

# Semantics-aware Exploration and Inspection Path Planning

Mihir Dharmadhikari and Kostas Alexis

**Abstract**—This paper contributes a novel strategy for semantics-aware autonomous exploration and inspection path planning. Attuned to the fact that environments that need to be explored often involve a sparse set of semantic entities of particular interest, the proposed method offers volumetric exploration combined with two new planning behaviors that together ensure that a complete mesh model is reconstructed for each semantic, while its surfaces are observed at appropriate resolution and through suitable viewing angles. Evaluated in extensive simulation studies and experimental results using a flying robot, the planner delivers efficient combined exploration and high-fidelity inspection planning that is focused on the semantics of interest. Comparisons against relevant methods of the state-of-the-art are further presented.

## I. INTRODUCTION

Robotic systems have long been utilized for remote sensing and inspection tasks [1–6]. Flying or ground robots, for example, are actively utilized to explore and inspect industrial facilities [1–5] or even demanding subterranean settings [7–14]. Building upon this success, the research community and the industry are currently actively looking towards means to completely automate the process of building and maintaining accurate “digital twins” [15] of the facilities of interest. In this framework, explicit focus on specific semantics of interest is of paramount importance, a task assisted by the progress in semantic segmentation and mapping [16–21]. Within a large-scale industrial facility, or other environment of interest, the vast majority of the surfaces may not represent informative or generally significant regions. It is in fact mostly specific structures that require precise and comprehensive monitoring. Accordingly, traditional methods on exploration or coverage path planning are not tailored to efficiently undertake the task as they are agnostic to semantics, rendering them inefficient and unable to deliver the necessary inspection behavior at scale.

Motivated by the above, in this work we present a novel Semantics-aWare exploration and inspection Planner (SWAP) that is tailored to environments with sparse semantics, assumes no prior map information, and can ensure as complete mesh reconstruction and surface inspection as possible at a set minimum resolution. The method is not exploiting merely a volumetric representation [22, 23], as commonly done in exploration planners [10, 24], and does not make the often unrealistic assumption of having access to an a priori available high-resolution overall mesh representation

This material was supported by the Research Council of Norway under project SENTIENT (NO-321435).

The authors are with the Autonomous Robots Lab, Norwegian University of Science and Technology (NTNU), O. S. Bragstads Plass 2D, 7034, Trondheim, Norway mihir.dharmadhikari@ntnu.no

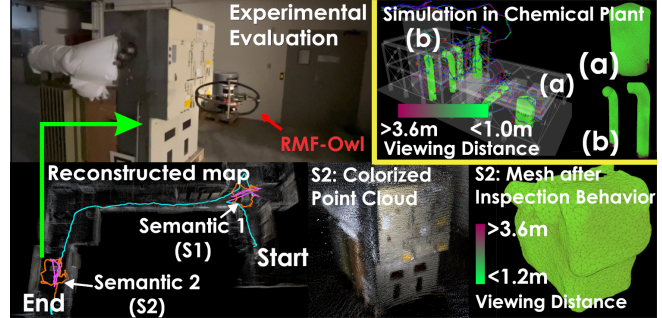


Fig. 1. Instance of an autonomous semantics-aware exploration and inspection mission in an environment with two sparsely distributed semantics (S1, S2). The path traversed by the robot and the reconstructed meshes colored by camera viewing distance are shown. Furthermore, a simulation run inside a chemical plant is depicted.

of the environment. Instead, SWAP combines a volumetric map with a disconnected set of meshes focused only on the semantics of interest which are also utilized to perform calculations as to coverage completeness or inspection fidelity in order to best plan its actions. Importantly, the method ensures autonomous exploration within unknown maps combined with complete and selectively detailed semantic inspection by means of transitioning among a set of behaviors as opposed to merely combining distinct objectives in an additive manner that would be highly susceptible to tuning. In particular, SWAP offers efficient volumetric exploration using a range sensor (e.g., 3D LiDAR), while at the same time for the detected semantics of interest it enables capturing the further point clouds necessary to derive a complete mesh model for each, and ensures that their surfaces are viewed by a –possibly separate– camera at sufficient resolution and through suitable viewing angles.

To verify the proposed solution for semantics-aware exploration and inspection path planning, a set of simulation and experimental studies are presented using flying robots, including comparison against a state-of-the-art exploration planner that does not explicitly consider semantics. We demonstrate that SWAP presents significant advantages for combined exploration and inspection tasks involving a sparse distribution of semantics, the mesh models of which have to be reconstructed and their surfaces be captured by a camera with image quality guarantees. Accordingly, we believe that the method is tailored to be used in complex facilities for which digital twins have to be built and then maintained.

In the remaining paper, Section II presents related work, followed by the problem statement in III. The proposed approach is detailed in Section IV, with evaluation studies in Section V and conclusions in Section VI.

## II. RELATED WORK

This work on semantics-aware exploration and inspection planning has relevance to a host of contributions that relate to its functionality. With respect to its elementary behavior on exploration planning it builds upon the relevant literature [9, 10, 24–28] and especially our prior work on graph-based exploration [9, 10]. Similarly, regarding its further core behavior of inspection planning over a reconstructed mesh it relates to works on coverage path planning [29–34].

With respect to the key innovation in semantics-aware planning, this work relates to two sets of contributions. First, it partially relates to works that on one hand only focus on planning for safe navigation among predefined waypoints but also simultaneously consider the semantics in the environment [35–37]. Second, and most importantly, the proposed method relates to the narrow niche of contributions that aim to co-optimize exploration and mapping of objects/semantics of interest [38–41]. Our earlier work in [38] performs autonomous exploration but also allows for small deviations from the exploration path in order to improve the resolution of observation of detected objects based on a volumetric map. The contribution in [39] considers an a priori known map and formulates the planning problem as an orienteering one further considering the desired reconstructed model resolution. The method in [40] uses a 3-layer semantic map involving a costmap, a classical exploration grid and a binary grid that monitors the observation over specific semantic classes and derives semantics-aware adaptations of A\*, transition-based RRT, and a shortcut algorithm which are then tested on a combined exploration and observation task. The authors in [41] use object detection and explicitly combine semantic information with exploration planning techniques to improve the quality of the 3D reconstruction.

