A Case Study of Semantic Mapping and Planning for Autonomous Robot Navigation

Silya Achat, Quentin Serdel, Julien Marzat and Julien Moras

DTIS, ONERA, Université Paris-Saclay, 91123, Palaiseau, France.

Contributing authors: firstname.lastname@onera.fr;

Abstract

This paper presents an approach to take into account semantic information for autonomous robot tasks which require planning capabilities, e.g. to determine a path or a next-best-view configuration. A semantic map can be constructed from labeled pointclouds acquired at successive sensor poses. This map then serves as input for generating a multi-layered structure, which can be exploited by multiple planners to address various navigation goals and constraints. Semantic-aware adaptations of A*, Transition-based RRT and a shortcut algorithm are derived in this framework, and evaluated numerically on an exploration and observation task using a reference dataset with multiple semantic classes as an illustrative test environment. The performance of real-time construction of the corresponding semantic map is also evaluated on the same dataset.

Keywords: Autonomous Robot Navigation, Planning Algorithms, Semantic Mapping, Semantic Scene Understanding, Exploration and Observation

1 Introduction

The combined recent progress in learning algorithms and computational power have led to the development of efficient semantic segmentation capabilities based on data from embedded LiDAR and/or RGB-D sensors, processed by (deep) neural networks to produce 2D annotated images [1] or 3D pointclouds [2–5]. The fully-embedded processing of this key perception asset is now also reported for Unmanned Aerial Vehicles (UAV) [6, 7]. This has paved the way for the development of semantic mapping algorithms, where the 3D representation of a surrounding environment also includes the categorization of the perceived 3D points or even 3D object segmentation at a higher level [8–11]. Compared to classical mapping structures which provide binary occupancy information, e.g. Octomap [12] or TSDF-based mapping [13], this opens new possibilities for autonomous robot navigation to directly take into account multiple mission objectives and constraints in interaction with the environment. Ground robots are incrementally used for exploration and various missions in unstructured and hazardous environments, where safety is a primary concern. While most actual methods for robot mapping and navigation rely solely on geometric information, the recent development of efficient semantic segmentation neural networks allow the inclusion of semantic information in robot mapping [14, 15]. This new capability can help the robots to identify and avoid problematic terrains during navigation as well as perform semantically assisted tasks such as identifying the position of specific instances. Since the nature of a terrain cannot always be estimated from its geometry alone, image semantic segmentation would represent a necessary addition to standard 3D mapping.

In this context, we propose a systematic approach to incorporate semantic mapping information in planning algorithms, in particular with the derivation of semantized versions of the A* and Transition-based Rapidly exploring Random Tree (T-RRT) algorithms as well as a post-processing shortcut procedure. The semantic classes are ranked by a user-defined cost representing traversability or observability constraints, and some classes are also identified as of particular interest for observation. The semantic map and the corresponding task objectives and constraints are then included in a multi-layered structure which can be exploited online by the planning algorithms. The proposed algorithms are evaluated and compared on typical autonomous robot navigation missions, namely waypoint rallying in the presence of obstacles and different types of terrain, and the exploration of an uncharted environment with observation of detected points of interest[1]. This work is an extended version of the ICINCO 2022[2] conference paper [16], with a consolidated system architecture including a method for semantic map construction and its numerical evaluation on the same reference 3DRMS dataset [17], as well as the description of the navigation graph interface between the map and the planners. More precisely, the newly presented mapping process relies on the semantic Octomap method [18] which takes in input a semantic pointcloud generated from the combination of depth and semantically-annotated images available in pre-recorded sequences of the dataset. The obtained map has then been evaluated at several resolutions in terms of geometric and classification precision compared to the ground-truth semantic pointcloud, and the corresponding computational cost has also been recorded. Another contribution of this paper is the transformation of the semantic map into a navigation grid for the aforementioned semantic-aware path-planners. The combination of all these building blocks are thus evaluated on a single case study to demonstrate the feasibility of a full semantic-aware mapping-and-planning process for an autonomous robot.

[1] Video at https://tinyurl.com/SemanticPlanning

[2] https://icinco.scitevents.org/?y=2022

2 Related work

A limited number of previous works have studied the exploitation of semantic maps for planning the motion of autonomous robots dealing with complex tasks. As described in the survey papers [14, 15], a lot of effort has been put on defining several semantic map representations for various tasks but there are still few complete semantic navigators linking the proposed maps with planning algorithms, and more extensive simulation and real-world experiments should be conducted. In the early work by [19], a 2D costmap associated with the traversability of an off-road terrain was obtained by an aerial vehicle to allow the navigation of a ground robot in a natural environment. This map was combined with a local one derived from the on-board sensors of the ground robot, and a path was then computed with a D* algorithm exploiting the local and global traversability data. However, the top view can bias the real traversability of the terrain, for example in the case of dense foliage trees on a flat and passable terrain.

In the context of inspection tasks carried out by an autonomous ground vehicle in a nuclear storage environment, the authors of [20] proposed to exploit a 2D binary custom map of obstacles containing the locations and orientations of objects of interest so as to build an enriched map that can be exploited for inspection-oriented path planning. In [21], a Geographic Information System (GIS) with geometrical and semantic layers is used to build costmaps that are exploited by the ROS *move_base* package to carry out a simplified fetch-and-deliver task with a ground platform. This was a first successful attempt demonstrating that a semantic map can be used within navigation modules and not only for data visualization. In [22], an active-vision approach has been proposed for an indoor exploration mission by a mobile robot so as to generate successive next best views based on the detected segmented objects and associated geometrical priors on their respective sizes.

The navigation of a rover in the framework of a Martian mission has been addressed in [23] and in a similar way in [24]. The onboard sensors provide raw images and a Digital Elevation Map to plan the rover's path over rocky terrain, where each recognized rock is classified. An RRG algorithm derived from RRT allows to define waypoints depending on the rock types to be avoided by taking into account the positioning of the wheels, and these waypoints are included in a graph to obtain optimal paths with an A* algorithm. These considerations on the rover model and the associated path planning architecture remain however very specific and present a high computational cost for real-time exploration-and-observation tasks.

A semantic 2D grid has also been exploited in [25] for traversability evaluation, along with a D* path planning algorithm to reach a destination designated by a human operator in the context of a rescue mission. An active perception approach has been derived in [26] for the autonomous navigation of a UAV using on-board visual odometry. The idea is to use semantic classes available in a 3D voxel map to evaluate perceptually-informative scenes and therefore maintain a reliable localization. A hierarchical structure is proposed, combining an A* path planner and B-Spline trajectory optimization with a penalty term to keep the most informative landmarks in the field of view. A similar problem has also been tackled in [27] using model predictive control as an online planning strategy.

A semantic planner for a UAV equipped with a RGB camera navigating in an unknown urban environment has been proposed and evaluated experimentally in [28]. The acquired images are first segmented by a convolutional network, and then used to build a projected probabilistic map giving preference to roads, which are assumed to be safer to fly over. A high-level long-distance traversability graph is finally inferred by a deep neural network as a combination of pre-defined geometrical primitives and the UAV path is extracted by direct graph search. In [29], a neural network architecture aims at providing probabilistic semantic occupancy layers including current and predicted locations of vehicles and obstacles for a self-driving vehicle in an urban environment, using voxelized LiDAR data and a prior mapping. The vehicle trajectory is then selected among a set of motion primitives by optimizing a cost function including penalty terms related to safety (computed using the predicted semantic segmentation), and other terms related to driving comfort and traffic rules which are independent from the semantic information. In [30], a hybrid version of A* relying on a distance map (instead of occupancy) and a vehicle collision model has been proposed for path planning in an urban environment. This allows safe navigation for this specific self-driving car application but cannot generalize to the navigation of any robotic system in any environment that seeks to optimize the nature of the areas to be traversed. In [31], a 2.5D semantic map is built by combining semantically segmented images and LiDAR depth information. This grid is centered on the position of an autonomous vehicle as it evolves in an outdoor unstructured environment, and allows to select a feasible instantaneous path from a set of primitives based on the evaluation of traversability costs associated to the different classes.

