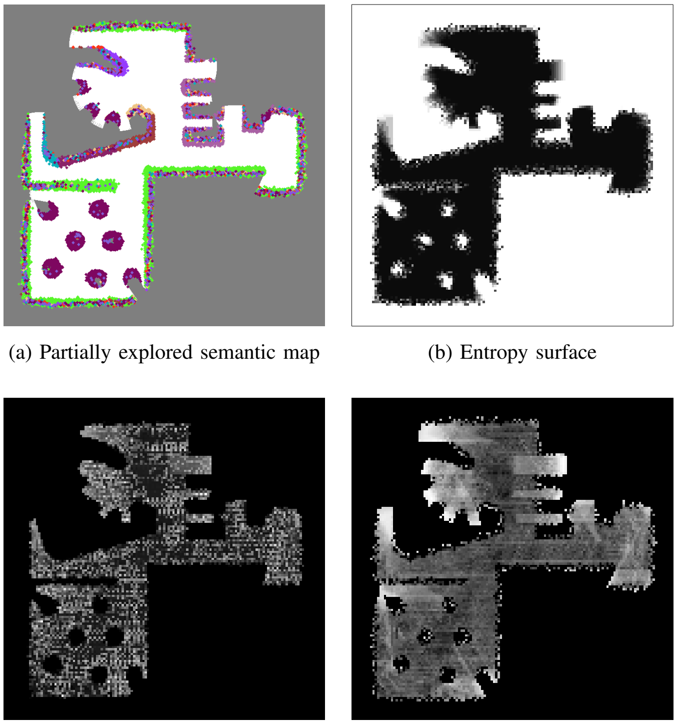
Next, we use the same 2-D environments in Fig. 7 but introduce range-category measurements. Range measurements are subject to additive Gaussian noise $\mathcal{N}(0, 0.1)$, while category measurements have a uniform misclassification probability of 0.35. Fig. 8b and Fig. 8d compare the semantic exploration performance for all three strategies. SSMI reaches the same level of map entropy as FSMI and Frontier but traverses a noticeably shorter distance. This can be attributed to the fact that only SSMI distinguishes map cells whose occupancy probabilities are the same but their per-class probabilities differ from each other. To further illustrate this, we visualize the entropy and information surfaces used by FSMI and SSMI. Fig. 9a shows a snapshot of semantic exploration while Fig. 9b visualizes the entropy of each pixel i computed as:

$$H(m_i|\mathcal{Z}_{1:t}) = -\sum_{k=0}^{K} p_t(m_i = k) \log p_t(m_i = k), \tag{19}$$

where $\mathcal{Z}_{1:t}$ denote realized observations until time t. The task of exploration can be regarded as minimizing the conditional entropy summed over all pixels, i.e., map entropy. However, since the observations are not known in advance, we resort to estimate the reduction in uncertainty by computing the expectation over the observations. Accounting for the prior uncertainty in map, we arrive at maximizing mutual information as our objective, which is related to entropy as follows:

$$H(m_i) - \mathbb{E}_{\mathcal{Z}_{1:t}}\{H(m_i|\mathcal{Z}_{1:t})\} = I(m_i; \mathcal{Z}_{1:t}). \tag{20}$$

Therefore, the exploration performance is highly dependent upon the mutual information formulation, since it directly dictates how the uncertainty is quantified. As shown in Fig. 9d and resulted from capturing per-class uncertainties, semantic



(a) Partially explored semantic map

(b) Entropy surface

(c) Occupancy mutual information surface

(d) Multi-class mutual information surface

Fig. 9: Comparison between different mutual information formulations used for exploration. (a), (b): A snapshot of 2-D exploration showing the map estimate and the corresponding uncertainty, where the entropy for each pixel i is computed as (19). (c), (d): The mutual information used to find the informative trajectory by FSMI and SSMI, respectively. Brighter pixels indicate larger values.

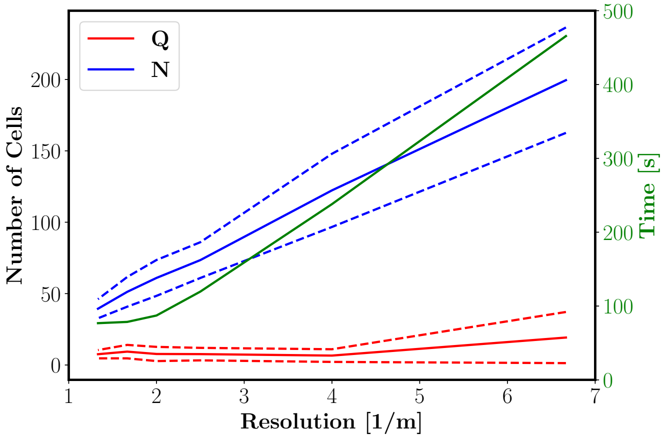


Fig. 10: Variation of the visited OctoMap cells and OcTree elements denoted as Q and N, respectively, with respect to the map resolution. Solid blue and red lines represent the average values for Q and N over all ray castings, while the dashed lines show one standard deviation from the average. The green curve shows the total exploration time for each map resolution. All measurements are accumulated in the course of 5 exploration iterations.

