Semantic Target Search and Exploration using MAVs in Cluttered Environments

Nikhil Sethi

TU Delft

STEM: Semantic Target Search and Exploration using MAVs in Cluttered Environments

Master of Science Thesis

Nikhil Sethi

August 27, 2024

Faculty of Mechanical Engineering, Delft University of Technology

For the degree of Master of Science in Robotics at the Delft University of Technology,

To be defended publicly on Tuesday August 27, 2024 at 14:00 CEST.

Student number:	5711428	
Project duration:	January 2024 – August 2024	
Thesis committee:	Dr. J. Alonso-Mora,	TU Delft ME, Primary supervisor
	Ir. M. Lodel,	TU Delft ME, Daily supervisor
	Dr. M. Popovic,	TU Delft AE, Assistant Professor
	Dr. C. Hernandez,	TU Delft ME, Assistant Professor

.

TU Delft Delft University of Technology

Cognitive Robotics

Acknowledgements

To shoulders, concrete, nature, and song.

Delft, University of Technology
August 12, 2024

Nikhil Sethi

Abstract

Autonomous target search is crucial for deploying Micro Aerial Vehicles (MAVs) in emergency response and rescue missions. Existing approaches either focus on 2D semantic navigation in structured environments – which is less effective in complex 3D settings, or on robotic exploration in cluttered spaces – which often lacks the semantic reasoning needed for efficient target search. This thesis overcomes these limitations by proposing a novel framework that utilizes semantic reasoning to minimize target search and exploration time in unstructured environments using a MAV. Specifically, the open vocabulary inference capabilities of Large Language Models are employed to embed semantic relationships in segmentation images. An active perception pipeline is then developed to guide exploration toward semantically relevant regions of 3D space by biasing frontiers and selecting informative viewpoints. Finally, a combinatorial optimization problem is solved using these viewpoints to create a plan that balances information gain with time costs, facilitating rapid location of the target. Evaluations in complex simulation environments show that the proposed method consistently outperforms baselines by quickly finding the target while maintaining reasonable exploration times. Real-world experiments with a MAV further demonstrate the method's ability to handle practical constraints like limited battery life, small sensor range, and semantic uncertainty.

Keywords: Search and Rescue, Drones, LLM, Semantic priority, TSP, Mapping, Visual Attention, Informative path planning

Contents

STEM: Semantic Target Search and Exploration using MAVs in Cluttered Environments

Nikhil Sethi *

1 Introduction

Searching for targets is a crucial task in emergency response environments such as search and rescue, and human involvement in these environments can be a cause of concern. For instance, mine rescues under harsh conditions pose health risks, including a high potential for heat-related illnesses [1]. First responders in earthquakes and nuclear disasters face lifelong mental trauma from disturbing conditions like gore and unpleasant smells [2]. Organized crime and law enforcement can incur significant economic costs, thus affecting victims, offenders, and society at large [3]. Such challenges raise ethical questions about labor practices and underscore the need to reevaluate our approach to these demanding tasks.

The integration of Micro Aerial Vehicles (MAVs) offers a promising approach for efficient, safe, and ethical solutions to search for targets in extreme environments. However, this integration is challenging due to the unique constraints of MAVs, such as limited flight time, computational budgets, and a small sensor range. In cluttered environments, these constraints are magnified, making it essential to develop a target search framework that can solve perception and planning as a unified task, commonly known as *Active Perception* [4].

Problem Domain: There are three common approaches to solving this problem under the active perception framework. First, coverage-driven exploration methods can be used to cover the full map and search every possible location [5], [6]. These approaches take a long time on average to find the target and result in large variances because they rely on chance. Second, explicit target search methods can be utilized that leverage semantic features within a *known* environment. These approaches reduce search times but rely on domain-specific priors learned from preexisting datasets of structured indoor environments [7], [8]. Third, a hybrid approach can be used where the drone needs to simultaneously explore an unknown environment and search promising regions of space to find the target [9]. This approach also uses semantic reasoning, but partial observability poses a unique challenge in emergency response situations, where exploration and target search must be balanced optimally to minimize search time. Some studies aim to find the correct room for a target object using object-room relationships [10], while others focus on refined local search within a cluttered room [11]. Our work focuses on the latter and uses the hybrid approach to search for targets.

Contribution statement In this work, we develop STEM: a **S**emantic **T**arget Search and **E**xploration framework that uses **M**AVs to find targets without assuming any structure or knowledge of the environment. The primary contribution of this thesis is the development of an active perception pipeline that can embed semantic priorities in 3D, generating a rich set of viewpoints with balanced coverage and semantic information gains. This pipeline is supplemented by extending a combinatorial target search planner [12], to create efficient global plans through 3D viewpoints. Additionally, we introduce a novel semantic priority masking scheme that uses Large Language Models to compress semantic segmentation images into priority masks. Finally, we conduct extensive experiments in both simulation and real-world environments using a Micro Aerial Vehicle (MAV). The results can be reproduced using publicly available ROS-compatible software at https://github.com/nikhil-sethi/thesis.

Summary The rest of the thesis is organized as follows. Section 2 covers related work and highlights how this study addresses the gaps in the literature. Section 3.1 and 3.2 establish technical preliminaries and a formal problem formulation, respectively. Further, Section 4 presents a detailed methodology for the work, with the experimental setup following in Section 5. Section 6 demonstrates qualitative and quantitative simulation results along with ablation studies that analyze the deeper aspects of the algorithm. Section 7 shows real-world experiments with a MAV in various cluttered configurations. Finally, Section 8 summarizes the thesis and presents recommendations for future work.

2 Related Work

The literature survey is divided into coverage-based exploration and target search methods in Sections 2.1 and 2.2 respectively. Target search is further divided into structured (Section 2.2.1) and unstructured environments (Section 2.2.2) because they require a different set of assumptions.

*MSc. Robotics, Delft University of Technology, Delft, The Netherlands

2.1 Coverage Exploration

Exploration is a fundamental component for deploying robots in domains such as search and rescue [13] and law enforcement [14]. Unlike traditional methods where maps are pre-built offline, exploration focuses on dynamically uncovering new regions of the environment and planning efficiently under partial observation. Coverage-based exploration methods focus on reducing the unknown space in a bounded volume, thus fully 'covering' the environment.

Earlier works use the Rapidly Exploring Random Tree (RRT) planning algorithm for coverage exploration [15]–[17]. *Viewpoints* (virtual camera poses in 3D) are sampled in free space to incrementally build a tree (graph), and each viewpoint is evaluated using a utility function such as unmapped volume or information gain. The first edge of the most informative branch of the tree is then selected as the goal and the process is repeated in a receding horizon manner. Such approaches struggle in open spaces because the number of samples can be large, and calculating information gain for each random sample in 3D space can be computationally expensive. This drawback thus limits the planning horizon, making these approaches greedy in their decision-making [18].

Recent state-of-the-art works such as FUEL [5] and TARE [18] systematically sample a minimum set of viewpoints and solve a combinatorial optimization problem to create a global plan. In TARE, the researchers reduce the computational complexity of large-scale 3D exploration by creating a refined local plan *close* to the robot and maintaining a coarse global plan using a Travelling Salesman Problem (TSP). The work is relevant because it uses non-myopic strategies for planning, but mainly addresses large-scale exploration with long-range LIDAR sensing ($\approx$ 100m) and a larger aerial robot. In contrast, we address the challenge of target search with range-constrained sensing (3m) using a single front-facing depth camera and a 1Kg MAV.

FUEL employs a novel formulation for the TSP that incorporates kinematic costs in the objective function. When combined with a local trajectory planner, this approach results in more efficient exploration because the global plan penalizes large changes in motion. This work is most relevant to ours because it relies on frontiers for exploration (as opposed to surfaces for TARE) and also uses a front-facing depth camera for sensing. Since the framework focuses on volumetric coverage, we leverage it for the exploration component in our work and integrate our semantic target search capability with it. Additionally, we use this method as a baseline for comparisons.

The main drawback of FUEL and TARE is that a standard TSP only solves for metric costs such as time or distance between viewpoints. However, efficient planning under partial observability requires balancing a viewpoint's utility and the time it takes to arrive at it. This trade-off is explored in works such as [6] and [19]. In [19], the cost between two viewpoints is equal to the distance inversely weighted by a directed information gain. This heuristic allows edges to be connected such that viewpoints with higher information gain come earlier in the tour. However, such heuristics need to be tuned to the situation and are less robust to uncertainty in the information gain. The work in [6], called FAEL, solves this problem by using a variant of the TSP called the Minimum Latency Problem (MLP) that treats information gain and movement distance in a joint objective. This concept is relevant to our work because target search also involves balancing multiple objectives like time, and semantic priority. However, unlike FAEL which only focuses on 2D exploration with ground robots, we perform semantic target search in 3D using drones. We achieve this by creating a novel 3D information gain calculation, and an MLP formulation that balances metric and semantic costs.

2.2 Target Search

Humanitarian scenarios such as SAR often require searching for specific, task-relevant objects, and coverage-based exploration methods can be inefficient as they search every possible location. Finding targets in such situations requires a robot to reason in the environment and narrow the search only toward interesting regions. For example, in an earthquake, the search for a victim might start by identifying the living room, then progress to locating objects like tables, and ultimately find humans who might be trapped underneath. Locating the living room and then carrying out a refined search within the room require different assumptions, and thus, we distinguish between target search in structured and unstructured environments in sections 2.2.1 and 2.2.2 respectively.

2.2.1 Structured Environments

Structured environments refer to closed indoor spaces with well defined boundaries like perpendicular walls and consistent object placement (for example, searching for a computer screen in an office space). The works mentioned in this section use semantic reasoning to guide the search for targets.

Earlier works such as [20], investigated object-level semantic representations to predict the focus of human visual attention. The researchers used an open vocabulary Concept-Net to determine object-object relationships via cosine similarities. They discovered that objects that attracted more attention had higher conceptual similarity to nearby objects and the overall scene itself. This concept is the primary motivation for developing a semantic priority mask in our work.

Recent works, such as VLFM [9] and SemUtil [21] use foundation models to determine semantic object relationships for robot navigation tasks. Both methods use semantic knowledge to bias exploration frontiers but differ in the source of knowledge. VLFM uses a Vison-Language Model, while SemUtil uses an object detector to first get class labels from images and then infer the similarity scores with a Large Language Model

(LLM). Both approaches achieve competent results, but we use the latter method because of its simplicity and greater control over the computer vision pipeline. An even more recent work called SEEK [10], develops a separate model called the Relational Semantic Network which is trained using offline queries to an LLM and directly estimates the probability of finding the target object.

Notably, these methods have two main drawbacks. First, VLFM and SemUtil are limited to simple structured environments. The results are demonstrated on datasets of indoor spaces [7] and risk overfitting on domain-specific priors od indoor environments. SEEK demonstrates results in challenging outdoor spaces but uses prior map knowledge. Secondly, all the methods use planning strategies that do not scale well to cluttered environments. VLFM and SemUtil use greedy frontier selection which can be problematic when there is uncertainty in the output of the foundation model. SEEK uses Markov Decision Processes (MDP) for lower dimension problems like optimal room selection. Moreover, MDPs suffer in multi-resolution information-gathering tasks like exploration with a 3D action space [22]. We overcome these limitations by proposing a 3D target search framework that makes no assumption on the structure of the environment and uses combinatorial optimization to create non-greedy plans that are robust even under semantic uncertainty.

2.2.2 Unstructured Environments

Unstructured environments are complex 3D spaces with unpredictable geometric and semantic features. These environments are often encountered during disaster response or inspections in subterranean settings. According to Guivant et. al. [23], one of the key challenges of deploying robots in such environments is to develop appropriate map representations.

In works such as [24] and [25], the targets in the environment are represented on an object level. In [24], the researchers address robotic exploration in challenging subterranean environments, such as tunnels. An online object detection pipeline is used to project 2D bounding boxes into the 3D world, updating location and confidence estimates through a Bayesian log-odds update. However, multi-class Bayesian updates in 3D can be memory intensive, and works such as [24] or [26] mitigate this problem by storing only a finite number of most-likely classes (5 and 3 respectively). In contrast, our approach compresses semantic labels to discrete priorities directly at the 2D stage, and performs simple weighted updates, thereby avoiding the high memory demands of 3D semantic maps.

In [25], viewpoints are sampled around frontiers and as well as objects to reconstruct objects as point clouds. Our work does not focus on reconstruction but we also use object-centric viewpoint sampling to perform parallel inspection and exploration. Additionally, the work in [25] does not use semantics to reason within the environment, and the authors have not yet provided the full code.

In [27], the concept of visual attention is used to guide search towards objects of interest. The target object's visual features are used to create a probability map in the 2D plane of the ground robot, and belief space planning is used to navigate. Although the work is a good resource for creating disaster response environments, projecting 3D information to 2D probability maps can result in loss of information, particularly in unstructured environments [28]. In contrast, the work in [29] creates a 3D volumetric saliency map for drone-based exploration. The authors use a 2D saliency mask to mark important regions in 3D, and then prioritize high-utility viewpoints that can guide exploration to salient regions. However, they only use saliency as the source of importance without incorporating any semantic reasoning. In contrast, we use semantic relationships inferred from an LLM and evaluate viewpoint utility directly using biased exploration frontiers. Our work is conceptually related to [29], which we use as a baseline for comparison.

3 Preliminaries

3.1 Background

3.1.1 Frontier Exploration and Mapping

The goal of 3D robotic exploration is to traverse a defined volume V and incrementally reduce the unknown space $V \setminus \mathcal{M}$ by creating an occupancy map $\mathcal{M}$. This map is a 3D volumetric grid of voxels, with each voxel $m_k \in \mathcal{M}$ storing the probability of occupancy P_k. These probabilities are updated using an inverse camera sensor model and Bayesian Inference [30].

Among various methods for exploration, frontier-based methods are most relevant to our work [5], [9], [31]. These methods first detect a set $\mathcal{F}$ of *frontiers* – which are boundaries between known and unknown space, and then sample a set $\mathcal{V}$ of *viewpoints* – which are poses in free space that can 'view' the frontiers. Exploration frameworks then focus on two main problems: (a) Computing the order in which these frontiers are visited – which dictates the efficiency of exploration, and (b) Creating a function that quantifies the utility of a viewpoint – which dictates the quality of exploration. This function can be based on information gain [32], volumetric coverage [5], or semantic importance [9]. Unknown space in V can then be reduced by finding the most efficient path through high-quality viewpoints.

To maintain efficiency, frontiers are typically clustered into groups of voxels with each group having a minimum size F_{min}, as demonstrated in works such as [5]. Additionally, the utility of a viewpoint is required to be at least ν_{min}.

3.1.2 Visual Attention

Visual attention refers to the process of selectively focusing on specific visual information within our perceptions [33]. In literature, visual attention is

typically classified into two types: Bottom-Up and Top-Down [27], [34], [35]. Bottom-up attention is an object's intrinsic drive to attract attention, often called saliency. This includes features like brightness, color, texture, etc. Top-down attention refers to an actor's deliberate decision to focus on specific objects guided by prior knowledge and experience. For instance, a human searches for keys in a room even though they are not salient. Humans use both contexts when navigating in unknown environments and it is essential to impart robots with such capabilities. In the context of this research, a region of high visual attention can be seen as 3D space that has a higher likelihood of containing targets and should prioritized for further investigation and refinement of the search.

