IEEE Robotics and Automation Letters (RAL) paper presented at
2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Semantics-Aware Receding Horizon Planner for Object-Centric Active Mapping

Liang Lu, Yinqiang Zhang, Peng Zhou, *Member, IEEE,* Jiaming Qi, Yipeng Pan, Changhong Fu, *Member, IEEE,* Jia Pan, *Senior Member, IEEE,*

***Abstract*—The escalating demands for real-time scene comprehension in modern industries underscore the growing significance of semantic information in the daily tasks of robots, particularly in areas like autonomous inspection and target searching. This letter introduces a semantics-aware receding horizon planner (SARHP) for efficiently building the object-centric volumetric map. It includes a multi-layer mapping strategy and a semantics-aware frontier detection and planning method. With the multi-layer map, the semantics-aware frontier detection is conducted in the local layer, and the route assessment is conducted in the Field-of-View layer, which can reduce the time cost of the planning stage. Moreover, kinematic cost, geometric cost, and semantic cost are considered in the planner to ensure high search performance for semantic objects without affecting the overall mapping efficiency. The effectiveness of the proposed mapping and planning algorithm is validated in simulation and real-world experiments.**

***Index Terms*—Mobile Robotic System, Perception and Autonomy, Semantic Scene Understanding**

I. Introduction

Traditional active mapping technology that only builds a map with obstacle and non-obstacle areas is insufficient to cope with increasingly sophisticated tasks in current society [1], [2]. In the construction industry, the robot needs to distinguish between parts belonging or not belonging to the building information modeling to achieve construction progress monitoring, and in nuclear plant inspection tasks, the robot needs to find dangerous objects, both of which require semantic information to be considered in planning and mapping. Thus, to enhance the ability of real-time scene understanding lacking in traditional active mapping technology, object-centric active mapping becomes a key research topic in the robotic community [3], [4]. Properly integrating semantic information into the planner can significantly improve the efficiency of object-centric planning and mapping.

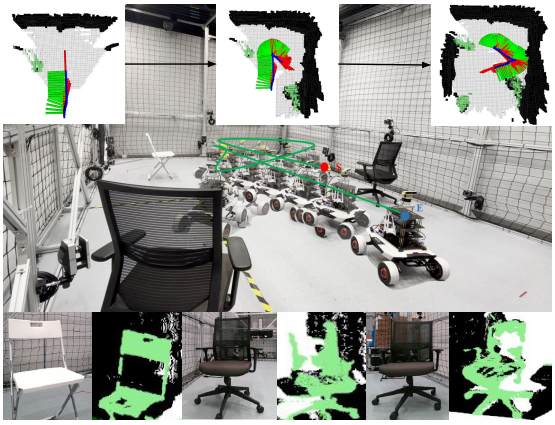


Fig. 1: (Up) The semantic planning and mapping process. The green, black, and grey grids represent the target, nuisance, and ground objects. (Middle) The side view of the exploration process using SARHP and the vehicle's path is in green. The start point and end point are in red and blue, respectively. (Bottom) The RGB image of the target objects and their corresponding semantic point cloud.

Most of the time, the robot has no prior knowledge or only a little knowledge about the environment. Thus, exploring the unknown environment and constructing the object map in real-time is required while considering collision and kinematic constraints. As shown in Fig. 1, unlike traditional active mapping, the object-centric active mapping problem in this letter requires a robot to effectively explore and build an object map of the unknown environment, relying on onboard sensing for object-level observations generated by semantic segmentation over RGBD images. Our work differs from other receding horizon planning approaches [5]–[7]; instead of evaluating branches, we assess the receding horizon routes that satisfy kinematic constraints to reach sub-goals. The benefits of evaluating the routes are that the planner not only considers how to reduce the map's uncertainty but also considers the kinematic constraints in a short horizon. Moreover, in reducing map uncertainty, the entire route is evaluated to ensure the completeness of the assessment process. A sliding window approach is used to iterate the route points to maintain the evaluation efficiency. Compared with the sum of information gain value from the points at the previous route point's Field-

Manuscript received November 23, 2023; revised February 13, 2024; accepted February 22, 2024.

This paper was recommended for publication by Editor Markus Vince upon evaluation of the Associate Editor and Reviewers' comments.

*This work was partially supported by the Innovation and Technology Commission of the HKSAR government under the InnoHK initiative. (Corresponding author:Jia Pan.)

Liang Lu, Peng Zhou, Jiaming Qi, Yipeng Pan, and Jia Pan are with Centre for Transformative Garment Production, Bldg 19W, Hong Kong Science Park, Hong Kong, China and also with the Department of Computer Science, The University of Hong Kong, Pok Fu Lam, Hong Kong, China. (llu92@hku.hk; jeffzhou@hku.hk; qijm_hit@163.com; yppancs@connect.hku.hk; jpan@cs.hku.hk)

Yinqiang Zhang is with the Department of Computer Science, The University of Hong Kong, Pok Fu Lam, Hong Kong, China. (zyq507@connect.hku.hk)

Changhong Fu is with the School of Mechanical Engineering, Tongji University, Shanghai 201804, China. (changhongfu@tongji.edu.cn)

Of-View (FOV), only a few points need to be newly evaluated at the current route point's FOV. This work also adopts multi-layer mapping to evaluate candidate routes in the FOV and frontier detection in a local map to reduce the time cost. In sum, the main contributions are as follows.

1) An efficient multi-layer object mapping that constructs the object-centric map from semantic segmentation over RGBD observation provides the planner with a suitably scaled map for effective planning.
2) A semantics-aware receding horizon planner (SARHP) considers kinematic constraints, geometric mapping uncertainty, and semantic mapping uncertainty of the candidate route that can efficiently map the semantic objects while maintaining an overall mapping performance.

The rest of the letter is organized as follows. In Section II, the related works are introduced. The problem statement and system overview are explained in Section III-A and Section III-B, respectively. The object-centric mapping and semantics-aware planning approach are described in Section IV. The simulation and real-world experimental results are presented in Section V. Finally, Section VI concludes the letter and proposes future research directions.

II. Related Work

In this section, we overview the state-of-the-art (SOTA) of two significant parts of this work: receding horizon planning and object-centric active mapping.