Compared to the existing literature, SWAP contributes multiple innovations and key features. First, it makes no assumption about the shape, size and location of the semantics or any prior map knowledge. Second, it combines exploration and semantic inspection in a principled manner through explicit behaviors as opposed to techniques that merely aggregate goals in an additive objective. Accordingly, it ensures complete semantic mapping using a range sensor (no gaps in the reconstructed mesh) and inspection with a camera at any desired resolution up to what is possible through collision-free configurations. Finally, it realizes a task-specific multi-facet environment representation involving a sparse volumetric map, a semantics-specific locally dense mesh representation and a sparse global graph for replanning in the mapped space.

## III. PROBLEM FORMULATION

The overall problem considered in this work is that of combined exploration of a bounded volume  $V \subset \mathbb{R}^3$  with a depth sensor  $\mathcal{Y}_D$  and detailed inspection of a set  $\Lambda$  of structures of interest within it, called semantics, in terms of reconstructing meshes of their surfaces using segmented data from  $\mathcal{Y}_D$ , and scanning them using a visual camera sensor  $\mathcal{Y}_C$ . The problem can be cast globally as that of a)

determining which parts of the initially unexplored volume  $V_{une} \stackrel{init.}{=} V$  are free  $V_{free} \subseteq V$  or occupied  $V_{occ} \subseteq V$  b) creating complete mesh reconstructions  $\mathcal{M}^j$  of the surfaces  $S_S^j$  of all the semantics  $\mathbb{S}^j$ , and c) inspecting every face  $f$  of every  $\mathcal{M}^j$  with  $\mathcal{Y}_C$  at resolution  $r$ , defined as the number of pixels per unit area, higher than or equal to  $r_{min}$  and viewing angle  $\theta_I$  with respect to the outward normal  $\mathbf{n}_f$  to  $f$  less than  $\theta_{I,max}$ . The environment is represented as a volume discretized in an occupancy map  $\mathbb{M}$  consisting of cubical voxels  $m \in \mathbb{M}$  with edge length  $\lambda_V$ , and a set of surface meshes  $\{\mathcal{M}^j\}$ ,  $\forall \mathbb{S}^j \in \Lambda$ . The operation is subject to the robot dynamics and the sensors' visibility constraints based on two possibly distinct frustum models for  $\mathcal{Y}_D, \mathcal{Y}_C$  defined by the Field of View (FoV)  $[F_h^D, F_v^D], [F_h^C, F_v^C]$  respectively and maximum ranges  $d_{max}^D, d_{max}^C$ . Hence, certain volume and surface cannot be mapped or inspected resulting in residual volume and surface. Given the above, we have the following definition and problem formulation cast globally.

*Definition 1 (Residual Volume and Semantic Surface)* Let  $\Xi$  be the simply connected set of collision free configurations and  $\check{V}_m \subseteq \Xi$  the set of all configurations from which the voxel  $m$  can be perceived by  $\mathcal{Y}_D$ . Then the residual volume is  $V_{res} = \bigcup_{m \in \mathbb{M}} (m | \check{V}_m = \emptyset)$ . Similarly the residual semantic surface  $S_{res}$  can be defined as the part of  $\{S_{\mathbb{S}^j}\}$ ,  $\forall \mathbb{S}^j \in \Lambda$  that cannot be mapped by  $\mathcal{Y}_D$  or seen by  $\mathcal{Y}_C$ .

*Problem 1 (Volumetric Exploration and Semantic Inspection Problem)* Given a volume  $V$  and an initial configuration  $\xi_{init} = [x, y, z, \psi] \in \Xi$  find a collision-free path  $\sigma$  that when traversed by the robot leads to a) identifying  $V_{free}$  and  $V_{occ}$  based on  $\mathcal{Y}_D$  b) complete mesh reconstructions  $\{\mathcal{M}^j\}$  of  $\{S_{\mathbb{S}^j}\}$ ,  $\forall \mathbb{S}^j \in \Lambda$ , and c) inspection of every face  $f$  of  $\mathcal{M}^j$  with  $\mathcal{Y}_C$  at the required resolution and viewing angle.

## IV. PROPOSED APPROACH

The proposed semantics-aware exploration and inspection planner (SWAP) utilizes three planning behaviors, namely Volumetric Exploration, Semantics Hole Coverage, and Semantics Inspection (Figure 2), for efficient exploration of the unknown environment combined with targeted inspection of the surfaces of the semantics of interest. The planner starts in the exploration mode to efficiently explore and volumetrically map the unknown environment. During exploration, possible semantic detections –defining which points of the  $\mathcal{Y}_D$  measurements belong to a certain semantic– are used to generate meshes as described in Section IV-A. The planner performs exploration for an allotted time  $T_e$ . After that, it switches to the hole coverage mode and identifies the holes possibly left in the semantic meshes. The semantic closest to the robot is selected first and viewpoints considering the onboard depth sensor  $\mathcal{Y}_D$  are planned to iteratively fill these holes in meshes of all of the detected semantics as described in Section IV-C. Once all holes in the meshes of all the semantics detected so far are filled, no more collision-free configurations can be found to fill the remaining holes, or an allotted time for that semantic has expired, the planner switches to the inspection mode

in order to provide detailed inspection of the semantics of interest considering the onboard camera sensor  $\mathcal{V}_C$  driven by certain metrics. Once this process is also complete within the currently explored map, the method switches back to exploration mode to explore further and where and when needed switch to the semantics-driven behaviors.

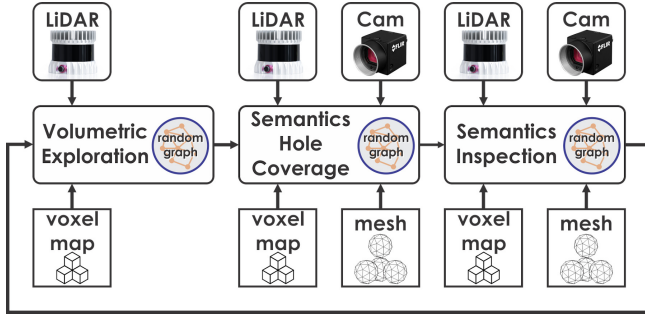


Fig. 2. The behaviors of the semantics-aware exploration and inspection planner. SWAP transitions between autonomous volumetric exploration, semantics mesh hole coverage and semantics inspection considering onboard depth and camera sensors and exploiting a rich environment representation combining volumetric maps and a set of semantic-specific meshes. For all planning stages, a random graph data structure is used for path planning.