3.1.3 Semantic Relationships

Semantics are labels or categories that humans use to classify objects. Humans use accumulated semantic knowledge to derive relationships between objects of interest when looking for targets. For instance, when searching for a `laptop`, we first look for a `table` as opposed to a `toilet`, because the former is more correlated with the target object. More formally, these relationships can be defined using a semantic relationship function $F : \mathcal{S} \times \mathcal{S} \rightarrow \mathbb{R}^+$ that maps a set $\mathcal{S}$ of natural language classes to scalar-valued similarity scores. State-of-the-art large language models like CLIP [36] or BERT [37] use open vocabulary concept databases and contextual understanding to infer such relationships. These models use a neural network to first transform the labels to *vector embeddings* which are real-valued representations of the text labels in a high-dimensional feature space. Then, the cosine similarity score S_c between two labels is obtained by calculating the dot product of their vector embeddings A and B:

$$S_c(A, B) = \frac{A.B}{\|A\|\|B\|} \tag{1}$$

3.1.4 Combinatorial Planning

Recent successes in robotic exploration use combinatorial optimization methods to formulate a non-myopic global path through the viewpoints in $\mathcal{V}$ [5], [6], [18], [38].

A traveling salesman problem (TSP) is often used to create this global path. The TSP is a combinatorial optimization problem that decides the order to visit a set of nodes $\mathcal{V}$ connected with edges $\mathcal{E}$ on a graph $\mathcal{G} : (\mathcal{V}, \mathcal{E})$. In its simplest form, nodes in $\mathcal{V}$ are positions in $SE(3)$, which means that the cost $c(e)$ to travel along edge $e \in \mathcal{E}$ is symmetric and equal to the travel distance. To make calculations efficient, the costs between nodes are represented using a cost matrix $\mathbf{C}$. The cost of a complete tour σ is defined as $C(\sigma)$.

In [5], the authors expand $c(e)$ to include the time cost for switching between two viewpoints. This makes the resulting plan more consistent and decreases oscillation because large changes in motion are penalized. We refer to this as the kinematic TSP. Recent works in literature use variants of the TSP, incorporating additional objectives like information gain [6][12]. We expand on the work in [5] and [12] to create efficient plans that incorporate both metric and semantic costs in the optimization objective $C(\sigma)$.

3.2 Problem Formulation

The goal of this work is to use sensor data from a MAV to simultaneously explore a previously unseen 3D environment and create a global plan that leads to a possible target in minimum time.

3.2.1 Environment Features

The environment is modeled using metric and semantic features. The metric features are modeled using a 3D global occupancy map $\mathcal{M}$ bounded by a volume $V \subset \mathbb{R}^3$. This map can be used for object localization and local obstacle avoidance. The semantic features are available as a set $\mathcal{S}$ of possible Objects Of Interest (OOIs) understood by natural language semantic labels. Further, it is expected that the objects in $\mathcal{S}$ have semantic relationships defined by a function F which can be exploited to guide the robot towards a target object $o^* \in \mathcal{S}$. A target is considered found when its relative semantic segmentation area in the robot's field of view crosses a threshold λ_{min}.

3.2.2 MAV Model

An autonomous aerial robot equipped with an RGB-D front camera is used to perform the search task. $\mathbf{x}_t$ is defined as the robot's pose in $SO(3)$ at time instant t and the action space of the robot is (x, y, z, ψ) assuming differentiable flat control [39]. The MAV has a maximum linear velocity v_{max}, maximum acceleration a_{max}, and maximum yaw rate ω_{max}. At each time instance t the robot receives a measurement tuple $z_t = (\mathbf{x}_t, \mathcal{I}_c, \mathcal{I}_d)$, where $\mathcal{I}_c$ and $\mathcal{I}_d$ are the RGB and depth images, respectively.

Problem statement: Given a target object o^*, a bounded volume V, and the robot's initial configuration $\mathbf{x}_0$, use z_t to find a collision free global plan σ through V such that o^* is discovered in minimum time.

4 Methodology

4.1 Overview

The goal of the method is to take RGB-D images along with the robot's pose and output a global plan that leads the robot toward the target.

To motivate our method, consider how humans search for important objects. We infer target-object relations from the environment based on context and create a mental map of interesting objects in 3D space. Then, we conduct a refined local search *near* OOIs and rely on active perception maneuvers to find the target. For instance:

- Opening a wardrobe (OOI) to find clothes (target) in the bedroom (context), OR
- Looking underneath a table (OOI) to find a human (target) in an earthquake (context) (see Fig. 1).

Thus, the method is motivated by a simple realization: To quickly find a target, it is essential to minimize unknown space *near* objects of interest that are conceptually and spatially related to the target object. This could be seen as *prioritizing* search to specific regions, unlike coverage-based exploration, which tries to minimize *all* unknown space. Additionally, if we can balance both tasks optimally, we can ensure that the method works even under semantic uncertainty and does not incur significant costs in exploration time.

The pipeline consists of three components. The *Semantics* module (Section 4.2) processes the RGB image to segment objects and ranks them using a Large Language Model. This ranking, called priorities, is then used to compress the segmentation image into a 2D priority mask. The *Active Perception* module (Section 4.3) uses the priority mask, the depth image, and the drone's state to give (a) A set of 3D viewpoints in free space, and (b) A set of information gains corresponding to each viewpoint. This module is also responsible for fusing new measurements to maintain a consistent global map. The *Planning* module (Section 4.4) solves a combinatorial optimization problem over the 3D viewpoints to create a global plan that balances metric and semantic gains.

4.2 Semantic Priority Masking

The goal of this module is to use the RGB image $\mathcal{I}_c$ to generate a priority mask image $\mathcal{I}_p$ that has pixel-wise discrete priority values for each object of interest.

A semantic segmentation image $\mathcal{I}_s$ and a set $\mathcal{S}_t \in \mathcal{S}$ of natural language classes is generated using $\mathcal{I}_c$ at time t. The set $\mathcal{S}$ is a diverse but limited superset containing possible objects that can be encountered in a wide variety of scenarios. The image $\mathcal{I}_s$ contains unique pixel wise labels $\{1, 2, .., \|\mathcal{S}_t\|\}$ for each class in $\mathcal{S}_t$. In this work, we assume the existence of a learning-based method like Mask-RCNN [40] or Fast-SAM [41] that can generate $\mathcal{I}_s$ and $\mathcal{S}_t$.

The segmentation image often contains noise and spurious detections that are not relevant to downstream tasks like planning [42]. Motivated by this, instead of using $\mathcal{I}_s$ directly, we compress the semantic segmentation image into a priority mask $\mathcal{I}_p$. This image contains pixel-wise discrete integers for each class, indicating their relative importance to the target class. The process to generate this mask is demonstrated in Fig. 1, and is also explained as follows:

1. Sequence preparation: First, situational context about the scenario is added to the target object in the set $\mathcal{S}$. This is done by appending a context label to the target class using the formulation: [`label`] [`preposition`] [`context`]. For instance, in an `earthquake` scenario, where the target is a `human`, the sequence becomes `human in earthquake`. This helps to form more appropriate relationships between the target object and other environment objects.
2. Vector embedding: Next, each class in $\mathcal{S}$ is tokenized using the model's appropriate tokenization algorithm and passed through the model (LLM) to give an output tensor of size $\|\mathcal{S}\| \times n_s \times n_e$. Here, n_s is the size of the sequence (3 for the target; 1 for other objects), and n_e is the size of the embedding vector for a particular model (bert-large-uncased [37]). Since each vector along the sequence dimension (n_s) contains contextual information gathered from the entire sequence, the tensor is averaged along this dimension, which gives a tensor $\boldsymbol{\tau}$ of size $\|\mathcal{S}\| \times n_e$. Intuitively, each row of $\boldsymbol{\tau}$ is the real-valued representation of the semantic class label in a high-dimensional feature space. The target embedding vector is τ^*.
3. Cosine similarity: Since the embeddings are vectors in the feature space, comparing their directions can provide insights into their similarity. Each vector in $\boldsymbol{\tau}$ is compared to τ^*, yielding similarity scores for each class. These scores are real values ranging from 0 to 1. This process is similar to the semantic relationship function F mentioned in Section 3.1.3.
4. Priority masking: The similarity scores are thresholded to retain objects of interest and subsequently scaled to integer values within the range $[1, p_{max}]$. Here p_{max} is a maximum priority value and the rationale behind this range is elaborated in subsequent sections on mapping and planning. The scaling process creates a mapping from the set of classes ($\mathcal{S}$) to integer-valued priorities, which we refer to as the priority function $r : \mathcal{S} \to \mathbb{N}^+$. The labels for each class in the segmentation image are then pixel-wise replaced with their corresponding priority to create the priority mask image $\mathcal{I}_p$.

   In practice, the priorities are stored in an offline vector at the beginning of the episode for a given context, target, and set $\mathcal{S}$. At runtime, the classes in set $\mathcal{S}_t$ are queried for their corresponding priorities from the offline priority vector.

The parameter p_{max} controls the sensitivity of the search. When there is a large dataset of objects with distinct semantic meanings, the cosine similarities have a high variance. Even a small p_{max} value can distinguish between relevant and irrelevant priorities, thus guiding the search correctly. However, if the dataset is limited to a set of correlated objects, then p_{max} can be increased to refine the search and detect stronger correlations to the target object. However, a higher p_{max} also tends to make the search more greedy, and it can be tuned based on the expectation of finding semantically related objects close to the target. A $p_{max} = 1$ value implies complete compression where all objects are equally important.

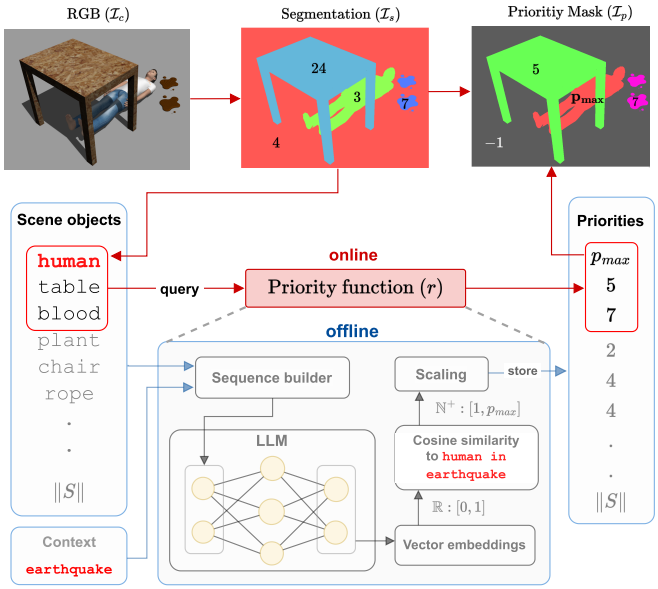


Figure 1: Semantic priority masking pipeline. Red arrows and blue arrows represent online and offline operations, respectively. At runtime, the priority of each class in $\mathcal{S}_t$ is queried from a pre-computed priority vector to create the priority mask $\mathcal{I}_p$.

4.3 Active Perception

The goal of the active perception module is to use the priority mask $\mathcal{I}_p$, depth image $\mathcal{I}_d$, and robot pose $\mathbf{x}_t$ to create (a) a set of viewpoints $\mathcal{V}$ in free space, and (b) a set of information gains I corresponding to each viewpoint in $\mathcal{V}$.

Section 4.3.1 provides a way to represent semantic priorities in 3D. Section 4.3.3 describes a method to diffuse semantic priorities to neighboring frontiers and Section 4.3.2 helps maintain a consistent global map for objects of interest. Section 4.3.4 describes the process of generating viewpoints in free space and Section 4.3.5 describes a novel procedure to calculate information gain for each sampled viewpoint. Figure 2 shows an overview of the complete active perception pipeline.

4.3.1 Priority Map

The priority values in $\mathcal{I}_p$ need to be represented on the 3D map where the drone will subsequently navigate and collect new observations. The priority mask $\mathcal{I}_p$, the depth image $\mathcal{I}_d$, and robot pose $\mathbf{x}_t$ are used to create a 4D depth-intensity point cloud observation Ω. Each point Ω_k in this point cloud carries the 3D position in the world frame (x_w, y_w, z_w) and the priority value p_w as the intensity channel. Let $\mathbf{d_i} = (u_i, v_i, z_i)$ be a point in the depth image $\mathcal{I}_d$ and p_w be the corresponding priority value from $\mathcal{I}_p$ at (u_i, v_i). The point cloud Ω is generated using projective transformation and a camera sensor model as follows:

$$\Omega_k = \begin{bmatrix} \mathbf{T_c^w K^{-1} d}_i \\ p_w \end{bmatrix} \tag{2}$$

Here $\mathbf{K}$ is the camera's intrinsic transformation matrix, and $\mathbf{T_c^w}$ is the transformation matrix that transforms a point in the camera frame to the world frame. The point cloud is also post-processed using voxel-grid filtering and statistical outlier removal. The priority value p_w at each 3D point k in Ω is then used to update a discrete volumetric grid $\mathcal{P}$ at the corresponding voxel $p_k \in \mathcal{P}$ using a simple weighted update (Eq. 3).

$$p_k \leftarrow (1-\alpha)p_k + \alpha p_w \quad \forall k : 1 \rightarrow \|\Omega\| \tag{3}$$

Here α is a learning rate that updates the map progressively and prevents noise from being integrated. Note that a Bayesian multi-class update, as utilized in [26], could also be used to store the priority values in the map, but this would increase memory complexity, scaling linearly with $O(p_{max})$. In our work, this approach does not offer a significant advantage because accurate map reconstruction is not the goal, and the priority map serves only as an intermediate representation for biasing frontiers (see Section 4.3.3).

Next, a local section $\mathcal{P}_l \subset \mathcal{P}$ centered around the drone is retrieved to keep the computational efficiency bounded. Points in this local section are clustered based on priority values using a region-growing algorithm to generate a distinct set of clusters in 3D space. Each cluster is referred to as an *Object* and is stored as a data structure containing detailed information such as the bounding box, centroid, and mean priority of the cluster. Thus, this process produces a local set of distinct 3D objects (O_l) that are clustered based on priority values.

4.3.2 Object Fusion

As the robot moves around and takes new observations, a new set of local objects is created at each iteration. This local set of objects O_l needs to be fused with already existing objects in the map, i.e., a global set O_g.

To achieve this, a custom bounding box fusion algorithm was developed based on geometric and semantic features. For each object $o_l \in O_l$, O_g is searched for a potential merging candidate o_g. Two objects can be merged into each other if (a) the ratio of the overlapping volume to each object's volume exceeds a threshold μ_l, for either object *and* (b) the objects have similar mean priorities. Note that some works, such as [25], use 3D Intersection Over Union (IOU) for merging, but we do not use that here because it only permits the merging of similar boxes. Even if o_l appears different from o_g, it might belong to the same object and thus needs to be merged (see Fig. 4). When a merging candidate o_g is found, it is expanded in place to incorporate o_l, and o_l is removed from the local set O_l. If a potential merge candidate is not found, the local object is simply added to the global set O_g.