A. Receding Horizon Planning

The concept of receding horizon is widely used in robotic control and planning. Instead of completely executing the best branch, Receding Horizon Planning (RHP) requires the robot to walk through the first edge of this branch and then recalculate the best branch based on the updated map information. A well-known work that used RHP is Receding Horizon Next Best View Planning (RH-NBVP) [5]. It has three main steps. 1) generating candidate points by Rapidly-exploring Random Trees (RRTs). 2) selecting the best branch by maximizing the utility function constructed by the exploration gain and the travelling distance. 3) exploring the first edge of this branch. After RH-NBVP was proposed, there were many extensions. For example, [6], [7] used RH-NBVP as a local planner and another global planner to guide the robot to the cached points with high value when there is no nearby potential gain, and [8] used an RRT*-inspired operation to continuously expand and maintain a single tree that can refine trajectories to maximize their utility. Instead of RRTs, [9]–[11] build branch graphs to store the connections between candidate points. [12]–[14] take perception quality into RHP's account. The uncertainty-aware receding horizon planning [12] that selected the branch could minimize the expected localization and mapping uncertainty. [13] considers visual saliency, the distinct subjective perceptual quality of certain objects, a main factor in RHP. [14] integrates active vision in a receding horizon navigation to improve state estimation accuracy in difficult scenes ($e.g.$, weak texture).

B. Object-centric Active Mapping

Object-centric active mapping, or called semantic active mapping, or autonomous object-centric exploration, is a new research direction in recent years. [15] used a neural network-based detector to detect the bounding box of target objects and maximize their resolution in a semantically–annotated occupancy map. The planning part followed the planning process of RH-NBVP with the exploration gain and object gain. [16] proposed a semantic visible voxel utility function and a semantic visited object of interest visual voxel utility function. The former was used to direct the robot toward the views containing occupied voxels with a small confidence value of objects. In contrast, the latter was used to control the robot toward occupied voxels classified as an object of interest but not visited sufficiently. [17] adopted a geometric and semantic reconstruction entropy to perform online semantic reconstruction. [18] applied Bayesian fusion of volumetric and semantic information in a 3D octree structure and an RH-NBVP planner, which considers these two pieces of information to map the environment. [19] formulated active domain adaptation as an information path planning problem and proposed an information gain considering unobserved and semantic surface voxels. [20] presented a semantic-centric exploration algorithm with an aerial robot to find things in an unknown environment. Their algorithms have high-quality object reconstructions while preventing the exploration from getting stuck when depth measurements consistently cannot be obtained. [21] proposed a semantics-aware exploration and inspection planner with three planning behaviors: volumetric exploration, semantics hole coverage, and semantics inspection. It advances GBPlanner [9] in reconstructing the semantic surface. More recently, [22] proposed an efficiently computable lower bound for the Shannon mutual information between a multiclass octree map and a set of range-category measurements. [23] adopted active learning in active semantic mapping that can reduce the human labeling effort for continuous robotic perception improvement. Other works, such as [24], [25], use multi-robot for semantic object mapping. Contemporary works in robot navigation that used Large Language Model (LLM)-based planning [26], [27], open-set mapping [28], [29] and end-to-end learning of local control [30] also show the potential to be used in the object-centric active mapping.

This letter considers the idea of perception-aware RHP from [14], using a lattice sampling method [31] to generate candidate collision-free trajectories while adding semantics-aware cost to improve the object-centric mapping ability. Unlike the literature above, the cost function's kinematic constraints are also considered to select the candidate trajectory. Moreover, compatible with multi-layer maps to provide a suitably scaled map, the time cost of frontier detection and candidate route selection does not increase when the size of the overall map is increased.

III. Preliminaries

In this section, the problem of semantic mapping and planning is first formulated, and then the robotic system used is introduced.

A. Problem Statement

The space is generally divided into occupied, free, and unknown areas, each classified based on occupancy probability. Occupied areas have an occupied probability higher than a threshold, free areas have an occupied probability lower than a threshold, and the other areas are unknown. This work divides occupied areas into Target Objects (TOs) and Nuisance Objects (NOs). The criterion to distinguish TOs from NOs is that objects that the task cares about are listed as TOs. In contrast, other objects the task does not care about are listed as NOs. The main purpose of this division is to utilize TOs to improve planner efficiency and reduce the interference of NOs on tasks. TOs and NOs are observed using onboard sensors through a pre-trained classifier operating on the map. The problem considered in this work is mapping a bounded 3D space $V \subset \mathbb{R}^3$ with target objects $V_{tos} \subset V$ and nuisance objects $V_{nos} \subset V$. This determines which parts of the initially unmapped space $V_{unm} \overset{init.}{=} V$ are free $V_{free} \subset V$, target objects $V_{tos} \subset V$, or nuisance objects $V_{nos} \subset V$ while minimize the uncertainty of $V_{tos} \subset V$. The operation is subject to vehicle kinematic constraints, localization uncertainty, and limitations of the employed sensor system with which the space is mapped. The entire space cannot be scanned due to the vehicle's kinematic constraints and sensing limitations. Thus, not all the space can be mapped properly. This residual space is denoted by V_{res}. The problem is considered fully solved when $V_{free} \cup V_{tos} \cup V_{nos} = V \setminus V_{res}$. To sum up, the planner's goal is to find a kinematic-friendly path within a short horizon that contains rich semantic information of TOs to explore and finally map the entire available space $V \setminus V_{res}$.

B. System Overview

The vehicle used in this work is a wheeled vehicle equipped with an RGB-D camera as a main sensor to perceive the surrounding environment and a 3D LiDAR for localization. The model of the wheeled vehicle is simplified as a square box of $w \times l$. The RGB-D camera has a resolution of $W \times H$, the focal length of f_x and f_y, the camera center (p_x, p_y), and the depth $d \in [d_{min}, d_{max}] \in \mathbb{R}^+$. The RGB image and the depth image are synchronized. The state $\boldsymbol{x}$ of the wheeled vehicle consists of its position vector $\boldsymbol{p} = [x, y]^T \in V$.

IV. Methodology

To deal with the formulated problem, we proposed a front end of an object-centric mapping approach to map V_{tos} and V_{nos} in a multi-layer volumetric map and a back end of the semantics-aware planning strategy that guides the vehicle to map V_{tos}. In this section, we first introduce the object-centric mapping approach and then present the semantics-aware planning strategy. The overview of the proposed methodology is shown in Fig. 2.