### A. Environment Representation

For efficient exploration and semantic inspection, SWAP utilizes a dual representation of the environment consisting of a volumetric map  $\mathbb{M}$  and a set  $\mathbb{D}$  of data structures  $D^j$  storing a reconstructed point cloud  $P_R^j$ , a surface mesh  $\mathcal{M}^j$ , and corresponding raw sensor data for each semantic  $\mathcal{S}^j \in \Lambda$ . In this work we use Voxblox [22] as the volumetric mapping framework for collision-free navigation and volumetric calculations. Along with the volumetric map, we generate surface meshes for each detected semantic in the environment using semantically segmented  $\mathcal{V}_D$  data. In this work, we assume that a semantic segmentation module providing segmented point cloud exists. However, the method does not make any other assumptions about the size, shape, or location of semantics in the environment. From every segmented point cloud  $P_{seg}$ , the points  $P_{seg}^j$  belonging to semantic  $\mathcal{S}^j$  are appended to  $P_R^j$ . At a fixed temporal frequency,  $P_R^j$  is subsampled, and the surface mesh  $\mathcal{M}^j$  is constructed using the Advancing Front Surface Reconstruction algorithm [42]. The faces of  $\mathcal{M}^j$  that are observed by  $\mathcal{V}_C$  at a desired resolution  $r_{min}$  and viewing angle  $\theta_{I,max}$  are further marked.

### B. Volumetric Exploration

SWAP implements its autonomous exploration functionality by interfacing our previous and open-sourced work on graph-based exploration (GBPlanner) [7, 9, 10]. The method, verified extensively in subterranean and industrial environments, offers efficient exploration within a volume of set bounds assuming no prior map knowledge. It operates over a volumetric representation of the environment based on [22] and functions in a bifurcated architecture of local- and global path planning. At the local stage the method exploits a dense

random graph  $\mathbb{G}_E^L$  around the robot to identify collision-free paths maximizing volumetric exploration. Simultaneously, as such local steps take place, the algorithm builds a sparse global graph  $\mathbb{G}_E^G$ , used by the global stage that is invoked when local exploration reports inability to find a path of significant gain or when the robot approaches its endurance limits. Accordingly, the method offers re-positioning to previously detected unexploited frontiers of the exploration space or timely auto-homing. In SWAP, the autonomous exploration behavior is invoked for  $T_e$  seconds, before the system possibly switches to its semantically-driven behaviors.

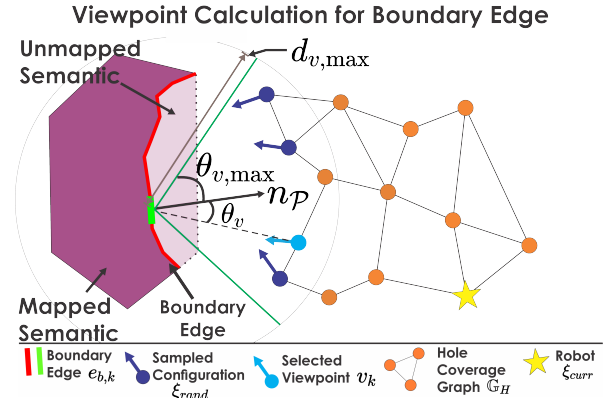


Fig. 3. Viewpoint generation procedure for a boundary edge of a hole during Semantics Hole Coverage mode. Robot configurations are sampled in a spherical coordinate system centered at the edge. The configurations that respect the viewing angle criteria (as described in Section IV-C) and can be connected to the hole coverage graph are extracted, and the configuration with the least viewing angle is selected as the viewpoint.

### C. Semantics Hole Coverage

After performing exploration for  $T_e$  seconds, the planner identifies the set of semantics,  $\Lambda_H$ , whose reconstructed meshes contain holes, and attempts to plan viewpoints to collect measurements to fill them considering one semantic at a time. The iterative procedure for hole coverage of one semantic is described in Algorithm 1.

The planner first selects the semantic  $\mathcal{S}_{C,H} \in \Lambda_H$  closest to the robot. For the mesh  $\mathcal{M}_{\mathcal{S}_{C,H}}$  of semantic  $\mathcal{S}_{C,H}$  the planner first identifies the set  $\mathbb{E}_b$  of mesh edges that belong to only one mesh face, called boundary edges  $e_{b,k} \in \mathbb{E}_b$ , and their corresponding faces called boundary faces  $f_{b,k}$ . We treat the boundary edges as frontiers on the mesh that need to be covered iteratively. For each boundary edge  $e_{b,k}$  a viewpoint  $v_k$  is derived that can view the edge and map the region around it as shown in Figure 3. Furthermore, the planner maintains a collision-free 3D graph, called the hole coverage graph  $\mathbb{G}_H$ , connecting the current robot configuration  $\xi_{curr}$  to all viewpoints  $v_k$ .

To generate the viewpoint for an edge, the planner randomly samples collision-free robot configurations  $\xi_{rand}$ , with  $\mathcal{V}_D$  pointing towards  $e_{b,k}$ , in a spherical coordinate system centered at the midpoint of the edge  $e_{b,k}$  with maximum radial distance  $d_{v,max}$ . The graph  $\mathbb{G}_H$  is extended towards  $\xi_{rand}$  and the configurations that cannot be connected to  $\mathbb{G}_H$

**Algorithm 1** Semantics Hole Coverage

---

```

1:  $\Lambda_H \leftarrow \text{extractSemanticsWithHoles}(\Lambda)$ 
2: while  $\Lambda_H \neq \emptyset$  do
3:    $\mathbb{S}_{C,H} \leftarrow \text{closestSemantic}(\Lambda_H)$ 
4:    $\mathbb{E}_b \leftarrow \text{extractBoundaryEdges}(\mathbb{S}_{C,H})$ 
5:   while  $\mathbb{E}_b \neq \emptyset$  and timeRemaining = True do
6:      $W_c \leftarrow \text{calculateViewpoints}(\mathbb{E}_b, \mathbb{M}, \mathbb{G}_H)$ 
7:     if  $W_c = \emptyset$  then
8:       exit loop
9:      $W_s \leftarrow \text{reduceNumberOfViewpoints}(W_c)$ 
10:     $\sigma_{hc} \leftarrow \text{pathToClosestViewpoint}(W_s, \mathbb{G}_H)$ 
11:    executePath( $\sigma_{hc}$ )
12:    updateMesh( $\mathbb{S}_{C,H}$ )
13:   $\Lambda_H \leftarrow \Lambda_H \setminus \mathbb{S}_{C,H}$ 
14:   $\Lambda_H \leftarrow \Lambda_H \cup \text{newSemanticsWithHoles}$ 