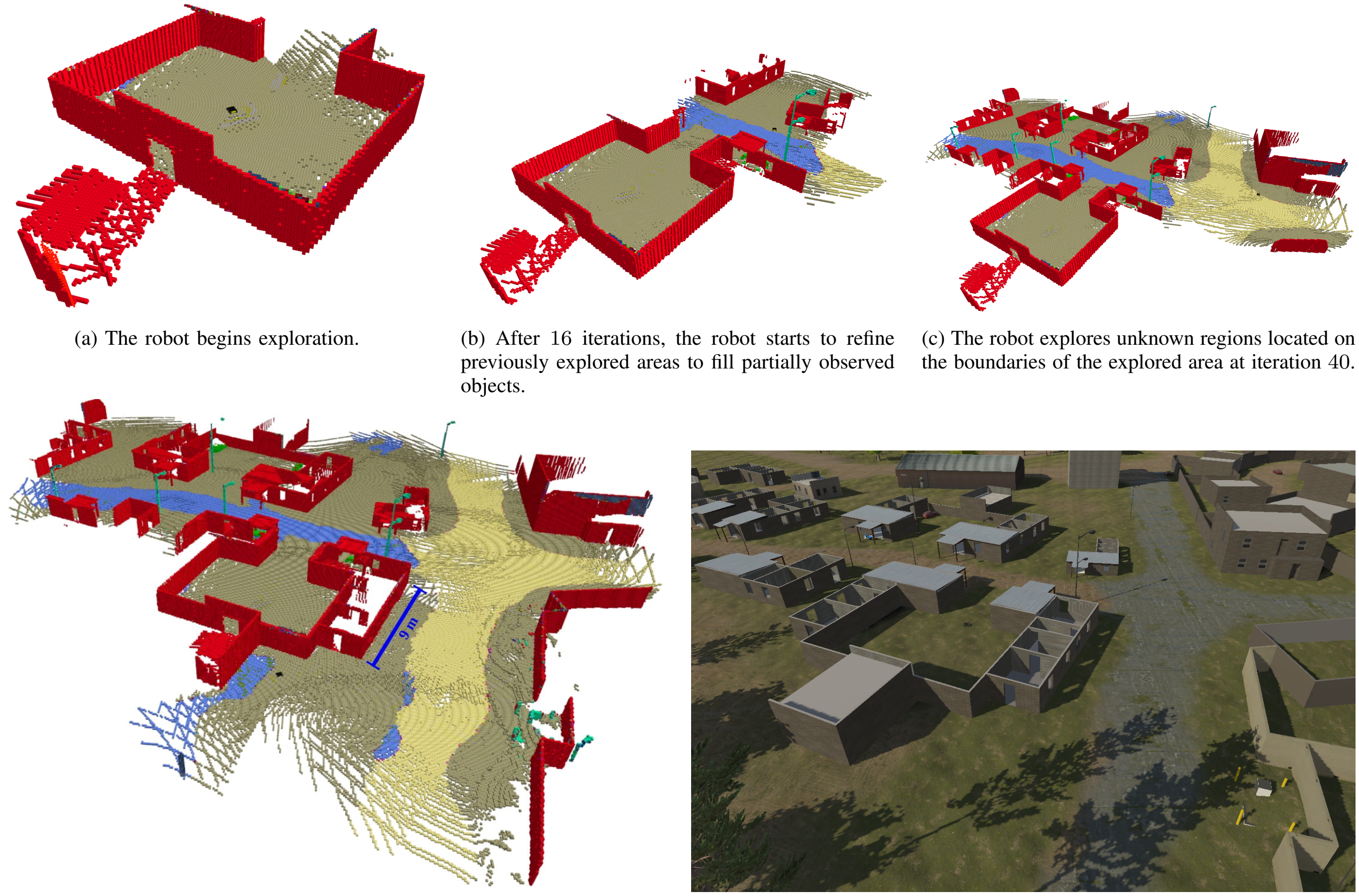
mutual information of SSMI, computed in (13) provides a smoother and more accurate estimation of information-rich regions compared to the binary mutual information formula used by FSMI (equation (18) in [10]) shown in Fig. 9c.

C. SRLE Compression for 3-D Ray Tracing

In this subsection, we evaluate the ray-tracing compression resulting from SRLE through an experiment in a photo-realistic 3-D Unity simulation, shown in Fig. 11e. We use a Husky robot equipped with an RGBD camera and run a semantic segmentation algorithm over the RGB images. In order to remove irrelevant randomness, the sensors and the semantic segmentation are defined as error-free. We define *map resolution* as the inverse of the dimensions of an OcTree element. For resolutions ranging from $1.3m^{-1}$ to $6.6m^{-1}$, we run 5 exploration iterations using the semantic OctoMap and information computation of Sec. VI and store all ray traces in SRLE format. Fig. 10 shows the change in distribution for the number of OctoMap cells Q and OcTree elements N visited during each ray trace, as well as the time required to execute each exploration episode as a function of map resolution. In other words, N represents the number of cells to be processed during mapping and information computation as if the environment was represented as a regular 3-D grid, while Q represents the actual number of processed semantic OctoMap cells. The pruning mechanism of the OcTree representation results in a substantial gain in terms of the number of cells visited for each ray tracing. As opposed to the almost linear growth of N, the distribution for Q is effectively independent of the map resolution, except for very fine resolutions where void areas between observations rays prevent efficient pruning. However, for map resolutions larger than $2m^{-1}$, the exploration time tends to grow larger with the increase of map resolution. This is attributed to the recursive ray insertion method of OctoMap in which it is required to re-compute log odds for each OcTree element along an observation ray whenever an observation ray does not carry the same (free or object class) state as the visited cell. In the subsequent 3-D experiments, we choose map resolution of $2m^{-1}$ in order to balance between performance and map accuracy.

D. Mapping Time vs. Number of Stored Classes

We analyse the influence of the number of stored classes in the semantic OctoMap on the mapping time. Let K_s denote the number of stored semantic classes. Alg. 2 has $O(K_s)$ memory and $O(K_s \log K_s)$ computational complexity (due to sorting in line 14). Furthermore, let p_{miss} be the misclassification probability assumed to be uniformly distributed among all incorrect classes. Regarding accuracy, for a classifier with $p_{\text{miss}} < \frac{K-1}{K}$, where K is the number of all object classes, the true class will be always asymptotically recoverable as long as $K_s \geq 2$, thanks to the auxiliary *others* class that stores the accumulated probability of the $K - K_s$ least likely classes (see line 16 of Alg. 2). In general, K_s controls how fast the true class will be detected with the cost of additional memory use and computation. In order to quantitatively evaluate the effect of K_s on mapping time, we consider the same Husky robot as the previous subsection with a fixed trajectory, shown in Fig. 12 (top), and measure the mapping frequency as a function of K_s. Fig. 12 (bottom) shows the decrease in average mapping frequency as K_s increases. It is important to mention



(a) The robot begins exploration.

(b) After 16 iterations, the robot starts to refine previously explored areas to fill partially observed objects.

(c) The robot explores unknown regions located on the boundaries of the explored area at iteration 40.

(d) Multi-class occupancy map after 60 exploration iterations

(e) Photo-realistic Unity simulation environment

Fig. 11: Time lapse of autonomous exploration and multi-class mapping in a simulated Unity environment. The robot is equipped with an RGBD sensor and runs semantic segmentation. Different colors represent different semantic categories (grass, dirt road, building, etc.).

that the trajectory along which the data is collected only visits 6 object classes, which explains the change in slope for $K_s > 6$.

E. 3-D Exploration in a Unity Simulation

We evaluate SSMI in the same 3-D simulation environment as two previous subsections, however, this time the range measurements have an additive Gaussian noise of $\mathcal{N}(0, 0.1)$ and the semantic segmentation algorithm detects the true class with a probability of 0.95 while the misclassification happens uniformly in the pixel space. Fig. 11 shows several iterations of the exploration process. For comparison, we implemented a 3-D version of FSMI [10] that utilizes run-length encoding to accelerate the information computation for a binary OctoMap. Moreover, we deploy the state-of-the-art hierarchical exploration method of TARE [21] in our 3-D Unity simulation environment. Fig. 13 shows the change in map entropy versus distance traveled and total elapsed time for all exploration strategies. We observe that SSMI is the most efficient in terms of solving the trade-off between path length and information gathered along the path. SSMI achieves the lowest entropy in the multi-class OctoMap. Similar to the discussion in Sec. VII-B, this observation can be ascribed to the fact that, among the compared methods, the only objective function which captures the uncertainty in both semantic classes and occupancy of the environment is the one used by SSMI. On the other hand, SSMI and FSMI require evaluation of mutual information along each candidate trajectory, which has the same cardinality as the number of all frontiers in the current map estimate $p_t(\mathbf{m})$, whereas the hierarchical planning method employed by TARE only requires local trajectory computation with a global coverage path obtained at a coarse level. As a result, TARE exploration can be performed over a relatively shorter time period compared to SSMI and FSMI in scenarios where the number of frontiers is large, e.g. outdoor areas. Parallel computation of mutual information for each candidate trajectory or using heuristics such as frontier size in order to sort candidate solutions would improve the computation time of SSMI; however we believe these are outside of the scope of this paper. Fig. 14 compares the mapping precision of various object classes for the tested methods. SSMI exhibits higher precision for object categories that appear rarely, such as the *Animal* or *Tree* classes while Frontier slightly outperforms SSMI when it comes to mapping the *Grass* and *Dirt Road* categories. This can be explained by the tendency of SSMI towards achieving high overall