A. Object-centric Mapping

This subsection introduces the object-centric mapping method encompassing semantic object perception and multi-layer volumetric mapping. Semantic object perception generates the semantic point cloud from the camera measurements,

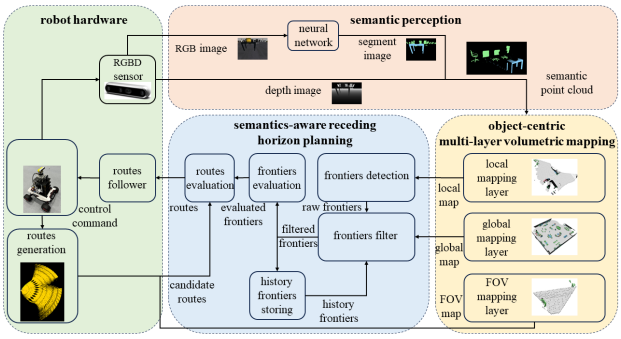


Fig. 2: The overview of the semantics-aware planning and the object-centric mapping methodology.

while multi-layer volumetric mapping maps the semantic point cloud into a multi-layer volumetric map.

1) Semantic Object Perception: The semantic object perception method uses a neural network that combines YOLOv8 detector[1] trained on the Microsoft COCO dataset[2] and Segment Anything [32] to produce semantic object masks from the RGB image. For the pixel (u_i, v_i) on the image plane, the detector outputs a probability distribution $p_o(u_i, v_i) \in P_N^K$, which is the probability that the image pixel (u_i, v_i) within the detected bounding boxes belongs to known classes and Segment Anything outputs the corresponding colour $c_o(u_i, v_i) \in C_N^K$ over the set of known classes $N \in \mathbb{N}^+$, where K represents the number of known classes and $\mathbb{N}^+$ represents the set of non-zero natural numbers. The masks of TOs with corresponding pixels in the synchronized depth image are selected to generate the semantic point cloud based on the extrinsic calibration parametric model. The equation to project 2D pixels into 3D in the camera frame is shown in (1).

$$\begin{bmatrix} x_{c_i} \\ y_{c_i} \\ z_{c_i} \end{bmatrix} = \begin{bmatrix} \frac{1}{f_x} & 0 & -\frac{p_x}{f_x} \\ 0 & \frac{1}{f_y} & \frac{p_y}{f_y} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_i \cdot z_{c_i} \\ v_i \cdot z_{c_i} \\ z_{c_i} \end{bmatrix}, \tag{1}$$

Where $(x_{c_i}, y_{c_i}, z_{c_i})$ are the coordinates of 3D points regarding the TOs' masks in the camera frame, z_{c_i} is the depth of the pixels on the depth image.

$$SP_w = T_c^w \cdot SP_c, \tag{2}$$

In (2), T_c^w denotes the homogeneous transformation matrix from the camera to the world frame, and the coordinates of the semantic point cloud in the world frame are $SP_w = [x_{w_i}, y_{w_i}, z_{w_i}, c_{tos}(u_i, v_i), p_{tos}(u_i, v_i)]$, $i \in \mathbb{N}^+$, where the $c_{tos}(u_i, v_i)$ is the objects' colour of the pixels (u_i, v_i) on the TOs' mask while the $p_{tos}(u_i, v_i)$ is the probability distribution of the TOs. SP_w are calculated using T_c^w multiply the semantic point cloud in the camera frame SP_c. SP_c is defined as $(x_{c_i}, y_{c_i}, z_{c_i}, c_{tos}(u_i, v_i), p_{tos}(u_i, v_i)$, $i \in \mathbb{N}^+$. A voxel filter[3] is then used to down-sample the semantic point cloud before sending it to the volumetric mapping process.

2) Multi-layer Volumetric Mapping: In the multi-layer volumetric mapping method, the semantic occupancy grid map stores the probability of occupancy, the semantic colour, and

[1] https://github.com/ultralytics/ultralytics
[2] https://cocodataset.org/#home
[3] https://pointclouds.org/documentation/tutorials/voxel_grid.html

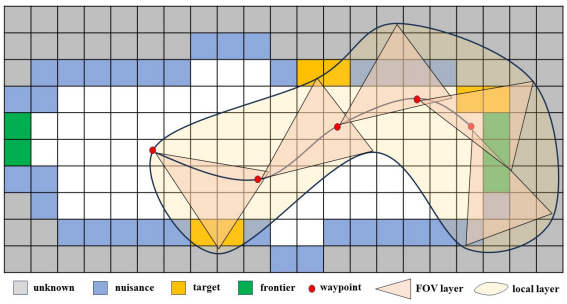


Fig. 3: Multi-layer volumetric mapping strategy. The global layer is the whole map.

the probability distribution of the TOs. It keeps integrating new semantic occupancy information data obtained from raycasting when the new semantic point cloud measurement is given. The probabilistic occupancy information, the semantic colour information, and the probabilistic semantic information are stored in an octree structure of voxels [33]. The updating of the semantic map follows a maximum fusion strategy [34]. Suppose the current observation of the semantic colour of the voxel is the same as the last one. In that case, the semantic colour is kept, and the probability is the average of the maximum likelihood of the two semantics. On the contrary, the semantics with a higher probability are preserved, and we decrease the voxel's maximum probability by p_{dis} as a punishment for disagreement.

As shown in Fig.3, we propose a three-layer volumetric map which includes ***FOV mapping layer***, the ***local mapping layer***, and the ***global mapping layer*** to represent the surrounding environment. Given semantic volumetric map measurements $m^w_{1:T} = \{m^{tos}_{1:T}; m^{nos}_{1:T}\}$ in the FOV and camera poses $p^w_{1:T}$ in the world frame from time 1 to T, the FOV mapping layer builds the semantic volumetric map $m^w_{t_k}$ at a certain time $t_k \in \{1:T\}$, the local mapping layer builds the semantic volumetric map $m^w_{t_i:t_j}$ within a certain time interval $\{t_i:t_j\} \in \{1:T\}$ and the global mapping layer builds the semantic volumetric map $m^w_{1:T}$ throughout the time period $\{t_1:t_T\} \in \{1:T\}$. The multi-layer volumetric mapping aims to accelerate the process of semantics-aware frontier detection and receding horizon planning. Taking into account that the time complexity of searching in the octree is $\mathcal{O}(\log(n))$, the iteration time increases dramatically when $m^w_{1:T}$ increases. Instead of iterating $m^w_{1:T}$ at each planning loop, it is more efficient to iterate $m^w_{t_k}$ for receding horizon planning and iterate $m^w_{t_i:t_j}$ for frontiers detection. Details related to receding horizon planning and frontiers detection are introduced in the following section.