```

---

are discarded. Out of the remaining configurations, the one whose viewing angle  $\theta_v$ , defined as the angle between the tangential vector  $\mathbf{n}_P$  to  $f_{b,k}$  and the vector from the center of  $e_{b,k}$  to  $\xi_{rand}$  (Figure 3), is lowest and less than a threshold  $\theta_{v,max}$  is selected as the viewpoint for that edge and added to the set  $W_c$  of candidate viewpoints. To reduce the number of viewpoints, a viewpoint  $v_l$  is selected from  $W_c$  and added to the set  $W_s$  of selected viewpoints. All boundary edges within a distance  $d_{v,max}$  from it for which  $\theta_v$  of  $v_l$  is less than  $\theta_{v,max}$  are extracted, and the corresponding viewpoints are removed from  $W_c$ . This procedure is repeated till  $W_c$  is empty. The path  $\sigma_{hc}$  to the viewpoint  $v_{close} \in W_s$  having the least path length from  $\xi_{curr}$  along  $\mathbb{G}_H$  is calculated and executed by the robot. This procedure is repeated until either no more boundary edges exist or no admissible viewpoints can be found for any boundary edge. Additionally, another threshold is introduced for the size of the hole removing small gaps, as well as a maximum time  $T_{hc}$  is allotted for hole coverage of each semantic. Once the hole coverage process is finalized for one detected semantic, the planner proceeds to the next detected semantic before eventually switching to inspection mode.

**D. Semantics Inspection**

For the reconstructed mesh of each semantic  $\mathbb{S}_I$  in the set of detected and partially inspected semantics  $\Lambda_I$ , the inspection mode of the planner finds an efficient path such that every face of the semantic mesh is inspected from at least one configuration in that path at the desired resolution and viewing angle. A face  $f$  is said to have been inspected from a configuration  $\xi$  if a) the centroid  $\mu_f$  of  $f$  lies within the modeled sensor  $\mathcal{Y}_C$  frustum, b) the distance  $d_f$  from  $\xi$  to  $\mu_f$  is within a threshold distance  $l_{max}$  (calculated based on  $r_{min}$ ), and c) the angle  $\theta_I$  between the vector from  $\mu_f$  to  $\xi$  and the outward normal  $\mathbf{n}_f$  is less than a set limit  $\theta_{I,max}$ . The inspection planning procedure is detailed in Algorithm 2 and illustrations are shown in Figure 4. The planner first selects the semantic  $\mathbb{S}_I \in \Lambda_I$  that is closest to the robot, calculates an oriented bounding box  $V_s$  around the

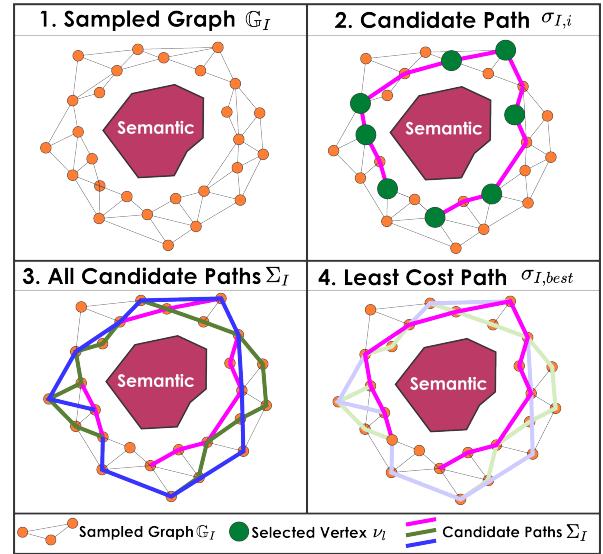


Fig. 4. Illustration of the steps involved in the Semantics Inspection mode of SWAP. A collision-free graph is built around the semantic.  $k$  candidate paths providing complete coverage are calculated from the graph, and the path having the least cost is executed by the robot.

**Algorithm 2** Semantics Inspection

---

```

1:  $\Lambda_I \leftarrow \text{extractUninspectedSemantics}(\Lambda)$ 
2: while  $\Lambda_I \neq \emptyset$  and  $\Lambda_H = \emptyset$  do
3:    $\mathbb{S}_{C,I} \leftarrow \text{closestSemantic}(\Lambda_I)$ 
4:    $\mathbb{G}_I \leftarrow \text{buildGraph}(\mathbb{S}_{C,I}, \mathbb{M})$ 
5:    $c_b \leftarrow \infty$ ,  $\sigma_{I,best} \leftarrow \emptyset$ 
6:   for  $i = 1$  to  $k$  do
7:      $V_c \leftarrow \text{selectCompleteCoverageVertices}()$ 
8:      $\sigma_{I,i} \leftarrow \text{solveTSP}(V_c)$ 
9:     if  $\text{cost}(\sigma_{I,i}) < c_b$  then
10:       $c_b \leftarrow \text{cost}(\sigma_{I,i})$ ,  $\sigma_{I,best} \leftarrow \sigma_{I,i}$ 
11:    $\Lambda_I \leftarrow \Lambda_I \setminus \mathbb{S}_{C,I}$ 
12:    $\Lambda_H \leftarrow \Lambda_H \cup \text{newSemanticsWithHoles}$ 

```

---

considered semantic mesh, and samples a 3D collision-free graph  $\mathbb{G}_I$ . For each vertex  $\nu_i$  in  $\mathbb{G}_I$ , the expected unobserved faces seen by  $\mathcal{Y}_C$  from  $\nu_i$  are calculated. The set  $L_i$  of faces seen by  $\nu_i$  is referred to as the visibility of  $\nu_i$ . Next, a path that ensures full inspection of the mesh, within the limitation of traveling in collision-free space, needs to be searched from this graph. To this end, the planner calculates  $k$  candidate paths  $\sigma_{I,i} \in \Sigma_I$ ,  $i = 1 \dots k$  providing complete coverage of the semantic mesh and selects the one with the least cost in terms of the path execution time. Each candidate path is calculated as follows. All vertices  $\nu_i$  in  $\mathbb{G}_I$  are sorted by the cardinality of  $L_i$ . The top  $\eta\%$  of vertices are selected, one vertex  $\nu_l$  is chosen from them at random and added to the set  $V_c$  of vertices in the final path. The visibility of the remaining vertices is re-evaluated to account for the overlap with the vertices in  $V_c$ . This process is continued till none of the remaining vertices have non-empty visibility. The order in which the vertices in  $V_c$  are to be visited is determined by solving the Traveling Salesman Problem (TSP), by means of the Lin-Kernighan-Helsgaun (LKH) heuristic [43], where the