These previous works mainly focused on a single specific objective or constraint related to the exploitation or data acquisition of semantic information. We propose to incorporate data from the semantic map into a multi-layered structure that can be readily adapted to multiple planners, such that different autonomous navigation tasks can be addressed in a multi-class environment. This makes it possible to derive variations of standard planning algorithms in a more systematic way than in the above-referenced related work, so as to efficiently carry out missions with multiple objectives and constraints. A numerical evaluation of the construction of the semantic map and its exploitation by a high-level Next-Best-View planner and a low-level path planner is also demonstrated on the same reference dataset.

3 Proposed approach

3.1 Problem formulation

A large majority of robotic tasks carried out by autonomous robots can be split into the definition of a high-level goal, followed by the generation of a path that the robot should follow to reach this goal while respecting a set of constraints. Examples of such tasks are the rallying of an arbitrary waypoint with obstacle avoidance or taking into account perception constraints, next-best-view exploration where successive goals are generated on the currently known frontier, fetch-and-deliver tasks where detected objects are defined as targets. Such systems are usually implemented using a set of planners organised in a hierarchical structure. The information used in this kind of process is classically included in discretized 2D or 3D maps, such as binary occupancy grids encoding the presence of obstacles, exploration grids storing the explored locations, or more generally costmaps that can encode arbitrary potential functions depending on the state of the vehicle [32]. Multi-layered structures [33] can then be defined to combine different types of data that can be accessed simultaneously to evaluate the quality of the high-level goals or the generated paths. High-level semantic maps have also been defined, e.g. in [34], where the labels correspond to different areas or rooms in a simplified navigation graph.

We define a configuration q in an associated bounded space $\mathbb{S}$ where the tasks are executed, and the position state components of q are denoted by $\xi \in \mathbb{R}^n$, with $n = 2$ for mobile robots and $n = 3$ for UAVs. The other components of q, left unspecified, could represent orientation and additional multi-body coordinates depending on the task.

A multi-layer map structure $\mathcal{M}$ is defined as a set of n_m layers $\mathcal{M}_i$, which can be used to evaluate an arbitrary cost function $c_j(q, \mathcal{M}_i)$ at any configuration q. Each planner can therefore use this multi-layered map to either compute an optimal path or to select an optimal goal by evaluating a utility function $u(\mathcal{C}(q_{\text{goal}}, \mathcal{M}))$, where $\mathcal{C}(q, \mathcal{M}) = [c_j(q, \mathcal{M}_i)]^T$ contains all the relevant cost evaluations c_j $(j = 1, ..., n_c)$ from appropriate layers $\mathcal{M}_i$ for this given task. The focus is put on how the information stored in a semantic pointcloud or map can be further included into this process and how the definition of goals and paths can be adapted from classical algorithms. It is assumed that the semantic information on the environment can be represented by a closed frame of discernment $\Omega = \{l_1, l_2, ..., l_n\}$ containing all the labels considered l_i. This makes it possible to create a semantic grid $\mathcal{S}$ of the environment (as detailed in Section 3.2), where each cell of index j of the map contains a label value $\mathcal{S}_j \in \Omega$. This semantic grid will be used as a proxy to define semantic-aware map layers.

This global approach has been applied, in this work, to a system using a 2-level planning architecture applied to a double-objective task: environment exploration and inspection of objects of interest. A Next-Best-View algorithm will be used at the top level in order to manage the two objectives whereas a lower-level planner will generate a path considering traversability constraints. For the latter, both weighted A* and T-RRT will be evaluated.

The perception information is aggregated into a map structure containing three layers depicted on the left of Figure 1. The semantic grid $\mathcal{S}$ is projected in the first layer $\mathcal{M}_1$, where a label is assigned to each cell that could be updated using sensor inputs. The second layer $\mathcal{M}_2$ is a binary exploration grid, where the observed cells are set to 1 and the unknown cells are set to 0. The last layer $\mathcal{M}_3$ is dedicated to monitor the observation of objects of interest, represented, as follows, by a three-valued cell state.

$$\mathcal{M}_3(j) = \begin{cases} 0 & \text{not an object of interest} \\ 1 & \text{object of interest not observed} \\ 2 & \text{object of interest already observed} \end{cases} \tag{1}$$

Other types of map layers could be considered for alternative tasks.

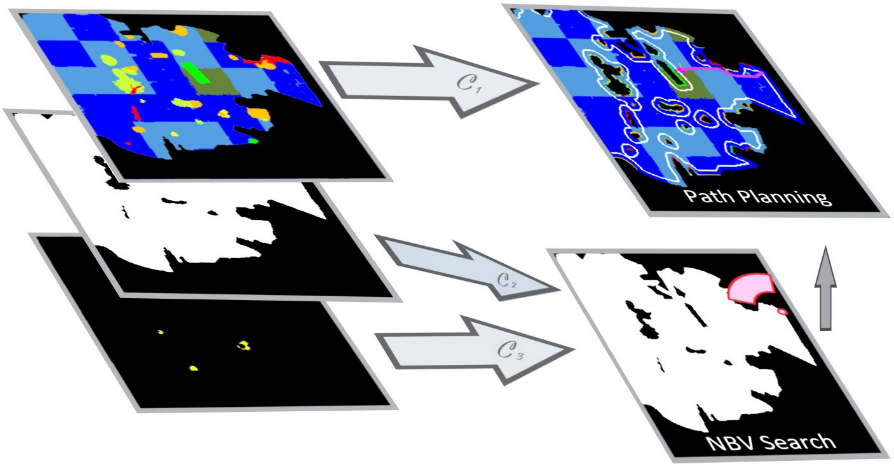

Fig. 1: Example of a three-layer map structure (left): (i) a semantic grid $\mathcal{M}_1$ derived from semantic pointcloud inputs to incorporate traversability or observation constraints. (ii) a classical exploration grid $\mathcal{M}_2$ with binary states $\{$*unknown* ; *explored*$\}$. (iii) a ternary grid $\mathcal{M}_3$ to monitor the observation of specific semantic classes. This structure is exploited by two planners: a NBV planner (bottom right) computes costs from $\mathcal{M}_2$ and $\mathcal{M}_3$ to determine the best goal location for an exploration-and-observation task, towards which a path is generated by a low-level planner (upper right) by taking into account traversability from $\mathcal{M}_1$.