B. Semantics-Aware Planning

In the subsection, we introduce semantics-aware planning, which includes a semantic frontier detector to find the optimal semantic frontier to guide the vehicle to explore it and SARHP to select the route with rich semantic information at the next moment to move.

1) Semantics-Aware Information Gain: Before introducing the frontier detector and the planner, we first introduce the semantics-aware information gain I_{sum}, the weighted sum of the geometric reconstruction information gain I_g, and the semantic reconstruction information gain I_s. I_{sum} measures how much geometric and semantic uncertainty is reduced at a potential scanning view. Both I_g and I_s are computed using Shannon's information entropy. According to [35], the definition of the geometric entropy of a single voxel is

$$\mathbb{H}_g(\mathbf{v}) = -P_o(\mathbf{v})\cdot\ln P_o(\mathbf{v}) - (1-P_o(\mathbf{v}))\cdot\ln(1-P_o(\mathbf{v})), \quad (3)$$

where $P_o(\mathbf{v})$ is the probability of voxel $\mathbf{v}$ being occupied. Similar to the geometric entropy, as reported in [17], the semantic entropy used in this work is

$$\mathbb{H}_s(\mathbf{v}) = -P_s(\mathbf{v})\cdot\ln P_s(\mathbf{v}) - (1-P_s(\mathbf{v}))\cdot\ln(1-P_s(\mathbf{v})), \quad (4)$$

where $P_s(\mathbf{v})$ is the probability of voxel $\mathbf{v}$ belongs to TOs or NOs. I_g and I_s are the geometric and semantic entropy sums at the FOV of a specific robot pose, respectively.

2) Semantics-Aware Frontiers Detecting, Updating, Filtering & Evaluating: The map frontiers $f \in F$ are the voxels between the known voxels $\{\mathbf{v}_{occ}; \mathbf{v}_{free}\}$ and unknown voxels $\mathbf{v}_{unk}$. The frontier detector runs when the vehicle explores frontiers and detects frontiers in the local map layer $m^w_{t_i:t_j}$. In other words, the frontier detector only detects a part of the global map to enable efficient frontier detection. The history frontiers are stored as a sorted list $F_{his} = \{F_{his_0}; F_{his_1}; \cdots; F_{his_n}\}$ while the newly detected frontiers are $F_{new} = \{F_{new_0}; F_{new_1}; \cdots; F_{new_n}\}, n \in N^*$. F_{new} is added to F_{his} and update it to $\{F_{his}; F_{new}\}$. A frontier filter is then applied to remove the fake frontiers F_{fake} which are not frontiers in the global map layer $m^w_{1:T}$ from $\{F_{his}; F_{new}\}$. The frontiers $F_{his} = \{F_{his}; F_{new}\} \backslash F_{fake}$ are sent for evaluation. In the evaluation process, as shown in the equation (5), we first calculate I_{sum} of the FOV of each face direction $\phi \in [-\pi, \pi]$ of the frontier $f_{his_i} \in F_{his}, i \in N^*$ and then choose the highest I_{sum} of the face direction ϕ as the information gain of f_{his_i}. Subsequently, the distance $dist$ from the current robot pose p_r to f_{his_i} is calculated, and the cost c_f is then calculated as (5). Finally, f_{his_i} with the highest c_f is the frontier to explore.

$$\begin{aligned}
I_g(f_{his_i}, \phi) &= \sum_{-\frac{FOV_\psi}{2}}^{\frac{FOV_\psi}{2}} \sum_{-\frac{FOV_\theta}{2}}^{\frac{FOV_\theta}{2}} \sum_{d_{min}}^{d_{max}} (\mathbb{H}_g(\mathbf{v}|(f_{his_i}, \phi))), \\
I_s(f_{his_i}, \phi) &= \sum_{-\frac{FOV_\psi}{2}}^{\frac{FOV_\psi}{2}} \sum_{-\frac{FOV_\theta}{2}}^{\frac{FOV_\theta}{2}} \sum_{d_{min}}^{d_{max}} (\mathbb{H}_s(\mathbf{v}|(f_{his_i}, \phi))), \\
c_f(f_{his_i}) &= (w_s \cdot I_s(f_{his_i}, \phi) + w_g \cdot I_g(f_{his_i}, \phi)) e^{-k_d \cdot dist},
\end{aligned} \quad (5)$$

where FOV_ψ and FOV_θ are the horizontal FOV and vertical FOV while d_{min} and d_{max} are the minimum sensing distance and maximum sensing distance, respectively. w_g and w_s are the geometric and semantic entropy weights, respectively. k_d is the weight of $dist$.

3) SARHP: Fig. 4 shows the pipeline of SARHP. Unlike the traditional receding horizon planning approach, we evaluate the candidate branches instead of the candidate routes that satisfy kinematic constraints. The process to select the next

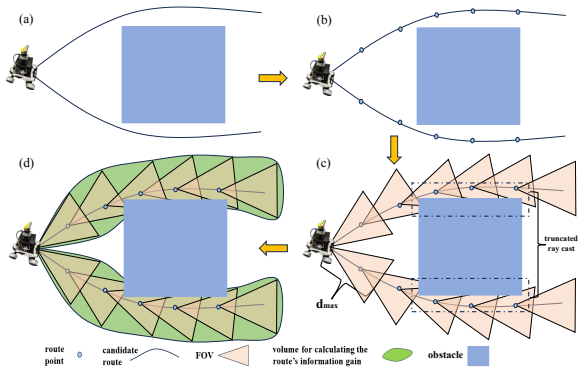


Fig. 4: SARHP. (a), (b), (c), and (d) are the obstacle-free candidate routes, route points generation, FOV calculating of each route point and volume generation, respectively. The candidate routes are generated in the robot frame.