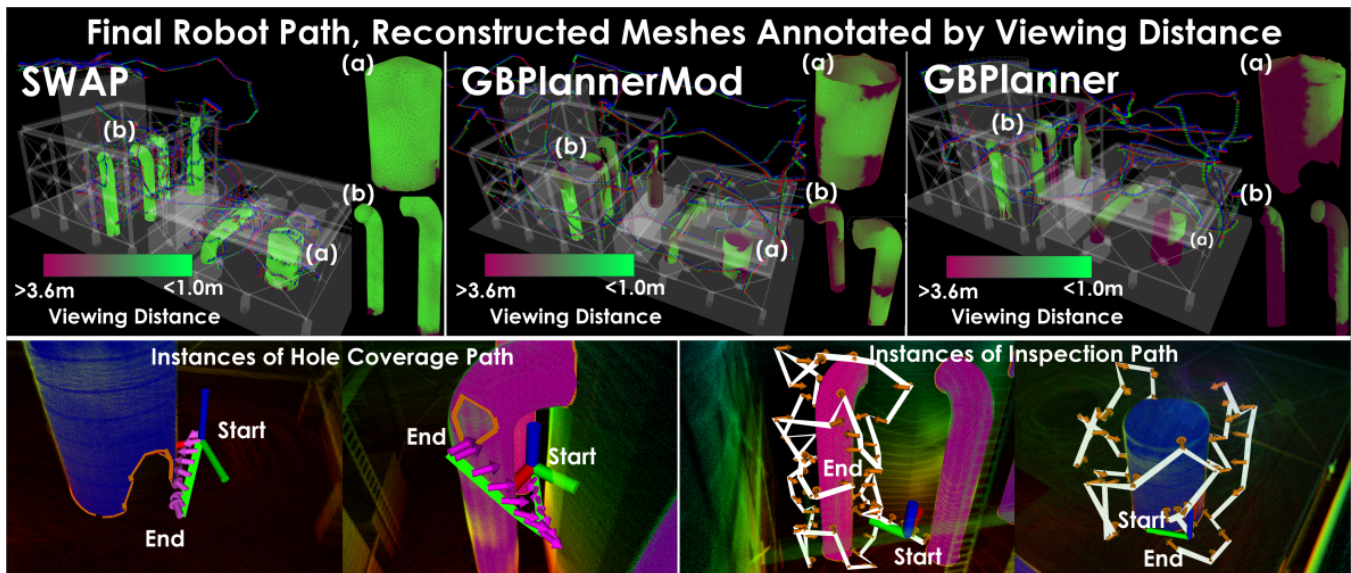


Fig. 5. This figure shows the results of the simulation conducted in the model of a chemical plant. The figures on the top show the overall path traversed by the robot along with the semantic mesh reconstructions annotated with the distance from which the faces of the meshes were viewed for all three planners. As seen from the mesh reconstructions, the proposed planner is able to provide more complete semantic meshes with near complete visual coverage than the state-of-the-art. The bottom row shows instances of hole coverage and inspection paths.

cost of travel between any two vertices is the length of the shortest path along  $\mathbb{G}_I$ . Finally, the path  $\sigma_{I,best} \in \Sigma_I$  with the lowest execution time cost  $c_b$ , is selected and executed by the robot. At the end of execution, if any new semantics  $\Lambda_H$  requiring hole coverage are detected, the planner switches to the hole coverage mode, otherwise moves to the next detected semantic, closest to the robot, for inspection.

## V. EVALUATION STUDIES

To evaluate the proposed semantics-aware exploration and inspection planner, both simulation and experimental studies were conducted. The computation times of various steps involved as well as the parameters used in both are presented in Tables I and II respectively.

### A. Simulation Studies

We present a simulation study for evaluating the planner inside a model of a chemical plant of size  $44 \times 28 \times 20$ m involving six semantics of interest. The simulation study utilized the Gazebo simulator [44] with a model of the RMF-Owl [45, 46] aerial robot with dimensions  $0.38 \times 0.38 \times 0.24$ m carrying a 3D LiDAR sensor as  $\mathcal{Y}_D$  with  $[F_h^D, F_v^D] = [360, 90]^\circ$ ,  $d_{max}^D = 50$ m, and a color camera as  $\mathcal{Y}_C$  having  $[F_h^C, F_v^C] = [120, 90]^\circ$ ,  $d_{max}^C = 7$ m. We utilize the semantic segmentation camera from the simulator to get segmented data. The simulations were conducted on a laptop with an Intel Core i9-10885H CPU.

The performance of the planner was compared against our previous Graph-based Exploration Planner (GBPlanner) [7, 9, 10] in both its original form (purely volumetric exploration given a depth sensor  $\mathcal{Y}_D$ ) and with a modified objective for surface coverage. The modified method (hereafter referred to as GBPlannerMod) annotates the mapped voxels that are also seen by  $\mathcal{Y}_C$  in the volumetric map. The method samples

a random graph as GBPlanner, and the information gain for each vertex  $v_i$  is defined as the number of unseen surface voxels inside  $\mathcal{Y}_C$  if the robot were at the configuration  $\xi_i$  corresponding to  $v_i$ . An unseen surface voxel is defined as an unknown voxel or an occupied voxel, that is mapped by  $\mathcal{Y}_D$  but not seen by  $\mathcal{Y}_C$ , both neighboring a free voxel. Using this new information gain formulation, the path is selected in a manner identical to GBPlanner.

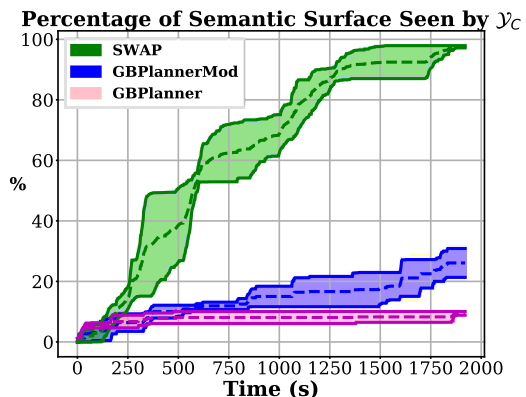


Fig. 6. This plot shows the percentage of the cumulative surface of all semantics seen by  $\mathcal{Y}_C$  over time by all three methods for the conducted simulation study. Each planner was run five times. The targeted inspection of SWAP enables it to significantly outperform the other methods.