The low-level planner has to compute a path P, defined as the ordered list of m successive configurations $q_k \in \mathbb{S}$, from the start q_0 to the goal $q_{m-1} = q_{\text{goal}}$. This problem can be formulated as finding an optimal path for some criteria (shortest path, no-collision, granting a sufficient safety level, ...) evaluated from the appropriate set of n_c functions $\mathcal{C}(q, \mathcal{M})$. For instance, a standard formulation to evaluate the quality of a path can be derived as:

$$J(P, \mathcal{M}) = \sum_{k=0}^{m-2} \sum_{j=1}^{n_c} c_j(q_{k+1}, \mathcal{M}) \cdot \|\xi_{k+1} - \xi_k\| \tag{2}$$

where each segment of the path is weighted by the sum of appropriate costs extracted from the data layers. A constraint depending on a semantic label can therefore be introduced by defining a cost value $c(q_j, \mathcal{M}_1)$ for each cell as a function of the corresponding semantic label l_i at the location q_j corresponding to the cell j of $\mathcal{M}_1$. A specific label *unknown*, which is not traversable, has been added to represent the state of the unseen cells. As an application example, we consider traversability constraints defined in the following way:

$$c(q_j, \mathcal{M}_1) = c(\xi_j, \mathcal{M}_1) = \begin{cases} \dfrac{v_{\text{ref}}}{v_i^{\max}} & \text{if } l_i \text{ is traversable} \\ +\infty & \text{if } l_i \text{ is not traversable} \end{cases} \tag{3}$$

where $v_i^{\max}$ is the maximum allowed velocity for a robot located in a cell with a traversable label l_i, and $v_{\text{ref}} = \max_i(v_i^{\max})$. This incorporates constraints related to the motion of a ground robot on a specific soil type or the presence of sufficient textures in the scene for vision-based localization, as in [26, 27] for the latter. For a simple navigation task towards an arbitrary waypoint considering only the information from this map layer $\mathcal{M}_1$ (as depicted in Figure 1), the proposed cost function (2) simplifies into a sum along the path of the single cost $c(\xi_k, \mathcal{M}_1)$ defined in (3). Thus, for a path section within an area labeled l_i such that $v_{\text{ref}} = v_i^{\max}$, the cost associated to this label would be equal to 1, so the cost for this path section would be equivalent to its Euclidean distance. It should then be minimized by a candidate planning algorithm to incorporate this semantic information in relation with the mission carried out. As a byproduct, it can be directly used to evaluate the paths obtained and thus compare the efficiency of different planners for such a task (see Section 4). Note that the infeasibility character of a path is directly obtained from the definition of non-traversable infinite costs in (3).

The second layer $\mathcal{M}_2$ is used to monitor a standard exploration mission, where the views are evaluated by considering the current frontier between explored and known cells (see Section 3.4). An independent subset $\mathcal{V} \subset \Omega$ contains a list of labels that should be observed during exploration, with a visit status updated in the corresponding layer $\mathcal{M}_3$ as defined in (1). The resulting NBV is computed by a high-level planner based on these two layers to fulfill the objectives of this simultaneous exploration-and-observation task.

The lower-level planner is then called upon to generate the path to reach this NBV using the process defined above, which takes into account traversability via layer $\mathcal{M}_1$.

3.2 Semantic mapping

The construction of a discretized metric-semantic map is a mandatory requirement for the presented planning approach. Several recent works in this area have given promising results. The mapping approaches proposed by [10, 18, 35] fuse directly image semantic segmentation produced by deep learning algorithms such as [1, 36] to build 3D semantic maps. Others works as [37, 38] on pointcloud semantic segmentation reach a high level of accuracy and could also be used to produce such a map.

3.2.1 Semantic dataset

The 3DRMS challenge dataset [17] has been selected as a case study. This dataset contains pre-recorded sequences of camera images and poses related to the navigation of a robot in a simulated garden-like outdoor environment. Depth images and ground-truth semantically-segmented images are available along with a semantic pointcloud representing the ground-truth for the 3D geometry and labeling of the environment. The inputs for the semantic mapping process have been generated from the fusion of the segmentation and depth images of the 4 cameras linked to the simulated mobile robot to produce a set of labeled pointclouds, each one associated to a robot pose and containing an average of 770K points. To evaluate the performances of the semantic mapping process, we have used the ground-truth semantic pointcloud of the complete environment to evaluate the quality of the mapping output and as a basis for the numerical experiments reported in Section 4. To obtain a ground-truth semantic map, the ground-truth labeled pointcloud has been voxelized following a transformation similar to the one proposed in [39]. For each voxel, its label $l \in \Omega$ is assigned following a majority vote including all the points belonging to this voxel. If no point belongs to a voxel then its label is assigned to *free*. A point $\{x, y, z\}$ belongs to a voxel $\{i, j, k\}$ of resolution h if it satisfies:

$$\begin{cases} x \text{ s.t. } ih \leq x < (i+1)h \\ y \text{ s.t. } jh \leq y < (j+1)h \\ z \text{ s.t. } kh \leq z < (k+1)h \end{cases} \tag{4}$$

3.2.2 Real-time construction of the semantic map

In this section, we present the implementation and evaluation of the semantic Octomap [18] online 3D mapping algorithm. This method takes labeled pointclouds associated to sensor poses as input. The received pointclouds are filtered, then 3D mapping is performed using Octomap [12] which provides a volumetric representation of the scene as a hierarchical structure of voxels with assigned semantic labels. This data structure allows fast and efficient update and access to the 3D cells. The output of the mapping process is used to build online a dense 2.5D navigation graph (see Section 3.2.3) in which the robot can plan paths and make exploration decisions. Figure 2 shows the complete semantic volumic reconstruction obtained with Octomap on Sequence 1 of the 3DRMS challenge dataset, along with the corresponding navigation graph.

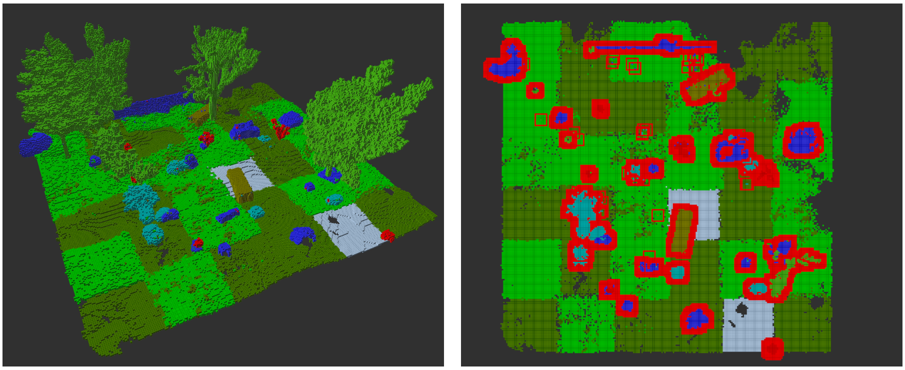

Fig. 2: Output of the 3D semantic Octomap (left) and of the navigation graph builder (right) on Sequence 1 of the 3DRMS dataset with voxel and node resolutions of 5 cm. Displayed colors correspond to the RGB values from the segmentation input images.

In order to evaluate the real-time capacities of the proposed mapping method, its average CPU usage, maximum RAM usage and pointcloud integration time have been measured while processing the 3DRMS challenge dataset. This evaluation was performed using an Intel Xeon(R) W-2123 8-core 3.60GHz CPU with 16 GB of RAM. Pointclouds and poses from the dataset sequence are produced at a fixed rate of 2 Hz with a total sequence duration is of 51.2 s. Figure 3 reports the computational cost of the mapping process with different voxel resolutions and Figure 4 displays the average integration time of a single pointcloud. This shows that the mapping process could be updated at a rate greater than 2 Hz for resolutions equal or greater than 2 cm. The real-time performances of the approach can then be guaranteed with resolutions that turn out to be sufficiently small for typical navigation tasks.