best route is as follows. Firstly, a lattice sampling method [31] is used to generate collision-free routes to reach a short-range goal. We calculate each candidate route's cost c_{route}, which is the weighted sum of c_{fov} and the kinematic cost c_k. The c_{fov} is calculated using the weighted sum of the geometric reconstruction cost c_g, and the semantic reconstruction cost c_s. As shown in Fig. 4d, the route pipe comprises a series of interconnected FOVs; in this work, c_g and c_s are calculated by iteratively adding the geometric information gain I_g and the semantic information gain I_s of the truncated ray cast in the route points' FOV (the points in the green volume of Fig. 4d). The truncated ray cast means the ray cast in the FOV is cut off when it reaches d_{max} or encounters the occupied voxel. As the distance between adjacent FOVs in the candidate route is small, the endpoint of the truncated ray cast of the current route point's FOV only has a small value concerning the endpoint of the truncated ray cast of the previous route point's FOV. It is noted that the voxel is not considered when a part of it is in the candidate route's volume for calculating I_g and I_s in interconnected FOVs. As c_{fov} is the weighted sum of c_g, and c_s, we introduce the formulations to calculate c_g, and c_s, as shown in the following equation.

$$
\begin{aligned}
I_g(p_i,\phi_i,d_i^j) &= \sum_{-\frac{FOV_\psi}{2}}^{\frac{FOV_\psi}{2}} \sum_{-\frac{FOV_\theta}{2}}^{\frac{FOV_\theta}{2}} \sum_{d_i^j-\delta_j}^{d_i^j} (\mathbb{H}_g(\mathbf{v}|(p_i,\phi_i))),\\
I_s(p_i,\phi_i,d_i^j) &= \sum_{-\frac{FOV_\psi}{2}}^{\frac{FOV_\psi}{2}} \sum_{-\frac{FOV_\theta}{2}}^{\frac{FOV_\theta}{2}} \sum_{d_i^j-\delta_j}^{d_i^j} (\mathbb{H}_s(\mathbf{v}|(p_i,\phi_i))),\\
c_g &= I_g(p_0,\phi_0) + \sum_{i=0}^{n_r}\sum_{j=0}^{n_{ray}} (I_g(p_i,\phi_i,d_i^j)),\\
c_s &= I_s(p_0,\phi_0) + \sum_{i=0}^{n_r}\sum_{j=0}^{n_{ray}} (I_s(p_i,\phi_i,d_i^j)),\\
i &\in \{1,2,\cdots,n_r\}, j \in \{1,2,\cdots,n_{ray}\},
\end{aligned}
\tag{6}
$$

where d_i^j is the distance from the endpoint of the $j\ th$ truncated ray cast of the $i\ th$ route point's FOV to the $i\ th$ route point in the candidate route. p_i and ϕ_i are the position and face direction of the $i\ th$ route point in the candidate route. δ_j is the distance between the $i\ th$ route point and its truncated point of $j\ th$ ray cast by the FOV of $(i-1)\ th$ route point. n_r is the number of route points in each candidate route while n_{ray} is the number of the ray cast in the FOV. The equation to calculate c_{fov} is shown as

$$
c_{fov} = w_{rg}\cdot c_g + w_{rs}\cdot c_s, \tag{7}
$$

where w_{rg} and w_{rs} are the weights for c_g and c_s, respectively. If the value of w_{rg} is far higher than w_{rs}, SARHP degenerates into RHP (c_k + c_g). w_{rs} is tuned to decide how much SARHP pay attention to map the target objects before exploring unknown areas.

According to [36], [37], c_k is defined as

$$
c_k = w_k\cdot(1-(w_{dir}\cdot d_{dif})^{\frac{1}{4}}), \tag{8}
$$

where d_{dif} is the difference between the direction of the vehicle to the target frontier pose and the direction of the vehicle to the end pose of the candidate route. w_{dir} and w_k is the weights of d_{dif} and c_k, respectively. In our work, both w_{dir} and w_k are set as 0.02. The change of the vehicle's direction is larger if the value of w_{dir} is higher, and vice versa. If w_k is far higher than c_{rg} and c_{rs}, SARHP degenerates into RHP (c_k).

Finally, c_{route} is calculated using the weighted sum of c_{fov} and the kinematic cost c_k and the candidate route with the highest c_{route} is chosen as the next best view route for the vehicle to move.

TABLE I: Parameters of SARHP in simulation environments.

Parameter	Value	Parameter	Value
v_{max}	$0.3m/s$	w_s	0.9
ω_{max}	$0.2rad/s$	w_g	0.1
FOV	$[70.5°, 94.5°]$	*Mounting Pitch*	$30°$
$[d_{min}, d_{max}]$	$[0.1m, 2m]$	w_{dir}	0.02
Map Resolution	$0.1m$	w_k	0.02
Vehicle Length	$0.75m$	*Vehicle Width*	$0.6m$
k_d	0.02	*Weighted Gain* (w_{rg}, w_{rs})	$(0.5, 0.5)$

V. Experimental Results

In this section, we introduce the details of the simulation and real-world experiments performed to evaluate the proposed mapping and planning algorithms.

A. Simulation Experimental Setup and Results

The simulation experiments use the Robot Operating System (ROS) running on Ubuntu 20.04. The simulation runs in Gazebo, where the robot physical model used in the simulation is a wheeled vehicle. The simulation environments are constructed based on [38]. All the experiments run on a PC with a CPU of Intel Core i9-13900K and a GPU of NVIDIA GeForce RTX 4070. The primary perception sensor is an RGB-D camera mounted at the wheeled vehicle's front. The low-level control of the robot is developed based on the autonomous exploration development environment[4].

[4]https://www.cmu-exploration.com/

The detailed configurations of the proposed algorithm for simulation experiments are shown in Tab. I. The robot's pose is provided using the ground truth in the simulations. The environments used in the simulation experiments are two different office environments with rich objects. The height of the environments is fixed as $0.75m$. The width and length of the maze environment are $24m$ and $24m$, respectively. In the simulations, the robot starts at $(0, 0)m$, moves $2m$ in the front, and then starts exploring the environment.

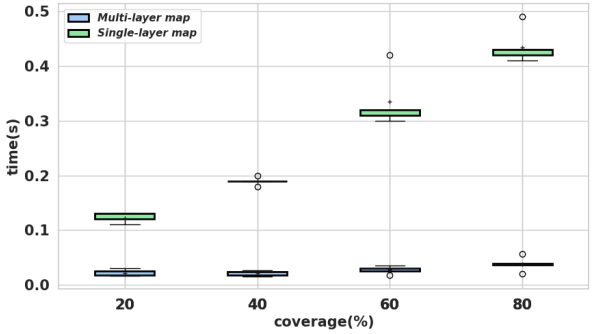


Fig. 5: The computing time of the frontier detection using the multi-layer map and single-layer map.