A total of five simulation runs of each planner were conducted in the same environment starting from the same location. Each run lasted for 1850s. The overall paths traversed by the robot, the reconstructed meshes for few of the semantics, and indicative paths from the hole coverage and inspection steps can be seen in Figure 5. Figure 6 shows the semantic visibility comparison between the three methods in terms of the percentage of all semantic surfaces seen by  $\mathcal{Y}_C$ , within the required quality metrics, as a function of time. Due

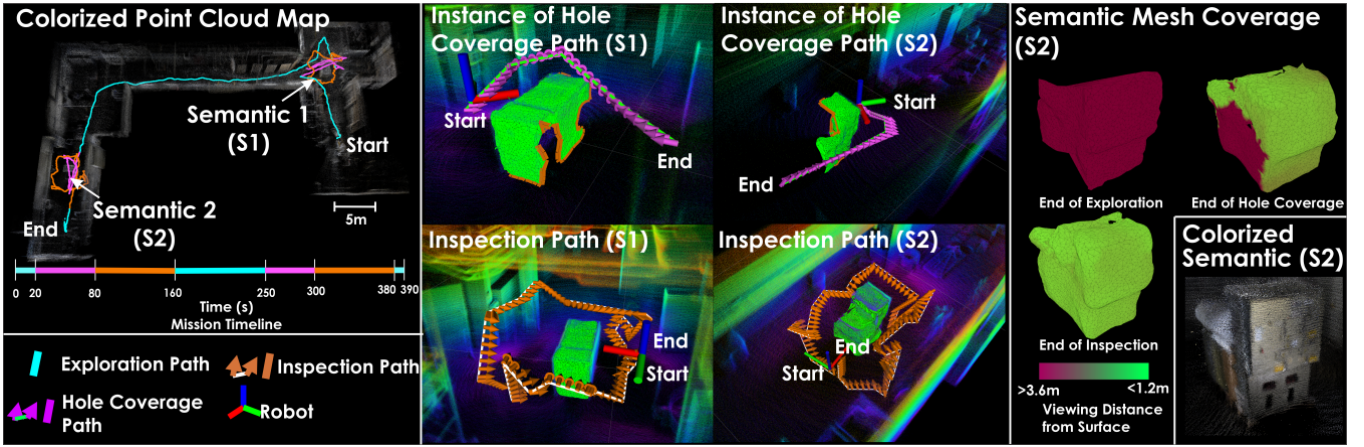


Fig. 7. The method was verified experimentally by deploying onboard the RMF-Owl aerial robot inside the basement of NTNU’s Elektro building. The experiment involved inspection of two machines distributed along three connected corridors connected via two  $90^\circ$  turns. In this figure, the top right sub-figure shows the overall path traversed by the robot along with a timeline of the mission colored according to the active mode. The sub-figure on the top left presents the mesh reconstruction of one of the semantics with each face colored by the distance from which it was seen by the onboard camera at the end of exploration (up to the beginning of hole coverage for that semantic), hole coverage, and inspection modes. The two sub-figures at the bottom right show the inspection and one of the hole coverage paths for one semantic. Finally the sub-figure on the bottom left shows the point cloud reconstruction of the semantic colored based on the onboard camera image. The robot was successfully able to explore the environment and inspect both the semantics at the required metrics due to the planner’s ability to perform combined exploration and targeted inspection of the semantics.

TABLE I

COMPUTATION TIMES FOR VARIOUS STEPS INVOLVED.

Computation Step	Simulation	Experiment
Volumetric Exploration (s)	0.648	0.446
Semantics Hole Coverage (s)	0.027	0.121
Semantics Inspection (s)	2.689	1.527
Mesh computation (s)	0.024	0.054

TABLE II

PARAMETERS USED IN THE SIMULATION AND EXPERIMENT.

Parameter	Simulation	Experiment
$\lambda_V$ (m)	0.2	0.3
$r_{min}$ (pixels/cm <sup>2</sup> )	20.67	5.06
$\theta_{I,max}$ (deg)	45	45
$\theta_{H,max}$ (deg)	75	75
$k$	10	3

to the ability of SWAP to perform targeted inspection of the semantics, it is able to outperform both the other methods. Even GBPlannerMod, with its new formulation for surface inspection, is unable to capture all surfaces of the semantic at the desired quality.

### B. Experimental Studies

For experimental verification, SWAP is deployed onboard RMF-Owl [45], a small-sized ( $0.38 \times 0.38 \times 0.24$ m) and lightweight (1.4kg) aerial robot with approximately 10min of endurance integrating a) a multi-modal sensing suite involving an OUSTER OS0 3D LiDAR with 64 channels used as  $\mathcal{Y}_D$  ( $[F_h^D, F_v^D] = [360, 90]^\circ$ ,  $d_{max}^D = 20$ m), a FLIR Blackfly S 0.4MP visual camera used as  $\mathcal{Y}_C$  ( $[F_h^C, F_v^C] = [85, 64]^\circ$ ,  $d_{max}^C = 7$ m), and an IMU, and b) a Khadas VIM3 Pro Single Board Computer (SBC) incorporating  $\times 4$  2.2Ghz Cortex-A73 cores, paired with  $\times 2$  1.8Ghz Cortex-A53 cores implementing an A311D big-little architecture. The robot was developed as part of the activities of Team CERBERUS in the DARPA Subterranean Challenge [7, 8,

47] and integrates a robust localization and mapping method as presented in [48] upon which autonomous path planning can take place.

The conducted experiment took place in the basement of NTNU’s Elektro Building and involved two industrial machines as the semantics of interest distributed along three corridors connected via two  $90^\circ$  turns. The robot started at the beginning of one of the corridors, in the exploration mode with  $T_e = 20$ s, and after that switched to the semantic modes. Upon completion, it continued exploration and switched to semantic modes after detecting the second semantic. It is noted that as this work is focused on path planning, the semantic detection and point cloud segmentation is considered given and it in the experiment it was implemented by using an AprilTag unique to each semantic object. Figure 7 presents the result of this mission showing the path traversed by the robot, the mission timeline, instances of hole coverage and inspection paths, the generated mesh annotated with the camera viewing distance, and the colorized reconstructed point cloud of one of the semantics. The average resolution at which the semantics S1 and S2 were viewed was 9.81 and 14.40 pixels/cm<sup>2</sup>.