This process is efficient because only a small set, O_l, is traversed once to identify potential merges, and merging happens in place. However, this approach may overlook candidates for merging within the global set itself. Therefore, the same fusion procedure is applied to each object in the global set O_g with a separate threshold

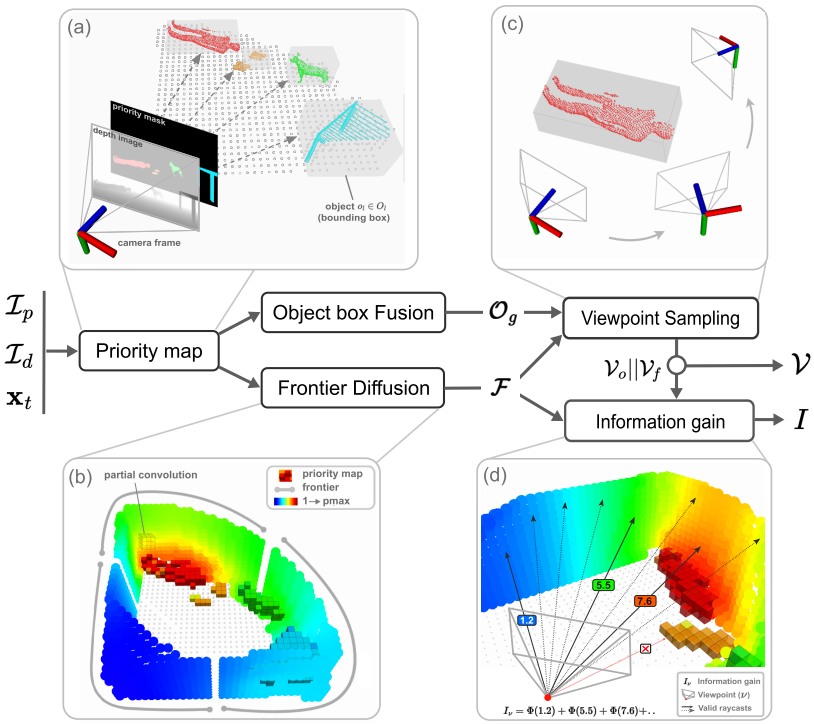


Figure 2: Active Perception pipeline. In (b), the priority map voxels are cubes, and frontier voxels are spheres. (c) shows sampling for object viewpoints, see Fig. A.1 for frontier viewpoints.

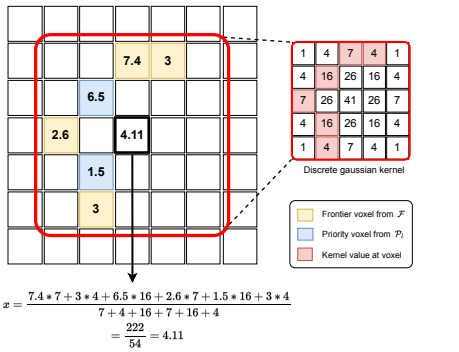


Figure 3: Partial convolution process in 2D

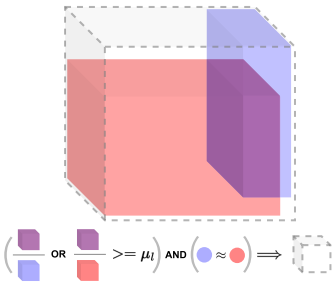


Figure 4: Object box fusion schematic. Cuboids are volumes and circles are priorities

μ_g. Since this is done after merging the local objects, $\mathcal{O}_g$ remains bounded in size, and this second iteration of fusion is also efficient in practice.

The object fusion module results in a global set of distinct objects O_g that remains consistent over time and is updated with the most recent point cloud measurement Ω.

4.3.3 Frontier Diffusion

The goal of this module is to get a set $\mathcal{F}$ of prioritized frontiers. This motivation for this section directly draws from the goal of minimizing unknown space *near* objects of interest (Section 4.1). In exploration terminology (Section 3.1.1), if frontiers can be biased to have higher weights near objects of interest, we can refine search to interesting regions of the space and find the target faster.

To implement this mathematically, the priority values from the local map section $\mathcal{P}_l$ are diffused into neighboring frontier voxels using a 3D partial convolution. A partial convolution was chosen for this diffusion process because frontiers are sparse structures in the voxel grid and a partial convolution allows normalization for only valid voxels [43]. A Gaussian kernel with spread σ, and size W is used, thus making it a 3D Gaussian filter. Figure 3 shows an example calculation in 2D, and Fig. 2b shows a simulation from RViz, where the diffusion process is applied to 3D frontiers.

The diffusion process is applied to each frontier voxel in a *local* region surrounding the drone to maintain computational efficiency. Note that coverage-based exploration frameworks such as [5] keep a minimum size for frontier clusters. We avoid this thresholding because, in semantic target search, even small regions of space can hold significance.

The frontier diffusion module thus results in a set of $\mathcal{F}$ of exploration frontiers embedded with semantic priorities which can further be used for downstream tasks like informative path planning.

4.3.4 Viewpoint Sampling

This module generates a set of viewpoints $\mathcal{V}$ which are candidate poses sampled in free space to 'view' the set of frontiers $\mathcal{F}$.

We use frontiers $\mathcal{F}$ and objects O_g to generate two distinct viewpoint sets called Frontier Viewpoints ($\mathcal{V}_f$) and Object Viewpoints ($\mathcal{V}_o$). Frontier viewpoints are generated by uniformly sampling poses around frontier clusters, the same as the process in [5]. Object Viewpoints are generated around each object in the global set of objects O_g.

When planning a path through viewpoints, $\mathcal{V}$ contains either $\mathcal{V}_o$ or $\mathcal{V}_f$. $\mathcal{V}_o$ is used first as a priority and $\mathcal{V}_f$ is used if $\mathcal{V}_o = \phi$. This approach works well in practice because sampling around objects helps reduce uncertainty in semantically important regions and provides high-quality observations of the objects [25].

The uniform sampling process is defined as follows. Let $\mathbf{r}_c$ be the origin of the cylindrical coordinate system used for sampling. A viewpoint $\nu_{r,\theta,z}$ is then generated at angle θ, radius r, and height z using Eq, 4.

$$\nu_{r,\theta,z} = \mathbf{r}_c + r * (cos(\theta), sin(\theta), z) \quad (4)$$

where,

$$r = r_{min} + i(r_{max} - r_{min})/n_r \ \forall i = 1 \rightarrow n_r \quad (5)$$
$$\theta = \theta_{min} + j(\theta_{max} - \theta_{min})/n_\theta \ \forall j = 1 \rightarrow n_\theta \quad (6)$$
$$z = r_z^c \quad (7)$$

For viewpoints in set $\mathcal{V}_f$, $\mathbf{r}_c$ is equal to the centroid of the points in a frontier cluster, and for set $\mathcal{V}_o$, $\mathbf{r}_c$ is the centroid of an object's axis-aligned bounding box (see Fig. A.1 and 2b). The variables r_{min} and r_{max} define the range for sampling the radius, and n_r is the number of samples taken. Similar notation is used for θ. The sampling height is just the z component of $\mathbf{r}_c$.

Additionally, for object viewpoints, we propose a perception-aware filtering process that discards viewpoints unable to fully see the object o in its field of view. The camera's sensor model is used to achieve this. First, the eight corner positions of an object's bounding box are projected to the camera image plane using the inverse of the procedure from Section 4.3.1 to create a set $\mathcal{R}_o$. An object is considered 'in view' when all 2D points in set $\mathcal{R}_o$ are inside the image bounds. Equation (8) describes this condition.

$$|u| >= \mu_s.w \text{ AND } |v| >= \mu_s.h \quad \forall (u,v) \in \mathcal{R}_o \quad (8)$$

Viewpoints that do not satisfy the condition in Eq. (8) are discarded. The tolerance parameter μ_s is a small value that keeps the detected object safely within the bounds and thus accounts for imperfect sensing. The perception-aware filtering allows the drone to maintain consistent object detections across time and enables simultaneous inspection of objects, which is often an auxiliary goal when looking for targets.

4.3.5 Information Gain

This section describes a novel formulation for computing a balanced coverage and semantic information gain for a viewpoint $\nu \in \mathcal{V}$, using the prioritized frontier set $\mathcal{F}$.

Consider Fig. 2d, where the frontier voxels are colored based on their priorities. Rays are cast from a candidate viewpoint ν toward the voxels in $\mathcal{F}$ to determine the priority value at the ends of valid rays. A ray is considered valid when it is unobstructed by occupied or unknown space. Voxels at the end of valid rays create a new *visible* frontier set $\mathcal{F}_\nu \subset \mathcal{F}$. Each priority value in $\mathcal{F}_\nu$ is then passed through a transfer function Φ and summed up to give the total information gain I_ν of the viewpoint ν (see Eq. (10)).

$$\Phi(f) = max(e^{\gamma(f-1)}, 1) \quad (9)$$
$$I_\nu = \sum_{f \in \mathcal{F}_\nu} \Phi(f) \quad (10)$$

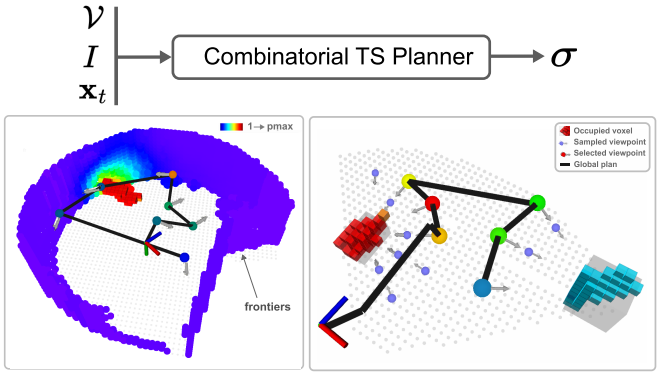


Figure 5: Global plan created for frontier viewpoints $\mathcal{V}_f$ (left) or object viewpoints $\mathcal{V}_o$ (right)

Here, γ is a parameter that decides the rate at which priority values are weighted on an exponential curve. The function Φ works with the range of priority values $[1, p_{max}]$ to balance coverage exploration and semantic target search. To demonstrate this, consider an exploration frontier that is far from a semantically interesting object and thus consists of voxels that have the least priority, i.e. 1. When Eq. (9) is used to calculate the gain for such a frontier, it will equal volumetric coverage because $\Phi(1) = 1$. This situation is equivalent to coverage-based exploration methods.

Contrarily, when a frontier is close to a semantic object (Fig. 2d), the raycasting procedure will weigh high-priority voxels exponentially, thereby prioritizing viewpoints that are facing semantically meaningful regions of 3D space. The parameter γ determines the greediness of the viewpoint evaluation. A higher γ selects viewpoints oriented more towards semantically interesting regions, while a lower gamma makes the search more coverage-based. Note that a γ value of 0 equals coverage-based exploration.

Thus, our method of calculating information gain from frontier voxels unifies coverage and semantic exploration in a single calculation at the viewpoint level and provides a parameter to bias search towards the target.

The Active Perception module thus results in a set $\mathcal{V}$ of viewpoints sampled around either Frontiers or Objects and a set I of information gains with balanced volumetric and semantic utility.

4.4 Combinatorial Target Search Planner

The goal of the global planner is to use a set of viewpoints $\mathcal{V}$, their respective information gains I, and the drone's state $\mathbf{x}_t$ to plan a global path that minimizes time.

Owing to recent successes in combinatorial planning for exploration, we solve a combinatorial optimization problem to create a global plan. Contrary to a classical TSP which minimizes tour distance, semantic target search focuses on prioritizing viewpoints with high semantic gains. To achieve this, the works in [12] and

[6] modify the tour cost $C(\sigma)$ to minimize the sum of waiting times (or latency) for all nodes, weighted by the information gain. This can be seen as a *Weighted Minimum Latency Problem*. The planner in [12] is most relevant to our work and we extend it to create efficient time-optimal tours through 3D viewpoints.

Let σ be a potential tour represented as a permutation of the viewpoints in set $\mathcal{V}$. The information gains in set I are used as weights for each corresponding viewpoint's waiting time in the MLP formulation [12]. The cost function used in the weighted MLP for a tour σ is then defined as:

$$C(\sigma) = \sum_{i=1}^{\|\sigma\|} I(\sigma(i)) \sum_{j=1}^{i} \mathbf{C}_{\sigma(i)\sigma(j)} \tag{11}$$

Intuitively, solving the objective from Eq. (11) means that viewpoints with higher information gain get scheduled earlier in the tour. Since the information gain was calculated by balancing coverage and semantic priorities in section 4.3.5, this makes the plan more robust under semantic uncertainty, ensuring that exploration is not significantly compromised in pursuit of the target. This is better than the greedy Next Best Viewpoint planner [15] because it minimizes the average waiting time across *all* relevant viewpoints.

The work in [12] uses the distances between two viewpoints as elements of the cost matrix $\mathbf{C}$. However, when using robots with complex dynamics such as MAVs, the time to switch between two viewpoints is a more appropriate cost function. Thus, we use the kinematic cost function from [5] to create the elements of the cost matrix.

Let $\mathcal{V}^* = \mathcal{V} \cup \mathbf{x}_t$ be the modified set of viewpoints containing the agent's pose at time t. $\mathbf{C}_{ij}$ is then defined as the maximum time required to switch between two viewpoints $\nu_i, \nu_j \in \mathcal{V}^*$.

$$\mathbf{C}_{ij} = max\left(\frac{\text{length}(\nu_i^{\mathbf{P}}, \nu_j^{\mathbf{P}})}{v_{max}}, \frac{|\nu_i^{\psi} - \nu_j^{\psi}|}{\omega_{max}}\right) \tag{12}$$

Here, $\nu_i^{\mathbf{P}}$ is the 3D position, and ν_i^{ψ} is the yaw angle of the i^{th} viewpoint from set $\mathcal{V}^*$. The cost function in Eq. (11) using the cost matrix from Eq. (12) is minimized using the Large Neighbourhood Search and 2-Opt meta heuristics to obtain a near-optimal tour σ. Figure 5 shows the resulting tour for both Frontier Viewpoints $\mathcal{V}_f$ and Object Viewpoints $\mathcal{V}_o$.

In summary, the tour σ minimizes the time to arrive at semantically important regions of the environment, thus providing situational awareness in time-critical emergency response scenarios.

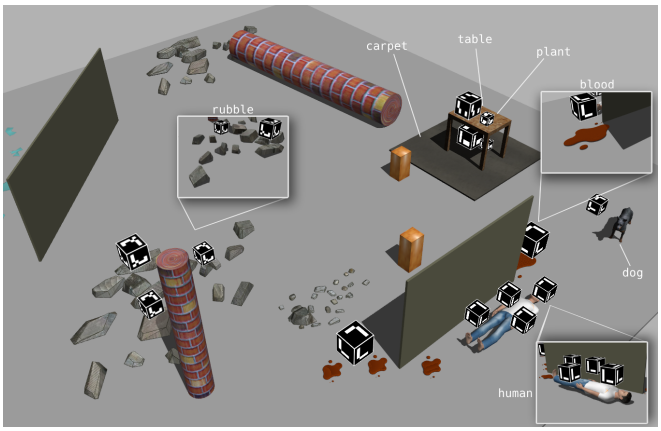


Figure 6: Earthquake environment(Gazebo). Top view in Fig. B.2

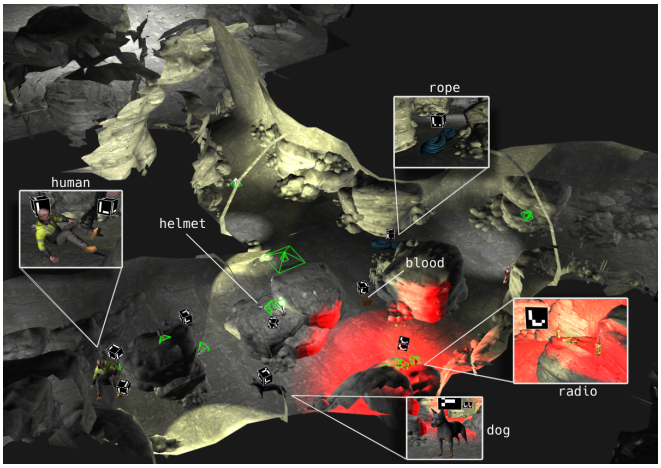


Figure 7: Cave environment(Gazebo). Top view in Fig. B.3

5 Experimental Setup

5.1 Simulation Environments

Two realistic simulation environments were used to evaluate the algorithm in the PX4-Gazebo SITL simulator[1]. The *Earthquake* is a custom environment (Fig. 6) and the *Cave* (Fig. 7) is a section of the 'Cave Circuit 02' world from the DARPA SubT Challenge[2].