Fig. 5 shows the computing time for frontier detection changes with different map coverage. We also calculate the average time of frontier detection using the multi-layer map and single-layer map during the exploration process, which is $0.013s \pm 0.007s$ and $0.470s \pm 0.093s$, respectively. As shown in Fig. 6a and 6b, we chose the chair, sofa, and table as three TOs while the others as NOs. The semantic colours for the chair, sofa, table, and NOs are green, pink, blue, and black. It is noted that we segment the ground as silver to make it different from TOs and NOs. The semantic volumetric representations of TOs are shown in green, blue, and pink occupied grids while NOs are shown in black occupied grids of Fig. 6a and Fig. 6b (i), (ii), (iii), (iv). The right part of Fig. 6a and Fig. 6b show the comparison of the map coverage efficiency and semantic coverage efficiency of SARHP and the SOTA algorithms. The SOTA algorithms we used for comparison are the Next Best View Planner (NBVP) [5], RHP using kinematic cost, and RHP using geometry and kinematic cost. All the data is gathered after 5 runs of SARHP and SOTA algorithms in two different simulations. The quantitative assessment of the map coverage and semantic coverage of SARHP and SOTA algorithms, when SARHP reaches 90 $\%$ coverage of the two simulations, are shown in Tab .II.

B. Real-world Experimental Setup and Results

The real-world experiments are achieved using a customized wheeled vehicle as shown in Fig. 7; the robot has onboard sensors, including an OS0-128 LiDAR[5] used to detect the surrounding obstacles and perform localization and a D455 camera[6] used to detect the target objects. The localization of the robot is achieved using [39]. The learning and main

[5]https://ouster.com/products/hardware/os0-lidar-sensor
[6]https://www.intelrealsense.com/depth-camera-d455/

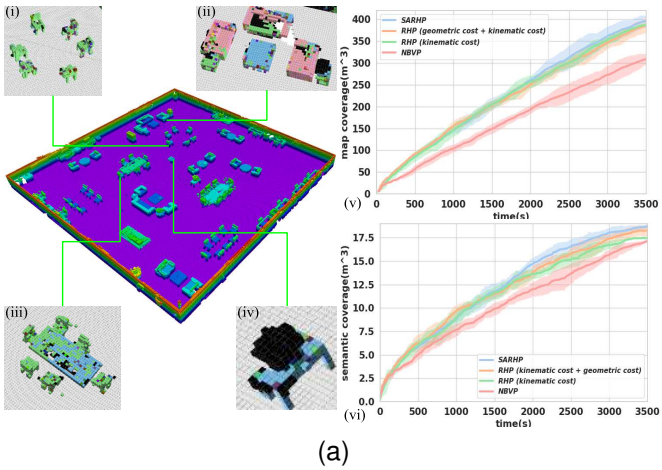


Fig. 6: The simulation results of SARHP and SOTA algorithms in two different office environments. SARHP is the RHP using the hybrid cost (semantic cost + geometric cost + kinematic cost). The colour line and colour shades represent the average and the standard deviation of the coverage of the 5 runs of each algorithm, respectively.

TABLE II: The map coverage and semantic coverage of SARHP and SOTA when SARHP reaches 90$\%$ coverage. env 1 and env 2 are the office environment in Fig. 6a and 6b.

Environment	Coverage type	Algorithm	Coverage (m^3)
env 1	Map Coverage	*SARHP*	***396.08 ± 12.9***
		RHP (geometric cost + kinematic cost)	*386.44 ± 15.89*
		RHP (kinematic cost)	*386.76 ± 5.81*
		NBVP [5]	*308.94 ± 11.82*
	Semantic Coverage	*SARHP*	***18.69 ± 0.24***
		RHP (geometric cost + kinematic cost)	*18.24 ± 0.62*
		RHP (kinematic cost)	*17.47 ± 0.18*
		NBVP [5]	*17.09 ± 0.17*
env 2	Map Coverage	*SARHP*	*389.19 ± 8.72*
		RHP (geometric cost + kinematic cost)	***389.84 ± 27.71***
		RHP (kinematic cost)	*380.24 ± 17.65*
		NBVP [5]	*250.93 ± 28.71*
	Semantic Coverage	*SARHP*	***18.29 ± 0.32***
		RHP (geometric cost + kinematic cost)	*17.64 ± 0.82*
		RHP (kinematic cost)	*16.72 ± 0.40*
		NBVP [5]	*14.11 ± 2.61*

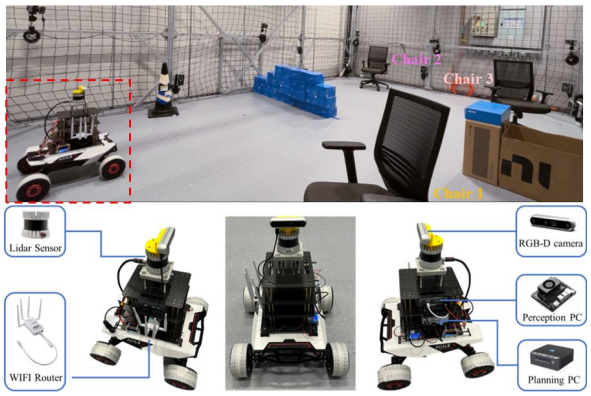
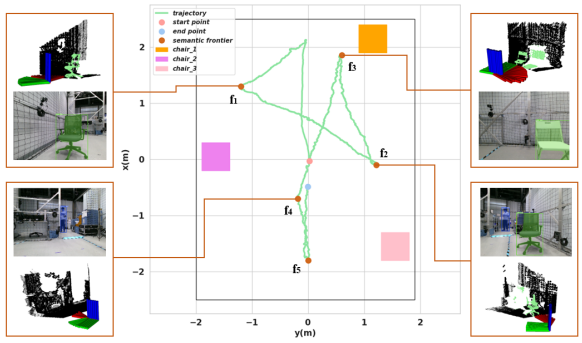
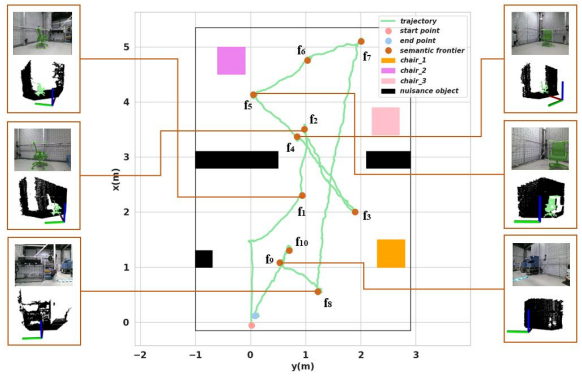


Fig. 7: (Up) The multi-room environment. The wheeled vehicle used is enclosed in the red dashed box. (Down) The details of the wheeled vehicle used in real-world experiments.