## VI. CONCLUSIONS

A semantics-aware path planner for exploration of an unknown environment combined with mesh reconstruction and inspection of the semantics of interest is presented. The behavior-based approach allows the planner to volumetrically explore, generate complete mesh reconstructions of the semantics, and perform an inspection of their faces given specific image quality metrics without any prior knowledge or assumptions about the environment. Both simulation studies comparing the method with a state-of-the-art exploration planner and an experiment using a flying robot are conducted to verify the new method.

## REFERENCES

- [1] I. Sa and P. Corke, "Vertical infrastructure inspection using a quadcopter and shared autonomy control," in *Field and service robotics*. Springer, 2014, pp. 219–232.
- [2] C. Gehring, P. Fankhauser, L. Isler, R. Diethelm, S. Bachmann, M. Potz, L. Gerstenberg, and M. Hutter, "Anymal in the field: Solving industrial inspection of an offshore hvdc platform with a quadrupedal robot," in *12th Conference on Field and Service Robotics (FSR 2019)*, 2019.
- [3] G. Caprari, A. Breitenmoser, W. Fischer, C. Hürzeler, F. Tâche, R. Siegwart, O. Nguyen, R. Moser, P. Schoeneich, and F. Mondada, "Highly compact robots for inspection of power plants," *Journal of Field Robotics*, vol. 29, no. 1, pp. 47–68, 2012.
- [4] B. Chan, H. Guan, J. Jo, and M. Blumenstein, "Towards uav-based bridge inspection systems: A review and an application perspective," *Structural Monitoring and Maintenance*, vol. 2, no. 3, pp. 283–300, 2015.
- [5] A. Bircher, K. Alexis, M. Burri, P. Oettershagen, S. Omari, T. Mantel and R. Siegwart, "Structural inspection path planning via iterative viewpoint resampling with application to aerial robotics," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 6423–6430. [Online]. Available: <https://github.com/ethz-asl/StructuralInspectionPlanner>
- [6] G. A. Hollinger, B. Englot, F. Hover, U. Mitra, and G. S. Sukhatme, "Uncertainty-driven view planning for underwater inspection," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 4884–4891.
- [7] M. Tranzatto, T. Miki, M. Dharmadhikari, L. Bernreiter, M. Kulkarni, F. Mascarich, O. Andersson, S. Khattak, M. Hutter, R. Siegwart, and K. Alexis, "Cerberus in the darpa subterranean challenge," *Science Robotics*, vol. 7, no. 66, p. eabp9742, 2022.
- [8] M. Tranzatto, M. Dharmadhikari, L. Bernreiter, M. Camurri, S. Khattak, F. Mascarich, P. Pfreundschuh, D. Wisth, S. Zimmermann, M. Kulkarni *et al.*, "Team cerberus wins the darpa subterranean challenge: Technical overview and lessons learned," *arXiv preprint arXiv:2207.04914*, 2022.
- [9] M. Kulkarni, M. Dharmadhikari, M. Tranzatto, S. Zimmermann, V. Reijgwart, P. De Petris, H. Nguyen, N. Khedekar, C. Papachristos, L. Ott, R. Siegwart, M. Hutter, and K. Alexis, "Autonomous teamed exploration of subterranean environments using legged and aerial robots," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 3306–3313.
- [10] T. Dang, M. Tranzatto, S. Khattak, F. Mascarich, K. Alexis, and M. Hutter, "Graph-based subterranean exploration path planning using aerial and legged robots," *Journal of Field Robotics*, vol. 37, no. 8, pp. 1363–1388, 2020.
- [11] N. Hudson, F. Talbot, M. Cox, J. Williams, T. Hines, A. Pitt, B. Wood, D. Frousheger, K. L. Surdo, T. Molnar *et al.*, "Heterogeneous ground and air platforms, homogeneous sensing: Team csiro data61's approach to the darpa subterranean challenge," *arXiv preprint arXiv:2104.09053*, 2021.
- [12] A. Agha, K. Otsu, B. Morrell, D. D. Fan, R. Thakker, A. Santamaria-Navarro, S.-K. Kim, A. Bouman, X. Lei, J. Edlund *et al.*, "Nebula: Quest for robotic autonomy in challenging environments; team costar at the darpa subterranean challenge," *arXiv preprint arXiv:2103.11470*, 2021.
- [13] T. Rouček, M. Pecka, P. Čížek, T. Petříček, J. Bayer, V. Šalanský, D. Heřt, M. Petrлік, T. Báča, V. Spurný *et al.*, "Darpa subterranean challenge: Multi-robotic exploration of underground environments," in *International Conference on Modelling and Simulation for Autonomous Systems*. Springer, Cham, 2019, pp. 274–290.
- [14] S. Scherer, V. Agrawal, G. Best, C. Cao, K. Cujic, R. Darnley, R. DeBortoli, E. Dexheimer, B. Drozd, R. Garg, I. Higgins, J. Keller, D. Kohanbash, L. Nogueira, R. Pradhan, M. Tatum, V. K. Viswanathan, S. Willits, S. Zhao, H. Zhu, D. Abad, T. Angert, G. Armstrong, R. Boirum, A. Dongare, M. Dorman, S. Hu, J. Jaekel, R. Ji, A. Lai, Y. Hsuan Lee, A. Luong, J. Mangelson, J. Maier, J. Picard, K. Pluckter, A. Saba, M. Saroya, E. Scheide, N. Shoemaker-Trejo, J. Spisak, J. Teza, F. Yang, A. Wilson, H. Zhang, H. Choset, M. Kaess, A. Rowe, S. Singh, J. Zhang, G. A. Hollinger, and M. Travers, "Resilient and modular subterranean exploration with a team of roving and flying robots," *Field Robotics*, 2021.
- [15] F. Tao, H. Zhang, A. Liu, and A. Y. Nee, "Digital twin in industry: State-of-the-art," *IEEE Transactions on industrial informatics*, vol. 15, no. 4, pp. 2405–2415, 2018.
- [16] L. Schmid, J. Delmerico, J. Schönberger, J. Nieto, M. Pollefeys, R. Siegwart, and C. Cadena, "Panoptic multi-tdsfs: a flexible representation for online multi-resolution volumetric mapping and long-term dynamic scene consistency," in *2022 IEEE International Conference on Robotics and Automation (ICRA)*, 2022, pp. 8018–8024.
- [17] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, and A. Markham, "Randla-net: Efficient semantic segmentation of large-scale point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 108–11 117.
- [18] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez, "A review on deep learning techniques applied to semantic segmentation," *arXiv preprint arXiv:1704.06857*, 2017.
- [19] M. Grinvald, F. Furrer, T. Novkovic, J. J. Chung, C. Cadena, R. Siegwart, and J. Nieto, "Volumetric instance-aware semantic mapping and 3d object discovery," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 3037–3044, 2019.
- [20] J. McCormac, R. Clark, M. Bloesch, A. Davison, and S. Leutenegger, "Fusion++: Volumetric object-level slam," in *2018 international conference on 3D vision (3DV)*. IEEE, 2018, pp. 32–41.
- [21] A. Rosinol, M. Abate, Y. Chang, and L. Carlone, "Kimera: an open-source library for real-time metric-semantic localization and mapping," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 1689–1696.
- [22] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto, "Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.
- [23] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, 2013.
- [24] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova and R. Siegwart, "Receding horizon "next-best-view" planner for 3d exploration," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2016. [Online]. Available: <https://github.com/ethz-asl/nbvplanner>
- [25] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, "Receding horizon path planning for 3d exploration and surface inspection," *Autonomous Robots*, pp. 1–16, 2016.
- [26] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *CIRA'97*. IEEE, 1997, pp. 146–151.
- [27] C. Cao, H. Zhu, H. Choset, and J. Zhang, "Tare: A hierarchical framework for efficiently exploring complex 3d environments." in *Robotics: Science and Systems*, 2021.
- [28] G. Best, R. Garg, J. Keller, G. A. Hollinger, and S. Scherer, "Resilient multi-sensor exploration of multifarious environments with a team of aerial robots," in *Robotics: Science and Systems (RSS)*, 2022.
- [29] A. Bircher, M. Kamel, K. Alexis, M. Burri, P. Oettershagen, S. Omari, T. Mantel and R. Siegwart, "Three-dimensional coverage path planning via viewpoint resampling and tour optimization for aerial robots," *Autonomous Robots*, pp. 1–25, 2015.
- [30] A. Bircher, K. Alexis, U. Schwesinger, S. Omari, M. Burri, and R. Siegwart, "An incremental sampling-based approach to inspection planning: the rapidly exploring random tree of trees," *Robotica*, vol. 35, no. 6, p. 1327–1340, 2017.
- [31] D. E. Soltero, M. Schwager, and D. Rus, "Decentralized path planning for coverage tasks using gradient descent adaptive control," *The International Journal of Robotics Research*, vol. 33, no. 3, pp. 401–425, 2014.
- [32] H. Choset and P. Pignon, "Coverage path planning: The boustrophedon cellular decomposition," in *International conference on field and service robotics*, vol. 12, 1997.
- [33] E. Galceran and M. Carreras, "A survey on coverage path planning