A common set of objects was used as semantic clues for both environments. These objects are a combination of object classes from the DARPA SubT challenge, ChatGPT prompts, and common sense objects that are expected near a `human` target in search and rescue situations (see Table B.1). However, it was observed that including only expected objects could introduce bias, making it hard to conclude if using semantic relationships was beneficial. To reduce bias, unexpected objects like `toy` and `plant` were also included to create sufficient diversity in objects and ensure fairness of the priority inference model. Both environments contain a trapped human as the target, which was sufficiently occluded to make the problem challenging.

[1] docs.px4.io/main/en/simulation/ros_interface.html
[2] https://www.darpa.mil/program/darpa-subterranean-challenge

Note that Gazebo is not a photo-realistic simulator and existing detection pipelines performed poorly (see Fig. B.4). Since training a full semantic segmentation pipeline is beyond the scope of this work, ArUco markers were used as a proxy for 2D image segmentation. These markers were placed near their respective semantic objects, and the 2D segmentation image then contains a pixel-wise label for each marker $o \in \mathbb{S}$.

Both environments were made sufficiently large to ensure realistic exploration and the drone was started at the same pose for all simulation episodes (see Table 2). The start pose was chosen to be as far as possible from the target to observe the effect of semantic exploration. In the earthquake environment, this starting pose is at the door entry to the room, and in the cave environment, it is the default staging area used in the DARPA challenge. It was observed that there was sufficient variance in the SITL simulation because of drone dynamics, virtual sensors, or non-deterministic behavior in parts of the FUEL pipeline. Therefore, multiple start poses were not recorded.

5.2 Software Architecture

One of the primary goals of the thesis was to extensively validate simulation results with hardware experiments. This decision significantly influenced the choice of algorithms, software packages, and the general mindset of software development. It was important to invalidate potential software packages as early as the literature study phase. For instance, we rapidly tested mapping frameworks like HYDRA [44], RACER [45], and FUEL [5] and found FUEL to be most relevant to the active perception task. The choice of the appropriate simulation pipeline was made after invalidating gym-pybullet-drones [46] and choosing the PX4-SITL framework because of easier sim2real transfer.

Our universal software stack is based on the Robotics Operating System (ROS) and integrates target search capability into a modified exploration pipeline from [5]. The architecture is demonstrated in Fig. 8. A key capability of the software is that we use the same pipeline for both hardware and software experiments, with the only difference being the source of the measurement tuple z_t. For simulation, this measurement comes from the Gazebo simulator, whereas for hardware experiments this measurement comes from the onboard camera and positioning system. This setup establishes a general pipeline that can be expanded to other photorealistic simulators or more complex real-world environments with onboard position systems like Visual Inertial Odometry (VIO).

5.3 Hardware Setup

Hardware experiments were performed with a custom Micro Aerial Vehicle (MAV) built at the Mobile Robotics Laboratory at Delft University of Technology [47]. The MAV is equipped with an Intel Realsense D455 camera and an Nvidia Jetson Xavier NX onboard computer. A

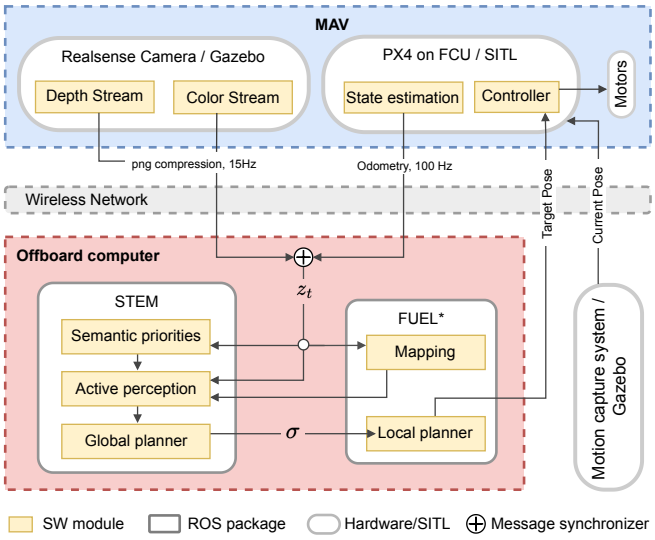


Figure 8: Software architecture. *A modified version of FUEL [5] is used for mapping and local planning.

Parameter	Value	Parameter	Value
α	0.9	μ_s	0.1
v_{max}	0.5 m/s	a_{max}	0.5 m/s^2
ω_{max}	0.7 rad/s	h	480
w	848	σ	2
W	5	p_{max}	8
γ	4	ν^o_{min}	25
ν_{min}(e.q.)	20	ν_{min}(cave)	10
$\|\mathcal{S}\|$	22	λ_{min}	0.01
μ_l	0.5	μ_g	0.5

Table 1: Parameter values for experiments. Note: (e.q. = earthquake). ν_{min} and ν^o_{min} are minimum information gain values for frontier and object viewpoints, respectively.

HolyBro Kakute F7 V2 flight controller was used with PX4 autopilot software. The MAV was localized in the environment via a Vicon motion capture system. ArUco markers were placed in the environment as semantic objects of interest and the experiments were conducted in sufficiently cluttered configurations with screens and boxes as obstacles. Figure 9 shows a potential environment for the experiment.

Hardware experiments with MAVs come with a unique set of challenges and some of them are briefly mentioned as follows:

- Our method strongly relies on color images for

Environment	Bounds (x, y, z)	$\mathbf{x}_0$ (x, y, z, ψ)
Earthquake	$(12, 12, 1.5)$	$(-4.5, -4.5, 0, 1.57)$
Cave	$(30, 30, 2.5)$	$(0, 0, 0, 0)$
Real-world	$(6, 6, 2)$	-

Table 2: Maximum volume bounds and start poses ($\mathbf{x}_0$) for the environments. Positions (x, y, z) are in meters, and yaw angle (ψ) is in radians. For hardware experiments, start pose was kept random.

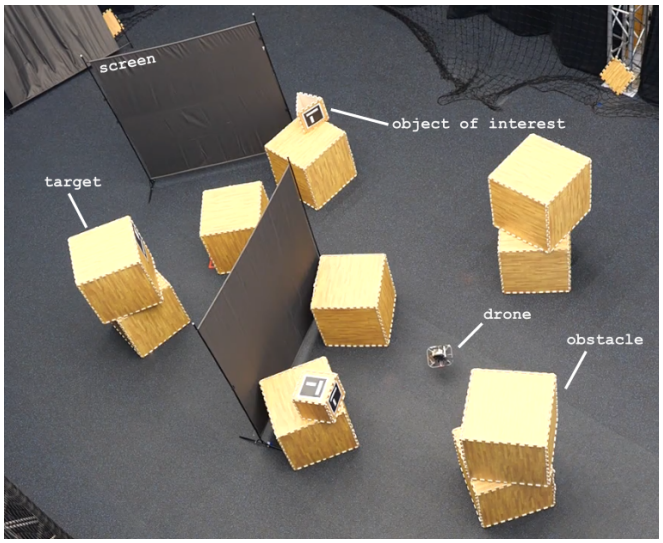


Figure 9: Lab environment for hardware experiments

creating the priority mask. PNG compression was used since the images were exchanged over a bandwidth-limited Wi-Fi Network. Level 5 PNG compression gave a sufficient frequency of 15 Hz, and a bandwidth of ≈ 1.5 MB/s without compromising on resolution. We used similar compression for the depth stream.

- When working with drones, dynamics strongly influence all parts of the pipeline. Even though the mapping framework reported good results for yaw rates up to 1 rad/s, they had to be reduced because motion blur caused inconsistent detections and poor obstacle avoidance.
- Since AruCo markers were used for detection, false detections resulted in oscillating semantic priorities (see Fig. B.6). However, progressive map updates and the diffusion process made the algorithm robust to semantic uncertainty.

6 Simulation Results

In this section, we present and discuss the results of the proposed method on the simulation environments from Section 5.1. Evaluation metrics are proposed in Section 6.1, and three baselines are chosen for comparison in Section 6.2. We provide the primary performance comparisons with baselines in Section 6.3 which are further supported using qualitative results in Section 6.4. Finally, we also analyze our method in depth using auxiliary analysis and ablation studies in Sections 6.5 and 6.6 respectively.

6.1 Evaluation Metrics

6.1.1 Performance Metrics

Success %: This metric calculates the percentage of episodes when the target was successfully found in a total of n trials. An object o (or target) is considered found when the fraction of object pixels λ_o crosses a threshold λ_{min} in the segmentation image (see Eq. 14). For example, a value of $\lambda_o = 0.02$ means that the marker occupies 2% of the field of view in the image plane (see Fig. B.5).

$$\lambda_o = \frac{\beta}{wh} \tag{13}$$

$$Success(o) = \begin{cases} 1 & \lambda_o \geq \lambda_{min} \\ 0 & \lambda_o < \lambda_{min} \end{cases} \tag{14}$$

Here, o is an object, β is the number of pixels belonging to the object's ArUco marker, and w and h are the segmentation image width and height respectively. The parameter λ_{min} depends on the environment complexity and camera intrinsic matrix $\mathbf{K}$.

Time to target: A commonly used metric for ObjectNav tasks is Success weighted by Path Length (SPL) [48]. A notable drawback of this metric is that it only considers travel distance in SE(3) and for robots with complex dynamics (like MAVs), the completion time is recommended [49]. Thus, we record the first time instant when a target o^* was successfully detected (i.e., $\lambda_{o^*} \geq \lambda_{min}$) and call this metric as the Time to target t^*.

Exploration time: Since balancing exploration and target search is a key goal for our method, we also measure the exploration time (t_f) in seconds. An environment is considered explored when no *visible* frontier can be found for 10 consecutive iterations. For a frontier to be considered *visible*, it must have (1) at least F_{min} number of clustered voxels, and (2) at least one viewpoint with minimum information gain ν_{min}. These conditions are directly based on the work in [5].

6.1.2 Auxiliary Metrics

Singular data points like completion time and success % are easy to record and compare, but they can have high variance and do not quantify the quality of search [48]. We propose two additional metrics in this section which are used for auxiliary analysis and ablation studies in further sections.

Time in view: We quantify the quality of object observations using a new metric called Time in View (TIV). This metric measures the time in seconds that an object was successfully detected (i.e., $\lambda_o > \lambda_{min}$) throughout the episode. We record the TIV for all objects in the environment, and a higher TIV indicates better performance. Intuitively, this metric acts as a proxy for temporal consistency of object detection scores [50]. The rationale for using this metric is further explained in appendix C.1.

Cumulative Information Gain: Semantic target search methods bias exploration toward objects of interest in the environment because searching all unknown space is inefficient when looking for targets. To quantify the search quality of such methods, we propose a new metric called the Cumulative Information Gain (CIG), I_t^w. CIG is defined as the cumulative

weighted information gain recorded at time instant t, calculated using the weighted entropy of a ground truth importance map $\mathcal{A}$. Intuitively, this metric measures how well the uncertainty reduces when certain regions of the map carry more semantic importance than others. The process to calculate I_t^w and create the importance map is delineated in appendix C.2.

A graph for I_t^w can also be plotted against time to show how uncertainty reduces. Additionally, the completeness of the search can be measured by the gain at the end of the episode i.e. $I_{t_f}^w$. The rationale for using this metric is further explained in appendix C.2.

6.2 Baselines

The proposed method is compared to two coverage-based exploration methods and a saliency-aware exploration method. The experiment conditions for the three baselines are as follows:

1. FUEL: This refers to the pure coverage-based exploration algorithm used in [5]. The sensor range R_{max} for FUEL was kept the same as our work (3 meters) to make comparisons fair. The minimum frontier size ($F_{min} = 100$) and minimum viewpoint coverage gain ($\nu_{min} = 20$) parameters were left unchanged from the original work. The episode ends when there are no new frontiers detected.

2. FUEL-complete: It was noticed that finding the target in small regions depended strongly on the F_{min} and ν_{min} parameters. Therefore, in the FUEL-complete method, these parameters were tuned optimally to balance target search and exploration. For the Earthquake scenario, $F_{min} = 0$ and $\nu_{min} = 10$ were used. For the Cave scenario, $F_{min} = 0$ and $\nu_{min} = 0$ were used. Additionally, frontier down-sampling (see [5]) also had to be turned off due to the narrow passages and small frontier sizes in the Cave environment.

3. VSEP: This refers to the exploration framework from [29] which guides the robot to important regions using a visual saliency mask. The original work uses color filtering operations to create this mask. However, this is not relevant to our work, and to keep the comparison fair, we use the priority mask generated from our method (Section 4.2) as the 2D saliency mask. It was also observed that VSEP was highly inefficient with the small sensor ranges in our study (Fig. C.9) and some relaxations were essential for the comparison. We use $R_{max} = 7m$, as opposed to 3 meters. This made the sensing less difficult and helped the RRT planner to create better plans. Note that the authors also use 7 meters to report results in their original paper. v_{max} was left unchanged ($0.5m/s$) but ψ_{max} was increased to $0.7rad/s$.

6.3 Performance Results

This section presents the performance results of our method compared to the three baselines from Section 6.2 based on the performance metrics from Section 6.1.1. Table 3 summarises results for both Earthquake and Cave simulation environments. Data was gathered from 10 simulation runs for the Earthquake environment and 5 simulation runs for the Cave environment. The results are discussed in the following paragraphs.

FUEL rarely finds the target but consistently completes exploration faster due to two reasons:

- It relies on volumetric coverage gains from frontiers, which do not vary significantly in size. This consistency in information gains across frontiers leads to global plans that are more stable over time, allowing the MAV to maintain high speeds throughout the episode.

- It solves a metric TSP using the LKH heuristic, which is widely recognized for its robustness and efficiency. This allows FUEL to converge to better solutions with fewer iterations, resulting in an overall improved global plan.

FUEL-complete has good target success rates but it takes longer to find the target as compared to STEM. This is because FUEL-complete does not use semantic information to guide search. Moreover, exploration times for FUEL-complete are higher compared to FUEL. This is expected because lowering F_{min} and ν_{min} significantly affects the exploration efficiency. A lower F_{min} allows clustering frontiers even in tight spaces, and a lower ν_{min} retains viewpoints for these small clusters. While this helps the MAV in finding the target, it also means the drone tries to cover small, unimportant areas, thus reducing speed and increasing time. This behavior underscores the necessity of prioritizing exploration *only* towards regions of high semantic importance, which is the primary motivation of our work.

In complex environments like the Cave which has significant occlusions, the success rate for FUEL-complete drops because the drone is unable to reach in tight spaces without the explicit object viewpoints that our method uses. This phenomenon is also observed when λ_{min} is increased (see Table C.2) because FUEL-complete only relies on discovering the target opportunistically without getting close to it.

VSEP performs poorly in both exploration and target search due to two reasons:

- First, VSEP gets stuck around irrelevant (low priority) semantics because it compresses the 3D saliency map into 3 discrete saliency modes i.e. `Salient`, `Not Salient`, or `Inhibited`. Thus, it cannot ascertain fine-grained priorities and takes a long time to find the target.

- Second, VSEP relies on a finite horizon sampling-based planner [15]. Under computation and time constraints, such planners often get

		Target search		Exploration
Env	Method	Success %	Time, t^*(s)	Time, t_f(s)
Earthquake $n = 10$	FUEL	10%	101.1 ± 0.0	$\mathbf{106.0 \pm 9.8}$
	FUEL-complete	90%	66.2 ± 21.8	121.7 ± 6.9
	VSEP	20%	165.5 ± 0.0	212.7 ± 25.4
	STEM (Ours)	**100%**	$\mathbf{47.4 \pm 6.5}$	143.4 ± 8.7
Cave $n = 5$	FUEL	0%	-	$\mathbf{108.8 \pm 4.8}$
	FUEL-complete	60%	89.3 ± 24.6	169.6 ± 24.6
	VSEP	0%	-	272.3 ± 84.1
	STEM (Ours)	**100%**	$\mathbf{77.9 \pm 10.1}$	153.9 ± 11.5

Table 3: Comparison study with baselines in the Earthquake and Cave environments. n is the number of statistical trials. Three baselines from section 6.2 were compared to our method on Success percentage, Time to target ($t*$), and exploration time (t_f).

stuck in local optima, resulting in greedy decision-making. Moreover, VSEP uses waypoint planning that causes the MAV to stop and recalculate a new solution at every iteration. This stop-and-go behavior is a known limitation of the receding horizon RRT planner [16]. Even with relaxed parameters, this behavior significantly reduced the speed, and resulted in larger exploration times. In contrast, FUEL uses a hierarchical planner which refines the global plan using a local kinodynamic BSpline, and maintains higher speeds.