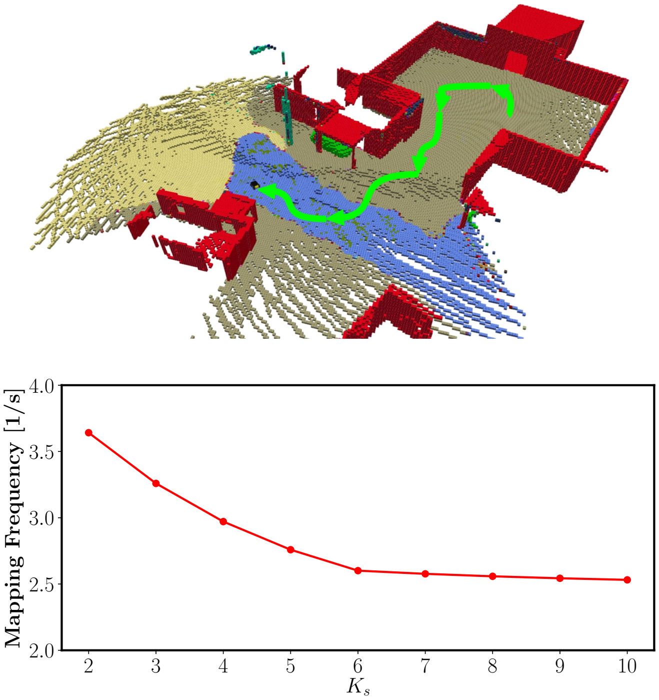


Fig. 12: Mapping time vs. number of stored classes. Top: Robot trajectory (in green) used for all mapping frequency evaluations. Bottom: Average mapping frequency as a function of the number of stored semantic classes K_s.

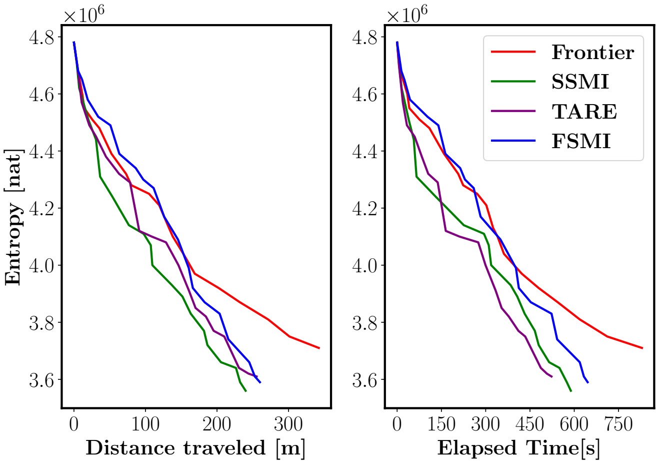


Fig. 13: Simulation results for exploration in Unity 3-D environment.

classification precision even if it requires slight reduction of precision for certain object categories. Furthermore, TARE achieves the highest precision for the *Building* class, which can be justified by the observation that the computed global coverage path tends to traverse near building walls.

F. 3-D Mapping in a Real-world Outdoor Environment

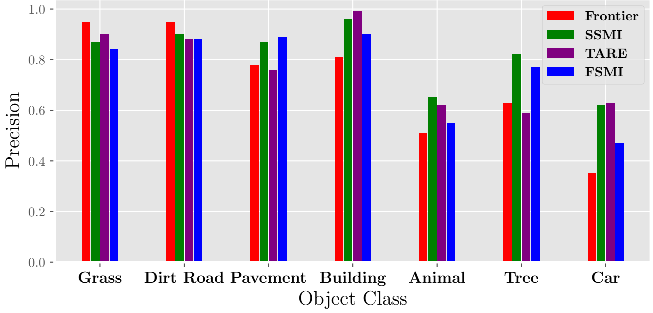


Fig. 14: Mapping precision for observed semantic classes.

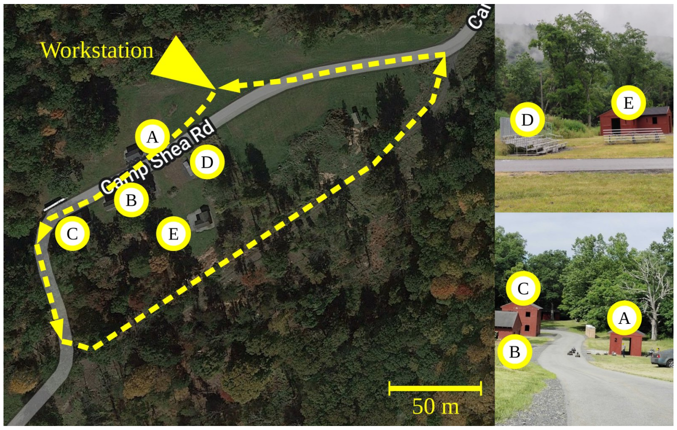


Fig. 15: Environment for the outdoor mapping experiment. Left: Satellite image of the experiment locale with robot trajectory shown in yellow. Right: Corresponding locations from the ground level point of view.

We deployed our semantic OcTree mapping approach on a Husky robot equipped with an Ouster OS1-32 LiDAR and an Intel RealSense D455 RGBD camera. Our software stack is implemented using the *Robot Operating System* (ROS) [50]. The LiDAR is used for localization via iterative closest point (ICP) scan matching [51]. A neural network based on a *FCHarDNet* architecture [52] and trained on the RUGD dataset [53] was used for semantic segmentation. The RGBD camera produces color and depth images with size 640×480 at 30 frames per second. The semantic segmentation algorithm takes a 2-D color image and outputs a semantic label for each pixel in the image, at an average frame rate of 28.7 frames per second. By aligning the semantic image and the depth map, we derive a semantic 3-D point cloud which is utilized for Bayesian multi-class mapping. Our implementation was able to update the semantic OctoMap every 0.12 s, on average, while all of the computations where performed on the mobile robot. The experiment was carried out in an approximately 6 acre forested area shown in Fig. 15. The environment contained various terrain features, including asphalt road, gravel, grass, densely forested areas, and hills. Additionally, a number of buildings and other structures such as bleachers, tents, and cars add to the diversity of the type of object categories within the locale. The robot was manually controlled via joystick, and traveled the path shown in Fig. 15 (left) while incrementally building the semantic OctoMap. Fig. 16 shows the semantic mapping result overlaying the satellite image obtained via 2-D projection of the semantic OctoMap. We computed the memory size of the semantic OctoMap, and compared it with the corresponding regular

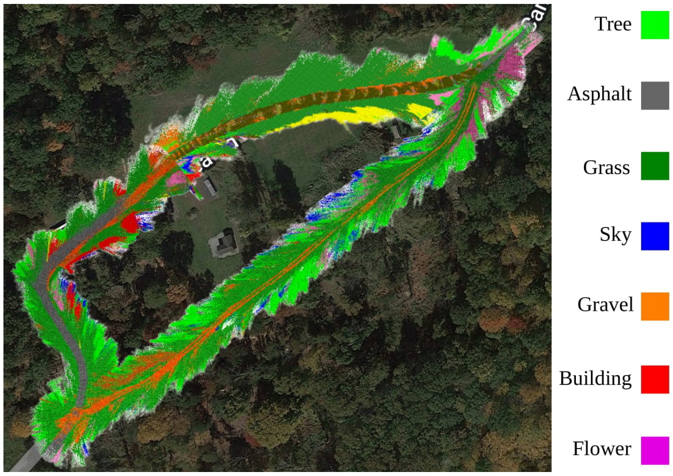


Fig. 16: Semantic mapping output overlaying the satellite image. The map is obtained via 2-D projection of the 3-D semantic OctoMap.