The mapping precision of the semantic Octomap method has also been evaluated with respect to the ground-truth labeled pointcloud, both in terms of geometry and labeling. The average distance between the center of each voxel of the produced Octomap and its closest point in the ground truth pointcloud has been computed for various voxel resolutions from 1 cm to 50 cm. The classification ratio has been calculated for each resolution as the total number of voxels of the Octomap whose closest point on the ground truth pointcloud is of same label over the total number of Octomap voxels. Figure 5 shows the mapping geometric precision and classification ratio for different voxel resolutions. The 3D mapping geometric precision and classification evaluation

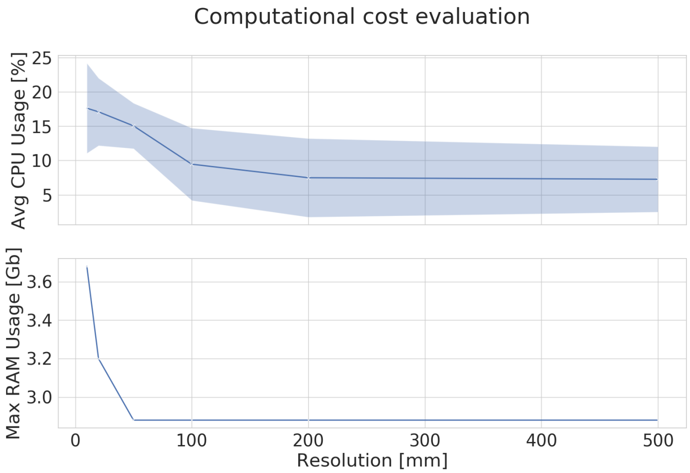

Fig. 3: Average CPU usage and maximum RAM usage by the mapping process on the 3DRMS dataset, for voxel resolutions of 1 cm, 2 cm, 5 cm, 10 cm, 20 cm and 50 cm.

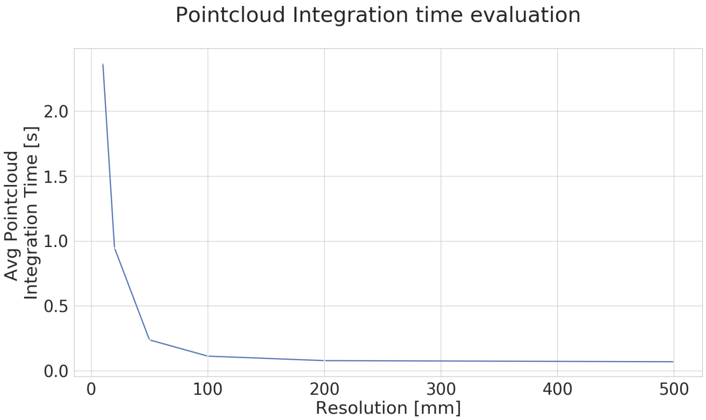

Fig. 4: Average time taken by the mapping process to integrate a single pointcloud from the 3DRMS dataset, for voxel resolutions of 1 cm, 2 cm, 5 cm, 10 cm, 20 cm and 50 cm.

shows satisfying results, with an average geometric error always inferior to the voxel resolution and more than 85% of voxels associated to the correct labels for all evaluated resolutions smaller than 50 cm. In particular for the 5-cm resolution, the average mapping error is equal to 2.93 cm, with a classification ratio of 96 %. Since the precision evaluation of the semantic mapping method demonstrates satisfying performances and in order to isolate errors that could be induced by the mapping process, the semantic voxel map obtained from the ground-truth labeled pointcloud with a 5-cm resolution has been used instead of the semantic Octomap output in the numerical evaluations of navigation and exploration methods (Section 4).

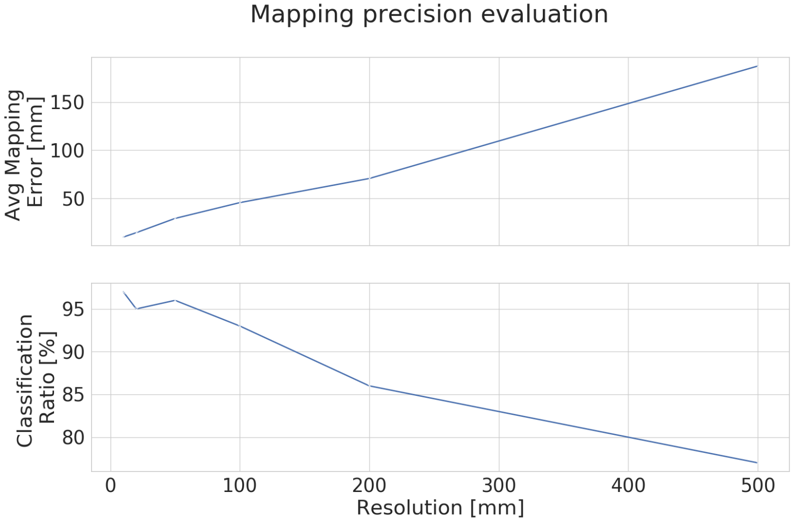

Fig. 5: Average geometric reconstruction error and classification ratio of the 3D semantic mapping process while running Sequence 1 of the 3DRMS dataset, calculated with voxel resolutions of 1 cm, 2 cm, 5 cm, 10 cm, 20 cm and 50 cm.

3.2.3 Navigation graph

Two steps of pre-processing are applied on the semantic map such that it could be used by the planning algorithms. First, in the case of a ground robot, a 2.5D representation is usually sufficient and also more suitable for traversability representation since it implicitly models the ground surface. This grid is computed by projecting the 3D map along the z axis and taking the label of the higher *non-free* voxel as the label of the 2D cell that does not exceed a threshold z_{th}. This threshold, above which the voxels are not taken into account, should be set to about the height of the robot. Taking the z_{th} threshold into account makes it possible to avoid, for example, the robot bypassing the projection of the branches of a tree if this class is considered non-traversable. Figure 6 illustrates the result of this pre-processing, where the left view shows the 3D voxelized map and the right view shows the projected navigation grid, each color representing a distinct label (see Table 1). The navigation grid is generated with the same resolution as the map voxels.

The second map pre-processing consists in the construction and the update of a graph during map integration. Because path planners usually rely on the construction of a graph (or a tree), we propose a method that builds it

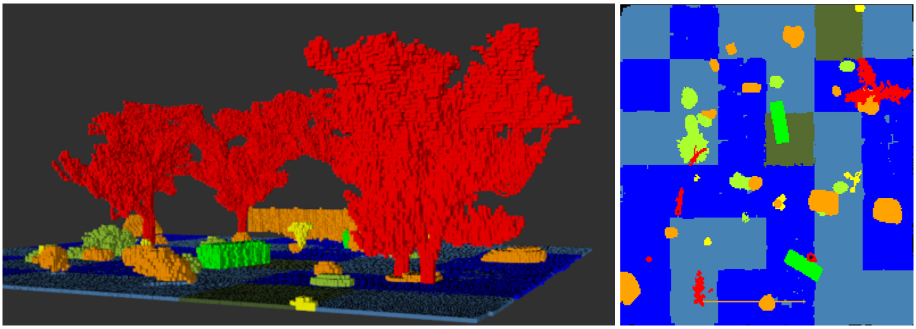

Fig. 6: Ground-truth voxelized 3D semantic map (left) and 2D projection for mobile robot path planning (right) for the 3DRMS dataset [17]. Colors represent semantic classes according to Table 1.

incrementally by updating only the cells that have been modified in the mapping process and their neighbors. This method projects the received points in 2D and allocates their labels, heights and (x, y) coordinates to nodes of corresponding positions into a 2D grid. These nodes are connected by storing pointers to their neighbor node(s), forming a navigable graph. The neighborhood considered depends on the kind of planner, for instance a weighted A* can use a 8-cell neighborhood whereas a RRT will extend the tree with one new sampled node. Figure 7 illustrates the described data structure. When nodes labeled as obstacle classes are integrated in the navigation graph, an extrusion step is performed for safe robot navigation. A square of size depending on a *robot radius* parameter is drawn around the obstacle and all nodes of that square are granted a corresponding *safety-zone* label, which is considered as an obstacle in the class table. Figure 8 summarizes this integration process.