STEM consistently finds the target more successfully and faster than all methods while keeping the exploration times under reasonable bounds. This is because of three reasons:

- The diffusion of priority values into frontier voxels helps in orienting viewpoints toward objects that are semantically related to the target.
- The object viewpoint sampling allows the drone to capture multiple viewpoints in semantically interesting regions, and often brings related objects (including the target) into view. For instance, in Fig. 11, inspecting the `blood` semantic allows the `human` to come into view. In contrast, VSEP uses the Inhibition Of Return mechanism [29] for salient objects, which only views a semantic from a single viewpoint and misses out on crucial information around it. Section 6.6.1 analyzes the viewpoint sampling method further via an ablation study.
- The combinatorial target search planner minimizes the priority-weighted latencies of viewpoints in the tour. This allows high semantic priority viewpoints to be scheduled earlier, enabling the MAV to reach semantically important regions more quickly. Since the priorities are inferred from target-object relationships, following these semantically important regions eventually leads the MAV toward the target. Section 6.6.2 analyzes the planning strategy further via an ablation study.

6.4 Qualitative Results

In this section, the drone's behavior is visualized to support the quantitative results from Section 6.3.

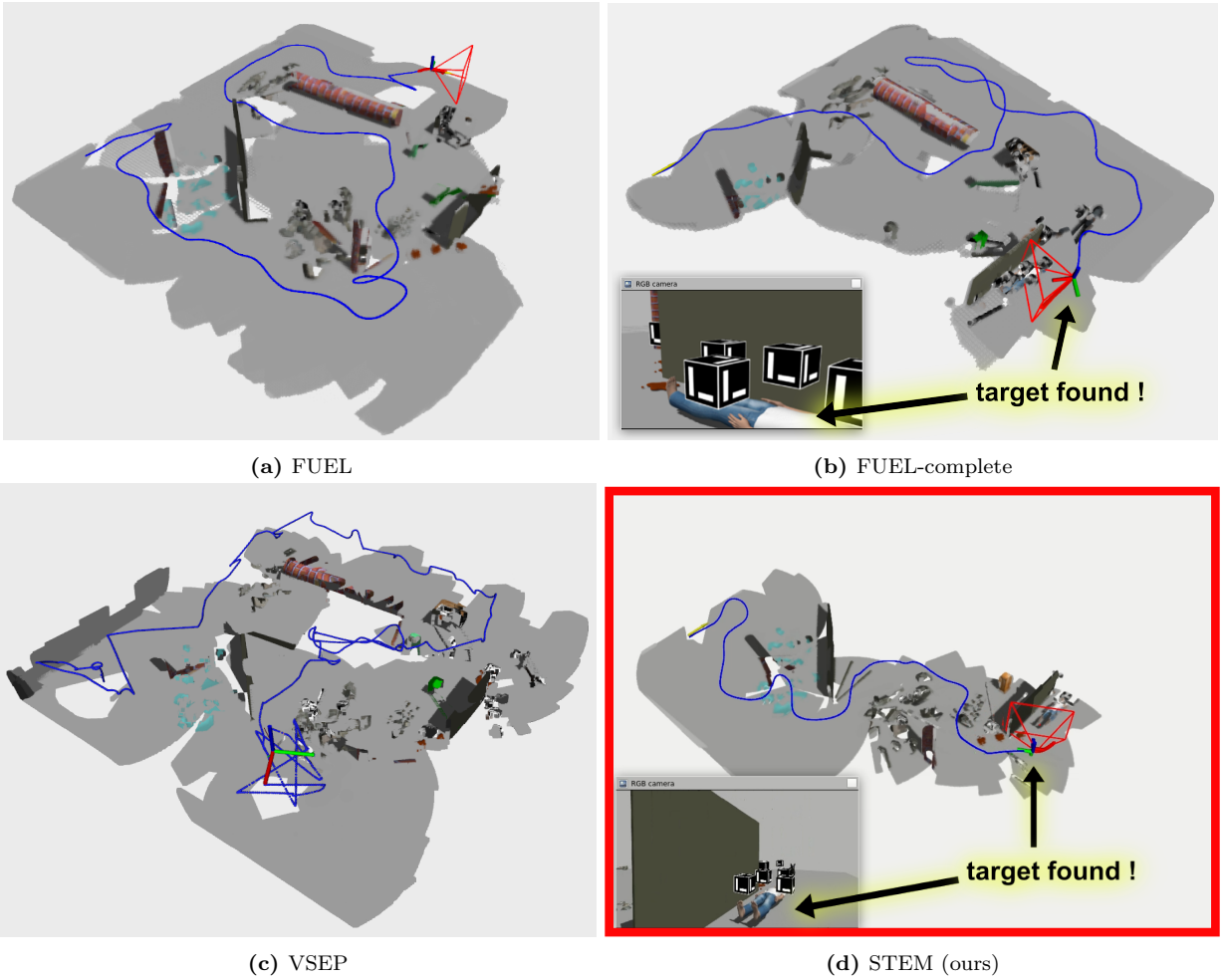
Figure 10 shows a qualitative comparison of the methods in the Earthquake environment. The episodes show the drone's trajectory and reconstructed point cloud from RViz. The episode completes when the target is detected or exploration is completed, whichever comes first. The RGB view of the front camera is also displayed whenever the target is discovered.

VSEP produces the worst trajectories because waypoint-based planning results in stop-and-go behavior. This produces inconsistent plans, path intersections, and reduces speed. It was also observed that VSEP often got confused by the semantic information rather than utilizing it effectively. In Fig. 10c, this behavior can be observed where the MAV spends a significant amount of time in the center, where there is a dense concentration of ArUco markers.

For **FUEL** (Fig. 10a) the episode terminates before completing the full map or finding the target. This is because the semantics are often located in tight spaces, where frontier sizes can get very small, and $F_{min} = 100$ is not small enough to reach these small spaces. In contrast, **FUEL-complete** (Fig. 10b) finds the target because setting F_{min} to 0 means it searches every possible place before the episode ends. However, FUEL-complete takes unnecessary detours because it only uses coverage gains and ignores semantic information.

STEM arrives at the target most efficiently because it utilizes frontier diffusion to move the drone to regions of high priority. Additionally, balancing semantic target search with exploration gives trajectories that not only inspect the OOIs sufficiently but also arrive at the target quickly.

Figure 11 further shows keyframes of our algorithm performing target search in the earthquake environment. The MAV starts at a disadvantaged position and explores the environment first to gather information. When it comes across semantically relevant objects such as `blood`, it samples informative viewpoints near these objects. Planning a path through these viewpoints using the target search planner allows the target to come into view,



(a) FUEL **(b)** FUEL-complete

(c) VSEP **(d)** STEM (ours)

Figure 10: Qualitiative comparison with baselines for the Earthquake environment. Episodes were recorded until t^* or t_f, whichever comes first. The reconstructed point cloud from RViz is shown along with the drone's trajectory in blue. The camera FOV is shown in red and the RGB image is displayed when the target is found

even in tight spaces (behind the brown wall).

Figure 12 shows the results from the Cave environment. Even under the presence of complex 3D geometry and occluded semantic objects, our method performs competently. Figure 12a shows a section of the Cave and the drone's trajectory as a motion trail. After exploring initially, the MAV comes across the `rope` and `radio` and `dog` semantics, which guides it to the target quickly. The top view of the complete trajectory in Fig. 12b shows that the semantics guide the drone into the right arm of the Cave where the target is located. Figure 12c further demonstrates that the drone does not compromise significantly on the overall exploration trajectory in pursuit of the target.

6.5 Auxiliary Analysis

The performance results from Section 6.3 are further analyzed in this section using the auxiliary metrics from Section 6.1.2.

Figure 13 shows the Time in View (TIV) for each semantic class in the Earthquake environment. STEM spends a higher time viewing each semantic on average as compared to other methods. Although VSEP may seem to outperform STEM on some classes like `rubble` and `table`, this is due to VSEP's tendency to get stuck near a particular object. The large episode completion time t_f and qualitative results in Section 6.4 confirm this behavior for VSEP.

Figure 14 shows the Cumulative Information Gain vs. time curve and shows how the exploration progresses. It can be noticed that for the early parts of the episode ($t < 50s$), STEM has a higher slope on the curve, signifying that it comes across important semantic regions faster than FUEL-complete. Although FUEL has a similar slope in this region, it is because it covers unimportant regions at a higher rate, and not because it utilizes semantic information. VSEP takes a significantly longer time to complete the episode and comes across objects of interest much later. This shows that an efficient exploration pipeline is crucial for competent target search performance.

An important point on this curve is the information gained at episode completion i.e. $I^w_{t_f}$. This point indicates the completeness of the search. FUEL achieves a $I^w_{t_f}$ of 0.95, while STEM and FUEL-complete achieve 0.99. While these values seem close, the actual difference is substantial due to the large normalization factor

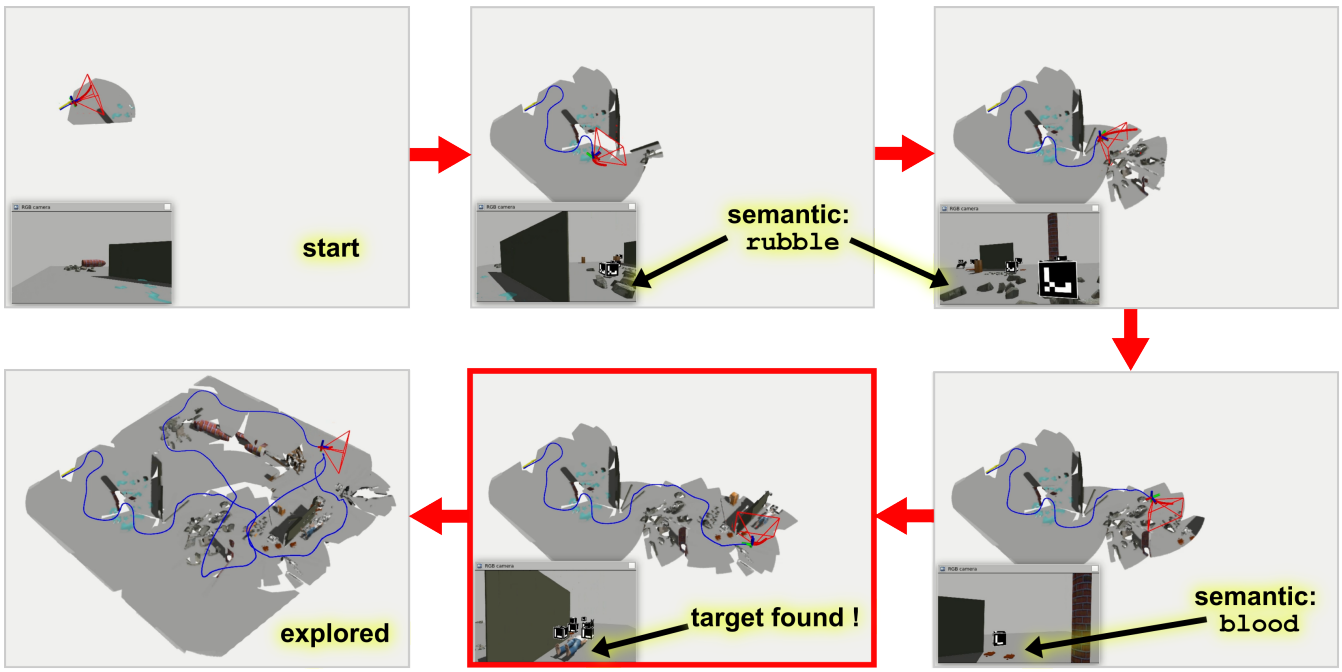
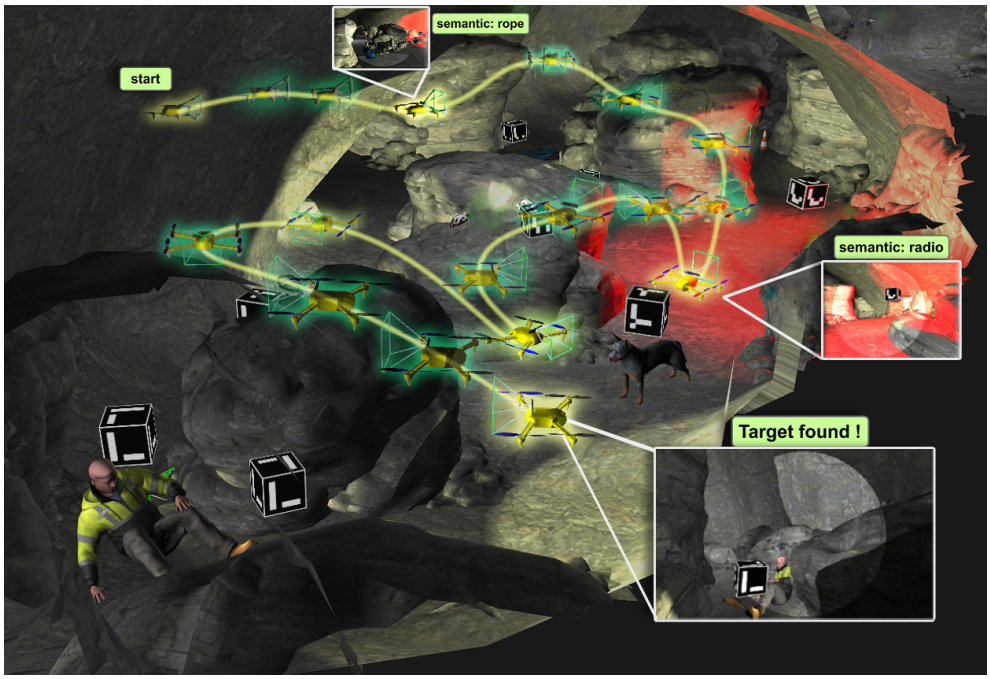
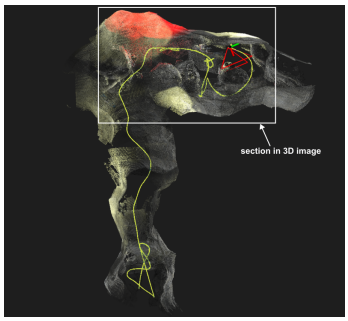


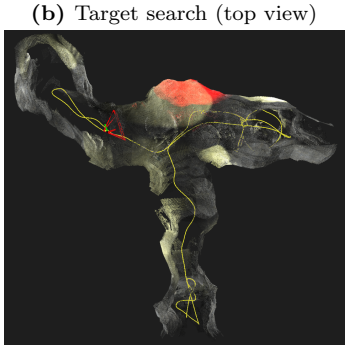
Figure 11: Target search in Earthquake environment. The drone starts at a disadvantaged position and first explores the volume to find semantic clues. The diffusion of priority values combined with a combinatorial planner help the drone find the target (human) very quickly. After locating the target, the full volume is also explored until no new viewpoints are found.