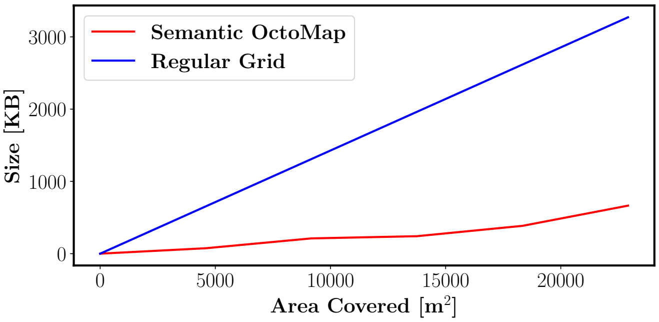


Fig. 17: Memory use of regular grid vs. semantic OctoMap

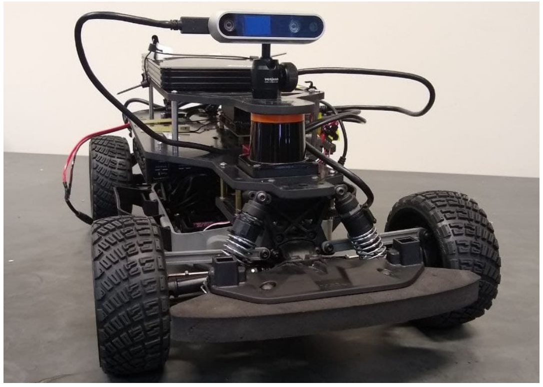


Fig. 18: Robot car used in indoor real-world experiments.

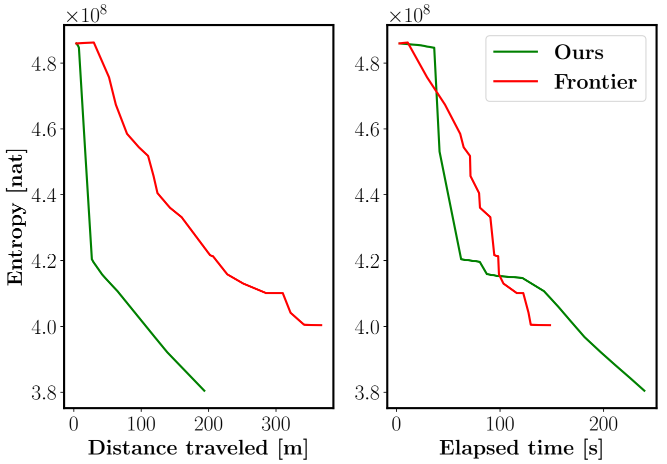


Fig. 19: Real-world experiment results for active mapping for 20 exploration iterations.

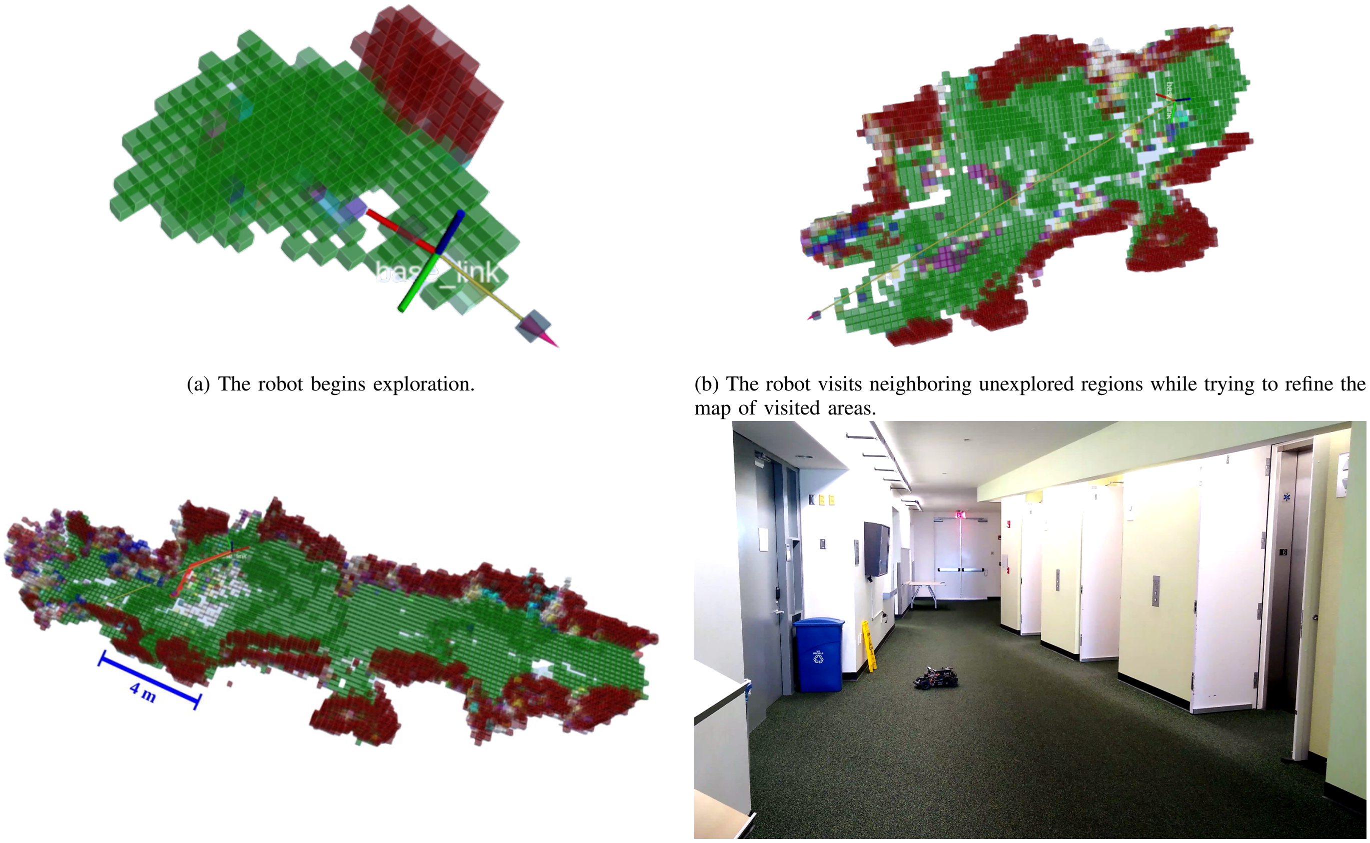
voxel grid representation, where each voxel contains the same amount of data as an OcTree leaf node at the lowest depth. Fig. 17 shows an almost five-fold saving in memory when using OcTree data structure. The importance of the memory savings of the OctoMap representation becomes more apparent when communication is considered. Our semantic OctoMap implementation resulted in a network bandwidth requirement of 238 KB/s for OctoMap, whereas a regular grid required 1173 KB/s for map communication.