Each node of the graph holds a label that can be mapped to a traversability coefficient and an interest coefficient. Therefore, we can compute the traversability cost from $\mathcal{M}_1$ and update the observation grid $\mathcal{M}_3$ from the node labels and positions in the 2D grid. The exploration grid $\mathcal{M}_2$ can be computed by extracting the graph frontier composed of the nodes for which at least one neighbor is unexplored. Overall, this navigation graph structure built online is one possible implementation of the multi-layered structure proposed in Section 3.1, and can be exploited by navigation and exploration planners as described in Sections 3.3 and 3.4.

3.3 Semantic-Aware Path Planning algorithms

Modified versions of A*, T-RRT and a shortcut procedure taking into account the proposed mapping structure are provided. Note that other graph-based or sampling-based path planning algorithms can be adapted in a similar fashion.

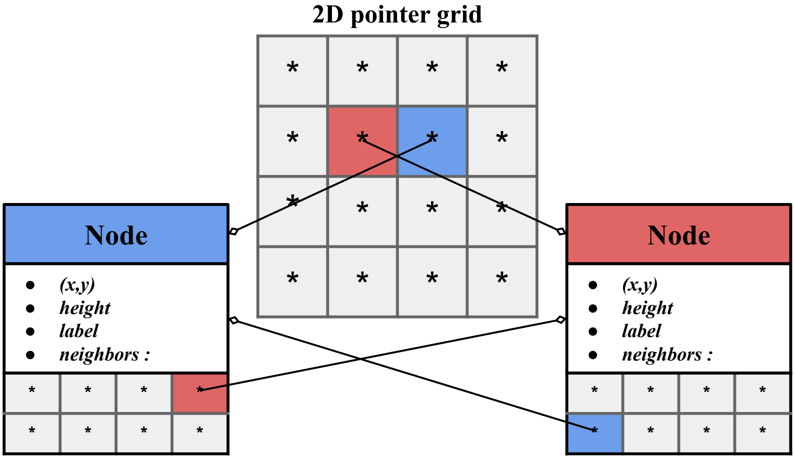

Fig. 7: Schematic representation of the navigation graph structure. Node pointers are stored in a 2D grid and each node contains its position (x, y), its height, its label and a set of pointers to neighbor nodes, for example its 8-neighborhood with an A* planner.

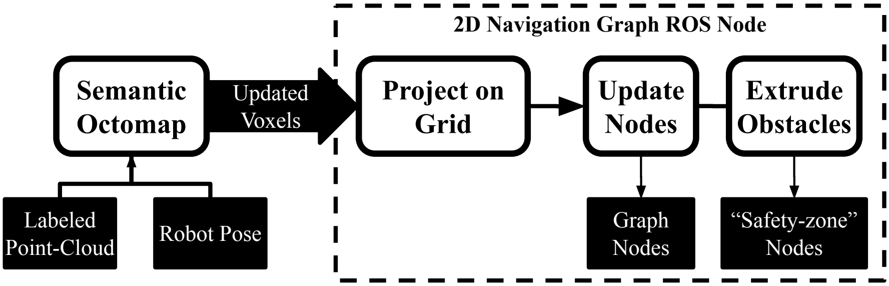

Fig. 8: Block diagram representing the 3D semantic mapping process from input data and the integration of mapping updates in the navigation graph.

3.3.1 Weighted A*

A weighted A* algorithm [40] is used to take into account the semantic-aware traversability layer $\mathcal{M}_1$ defined in Section 3.1 to compute a path P from a starting node q_0 to a destination q_{goal}, with an initially unknown path length (i.e., the number of intermediate points m), minimizing the cost function (2).

In the A* process, the transition function g from the current node q_{cur} (defined at the start as q_0 with value $g(q_0, \mathcal{M}_1) = 0$) to a candidate node q_{nei} among the eight neighbors is taken as the sum of the cumulated cost at q_{cur} and the distance between the corresponding positions ξ_{nei} and ξ_{cur} weighted by a cost corresponding to the semantic label of the node q_{nei}, as

$$g(q_{\text{nei}}, \mathcal{M}_1) = g(q_{\text{cur}}, \mathcal{M}_1) + c(q_{\text{nei}}, \mathcal{M}_1) \cdot \|\xi_{\text{nei}} - \xi_{\text{cur}}\| \tag{5}$$

The classical Euclidean distance heuristic for attraction to the goal q_{goal} is then applied without weight,

$$f(q_{\text{nei}}, q_{\text{goal}}, \mathcal{M}_1) = g(q_{\text{nei}}, \mathcal{M}_1) + \|\xi_{\text{goal}} - \xi_{\text{nei}}\| \tag{6}$$

Note that the cost $c(q_{\text{nei}}, \mathcal{M}_1)$ as defined in (3) is never less than 1, so the Euclidean distance $\|\xi_{\text{goal}} - \xi_{\text{nei}}\|$ is always lower than or equal to the actual optimal path cost, which guarantees the admissibility of the chosen heuristic [41].

3.3.2 T-RRT

Transition-based RRT (T-RRT) [32] is an extension of the Rapidly-Exploring Random Tree path finding algorithm which probabilizes the conservation of the new nodes of a tree, and thus the transitions of this tree, according to a costmap so as to favor the valleys and saddle points. The proposed semantic-aware version of this algorithm probabilizes the tree transitions depending on the cost function derived from the semantic map layer $\mathcal{M}_1$. The transition probability p from a new sampled node q_{new} to the nearest node included in the tree q_{near} of respective costs $c_{\text{new}} = c(q_{\text{new}}, \mathcal{M}_1)$ and $c_{\text{near}} = c(q_{\text{near}}, \mathcal{M}_1)$ is then taken as

$$p(q_{\text{new}}, q_{\text{near}}) = \begin{cases} 1 & \text{if } c_{\text{new}} < c_{\text{near}} \\ \exp\left(\dfrac{c_{\text{near}} - c_{\text{new}}}{T \cdot \|\xi_{\text{near}} - \xi_{\text{new}}\|}\right) & \text{otherwise} \end{cases} \tag{7}$$

where T is a temperature parameter. It can be noted that if q_{new} is not a traversable node (i.e. $c_{\text{new}} = +\infty$), then the transition probability p from q_{near} to q_{new} is equal to 0. In the implemented version, the *minExpandControl* function introduced in the classical formulation of the algorithm to maintain a minimal rate of expansion toward unexplored regions has been deactivated and replaced by a bias towards the target with uniform probability 5%, in which case the tree is not expanded towards a random node q_{rand} of the configuration space, but towards the target node q_{goal}.