(a)



(b)

Fig. 8: The results of the real-world experiments. (a) is the planning and mapping results of the single-room experiment, and (b) is the planning and mapping results of the multi-room experiment.

PCs provide onboard inference for the neural network for semantic segmentation and onboard computing for localization, mapping, and planning algorithms. The learning computer is a Jetson Orin NX module with 100 TOPS of AI performance, and the main computer is an Intel NUC mini PC with Intel Core i9-12900H. The WIFI router is used to build a network for communication among the offboard computer, onboard computers, and sensors. The real-world experiments are conducted in an area with $3.8m$ in width and $5.5m$ in length, as shown in Fig. 1, using onboard computing resources.

Fig. 8a shows three TOs in the single-room environment in the orange, purple, and pink boxes. The robot started at the start point, first moved $2m$ in the front, then started exploring the whole environment, and finally, stopped at the endpoint. f_1, f_2, f_3, f_4, and f_5 are five frontier points of the whole exploration process. The segmented image and the semantic point cloud observed at f_1, f_2, f_3, and f_4 are also given. The results show that the vehicle first explores the area with target objects (f_1, f_2, f_3) and then the area with nuisance objects (f_4). In the multi-room environment, we combined SARHP with the information RRT* algorithm [40] to perform an object-searching task as shown in Fig. 8b. The vehicle cannot observe the target objects (chairs in the experiment) at the starting point due to the limited sensing range ($2m$). When the algorithm starts running, the vehicle moves $1.5m$ in the front and then starts searching. The experiment results show the vehicle moves from f_1 to f_{10} and finally finds all the targets and builds their semantic representations. It shows that the proposed planning and mapping algorithm has the potential to be used in complex tasks such as object searching in an indoor environment. Videos of real-world experiments are given at https://youtu.be/jJrqUxhFKHw.

C. Discussions

Experimental results of the map and semantic coverage are shown in Fig. 6a (v) and (vi) and Fig. 6b (v) and (vi) and Tab .II. As shown in the figures and table, the efficiency of map coverage and semantic coverage using the SARHP is significantly higher than the NBVP, which shows the SARHP's efficiency in map coverage. In the ablating experiments, to test the influence of geometric and semantic cost on map and semantic coverage, we set weight values of w_k, w_{rg} and w_{rs} for RHP (kinematic cost), RHP (kinematic cost + geometric cost) and SARHP as (0.02, 0, 0), (0.02, 1.0, 0) and (0.02, 0.5, 0.5), respectively. For RHP (kinematic cost), the efficiency of map coverage and semantic coverage are slightly worse than SARHP and RHP (kinematic cost + geometric cost), which shows that the geometric and semantic cost can improve the planner's coverage performance. The map coverage performance of SARHP and RHP (kinematic cost + geometric cost) is close. However, SARHP shows its superiority in semantic coverage performance, which means w_{rs} can help to improve semantic coverage performance. Fig. 5 shows that even when the size of the map continues to expand, the multi-layer map method does not affect the frontier detection time and can maintain it at a relatively small value. As for the single-layer map method, as the size of the map continues to expand, the time for frontier detection increases significantly. This result shows that the multi-layer map method can effectively improve planning efficiency compared to the traditional single-layer map method. The multi-layer map is more efficient in frontier detection because the frontiers are detected in a local layer (only a part of the global map) whose size is much smaller and evaluated in the global layer.

VI. Conclusions and Future Work

This letter proposes SARHP and a multi-layer object-centric volumetric mapping strategy. The object-centric volumetric

map is divided into three layers, including the FOV mapping layer, local mapping layer, and global mapping layer. Instead of processing the planning algorithm in the global mapping layer, which is time-consuming, the semantic frontiers are generated and evaluated in the local mapping layer and filtered in the global layer. The candidate routes of the receding horizon planner are assessed in the FOV mapping layer. Moreover, evaluating the candidate routes considers kinematic, geometric, and semantic costs, which can guide the robot to explore TOs and unknown areas while avoiding significant changes in the yaw angle. The simulations show that SARHP performs well in exploring the target objects without affecting overall coverage performance. The generation and evaluation of frontiers in the multi-layer map are also efficient. The proposed planning and mapping strategies are tested in a wheeled vehicle in real-world experiments. In the future, new map representation methods, such as the neural radiance fields, and more semantic details, such as the language, will be added to improve the robot's intelligence in planning and mapping.

REFERENCES

[1] Julio A. Placed et al. A survey on active simultaneous localization and mapping: State of the art and new frontiers. *IEEE Transactions on Robotics*, 39(3):1686–1705, 2023.

[2] Liang Lu et al. A comprehensive survey on non-cooperative collision avoidance for micro aerial vehicles: Sensing and obstacle detection. *Journal of Field Robotics*, 40(6):1697–1720, 2023.

[3] Antoni Rosinol et al. Kimera: an open-source library for real-time metric-semantic localization and mapping. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2020.

[4] Antoni Rosinol, Arjun Gupta, Marcus Abate, J. Shi, and Luca Carlone. 3d dynamic scene graphs: Actionable spatial perception with places, objects, and humans. *ArXiv*, abs/2002.06289, 2020.

[5] Andreas Bircher et al. Receding horizon "next-best-view" planner for 3d exploration. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1462–1468, 2016.

[6] Liang Lu et al. An optimal frontier enhanced "next best view" planner for autonomous exploration. In *2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids)*, pages 397–404, 2022.

[7] M. Selin, M. Tiger, D. Duberg, F. Heintz, and P. Jensfelt. Efficient autonomous exploration planning of large-scale 3-d environments. *IEEE Robotics and Automation Letters*, 4(2):1699–1706, April 2019.