G. 3-D Exploration in a Real-world Office Environment

We implemented SSMI on a ground wheeled robot to autonomously map an indoor office environment. Fig. 18 shows the robot equipped with a NVIDIA Xavier NX computer, a Hokuyo UST-10LX LiDAR, and an Intel RealSense D435i RGBD camera. Similar to the outdoor experiments, ROS was used for software deployment on the robot, and ICP laser scan matching provided localization. This time, we utilized a *ResNet18* [54] neural network architecture pre-trained on the SUN RGB-D dataset [55] for semantic segmentation. In particular, we employed the deep learning inference ROS nodes provided by NVIDIA [56], which are optimized for Xavier NX computers via TensorRT acceleration. Due to limited computational power available on the mobile platform, we operated the RGBD camera at a lower frame rate of 15 Hz with color and depth image size set to 640 × 480. The semantic segmentation algorithm was able to produce pixel classification images (resized to 512 × 400) at an average rate of 9.8 frames per second. Our implementation was able to publish semantic OctoMap ROS topics every $0.34s$, on average, with all of the processing occurred on the mobile platform. Fig. 20 depicts the exploration process, while Fig. 19 shows the performance of SSMI compared to frontier-based over 20 exploration iterations. We observe that, similar to the simulations, SSMI outperforms frontier-based exploration in terms of distance traveled. Also, SSMI shows on par performance compared to Frontier in terms of entropy reduction per time. This can be explained by the fact that large depth measurement noise and classification error in the real-world experiments result in (a) the need for re-visiting explored areas in order to estimate an accurate map, leading to poor entropy reduction for the frontier-based method and (b) a small number of safe candidate trajectories, leading to fewer computations to be performed by SSMI. Overall, our experiments show that SSMI outperforms Frontier in indoor exploration scenarios where the number, and length, of candidate trajectories is constrained by the size of the environment.

VIII. Conclusion

This paper developed techniques for active multi-class mapping of large 3-D environments using range and semantic



(a) The robot begins exploration.

(b) The robot visits neighboring unexplored regions while trying to refine the map of visited areas.

(c) Semantic OctoMap after 20 exploration iterations.

(d) Office environment featuring corridors, furniture, signs, and doors.

Fig. 20: Time lapse of autonomous exploration and multi-class mapping in the environment shown in (d). The exploration is run for 20 iterations. Different colors represent different semantic categories (floor, wall, furniture, etc.).

segmentation observations. Our results enable efficient mutual information computation over multi-class maps and make it possible to optimize for per-class uncertainty. Our experiments show that SSMI performs on par with the state of the art FSMI method in binary active mapping scenarios. However, when semantic information is considered SSMI outperforms existing algorithms and leads to efficient exploration and accurate multi-class mapping even in the presence of domain shift due to the difference between the classification training data and the testing environment. Experiments in both simulated and real-world environments showed the scalability of SSMI for large-scale 3-D exploration scenarios.

Appendix A Proof of Proposition 1

Applying Bayes rule in (2) and the factorization in (5) to $p_t(\mathbf{m})$ for some $\mathbf{z} \in \mathcal{Z}_{t+1}$ leads to:

$$\prod_{i=1}^{N} p(m_i|\mathcal{Z}_{1:t}, \mathbf{z}) = \frac{p(\mathbf{z})}{p(\mathbf{z}|\mathcal{Z}_{1:t})} \prod_{i=1}^{N} \frac{p(m_i|\mathbf{z})}{p(m_i)} p(m_i|\mathcal{Z}_{1:t}). \quad (21)$$

The term $\frac{p(\mathbf{z})}{p(\mathbf{z}|\mathcal{Z}_{1:t})}$ may be eliminated by considering the odds ratio of an arbitrary category $m_i = k_i \in \mathcal{K}$ versus the free category $m_i = 0$ for each cell i:

$$\prod_{i=1}^{N} \frac{p(m_i = k_i|\mathcal{Z}_{1:t}, \mathbf{z})}{p(m_i = 0|\mathcal{Z}_{1:t}, \mathbf{z})} \quad (22)$$
$$= \prod_{i=1}^{N} \frac{p(m_i = k_i|\mathbf{z})}{p(m_i = 0|\mathbf{z})} \frac{p(m_i = 0)}{p(m_i = k_i)} \frac{p(m_i = k_i|\mathcal{Z}_{1:t})}{p(m_i = 0|\mathcal{Z}_{1:t})}.$$

Since each term in both the left- and right-hand side products only depends on one map cell m_i, the expression holds for each individual cell. Re-writing the expression for cell m_i in vector form, with elements corresponding to each possible value of $k_i \in \mathcal{K}$, and taking an element-wise log leads to:

$$\left[\log \frac{p(m_i=0|\mathcal{Z}_{1:t},\mathbf{z})}{p(m_i=0|\mathcal{Z}_{1:t},\mathbf{z})} \quad \cdots \quad \log \frac{p(m_i=K|\mathcal{Z}_{1:t},\mathbf{z})}{p(m_i=0|\mathcal{Z}_{1:t},\mathbf{z})}\right]^{\top} \quad (23)$$
$$= (\mathbf{l}_i(\mathbf{z}) - \mathbf{h}_{0,i}) + \mathbf{h}_{t,i}.$$

Applying (23) recursively for each element $\mathbf{z} \in \mathcal{Z}_{t+1}$ leads to the desired result in (8). □

Appendix B Proof of Proposition 2

Let $\mathcal{R}_{t+1:t+T}(r_{max}) := \cup_{\tau,b} \mathcal{R}_{\tau,b}(r_{max})$ be the set of map indices which can potentially be observed by $\underline{\mathcal{Z}}_{t+1:t+T}$. Using the factorization in (5) and the fact that Shannon entropy is additive for mutually independent random variables, the mutual information only depends on the cells whose index belongs to $\mathcal{R}_{t+1:t+T}(r_{max})$, i.e.:

$$\begin{aligned} I(\mathbf{m}; &\underline{\mathcal{Z}}_{t+1:t+T} \mid \mathcal{Z}_{1:t}) \\ &= \sum_{\tau=t+1}^{t+T} \sum_{b=1}^{B} \sum_{i \in \mathcal{R}_{\tau,b}(r_{max})} I(m_i; \mathbf{z}_{\tau,b} \mid \mathcal{Z}_{1:t}). \end{aligned} \tag{24}$$