3.3.3 Shortcutting

Paths generated by sampling-based algorithms usually require to be post-processed by a shortcut strategy [42] to remove intermediate nodes that were useful to explore the space but result in an increase of the overall length, which usually does not take into account the terrain traversability or any other related characteristic that could be obtained from semantic information. In the present context, nodes can be removed from a generated path P obtained either with the A* or the T-RRT strategy if two successive nodes of the path correspond to the same class label in the semantic layer (which also contains obstacle classes in our case). The paths are stored such that each node points to the next in the path, a node thus corresponds to a waypoint and the last node of the path q_{m-1} points to *NULL*. The proposed semantic-aware shortcut algorithm iterates on the path with three node pointers i, j_{prev} and j which initially point respectively to q_0, q_1 and q_2. Then, as long as there is no obstacle or label change between the nodes q_i and q_j (which is verified by a line iterator traversing $[q_i, q_j]$ in the semantic layer $\mathcal{M}_1$, then the pointers j and j_{prev} are moved forward to the next nodes of the path. When a label change is detected between i and j, then a shortcut is made between the nodes pointed by i and j_{prev} such that q_i points to $q_{j_{\text{prev}}}$ in the shortcutted path P_s. The procedure is then iterated until j points to the target node q_{m-1}.

3.4 Next-Best-View Exploration-and-Observation

An exploration task consisting in the observation of the entire bounded space by an autonomous robot using a sensor with a field of view (FOV) limited in detection angle and range (90 deg and 3 m in our experiments) has been studied as a case study including the computation of high-level goals and path planning from semantic information. The multi-layered map $\mathcal{M}$ defined in Section 3.1 is initially unknown and discovered during the task. A standard *Next-Best-View* (NBV) strategy [43, 44] has been defined, where the map layer $\mathcal{M}_2$ monitors the binary status (observed or not) of the cells of the environment. This layer is eroded with a circular structuring element larger than the radius of the robot to take into account its dimensions for obstacle avoidance. A complementary sub-task has been considered to observe objects belonging to a class of interest when they are detected during exploration (similar to the *exploration-and-observation* problem addressed in [45]), whose observation is monitored using the map layer $\mathcal{M}_3$ to store the ternary observation status defined in (1) based on this specific class (positions of the objects are updated when the corresponding cell has been observed). The NBV is primarily selected to observe the closest points of interest of the latter layer at a close distance in the robot FOV. When there is no remaining point of interest in $\mathcal{M}_3$ at a given instant, a random number of NBV candidates are sampled such that their position is on the *border* cells of the exploration layer $\mathcal{M}_2$ with a yaw reference given by the normal to the border (computed using a Sobel filter). The NBV is then chosen as the candidate with the maximum number of unobserved cells that could be seen in the FOV. After the NBV has been calculated, a path P between the current robot position and this NBV is computed using either the previously presented semantic-aware A* or the T-RRT algorithms followed by the semantic shortcut post-processing strategy, all relying on the map layer $\mathcal{M}_1$ from which the traversability costs are computed and with the additional constraint that unobserved areas (in map layer $\mathcal{M}_2$) are not traversable either.

4 Numerical experiments

Repeated simulations have been carried out to evaluate the strategies proposed in the previous section, based on the 2D projection of the ground-truth semantic grid defined in Section 3.2 from the 3DRMS reference semantized pointcloud, which contains 243×256 cells with a 5×5 cm^2 resolution. Table 1 summarizes the semantic classes, with the user-defined cost they have been assigned from the map layer $\mathcal{M}_1$ corresponding to traversability (3 classes with different allowed speeds, the other ones being non-traversable) and the observation interest which is used to fill the map layer $\mathcal{M}_3$, here on the single *flower* class. It could be noted that this class is non-traversable but is of interest for observation, which further motivates the use of the proposed multi-layered structure. Table 1 also lists the colors associated to the classes, which are used in all the figures displaying 2D or 3D views. The developed algorithms have been implemented in C++ and run within a Ubuntu 18.04 Virtual Machine with 4096MB RAM on a Intel Core i5 8th generation CPU.

Table 1: Semantic labels, display colors and costs for the 3DRMS challenge dataset

Label l_i	Label Name	Color	Cost c_i (from $\mathcal{M}_1$)	Visit interest (for $\mathcal{M}_3$)
l_1	grass	dark blue	1.0	0
l_2	ground	pale blue	2.0	0
l_3	paving	dark green	3.0	0
l_4	hedge	bright green	∞	0
l_5	topiary	light green	∞	0
l_6	flower	yellow	∞	1
l_7	stone	orange	∞	0
l_8	tree	red	∞	0

4.1 Path planning unitary tests

A first evaluation campaign was carried out to evaluate path planning performance with respect to the semantic-weighted cost function (2) and computation time. The weighted A* and T-RRT algorithms with shortcutting have been compared, along with a standard A* procedure (which does not use the semantic weight derived from $\mathcal{M}_1$) as a baseline. For this purpose, 100 pairs of sufficiently distant start and goal positions have been sampled in the free space of the map. As the distances between each pair of start and goal positions vary, the weighted path length has been normalized by the distance between the two positions of each pair (see Table 2). A first consistent result is that the semantic-aware weighted A* always obtains shorter weighted path lengths compared to the standard A* algorithm.

On the other hand, the weighted version takes on average more than twice as long to execute, and is faster in only 5% of cases. This is due to the fact that

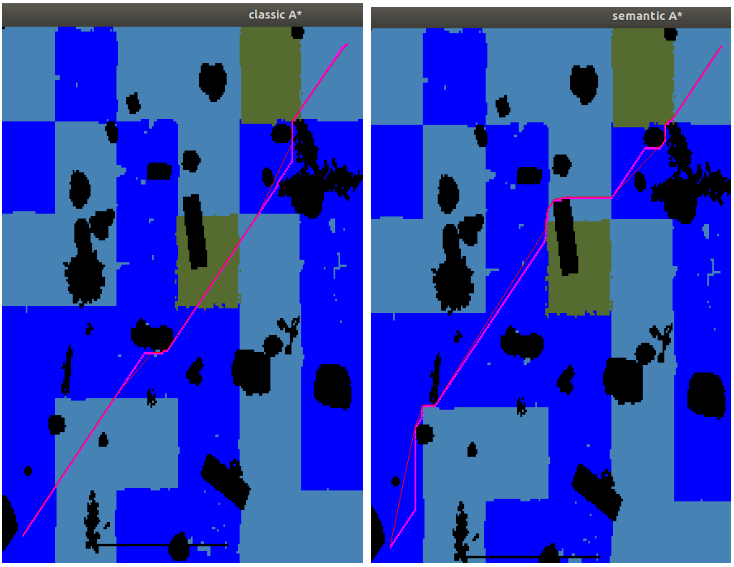

Fig. 9: Paths generated with classical (left) and semantic-aware (right) A*. Initial paths are displayed in pink and semantic-based shortcuts in red.

the paths obtained with a classical A* are shorter in terms of Euclidean distance, and thus fewer nodes of the navigation graph need to be evaluated before the algorithm reaches the target. Two respective paths generated by both algorithms are displayed in Figure 9. It can be seen that the less costly semantic class (dark blue) is favored by the weighted version. The T-RRT algorithm is on average faster to execute than the A* ones (in 82% of the runs), however the A* procedures provide shorter paths. This is a well-known trade-off when graph-based and sampling-based algorithms are compared on the same task, and this is also related to the relative simplicity of the test environment which presents many traversable areas. In larger 3D environments, the complexity of the graph-based A* strategies would induce more computational challenges and the T-RRT algorithm will probably be more robust to dimension increase.