[8] L. Schmid, M. Pantic, R. Khanna, L. Ott, R. Siegwart, and J. Nieto. An efficient sampling-based method for online informative path planning in unknown environments. *IEEE Robotics and Automation Letters*, 5(2):1500–1507, April 2020.

[9] Tung Dang et al. Graph-based path planning for autonomous robotic exploration in subterranean environments. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3105–3112, 2019.

[10] Daniel Duberg and Patric Jensfelt. Ufoexplorer: Fast and scalable sampling-based exploration with a graph-based planning structure. *IEEE Robotics and Automation Letters*, 7(2):2487–2494, 2022.

[11] Boyu Zhou, Yichen Zhang, Xinyi Chen, and Shaojie Shen. Fuel: Fast uav exploration using incremental frontier structure and hierarchical planning. *IEEE Robotics and Automation Letters*, 6(2):779–786, 2021.

[12] Christos Papachristos, Shehryar Khattak, and Kostas Alexis. Uncertainty-aware receding horizon exploration and mapping using aerial robots. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4568–4575, 2017.

[13] Tung Dang, Christos Papachristos, and Kostas Alexis. Visual saliency-aware receding horizon autonomous exploration with application to aerial robotics. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2526–2533, 2018.

[14] Zichao Zhang and Davide Scaramuzza. Perception-aware receding horizon navigation for mavs. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2534–2541, 2018.

[15] Tung Dang, Christos Papachristos, and Kostas Alexis. Autonomous exploration and simultaneous object search using aerial robots. In *2018 IEEE Aerospace Conference*, pages 1–7, 2018.

[16] Reem Ashour, Tarek Taha, Jorge Manuel Miranda Dias, Lakmal Seneviratne, and Nawaf Almoosa. Exploration for object mapping guided by environmental semantics using uavs. *Remote Sensing*, 12(5), 2020.

[17] Lintao Zheng et al. Active scene understanding via online semantic reconstruction. *Computer Graphics Forum*, 38(7):103–114, 2019.

[18] Rui Pimentel de Figueiredo, Jonas le Fevre Sejersen, Jakob Grimm Hansen, Martim Brandão, and Erdal Kayacan. Real-time volumetric-semantic exploration and mapping: An uncertainty-aware approach. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9064–9070, 2021.

[19] René Zurbrügg, Hermann Blum, Cesar Cadena, Roland Siegwart, and Lukas Schmid. Embodied active domain adaptation for semantic segmentation via informative path planning. *IEEE Robotics and Automation Letters*, 7(4):8691–8698, 2022.

[20] Sotiris Papatheodorou et al. Finding things in the unknown: Semantic object-centric exploration with an mav. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3339–3345, 2023.

[21] Mihir Dharmadhikari and Kostas Alexis. Semantics-aware exploration and inspection path planning. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3360–3367, 2023.

[22] Arash Asgharivaskasi and Nikolay Atanasov. Semantic octree mapping and shannon mutual information computation for robot exploration. *IEEE Transactions on Robotics*, 39(3):1910–1928, 2023.

[23] Julius Rückin, Federico Magistri, Cyrill Stachniss, and Marija Popović. An informative path planning framework for active learning in uav-based semantic mapping. *IEEE Transactions on Robotics*, pages 1–18, 2023.

[24] Xianmei Lei et al. Early recall, late precision: Multi-robot semantic object mapping under operational constraints in perceptually-degraded environments. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2017–2024, 2022.

[25] Sebastian Scherer et al. Resilient and modular subterranean exploration with a team of roving and flying robots. *Field Robotics*, 2022.

[26] Dhruv Shah et al. Navigation with large language models: Semantic guesswork as a heuristic for planning. In *Conference on Robot Learning*, pages 2683–2699. PMLR, 2023.

[27] Gengze Zhou, Yicong Hong, and Qi Wu. Navgpt: Explicit reasoning in vision-and-language navigation with large language models. *arXiv preprint arXiv:2305.16986*, 2023.

[28] Qiao Gu et al. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. *arXiv preprint arXiv:2309.16650*, 2023.

[29] Sourav Garg et al. Robohop: Segment-based topological map representation for open-world visual navigation. In *2nd Workshop on Language and Robot Learning: Language as Grounding*, 2023.

[30] Dhruv Shah et al. Vint: A foundation model for visual navigation. *arXiv preprint arXiv:2306.14846*, 2023.

[31] Ji Zhang, Chen Hu, Rushat Gupta Chadha, and Sanjiv Singh. Falco: Fast likelihood-based collision avoidance with extension to human-guided navigation. *Journal of Field Robotics*, 37(8):1300–1313, 2020.

[32] Chaoning Zhang, Dongshen Han, Yu Qiao, Jung Uk Kim, Sung-Ho Bae, Seungkyu Lee, and Choong Seon Hong. Faster segment anything: Towards lightweight sam for mobile applications. *arXiv preprint arXiv:2306.14289*, 2023.

[33] Armin Hornung, Kai M. Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 2013.

[34] Zhang Xuan and Filliat David. Real-time voxel based 3d semantic mapping with a hand held rgb-d camera. https://github.com/floatlazer/semantic_slam, 2018.

[35] Anna Dai et al. Fast frontier-based information-driven autonomous exploration with an mav. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9570–9576, 2020.

[36] C. Cao, H. Zhu, Z. Ren, H. Choset, and J. Zhang. Representation granularity enables time-efficient autonomous exploration in large, complex worlds. *Science Robotics*, 8(80):eadf0970, 2023.

[37] Chao Cao et al. Autonomous exploration development environment and the planning algorithms. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 8921–8928, 2022.

[38] Amir Rasouli and John K Tsotsos. The effect of color space selection on detectability and discriminability of colored objects. *arXiv preprint arXiv:1702.05421*, 2017.

[39] Wei Xu, Yixi Cai, Dongjiao He, Jiarong Lin, and Fu Zhang. Fast-lio2: Fast direct lidar-inertial odometry. *IEEE Transactions on Robotics*, 38(4):2053–2073, 2022.

[40] Ioan A. Şucan, Mark Moll, and Lydia E. Kavraki. The Open Motion Planning Library. *IEEE Robotics & Automation Magazine*, 19(4):72–82, December 2012. https://ompl.kavrakilab.org.