(a) The MAV's trajectory (yellow) inside the Cave



(b) Target search (top view)



(c) Exploration (top view)

Figure 12: Target search in the Cave environment. Figure 12a shows that our algorithm is also able to locate the target (human) in complex 3D environments like Subterranean caves. The camera's sensor FOV is also displayed which demonstrates that the drone performs simultaneous inspection of semantic objects. The section in Fig. 12a is highlighted white in the top view (Fig. 12b). Figures 12b and 12c show the top view of the Cave and the reconstructed point cloud after completing target search and exploration respectively. The red color is part of the simulation lighting and carries no weight.

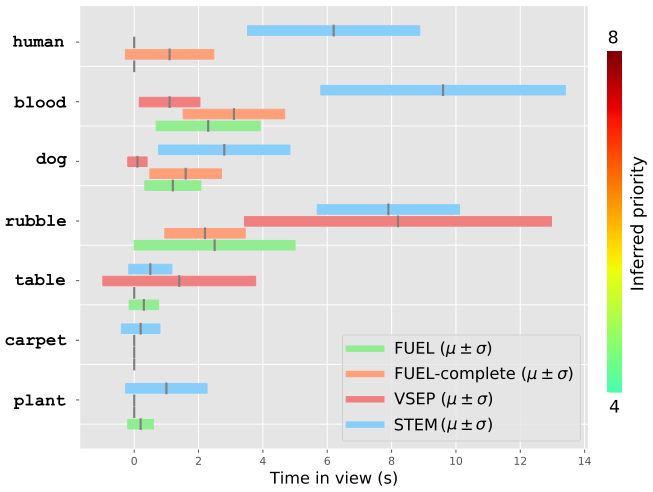


Figure 13: Auxiliary comparison: Class vs. Time in View (Earthquake)

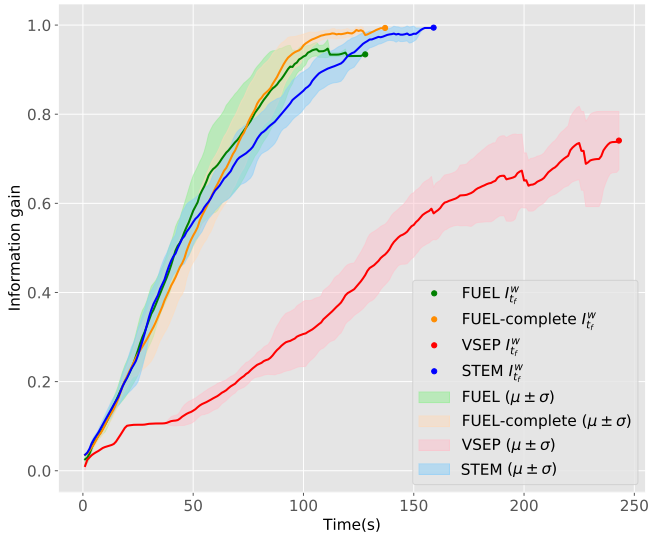


Figure 14: Auxiliary comparison: Cumulative Information gain vs. Time (s) (Earthquake)

H_0^w (equal to 115703.375 for the Earthquake scenario). Understanding this difference qualitatively is also helpful. Since $I_{t_f}^w$ is calculated using the weights from the ground truth importance map (Fig. C.7), leaving an important region unexplored (such as the location of the human), reduces this value. This is precisely why both FUEL and VSEP have a lower $I_{t_f}^w$, as they rarely find the target. In contrast, both FUEL-complete and STEM search important regions, but STEM takes the least time to arrive at the target.

6.6 Ablation Studies

Two ablation studies in Sections 6.6.1 and 6.6.2 are performed for the viewpoint sampling and planning methods, respectively. These studies demonstrate how each component of our pipeline contributes to its overall success or failure.

6.6.1 Viewpoint Ablation

Table 4 presents an ablation study for the viewpoint evaluation method. In this study, the combinatorial planner from Section 4.4 remains fixed, but viewpoints are sampled and evaluated differently.

1. FVP: This method only uses the frontier viewpoints set $\mathcal{V}_f$ without object viewpoints ($\mathcal{V}_o = \phi$). This study helps understand the effect of diffusing priority values into frontier voxels.

2. FVP+OVP: In this case, both sets $\mathcal{V}_f$ and $\mathcal{V}_o$ are used with the switching behaviour mentioned in Section 4.3.4.

FVP+OVP and **FVP** both successfully locate the target, but **FVP+OVP** requires more exploration time. This is because object viewpoints slow down the MAV to inspect objects. Interestingly, FVP is better than FUEL-complete from Table 3 in target search and similar in exploration time. This demonstrates that frontier diffusion contributes significantly to the algorithm's success.

In the Cave environment, FVP outperforms FVP+OVP by a significant margin in target search time because objects are often located in geometrically challenging spaces, and sampling viewpoints here reduces speed for FVP+OVP. However, FVP+OVP has a higher success rate because FVP can merely guide the drone to important regions, and not guarantee target detection. FVP+OVP creates a more focused tour around objects of interest, which allows the MAV to capture multiple viewpoints in semantically important regions, thereby increasing the likelihood of detecting the target.

Figure 15 captures these results graphically using the Cumulative Information Gain curve for a single episode in the Earthquake scenario. Both FVP and FVP+OVP have similar $I_{t_f}^w$, but FVP completes the exploration faster and comes across semantically important regions earlier in the episode. FVP+OVP runs slower but has more instances of target detection (red circles). Figure 16 also supports this notion and shows the viewing times for each semantic. As expected, FVP+OVP is more consistent and simultaneously inspects the objects, followed by FVP.

In summary, the results from this ablation show that the object viewpoints do not consistently improve target search time and result in worse exploration. However, they do improve object detection consistency, adding to the overall robustness of the target search framework.

6.6.2 Planner Ablation

Table 5 presents an ablation study for the planning algorithm. In this study, all methods use both Frontier Viewpoints $\mathcal{V}_f$ and Object Viewpoints $\mathcal{V}_o$ but different methods are used to create a global path.

1. Greedy: In the simplest case, the viewpoint with the highest information gain I_ν is chosen as the next goal.

		Target search		Exploration
Env	Method	Success %	Time, t^*(s)	Time, t_f(s)
Earthquake	FVP	100%	52.8 ± 6.2	122.7 ± 12.0
	FVP+OVP	100%	49.6 ± 8.1	143.4 ± 8.7
Cave	FVP	40%	70.8 ± 0.5	138.9 ± 25.0
	FVP+OVP	60%	94.0 ± 14.4	153.9 ± 11.5

Table 4: Viewpoint ablation study

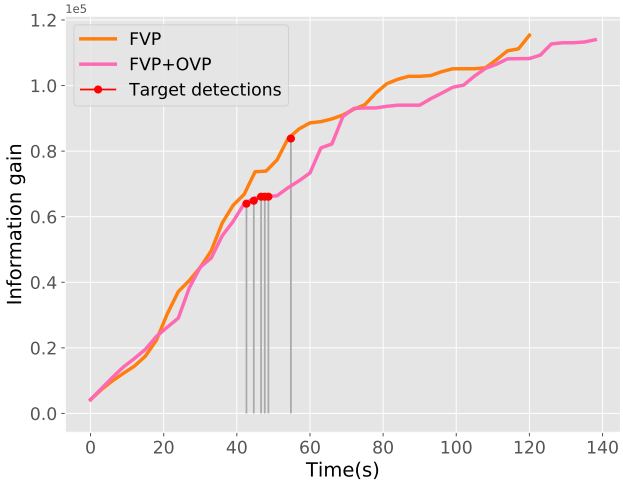


Figure 15: Viewpoint ablation: Cumulative Information Gain vs. Time(s). A single episode from the Earthquake scenario is shown here, to demonstrate the tradeoff between FVP+OVP and FVP.

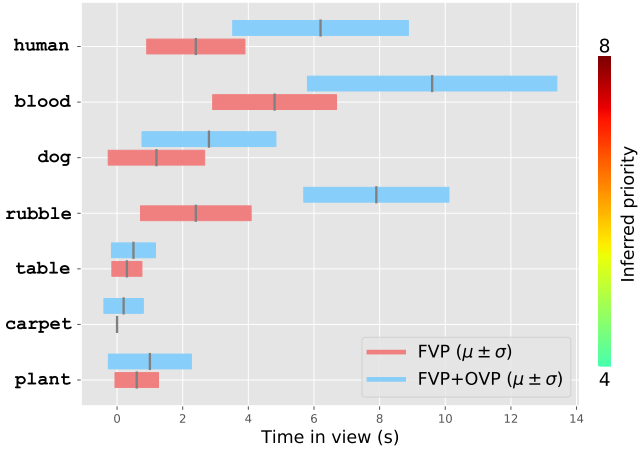


Figure 16: Viewpoint Ablation: Class vs. Time in view (Earthquake)

2. Traveling Salesman Problem (TSP): In this case, the kinematic TSP from [5] is used to create the global path between viewpoints. The Lin Kernighan Heuristic (LKH) is used to solve the optimization problem. Note that even though a metric TSP is solved, the chosen viewpoints are still evaluated using the information gain from section 4.3.5, unlike FUEL which only considers coverage.

3. Weighted Minimum Latency Problem (WMLP): In this case, the problem formulation from Section 4.4 is used to create the plan, and the solver from [12] is used. This situation is the same as FVP+OVP from Table 4 but named differently to emphasize the planner.

WMLP performs marginally better as compared to **TSP** in locating the target and exploring the entire volume. This indicates that the frontier diffusion and object viewpoints sampling are critical elements of the target search algorithm, rather than the planning strategy. The success of TSP is attributed to its use of the robust LKH heuristic. However, only performance metrics do not provide the complete picture, and auxiliary metrics are also discussed in further paragraphs.

The **Greedy** planner performed considerably worse than WMLP or TSP for both target search and exploration. This is expected because, when exploring a previously unseen environment, new regions are incrementally uncovered which may not be as interesting as regions already observed. Consequently, the current information gain could suddenly reduce and the next highest gain viewpoint might be very far, leading to oscillatory behavior (see Fig. C.8). The greedy planner's poor decision-making is further put to the test in the Cave environment, which features narrow passages and occlusions. Here, oscillations induced by the greedy planner have a large time cost, and advanced optimization techniques such as the TSP are crucial to reduce search times. Both WMLP and TSP perform competently in the Cave environment, with little differences in exploration time.

Figure 17 supports the above discussion using the CIG curve for the Cave environment. It can be observed that WMLP has a steeper slope than TSP which indicates that it consistently observes high-priority viewpoints early on in the episode. As expected, greedy search has a steep slope at the beginning because it always seeks out the most informative viewpoint. However, this greedy behavior causes oscillation, flattening the curve later in the episode, and significantly impacting the completion time. All planners have a similar information gain at the end of the episode ($I^w_{t_f}$) because they all use the same strategy for viewpoint planning.

In summary, this ablation study shows that combinatorial optimization is essential for efficient exploration in complex environments. Additionally, priority-weighted combinatorial planners quickly drive

		Target search		Exploration
Env	Method	Success %	Time, t^*(s)	Time, t_f(s)
Earthquake	Greedy	90%	65.4 ± 14.1	187.3 ± 15.6
	TSP	100%	53.2 ± 9.8	142.0 ± 15.2
	WMLP	100%	49.6 ± 8.12	143.4 ± 8.7
Cave	Greedy	40%	145.5 ± 14.9	233.4 ± 20.0
	TSP	60%	98.5 ± 10.3	157.6 ± 15.7
	WMLP	60%	94.0 ± 14.4	153.9 ± 11.5

Table 5: Planning ablation study

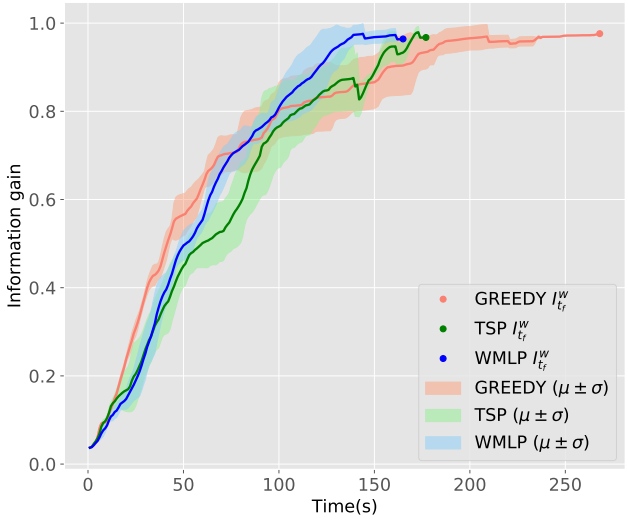


Figure 17: Planner Ablation: Cumulative information gain vs. Time (s) (Cave)

the drone toward semantically important regions of the volume, where a potential target might be located.

7 Real-world experiments

This section showcases the results of real-world experiments using the software architecture from Section 5.2 and the hardware setup from Section 5.3.

Figure 18 shows our algorithm performing simultaneous target search and exploration in a real-world laboratory setting. The MAV takes off from a disadvantaged position and starts exploring the environment, actively searching for semantic objects of interest. When it encounters a semantic object (Fig. 18a), it plans viewpoints around it and uses the diffusion process to prioritize frontiers near the object. This approach balances coverage and semantic information gain. Despite the empty space behind the screen offering greater coverage, the drone is still drawn to the inner space between the two screens, where it eventually finds the target (Fig. 18b). The object-centric viewpoint sampling further allows the drone to inspect the target object and reduce uncertainty in areas of interest. The last segment of the trajectory (Fig. 18c) shows that our algorithm can search in 3D (see Fig. C.10 also). Figure 19 illustrates that our algorithm can generalize across different scenarios with varied object placement and occlusions. In Figures 19b and 19c no target object was present, yet our method continued to explore the environment and observed semantics in search for the target. In Figure 19d, a target is located behind the screen. Despite the complex geometry of the environment occluding the semantic object, the drone successfully located the target by only relying on coverage gains. Finally, Fig. 19a demonstrates the method's ability to search in 3D, with semantic objects placed above a table. The diffusion process diffuses semantic priorities to frontiers below the table, enabling the drone to move along the z-axis and detect hidden targets underneath. Such maneuvers could be crucial in cluttered search and rescue scenarios.

8 Conclusion

In this work, a framework was developed for semantic target search and exploration using MAVs in unknown, cluttered environments. The central part of the framework was the use of appropriate 3D environment representations embedded with semantic priorities, which enabled informative path planning. Specifically, a masking technique was created to compress segmentation images into a priority mask using a Large Language Model. Following this, a comprehensive active perception pipeline was developed to guide the drone toward semantically interesting regions of 3D space. This was achieved by diffusing semantic priorities into exploration frontiers and generating informative viewpoints around objects of interest. Finally, an efficient global plan was formulated over the viewpoints with a combinatorial planner that balanced metric and semantic costs, allowing for the rapid location of the target.

We performed several simulation experiments to show the capability of our method in cluttered environments like the Earthquake and the Cave. A comparison study was performed on quantitative metrics such as time to target, success rate, and exploration time. Our method outperformed baselines by increasing the target success rate and reducing the completion time for target search while incurring acceptable costs in exploration time. In complex environments like the cave, which are dominated by occlusions, our method was still competent and simultaneously balanced exploration and target search very well, which are often competing objectives.