Table 2: Path planners with semantic information. 100 runs, average (± std.)

Planning method	Path weighted length (normalized)	Computation Time (ms/pix)
Standard A*	1.70 (± 0.259)	0.0687 (± 3.31e-02)
Weighted A*	1.40 (± 0.140)	0.153 (± 5.91e-02)
T-RRT	2.02 (± 0.407)	0.0968 (± 0.191)

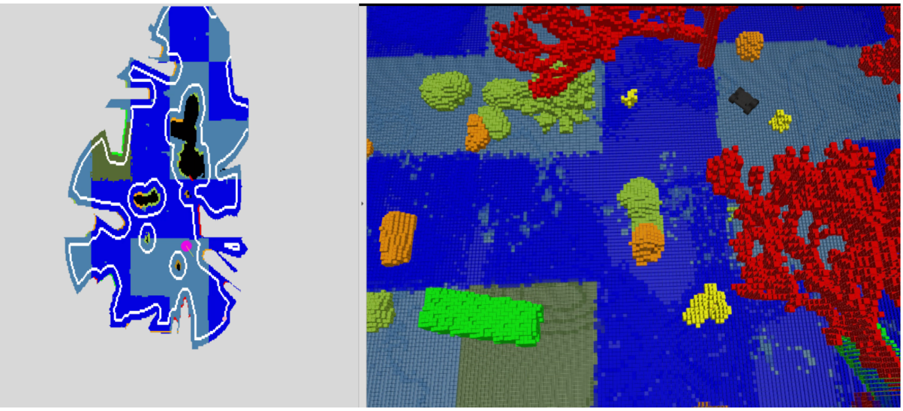

Fig. 10: Example of semantic-aware exploration and observation. Left: Real-time update of coverage layer (robot position in pink, frontier nodes in white). Right: Ground truth visualization of the semantic 3D voxel map from which information is extracted. Colors of semantic classes according to Table 1. Video available at https://tinyurl.com/SemanticPlanning.

4.2 Exploration-and-Observation Task

The NBV exploration-and-observation planner has then been evaluated in association with the above weighted A* and T-RRT procedures (Figure 10). Note that in the simulation, the robot speed depends on the traversability class label at its current position, according to the cost values listed in Table 1 and the inversely proportional rule defined by (3). The following two missions have been investigated: the case of a pure exploration of the map and the case of an exploration with observation of objects of interest (using layer $\mathcal{M}_3$), in this case those labeled l_6. One hundred simulations with random initialization have been performed for each path planning algorithm, half of which aimed at visiting the flowers in the semantic map as soon as they were detected. The exploration performance indicators correspond to the number of observed cells as a function of the number of algorithm iterations for both missions, and the additional number of iterations to visit all objects of interest for the observation task. Table 3 summarizes the performance indices obtained, while the cumulated performance indices during the missions are displayed in Figures 11 and 12. All the tasks have been successfully completed (with a stopping coverage criterion set to 90% of the free space) and the convergence curves are globally consistent. As a consequence of the unitary path planning tests, it follows that although the T-RRT planner is faster in calculating paths, exploration is more efficient in terms of number of iterations with the weighted A* algorithm because the computation time is compensated by the optimality of the path lengths. Indeed, the robot takes less time to reach the targets and this largely compensates for the slight additional path calculation time in this

2D evaluation case. The same trend is observed in the convergence rates to the maximum number of visited objects of interest.

Table 3: Performance indices for Exploration–Observation tasks. 50 runs, average (± std.)

NBV and Path planners	Weighted A*	T-RRT
Nb of iterations for 90% coverage, without visiting a class of interest	1381 (± 223)	1520 (± 438)
Nb of iterations for 90% coverage, while visiting a class of interest	1581 (± 321)	1682 (± 357)
Nb of iterations to visit all objects of interest	854 (± 361)	944 (± 401)

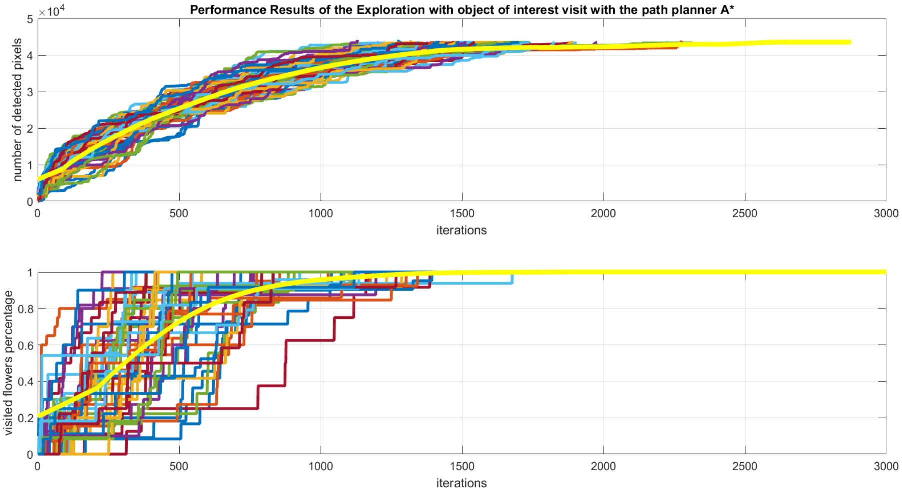

Fig. 11: Cumulated coverage and visit of classes of interest during the Exploration-and-Observation task with semantic-aware A* planner to reach NBVs (50 runs). The yellow curves correspond to the mean values.

5 Conclusions

An approach has been proposed in this paper to incorporate information available from semantic pointclouds and their registered poses into maps, which can be exploited for autonomous robot navigation tasks involving multiple planners with e.g. the definition of high-level goals followed by path planning. An example of construction of a multi-layered map structure has been proposed,

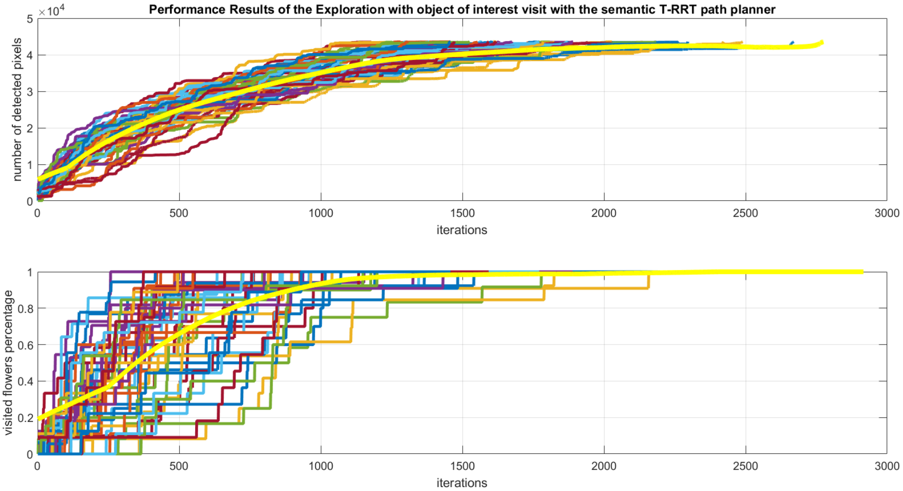

Fig. 12: Cumulated coverage and visit of classes of interest during the Exploration-and-Observation task with semantic-aware T-RRT planner to reach NBVs (50 runs). The yellow curves correspond to the mean values.

where semantic information can be used to derive cost functions or observation targets. It has then been shown how the classical A* and T-RRT planning algorithms can be adapted to handle semantic inputs.