This is true because the measurements $\mathbf{z}_{\tau,b} \in \underline{\mathcal{Z}}_{t+1:t+T}$ are independent by construction and the terms $I(m_i; \underline{\mathcal{Z}}_{t+1:t+T} \mid \mathcal{Z}_{1:t})$ can be decomposed into sums of mutual information terms between single-beam measurements $\mathbf{z}_{\tau,b}$ and the respective observed map cells m_i. The mutual information between a single map cell m_i and a sensor ray $\mathbf{z}$ is:

$$\begin{aligned} &I(m_i; \mathbf{z} \mid \mathcal{Z}_{1:t}) = \\ &\int p(\mathbf{z} \mid \mathcal{Z}_{1:t}) \sum_{k=0}^{K} p(m_i = k \mid \mathbf{z}, \mathcal{Z}_{1:t}) \log \frac{p(m_i = k \mid \mathbf{z}, \mathcal{Z}_{1:t})}{p_t(m_i = k)} d\mathbf{z}. \end{aligned} \tag{25}$$

Using the inverse observation model in (10) and the Bayesian multi-class update in (8), we have:

$$\begin{aligned} &\sum_{k=0}^{K} p(m_i = k \mid \mathbf{z}, \mathcal{Z}_{1:t}) \log \frac{p(m_i = k \mid \mathbf{z}, \mathcal{Z}_{1:t})}{p_t(m_i = k)} \\ &= (\mathbf{l}_i(\mathbf{z}) - \mathbf{h}_{0,i})^\top \sigma(\mathbf{l}_i(\mathbf{z}) - \mathbf{h}_{0,i} + \mathbf{h}_{t,i}) + \log \frac{p(m_i = 0 \mid \mathbf{z}, \mathcal{Z}_{1:t})}{p_t(m_i = 0)} \\ &= f(\mathbf{l}_i(\mathbf{z}) - \mathbf{h}_{0,i}, \mathbf{h}_{t,i}), \end{aligned} \tag{26}$$

where (10) and (8) were applied a second time to the log term above. Plugging (26) back into the mutual information expression in (25) and returning to (24), we have:

$$\begin{aligned} &I(\mathbf{m}; \underline{\mathcal{Z}}_{t+1:t+T} \mid \mathcal{Z}_{1:t}) \\ &= \sum_{\tau=t+1}^{t+T} \sum_{b=1}^{B} \sum_{y=1}^{K} \int_0^{r_{max}} \Bigg(p(\mathbf{z}_{\tau,b} = (r, y) \mid \mathcal{Z}_{1:t}) \\ &\qquad \sum_{i \in \mathcal{R}_{\tau,b}(r_{max})} f(\mathbf{l}_i((r, y)) - \mathbf{h}_{0,i}, \mathbf{h}_{t,i}) \Bigg) dr. \end{aligned} \tag{27}$$

For $\mathbf{z}_{\tau,b} = (r, y)$, the second term inside the integral above can be simplified to:

$$\begin{aligned} \tilde{C}_{\tau,b}(r, y) := &\sum_{i \in \mathcal{R}_{\tau,b}(r_{max})} f(\mathbf{l}_i((r, y)) - \mathbf{h}_{0,i}, \mathbf{h}_{t,i}) \\ &= f(\boldsymbol{\phi}^+ + \mathbf{E}_{y+1} \boldsymbol{\psi}^+ - \mathbf{h}_{0,i^*_{\tau,b}}, \mathbf{h}_{t,i^*_{\tau,b}}) \\ &\quad + \sum_{i \in \mathcal{R}_{\tau,b}(r) \setminus \{i^*_{\tau,b}\}} f(\boldsymbol{\phi}^- - \mathbf{h}_{0,i}, \mathbf{h}_{t,i}) \end{aligned} \tag{28}$$

because for map indices $i \in \mathcal{R}_{\tau,b}(r_{max}) \setminus \mathcal{R}_{\tau,b}(r)$ that are not observed by $\mathbf{z}_{\tau,b}$, we have $\mathbf{l}_i((r, y)) = \mathbf{h}_{0,i}$ according to (10) and $f(\mathbf{h}_{0,i} - \mathbf{h}_{0,i}, \mathbf{h}_{t,i}) = 0$.

Next, we apply the definition of (11) for the first term in the integral in (27), which turns it into an integration over $\tilde{p}_{\tau,b}(r, y) \tilde{C}_{\tau,b}(r, y)$. Note that $\tilde{p}_{\tau,b}(r, y)$ and $\tilde{C}_{\tau,b}(r, y)$ are piecewise-constant functions since $\mathcal{R}_{\tau,b}(r)$ is constant with respect to r as long as the beam $\mathbf{z}$ lands in cell m_{i^*}. Hence, we can partition the integration domain over r into a union of intervals where the beam $\mathbf{z}$ hits the same cell, i.e., $\mathcal{R}_{\tau,b}(r)$ remains constant:

$$\int_0^{r_{max}} \tilde{p}_{\tau,b}(r, y) \tilde{C}_{\tau,b}(r, y)\, dr = \sum_{n=1}^{N_{\tau,b}} \int_{r_{n-1}}^{r_n} \tilde{p}_{\tau,b}(r, y) \tilde{C}_{\tau,b}(r, y)\, dr,$$

where $N_{\tau,b} = |\mathcal{R}_{\tau,b}(r_{max})|$, $r_0 = 0$, and $r_N = r_{max}$. From the piecewise-constant property of $\tilde{p}_{\tau,b}(r, y)$ and $\tilde{C}_{\tau,b}(r, y)$ over the interval $(r_{n-1}, r_n]$, one can obtain:

$$\begin{aligned} &\int_{r_{n-1}}^{r_n} \tilde{p}_{\tau,b}(r, y) \tilde{C}_{\tau,b}(r, y)\, dr \\ &\qquad = \tilde{p}_{\tau,b}(r_n, y) \tilde{C}_{\tau,b}(r_n, y) \gamma(n) = p_{\tau,b}(n, y) C_{\tau,b}(n, y), \end{aligned} \tag{29}$$

where $p_{\tau,b}(n, y)$ and $C_{\tau,b}(n, y)$ are defined in the statement of Proposition 2. Substituting y with k and plugging the integration result into (27) yields the lower bound in (13) for the mutual information between $\mathbf{m}$ and $\mathcal{Z}_{t+1:t+T}$. □

Appendix C Proof of Proposition 3

Consider a single beam $\mathbf{z}_{\tau,b}$, passing through cells $\{m_i\}_i$, $i \in \mathcal{R}_{\tau,b}(r_{max})$. As shown in Appendix B, the mutual information between the map $\mathbf{m}$ and a beam $\mathbf{z}_{\tau,b}$ can be computed as:

$$I(\mathbf{m}; \mathbf{z}_{\tau,b} | \mathcal{Z}_{1:t}) = \sum_{k=1}^{K} \sum_{n=1}^{N_{\tau,b}} p_{\tau,b}(n, k) C_{\tau,b}(n, k). \tag{30}$$