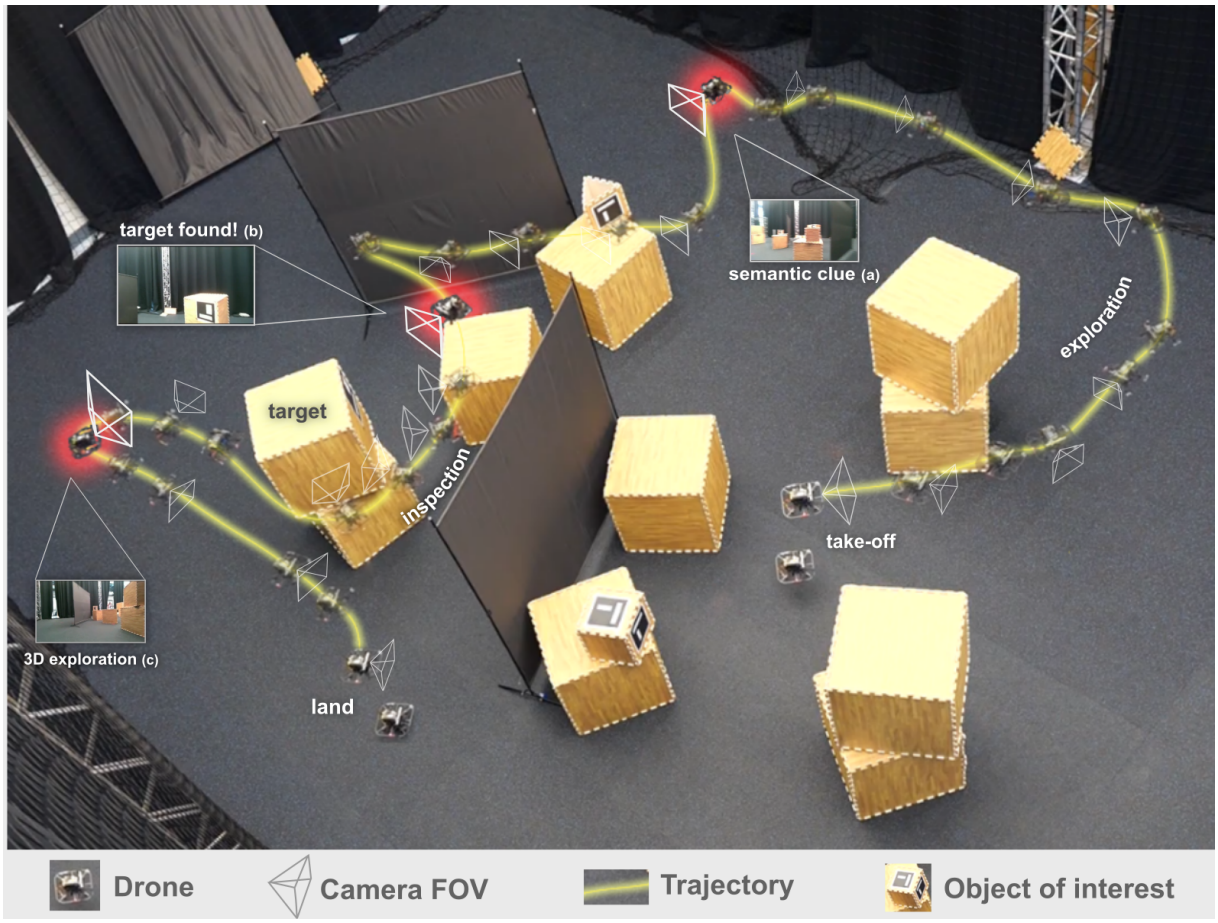


Figure 18: Real-world experiment of a MAV performing semantic target search and exploration inside a cluttered lab environment. Red color (a,b,c) shows important parts of the flight along with the front camera view. See also Fig. C.10.

To analyze the method in depth, we introduced two new auxiliary metrics called Time-in-View(TIV) and Cumulative Information Gain (CIG) that gave insights into how target search progresses. Two ablation studies for the viewpoint sampling and planning strategy were also performed. The findings from both studies underscored the importance of the diffusion process in directing the MAV towards high-priority frontiers in 3D space. Further, it was concluded that viewpoints sampled around objects were crucial for bringing novel objects (including the target) into view, thus facilitating target search and inspection.

Our algorithm was validated on a real drone in a lab environment with various random cluttered configurations. We showed that the method can quickly find the target and explore the space, even under practical constraints like sensor noise, and semantic uncertainty.

Future Work: Although we achieve competent performance and provide in-depth analysis, the field of target search for emergency response is still nascent and there is scope for improvement:

- Firstly, the **source of semantic priorities** is crucial. In our study, we explored a potential solution that infers cosine similarities using an LLM. However, it was observed that these models sometimes gave illogical cosine similarities because they lack deeper contextual understanding. Since these models are trained on open vocabulary data, future work could involve fine-tuning the model on a custom dataset specific to emergency response environments or integrating human feedback to refine priority assignments.

- Evaluating semantic exploration methods on **realistic simulation environments** free from biases is important. In fields such as disaster response, this task is particularly challenging due to the stochastic nature of disasters. Competitions like the DARPA SubT Challenge[3] and RoboCup Rescue[4] address some of these challenges. In this work, we attempt to create an unbiased simulation environment with semantic object placement using prompts from LLMs like ChatGPT[5], objects from the DARPA challenge, and common sense. However, such approaches have inherent limitations, the most notable one being the lack of ground truth data. Perhaps, the field of target search in unstructured environments could draw inspiration from the field of indoor ObjectNav

[3] https://www.darpa.mil/program/darpa-subterranean-challenge
[4] https://www.robocup.org/domains/2
[5] https://chatgpt.com/

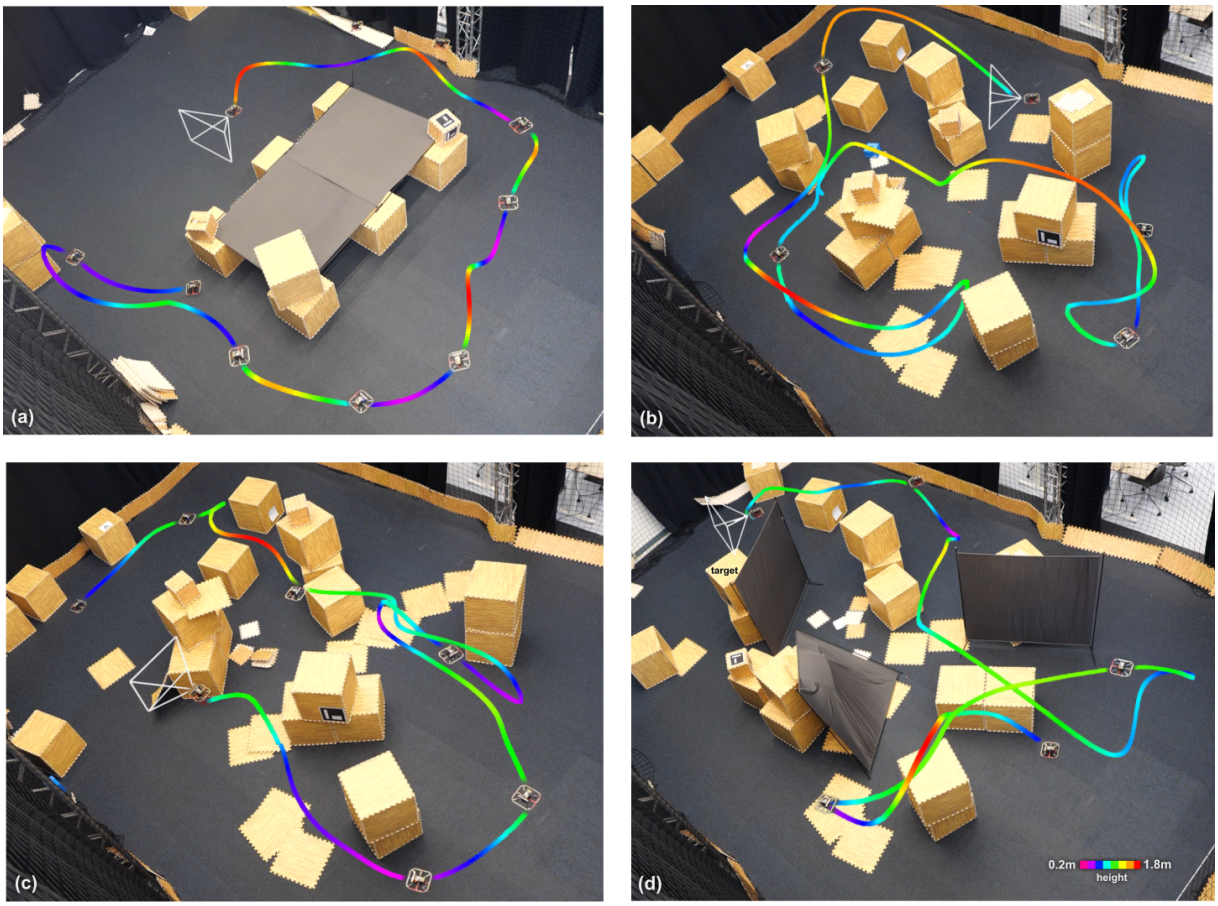


Figure 19: Real-world experiments in various cluttered configurations. Trajectory is colored based on drone height

that uses datasets like Habitat-MatterPort3D [7] or simulation environments like Gibson-env [51].

- We strongly believe that **appropriate environment representations** which are general enough to integrate high-level semantic reasoning while being conducive to informative planning are the key to solving problems in this domain. A potential solution was explored by compressing labels to priorities and biasing exploration frontiers. In practice, however, the diffusion process requires maintaining an additional voxel grid map. Since semantics occupy very little space in the map, it might be useful to explore sparse environment representations like Gaussian Mixture Model maps [52] to represent semantic priorities. Not only does this approach save memory, it also gives a probabilistic generative model that can be inferred at any time instant to reconstruct objects of interest.

References

[1] G. P. Kenny, J. Stapleton, A. Lynn, K. Binder, C. Allen, and S. G. Hardcastle, "Heat stress in canadian deep mechanized mines: Laboratory simulation of typical mining tasks performed in varying environments," in *Proceedings of the 13th International Conference on Environmental Ergonomics, Boston, USA*, 2009, pp. 441–445.

[2] A. Davids, "Urban search and rescue robots: From tragedy to technology," *IEEE Intelligent systems*, vol. 17, no. 2, pp. 81–83, 2002.

[3] S. Brand and R. Price, "The economic and social costs of crime," 2000.

[4] G. Best, "Planning Algorithms for Multi-Robot Active Perception," en, Jan. 2019.

[5] B. Zhou, Y. Zhang, X. Chen, and S. Shen, "Fuel: Fast uav exploration using incremental frontier structure and hierarchical planning," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 779–786, 2021.

[6] J. Huang, B. Zhou, Z. Fan, *et al.*, "FAEL: Fast Autonomous Exploration for Large-scale Environments With a Mobile Robot," en, *IEEE Robotics and Automation Letters*, vol. 8, no. 3, pp. 1667–1674, Mar. 2023.

[7] S. K. Ramakrishnan, A. Gokaslan, E. Wijmans, *et al.*, "Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai," *arXiv preprint arXiv:2109.08238*, 2021.

[8] A. C. Hernandez, E. Derner, C. Gomez, R. Barber, and R. Babuška, "Efficient Object Search Through Probability-Based Viewpoint Selection," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, ISSN: 2153-0866, Oct. 2020, pp. 6172–6179.

[9] N. H. Yokoyama, S. Ha, D. Batra, J. Wang, and B. Bucher, "VLFM: Vision-Language Frontier Maps for Zero-Shot Semantic Navigation," en, Oct. 2023.

[10] M. F. Ginting, S.-K. Kim, D. D. Fan, M. Palieri, M. J. Kochenderfer, and A.-a. Agha-Mohammadi, *SEEK: Semantic Reasoning for Object Goal Navigation in Real World Inspection Tasks*, en, arXiv:2405.09822 [cs], May 2024.

[11] Y. Luo, Z. Zhuang, N. Pan, *et al.*, "Star-searcher: A complete and efficient aerial system for autonomous target search in complex unknown environments," *IEEE Robotics and Automation Letters*, 2024.

[12] M. Lodel, N. Wilde, R. Babuska, and A.-M. Javier, "Target search planning with learned semantic priorities," Delft University of Technology, 2024.

[13] D. Calisi, A. Farinelli, L. Iocchi, and D. Nardi, "Autonomous exploration for search and rescue robots," *WIT Transactions on the Built Environment*, vol. 94, 2007.

[14] J. Kleinschmidt and L. Trenta, "Scanning the horizon: Drones and counter-narcotics in latin america," 2022.

[15] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, "Receding Horizon "Next-Best-View" Planner for 3D Exploration," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 1462–1468.

[16] T. Cieslewski, E. Kaufmann, and D. Scaramuzza, "Rapid exploration with multi-rotors: A frontier selection method for high speed flight," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, ISSN: 2153-0866, Sep. 2017, pp. 2135–2142.

[17] 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*, vol. 5, pp. 1–1, Jan. 2020.

[18] C. Cao, H. Zhu, H. Choset, and J. Zhang, "TARE: A Hierarchical Framework for Efficiently Exploring Complex 3D Environments," en, in *Robotics: Science and Systems XVII*, Robotics: Science and Systems Foundation, Jul. 2021.

[19] Z. Meng, H. Qin, Z. Chen, *et al.*, "A Two-Stage Optimized Next-View Planning Framework for 3-D Unknown Environment Exploration, and Structural Reconstruction," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1680–1687, Jul. 2017, Conference Name: IEEE Robotics and Automation Letters.

[20] T. R. Hayes and J. M. Henderson, "Looking for Semantic Similarity: What a Vector-Space Model of Semantics Can Tell Us About Attention in Real-World Scenes," en, *Psychological Science*, vol. 32, no. 8, pp. 1262–1270, Aug. 2021, Publisher: SAGE Publications Inc.

[21] J. Chen, G. Li, S. Kumar, B. Ghanem, and F. Yu, *How To Not Train Your Dragon: Training-free Embodied Object Goal Navigation with Semantic Frontiers*, en, arXiv:2305.16925 [cs], May 2023.

[22] S. Arora, S. Choudhury, and S. Scherer, "Hindsight is only 50/50: Unsuitability of mdp based approximate pomdp solvers for multi-resolution information gathering," *arXiv preprint arXiv:1804.02573*, 2018.

[23] J. Guivant, E. Nebot, J. Nieto, and F. Masson, "Navigation and mapping in large unstructured environments," *The International Journal of Robotics Research*, vol. 23, no. 4-5, pp. 449–472, 2004.

[24] M. Tranzatto, F. Mascarich, L. Bernreiter, *et al.*, *CERBERUS: Autonomous Legged and Aerial Robotic Exploration in the Tunnel and Urban Circuits of the DARPA Subterranean Challenge*, arXiv:2201.07067 [cs], Jan. 2022.

[25] S. Papatheodorou, N. Funk, D. Tzoumanikas, C. Choi, B. Xu, and S. Leutenegger, "Finding Things in the Unknown: Semantic Object-Centric Exploration with an MAV," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, May 2023, pp. 3339–3345.

[26] A. Asgharivaskasi and N. Atanasov, *Semantic OcTree Mapping and Shannon Mutual Information Computation for Robot Exploration*, arXiv:2112.04063 [cs], Feb. 2023.

[27] A. Rasouli, P. Lanillos, G. Cheng, and J. K. Tsotsos, "Attention-based active visual search for mobile robots," en, *Autonomous Robots*, vol. 44, no. 2, pp. 131–146, Jan. 2020.

[28] L. Wijayathunga, A. Rassau, and D. Chai, "Challenges and solutions for autonomous ground robot scene understanding and navigation in unstructured outdoor environments: A review," *Applied Sciences*, vol. 13, no. 17, p. 9877, 2023.

[29] T. Dang, C. Papachristos, and K. Alexis, "Visual Saliency-Aware Receding Horizon Autonomous Exploration with Application to Aerial Robotics," en, in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, QLD: IEEE, May 2018, pp. 2526–2533.