- for robotics,” *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1258–1276, 2013.
- [34] F. S. Hover, R. M. Eustice, A. Kim, B. Englot, H. Johannsson, M. Kaess, and J. J. Leonard, “Advanced perception, navigation and planning for autonomous in-water ship hull inspection,” *The International Journal of Robotics Research*, vol. 31, no. 12, 2012.
- [35] L. Bartolomei, L. Teixeira, and M. Chli, “Semantic-aware active perception for uavs using deep reinforcement learning,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 3101–3108.
- [36] —, “Perception-aware path planning for uavs using semantic segmentation,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 5808–5815.
- [37] M. Ryll, J. Ware, J. Carter, and N. Roy, “Semantic trajectory planning for long-distant unmanned aerial vehicle navigation in urban environments,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 1551–1558.
- [38] T. Dang, C. Papachristos, and K. Alexis, “Autonomous exploration and simultaneous object search using aerial robots,” in *2018 IEEE Aerospace Conference*. IEEE, 2018, pp. 1–7.
- [39] T. Koch, M. Körner, and F. Fraundorfer, “Automatic and semantically-aware 3d uav flight planning for image-based 3d reconstruction,” *Remote Sensing*, vol. 11, no. 13, p. 1550, 2019.
- [40] S. Achat, J. Marzat, and J. Moras, “Path Planning Incorporating Semantic Information for Autonomous Robot Navigation,” in *19th International Conference on Informatics in Control, Automation and Robotics (ICINCO) 2022*, Lisbonne, Portugal, Jul. 2022. [Online]. Available: <https://hal.science/hal-03831940>
- [41] V. Suriani, S. Kaszuba, S. R. Sabbella, F. Riccio, and D. Nardi, “S-ave: Semantic active vision exploration and mapping of indoor environments for mobile robots,” in *2021 European Conference on Mobile Robots (ECMR)*. IEEE, 2021, pp. 1–8.
- [42] D. Cohen-Steiner and F. Da, “A greedy delaunay-based surface reconstruction algorithm,” in *The Visual Computer*, vol. 22, 2004, pp. 4–16.
- [43] K. Helsgaun, “An effective implementation of the lin–kernighan traveling salesman heuristic,” *European journal of operational research*, vol. 126, no. 1, pp. 106–130, 2000.
- [44] Gazebo simulator. Open Robotics Organization.
- [45] P. D. Petris, H. Nguyen, M. Dharmadhikari, M. Kulkarni, N. Khedekar, F. Mascarich, and K. Alexis, “Rmf-owl: A collision-tolerant flying robot for autonomous subterranean exploration,” in *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2022, pp. 536–543.
- [46] M. Kulkarni and D. P. Paolo. (September) Cerberus rmf sensor config 1. Autonomous Robots Lab. [Online]. Available: [https://github.com/ntnu-arl/subt/tree/submitted\\_models/cerberus\\_rmf\\_obelix\\_sensor\\_config\\_1/submitted\\_models/cerberus\\_rmf\\_obelix\\_sensor\\_config\\_1](https://github.com/ntnu-arl/subt/tree/submitted_models/cerberus_rmf_obelix_sensor_config_1/submitted_models/cerberus_rmf_obelix_sensor_config_1)
- [47] M. Tranzatto, F. Mascarich, L. Bernreiter, C. Godinho, M. Camurri, S. M. K. Khattak, T. Dang, V. Reijgwart, J. Loeje, D. Wisth *et al.*, “Cerberus: Autonomous legged and aerial robotic exploration in the tunnel and urban circuits of the darpa subterranean challenge,” *Field Robotics*, 2021.
- [48] S. Khattak, H. Nguyen, F. Mascarich, T. Dang, and K. Alexis, “Complementary multi-modal sensor fusion for resilient robot pose estimation in subterranean environments,” in *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2020, pp. 1024–1029.