Assuming piece-wise constant class probabilities, we have:

$$\begin{aligned} &\sum_{n=1}^{N_{\tau,b}} p_{\tau,b}(n, k) C_{\tau,b}(n, k) = \\ &\qquad \sum_{q=1}^{Q_{\tau,b}} \sum_{n=\omega_{\tau,b,1:q-1}+1}^{\omega_{\tau,b,1:q}} p_{\tau,b}(n, k) C_{\tau,b}(n, k), \end{aligned} \tag{31}$$

where $\omega_{\tau,b,1:q} = \sum_{j=1}^{q} \omega_{\tau,b,j}$. For each $\omega_{\tau,b,1:q-1} < n \leq \omega_{\tau,b,1:q}$, the terms $p_{\tau,b}(n, k)$ and $C_{\tau,b}(n, k)$ are expressed as:

$$\begin{aligned} p_{\tau,b}(n, k) &= \pi_t(q, k) \pi_t^{(n-1-\omega_{\tau,b,1:q-1})}(q, 0) \prod_{j=1}^{q-1} \pi_t^{\omega_{\tau,b,j}}(j, 0), \\ C_{\tau,b}(n, k) &= f(\boldsymbol{\phi}^+ + \mathbf{E}_{k+1} \boldsymbol{\psi}^+ - \boldsymbol{\chi}_{0,q}, \boldsymbol{\chi}_{t,q}) + \\ &\qquad (n - 1 - \omega_{\tau,b,1:q-1}) f(\boldsymbol{\phi}^- - \boldsymbol{\chi}_{0,q}, \boldsymbol{\chi}_{t,q}) + \\ &\qquad\qquad \sum_{j=1}^{q-1} \omega_{\tau,b,j} f(\boldsymbol{\phi}^- - \boldsymbol{\chi}_{0,j}, \boldsymbol{\chi}_{t,j}). \end{aligned}$$

Plugging this into the inner summation of (31) leads to:

$$\begin{aligned} &\sum_{n=\omega_{\tau,b,1:q-1}+1}^{\omega_{\tau,b,1:q}} p_{\tau,b}(n, k) C_{\tau,b}(n, k) = \\ &\qquad \rho_{\tau,b}(q, k) \big[\beta_{\tau,b}(q, k) \sum_{j=0}^{\omega_{\tau,b,q}-1} \pi_t^j(q, 0) + \\ &\qquad\qquad f(\boldsymbol{\phi}^- - \boldsymbol{\chi}_{0,q}, \boldsymbol{\chi}_{t,q}) \sum_{j=0}^{\omega_{\tau,b,q}-1} j \pi_t^j(q, 0) \big], \end{aligned} \tag{32}$$

The summations in (32) can be computed explicitly, leading to the following closed-form expression:

$$\begin{aligned}
&\beta_{\tau,b}(q,k) \sum_{j=0}^{\omega_{\tau,b,q}-1} \pi_t^j(q,0) + \\
&\quad f(\boldsymbol{\phi}^- - \boldsymbol{\chi}_{0,q}, \boldsymbol{\chi}_{t,q}) \sum_{j=0}^{\omega_{\tau,b,q}-1} j\pi_t^j(q,0) = \\
&\beta_{\tau,b}(q,k) \frac{1-\pi_t^{\omega_{\tau,b,q}}(q,0)}{1-\pi_t(q,0)} + \\
&\quad \frac{f(\boldsymbol{\phi}^- - \boldsymbol{\chi}_{0,q}, \boldsymbol{\chi}_{t,q})}{(1-\pi_t(q,0))^2} \Big[(\omega_{\tau,b,q}-1)\pi_t^{\omega_{\tau,b,q}+1}(q,0) \\
&\qquad - \omega_{\tau,b,q}\pi_t^{\omega_{\tau,b,q}}(q,0) + \pi_t(q,0) \Big] = \Theta_{\tau,b}(q,k).
\end{aligned} \tag{33}$$

Therefore, the Shannon mutual information between a semantic OctoMap $\mathbf{m}$ and a range-category measurement $\mathbf{z}_{\tau,b}$ can be computed as in (15). □

References

[1] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, 1989.

[2] S. Thrun, "Learning occupancy grid maps with forward sensor models," *Autonomous Robots*, vol. 15, no. 2, pp. 111–127, 2003.

[3] 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.

[4] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser, "Semantic scene completion from a single depth image," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1746–1754.

[5] L. Gan, R. Zhang, J. W. Grizzle, R. M. Eustice, and M. Ghaffari, "Bayesian spatial kernel smoothing for scalable dense semantic mapping," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 790–797, 2020.

[6] T. Wang, V. Dhiman, and N. Atanasov, "Learning navigation costs from demonstration with semantic observations," *Proceedings of Machine Learning Research vol*, vol. 120, pp. 1–11, 2020.

[7] A. Milioto and C. Stachniss, "Bonnet: An Open-Source Training and Deployment Framework for Semantic Segmentation in Robotics using CNNs," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[8] B. J. Julian, S. Karaman, and D. Rus, "On mutual information-based control of range sensing robots for mapping applications," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013, pp. 5156–5163.

[9] B. Charrow, S. Liu, V. Kumar, and N. Michael, "Information-theoretic mapping using cauchy-schwarz quadratic mutual information," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 4791–4798.

[10] Z. Zhang, T. Henderson, V. Sze, and S. Karaman, "FSMI: Fast Computation of Shannon Mutual Information for Information-Theoretic Mapping," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019, pp. 6912–6918.

[11] A. Asgharivaskasi and N. Atanasov, "Active bayesian multi-class mapping from range and semantic segmentation observations," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 1–7.

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

[13] W. Burgard, M. Moors, C. Stachniss, and F. E. Schneider, "Coordinated multi-robot exploration," *IEEE Transactions on Robotics (TRO)*, vol. 21, no. 3, pp. 376–386, 2005.

[14] H. H. González-Baños and J.-C. Latombe, "Navigation strategies for exploring indoor environments," *The International Journal of Robotics Research (IJRR)*, vol. 21, no. 10-11, pp. 829–848, 2002.

[15] C. Gómez, M. Fehr, A. C. Hernández, J. Nieto, R. Barber, and R. Siegwart, "Hybrid Topological and 3D Dense Mapping through Autonomous Exploration for Large Indoor Environments," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[16] X. Kan, H. Teng, and K. Karydis, "Online exploration and coverage planning in unknown obstacle-cluttered environments," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5969–5976, 2020.

[17] R. Maffei, M. P. Souza, M. Mantelli, D. Pittol, M. Kolberg, and V. A. M. Jorge, "Exploration of 3D terrains using potential fields with elevation-based local distortions," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[18] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, "Receding horizon "next-best-view" planner for 3d exploration," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1462–1468.

[19] T. Dang, F. Mascarich, S. Khattak, C. Papachristos, and K. Alexis, "Graph-based path planning for autonomous robotic exploration in subterranean environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 3105–3112.

[20] H. Zhu, C. Cao, Y. Xia, S. Scherer, J. Zhang, and W. Wang, "Dsvp: Dual-stage viewpoint planner for rapid exploration by dynamic expansion," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 7623–7630.

[21] C. Cao, H. Zhu, H. Choset, and J. Zhang, "Tare: A hierarchical framework for efficiently exploring complex 3d environments." in *Robotics: Science and Systems (RSS)*, 2021.

[22] C. M and N. Michael, "Volumetric objectives for multi-robot exploration of three-dimensional environments," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 9043–9050.