[30] L. Han, F. Gao, B. Zhou, and S. Shen, "Fiesta: Fast incremental euclidean distance fields for online motion planning of aerial robots," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2019, pp. 4423–4430.

[31] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97.'Towards New Computational Principles for Robotics and Automation'*, IEEE, 1997, pp. 146–151.

[32] A. Dai, S. Papatheodorou, N. Funk, D. Tzoumanikas, and S. Leutenegger, "Fast Frontier-based Information-driven Autonomous Exploration with an MAV," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, ISSN: 2577-087X, May 2020, pp. 9570–9576.

[33] D. Walther, U. Rutishauser, C. Koch, and P. Perona, "Selective visual attention enables learning and recognition of multiple objects in cluttered scenes," *Computer Vision and Image Understanding*, vol. 100, no. 1-2, pp. 41–63, 2005.

[34] R. Desimone, J. Duncan, *et al.*, "Neural mechanisms of selective visual attention," *Annual review of neuroscience*, vol. 18, no. 1, pp. 193–222, 1995.

[35] S. Kastner and L. G. Ungerleider, "The neural basis of biased competition in human visual cortex," eng, *Neuropsychologia*, vol. 39, no. 12, pp. 1263–1276, 2001.

[36] A. Radford, J. W. Kim, C. Hallacy, *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*, PMLR, 2021, pp. 8748–8763.

[37] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[38] M. Dharmadhikari and K. Alexis, "Semantics-aware Exploration and Inspection Path Planning," en, in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, London, United Kingdom: IEEE, May 2023, pp. 3360–3367.

[39] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *2011 IEEE international conference on robotics and automation*, IEEE, 2011, pp. 2520–2525.

[40] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[41] C. Zhang, D. Han, Y. Qiao, *et al.*, "Faster segment anything: Towards lightweight sam for mobile applications," *arXiv preprint arXiv:2306.14289*, 2023.

[42] M. Bernhard, R. Amoroso, Y. Kindermann, *et al.*, "What's outside the intersection? fine-grained error analysis for semantic segmentation beyond iou," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 968–977.

[43] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro, "Image inpainting for irregular holes using partial convolutions," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 85–100.

[44] N. Hughes, Y. Chang, and L. Carlone, "Hydra: A real-time spatial perception system for 3d scene graph construction and optimization," *arXiv preprint arXiv:2201.13360*, 2022.

[45] B. Zhou, H. Xu, and S. Shen, "Racer: Rapid collaborative exploration with a decentralized multi-uav system," *IEEE Transactions on Robotics*, vol. 39, no. 3, pp. 1816–1835, 2023.

[46] J. Panerati, H. Zheng, S. Zhou, J. Xu, A. Prorok, and A. P. Schoellig, "Learning to fly—a gym environment with pybullet physics for reinforcement learning of multi-agent quadcopter control," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2021, pp. 7512–7519.

[47] S. Wu, "Risk-aware decentralized multi-mav planning in unknown and dynamic environments," *MSc. Thesis, Delft University of Technology*, 2023.

[48] D. Batra, A. Gokaslan, A. Kembhavi, *et al.*, *ObjectNav Revisited: On Evaluation of Embodied Agents Navigating to Objects*, en, arXiv:2006.13171 [cs], Aug. 2020.

[49] N. Yokoyama, S. Ha, and D. Batra, "Success weighted by completion time: A dynamics-aware evaluation criteria for embodied navigation," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2021, pp. 1562–1569.

[50] D. S. Chaplot, H. Jiang, S. Gupta, and A. Gupta, *Semantic Curiosity for Active Visual Learning*, arXiv:2006.09367 [cs], Jun. 2020.

[51] F. Xia, A. R. Zamir, Z. He, A. Sax, J. Malik, and S. Savarese, "Gibson env: Real-world perception for embodied agents," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2018.

[52] P. Z. X. Li, S. Karaman, and V. Sze, "Gmmap: Memory-efficient continuous occupancy map using gaussian mixture model," *IEEE Transactions on Robotics*, 2024.

[53] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, pp. 303–338, 2010.

[54] S. Guiaşu, "Weighted entropy," *Reports on Mathematical Physics*, vol. 2, no. 3, pp. 165–179, 1971.

Appendices

A Methodology

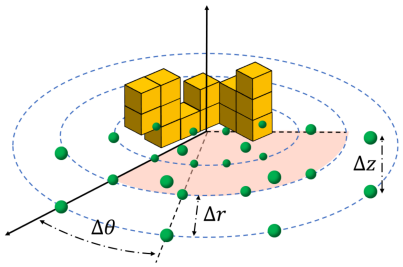


Figure A.1: Frontier viewpoint sampling. Figure taken from [5]

B Experimental Setup

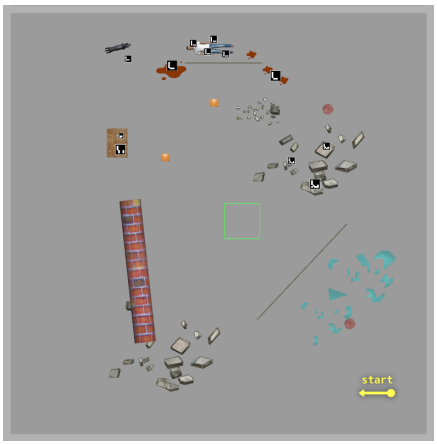


Figure B.2: Earthquake top view

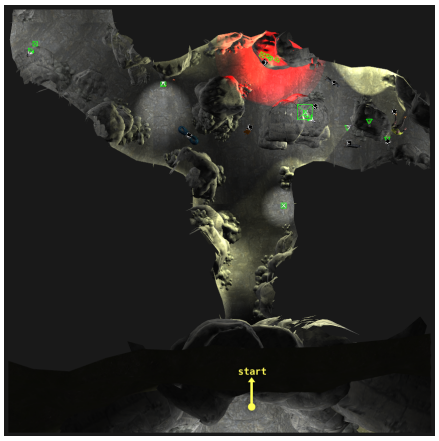


Figure B.3: Cave Top View

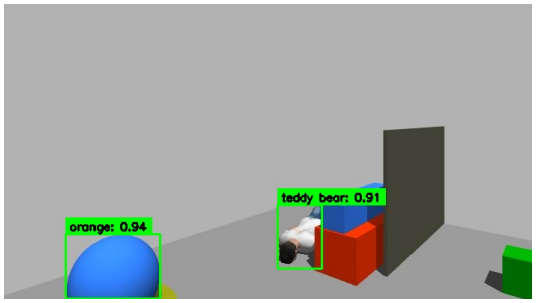


Figure B.4: Poor object detection scores in Gazebo using DETR

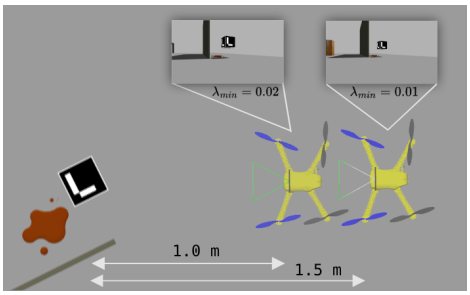


Figure B.5: Marker view at different detection thresholds

C Results

C.1 Time In View

Prior works that focus on object-centric exploration quantify the quality of object observations using metrics like observed object surface [38], or the error from ground truth object meshes [25]. Since our work focuses on object search, we quantify this using the time in seconds throughout the episode when an object was successfully detected. This metric is called Time In View (TIV) because an object o can only be successfully detected when it is fully in view (i.e. $\lambda_o > \lambda_{min}$).

It is also known that the object detection score varies significantly when the viewpoint changes [50][53]. In [50], the authors reward robot trajectories with inconsistent object detection scores, allowing them to explore an object from multiple 'bad' viewpoints. They also found that the temporal entropy of object detection score decreases when objects are attended more. Therefore, a higher time in view for semantic objects indicates better performance.

Object	Source	Object	Source
human	Darpa Challenge	toy	Common sense (Negative data)
blood	Common sense	ball	Common sense (Negative data)
rescue dog	ChatGPT prompt	plant	Negative data
flashlight	ChatGPT prompt	carpet	Negative data
radio	ChatGPT prompt	rope	Darpa Challenge
chair	ChatGPT prompt	drill	Darpa Challenge
table	ChatGPT prompt	rescue worker	ChatGPT prompt
helmet	Darpa Challenge	knife	Negative data
rubble	ChatGPT prompt	headphones	common sense (Negative data)
sofa	Common sense (Negative data)	gun	Common sense (Negative data)
wall	ChatGPT prompt	ground	ChatGPT prompt

Table B.1: Object set for both simulation environments. Negative data is added to the data to make the simulation free of biases.

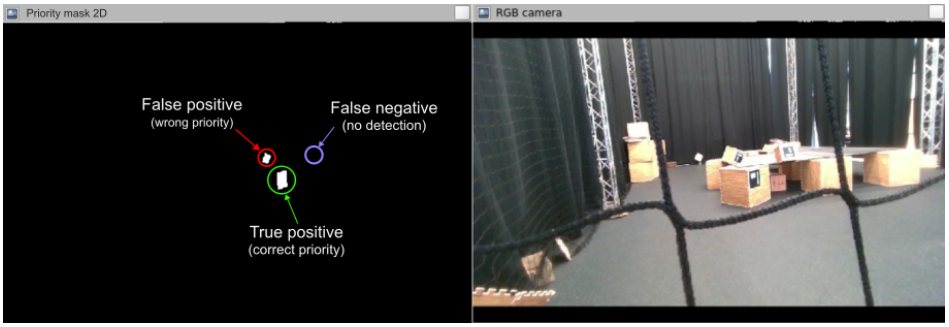


Figure B.6: False detections and oscillating priorities

C.2 Cumulative Information Gain

The weighted entropy is defined as the degree of uncertainty in a system defined by probabilistic events, where events are weighted based on importance [54]. For a discrete occupancy map $\mathcal{M}$, the event uncertainty is quantified using the probability of occupancy P_k at each voxel k. The weights w_k are calculated by creating a ground truth 3D attention map using the known locations of semantic objects in the environment. The standard entropy formula is then weighted at each voxel using these weights as follows:

$$H_t^w = -\sum_{k=0}^{N} w_k P_k \log_2(P_k) \; \forall \; k \in \mathcal{M} \tag{C.1}$$

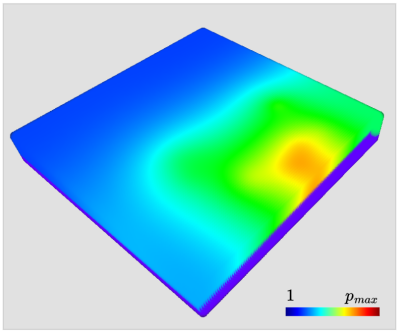


Figure C.7: Ground truth importance map (earthquake). Red areas indicate higher importance

To create the ground truth importance map, a priority voxel map is first generated for all ground truth objects O_g in V. The priority for each object is inferred using the priority function r from section 4.2. Then, the diffusion process from section 4.3.3 is applied over the full volume to create the importance map (Fig. C.7). The value at each voxel in this map acts as the weight w_k and quantifies the importance of regions in the volume V.

The Cumulative Information Gain I_t^w is then calculated as the change in weighted entropy from the start of the episode, i.e., H_0^w. The value is also normalized to yield a fraction (see Eq. C.2). Intuitively, I_t^w expresses the fraction of the ground truth importance map covered at time t.

$$I_t^w = (H_t^w - H_0^w)/H_0^w \tag{C.2}$$

Since $\mathcal{A}$ is non-zero everywhere, this metric can also evaluate coverage-based exploration planners on a target search task, unlike binary success metrics like SPL. A sharper slope on the CIG vs. Time curve indicates higher information gain and means that the robot comes across many *relevant* semantics before arriving at the target.

		Target search		Exploration
Env	Method	Success %	Time, t^*(s)	Time, t_f(s)
Earthquake $n = 10$	FUEL	0%	-	**106.0 ± 9.8**
	FUEL-mod	50%	57.2 ± 9.9	121.7 ± 6.9
	VSep	10%	205.9 ± 0.0	212.7 ± 25.4
	STEM (Ours)	**100%**	**49.6 ± 8.1**	143.4 ± 8.7
Cave $n = 5$	FUEL	0%	-	**108.8 ± 4.8**
	FUEL-mod	0%	-	169.6 ± 24.6
	Vsep	0%	-	272.3 ± 84.1
	STEM (Ours)	**40%**	**85.4 ± 9.51**	153.9 ± 11.5

Table C.2: Comparison study with baselines in the Earthquake and Cave environments with a $\lambda_{min} = 0.02$. Success rates for most methods drop significantly, but our method outperforms other methods on both Success % and Time to Target even under strict parameters. This is because object viewpoints inspect the target from a closer distance and keep object detection consistent.

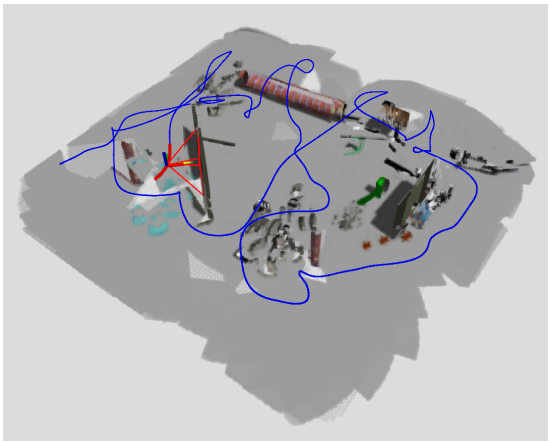


Figure C.8: Greedy Search in earthquake environment. The greedy planner has a lot of path intersections, and sharp points in the spline because of oscillating information gain. This increases completion times for both target search and exploration.

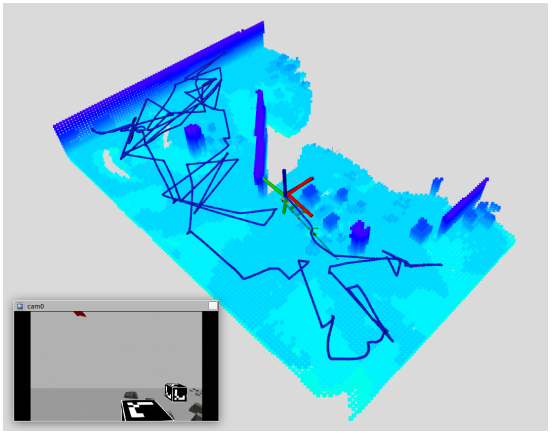


Figure C.9: VSep showing poor performance because the RRT planner finds it hard to escape cluttered spaces when the sensor range is limited ($R_{max} = 3m$). The drone starts at the upper left corner and takes a long time to move. Even when it finds a path, it is suboptimal and demonstrates stop-and-go behavior.

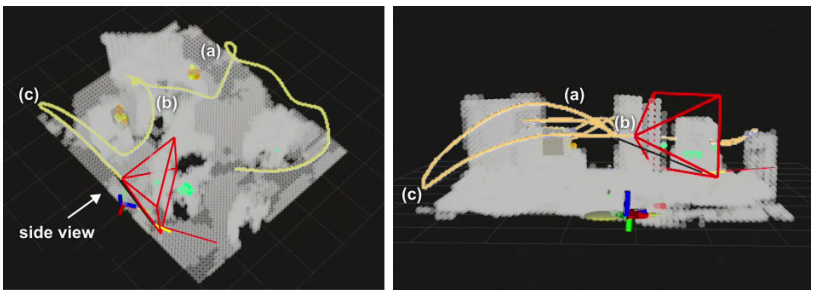


Figure C.10: Isometric view (left) and side view (right) of the reconstructed map in RViz for real-world experiments. The letters correspond to stages from Fig. 18. The side view shows the (c) stage, where the drone explores areas of interest near the target in 3D.