[23] A. Elfes, "Robot navigation: Integrating perception, environmental constraints and task execution within a probabilistic framework," in *Reasoning With Uncertainty in Robotics*, 1995, pp. 93–130.

[24] S. J. Moorehead, R. Simmons, and W. L. Whittaker, "Autonomous exploration using multiple sources of information," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2001, pp. 3098–3103.

[25] F. Bourgault, A. A. Makarenko, S. B. Williams, B. Grocholsky, and H. F. Durrant-Whyte, "Information based adaptive robotic exploration," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2002, pp. 540–545.

[26] A. Visser and B. Slamet, "Balancing the information gain against the movement cost for multi-robot frontier exploration," in *European Robotics Symposium*, 2008, pp. 43–52.

[27] T. Kollar and N. Roy, "Efficient optimization of information-theoretic exploration in SLAM," in *AAAI Conference on Artificial Intelligence*, 2008, p. 1369–1375.

[28] H. Carrillo, I. Reid, and J. A. Castellanos, "On the comparison of uncertainty criteria for active SLAM," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 2080–2087.

[29] L. Carlone, J. Du, M. K. Ng, B. Bona, and M. Indri, "Active SLAM and exploration with particle filters using Kullback-Leibler divergence," *Journal of Intelligent & Robotic Systems*, vol. 75, no. 2, pp. 291–311, 2014.

[30] N. Atanasov, J. Le Ny, K. Daniilidis, and G. J. Pappas, "Decentralized active information acquisition: Theory and application to multi-robot SLAM," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 4775–4782.

[31] J. Wang, T. Shan, and B. Englot, "Virtual maps for autonomous exploration with pose SLAM," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2019.

[32] C. Stachniss, G. Grisetti, and W. Burgard, "Information gain-based exploration using Rao-Blackwellized particle filters," in *Robotics: Science and Systems*, 2005, pp. 65–72.

[33] T. Henderson, V. Sze, and S. Karaman, "An efficient and continuous approach to information-theoretic exploration," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[34] F. Chen, P. Szenher, Y. Huang, J. Wang, T. Shan, S. Bai, and B. Englot, "Zero-shot reinforcement learning on graphs for autonomous exploration under uncertainty," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 5193–5199.

[35] M. Lodel, B. Brito, A. Serra-Gómez, L. Ferranti, R. Babuška, and J. Alonso-Mora, "Where to look next: Learning viewpoint recommendations for informative trajectory planning," *arXiv preprint arXiv:2203.02381*, 2022.

[36] E. Zwecher, E. Iceland, S. R. Levy, S. Y. Hayoun, O. Gal, and A. Barel, "Integrating deep reinforcement and supervised learning to expedite indoor mapping," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2022, pp. 10 542–10 548.

[37] H. Zhang, J. Cheng, L. Zhang, Y. Li, and W. Zhang, "H2GNN: Hierarchical-hops graph neural networks for multi-robot exploration in unknown environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3435–3442, 2022.

[38] J. Hu, H. Niu, J. Carrasco, B. Lennox, and F. Arvin, "Voronoi-based multi-robot autonomous exploration in unknown environments via deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 14 413–14 423, 2020.

[39] T. S. Veiga, M. Silva, R. Ventura, and P. U. Lima, "A hierarchical approach to active semantic mapping using probabilistic logic and information reward pomdps," *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 29, no. 1, pp. 773–781, May 2021.

[40] V. Suriani, S. Kaszuba, S. R. Sabbella, F. Riccio, and D. Nardi, "S-ave: Semantic active vision exploration and mapping of indoor environments for mobile robots," in *2021 European Conference on Mobile Robots (ECMR)*, 2021, pp. 1–8.

[41] H. Blum, S. Rohrbach, M. Popovic, L. Bartolomei, and R. Y. Siegwart, "Active learning for uav-based semantic mapping," *ArXiv*, vol. abs/1908.11157, 2019.

[42] G. Georgakis, B. Bucher, K. Schmeckpeper, S. Singh, and K. Daniilidis, "Learning to map for active semantic goal navigation," *ArXiv*, vol. abs/2106.15648, 2021.

[43] T. D. Barfoot, *State estimation for robotics*. Cambridge University Press, 2017.

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

[45] J. E. Bresenham, "Algorithm for computer control of a digital plotter," *IBM Systems Journal*, vol. 4, no. 1, pp. 25–30, 1965.

[46] H. Samet, "Implementing ray tracing with octrees and neighbor finding," *Computers & Graphics*, vol. 13, no. 4, pp. 445–460, 1989.

[47] M. Agate, R. L. Grimsdale, and P. F. Lister, "The hero algorithm for ray-tracing octrees," in *Advances in Computer Graphics Hardware IV*, 1991, pp. 61–73.

[48] K. Saulnier, N. Atanasov, G. J. Pappas, and V. Kumar, "Information theoretic active exploration in signed distance fields," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[49] A. Asgharivaskasi, S. Koga, and N. Atanasov, "Active mapping via gradient ascent optimization of shannon mutual information over continuous se(3) trajectories," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022.

[50] 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*, vol. 3, no. 3.2, 2009, p. 5.

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

[52] P. Chao, "Fully convolutional hardnet for segmentation in pytorch," https://github.com/PingoLH/FCHarDNet, accessed: 2022-07-01.

[53] M. Wigness, S. Eum, J. G. Rogers, D. Han, and H. Kwon, "A RUGD dataset for autonomous navigation and visual perception in unstructured outdoor environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 5000–5007.

[54] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[55] S. Song, S. P. Lichtenberg, and J. Xiao, "Sun RGB-D: A RGB-D scene understanding benchmark suite," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 567–576.

[56] NVIDIA, "Deep learning nodes for ROS/ROS2," https://github.com/dusty-nv/ros_deep_learning, accessed: 2021-07-10.

Arash Asgharivaskasi (S'22) is a Ph.D. student of Electrical and Computer Engineering at the University of California San Diego, La Jolla, CA. He obtained a B.S. degree in Electrical Engineering from Sharif University of Technology, Tehran, Iran, and an M.S. degree in Electrical and Computer Engineering from the University of California San Diego, La Jolla, CA, USA. His research focuses on active information acquisition using mobile robots with applications to mapping, security, and environmental monitoring.

Nikolay Atanasov (S'07-M'16) is an Assistant Professor of Electrical and Computer Engineering at the University of California San Diego, La Jolla, CA, USA. He obtained a B.S. degree in Electrical Engineering from Trinity College, Hartford, CT, USA, in 2008 and M.S. and Ph.D. degrees in Electrical and Systems Engineering from the University of Pennsylvania, Philadelphia, PA, USA, in 2012 and 2015, respectively. His research focuses on robotics, control theory, and machine learning, applied to active sensing using ground and aerial robots. He works on probabilistic perception models that unify geometry and semantics and on optimal control and reinforcement learning approaches for minimizing uncertainty in these models. Dr. Atanasov's work has been recognized by the Joseph and Rosaline Wolf award for the best Ph.D. dissertation in Electrical and Systems Engineering at the University of Pennsylvania in 2015, the best conference paper award at the International Conference on Robotics and Automation in 2017, and the NSF CAREER award in 2021.