A mapping process relying on semantic Octomap, the construction of a navigation graph and the evaluation of waypoint rallying and exploration-and-observation planning tasks have been carried out on the 3DRMS challenge dataset [17], which offered ground-truth semantic information as a reference. The promising results of this case study show that it is possible to run a full semantic-aware mapping-and-planning process for an autonomous robot.

The design and testing of such a process on board of mobile robots still require more investigation, including the training and integration of a semantic segmentation network to produce the labeled input pointclouds. The complete end-to-end pipeline allowing online robot navigation and exploration from embedded sensor data has yet to be implemented and thoroughly evaluated in increasingly challenging contexts. Studies concerning the best way to represent and store the map information and associated uncertainty should also be pursued in order to provide light, informative and reliable support for autonomous navigation. While the Octomap technique was employed for this purpose here, alternative environment representation methods have been developed and used in the recent years. The integration and comparative study of such methods in the context of the described architecture could lead to significant improvements for the task of semantic-aware robot navigation. The extension to multiple robots or heterogeneous teams, with the fusion of viewpoints to integrate semantic data into distributed 3D maps and their use by appropriate planning algorithms will also be considered in future work.

Conflict of Interest Statement

All authors state that there is no conflict of interest.

References

[1] He, K., Gkioxari, G., Dollar, P., Girshick, R.: Mask R-CNN. IEEE Transactions on Pattern Analysis and Machine Intelligence **42**(2), 386–397 (2020)

[2] Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 652–660 (2017)

[3] Qi, X., Wang, W., Liao, Z., Zhang, X., Yang, D., Wei, R.: Object semantic grid mapping with 2D LiDAR and RGB-D camera for domestic robot navigation. Applied Sciences **10**(17), 5782 (2020)

[4] Carvalho, M., Ferrera, M., Boulch, A., Moras, J., Le Saux, B., Trouvé-Peloux, P.: Technical Report: Co-learning of geometry and semantics for online 3D mapping. arXiv:1911.01082 (2019)

[5] Mascaro, R., Teixeira, L., Chli, M.: Diffuser: Multi-view 2D-to-3D label diffusion for semantic scene segmentation. In: IEEE International Conference on Robotics and Automation (ICRA) (2021)

[6] Nguyen, T., Shivakumar, S.S., Miller, I.D., Keller, J., Lee, E.S., Zhou, A., Özaslan, T., Loianno, G., Harwood, J.H., Wozencraft, J., Taylor, C.J., Kumar, V.: Mavnet: An effective semantic segmentation micro-network for MAV-based tasks. IEEE Robotics and Automation Letters **4**(4), 3908–3915 (2019)

[7] Bultmann, S., Quenzel, J., Behnke, S.: Real-time multi-modal semantic fusion on unmanned aerial vehicles. In: European Conference on Mobile Robots (ECMR) (2021)

[8] Jadidi, M.G., Gan, L., Parkison, S.A., Li, J., Eustice, R.M.: Gaussian processes semantic map representation. arXiv preprint arXiv:1707.01532 (2017)

[9] McCormac, J., Clark, R., Bloesch, M., Davison, A., Leutenegger, S.: Fusion++: Volumetric object-level SLAM. In: International Conference on 3D Vision (3DV), pp. 32–41 (2018)

[10] Rosinol, A., Abate, M., Chang, Y., Carlone, L.: Kimera: an open-source library for real-time metric-semantic localization and mapping. In: IEEE

International Conference on Robotics and Automation (ICRA), pp. 1689–1696 (2020)

[11] Grinvald, M., Tombari, F., Siegwart, R., Nieto, J.: TSDF++: A multi-object formulation for dynamic object tracking and reconstruction. In: International Conference on Robotics and Automation (ICRA) (2021)

[12] Hornung, A., Wurm, K.M., Bennewitz, M., Stachniss, C., Burgard, W.: OctoMap: An efficient probabilistic 3D mapping framework based on octrees. Autonomous Robots **34**(3), 189–206 (2013)

[13] Millane, A., Taylor, Z., Oleynikova, H., Nieto, J., Siegwart, R., Cadena, C.: C-blox: A scalable and consistent TSDF-based dense mapping approach. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, Madrid, Spain, pp. 995–1002 (2018)

[14] Kostavelis, I., Gasteratos, A.: Semantic mapping for mobile robotics tasks: A survey. Robotics and Autonomous Systems **66**, 86–103 (2015)

[15] Crespo, J., Castillo, J.C., Mozos, O.M., Barber, R.: Semantic information for robot navigation: A survey. Applied Sciences **10**(2), 497 (2020)

[16] Achat, S., Marzat, J., Moras, J.: Path planning incorporating semantic information for autonomous robot navigation. In: 19th International Conference on Informatics in Control, Automation and Robotics (ICINCO), Lisbon, Portugal, pp. 285–295 (2022). https://doi.org/10.5220/0011134300003271

[17] Tylecek, R., Sattler, T., Le, H.-A., Brox, T., Pollefeys, M., Fisher, R.B., Gevers, T.: The second workshop on 3D reconstruction meets semantics: Challenge results discussion. In: Proceedings of the European Conference on Computer Vision (ECCV) Workshops (2018)

[18] Xuan, Z., David, F.: Real-time voxel based 3D semantic mapping with a hand held RGB-D camera. https://github.com/floatlazer/semantic_slam (2018)

[19] Sofman, B., Lin, E., Bagnell, J.A., Cole, J., Vandapel, N., Stentz, A.: Improving robot navigation through self-supervised online learning. Journal of Field Robotics **23**(11-12), 1059–1075 (2006)

[20] Wang, M., Long, X., Chang, P., Padlr, T.: Autonomous robot navigation with rich information mapping in nuclear storage environments. In: IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR) (2018)

[21] Deeken, H., Puetz, S., Wiemann, T., Lingemann, K., Hertzberg, J.:

Integrating semantic information in navigational planning. In: 41st International Symposium on Robotics, pp. 1–8 (2014)

[22] Suriani, V., Kaszuba, S., Sabbella, S.R., Riccio, F., Nardi, D.: S-AVE: Semantic active vision exploration and mapping of indoor environments for mobile robots. In: European Conference on Mobile Robots (ECMR) (2021)

[23] Ono, M., Fuchs, T.J., Steffy, A., Maimone, M., Yen, J.: Risk-aware planetary rover operation: Autonomous terrain classification and path planning. In: IEEE Aerospace Conference, Big Sky, MT, USA, pp. 1–10 (2015)

[24] Chiodini, S., Torresin, L., Pertile, M., Debei, S.: Evaluation of 3D CNN semantic mapping for rover navigation. In: IEEE 7th International Workshop on Metrology for AeroSpace (MetroAeroSpace), pp. 32–36 (2020)

[25] Delmerico, J., Mueggler, E., Nitsch, J., Scaramuzza, D.: Active autonomous aerial exploration for ground robot path planning. In: IEEE Robotics and Automation Letters, vol. 2, pp. 664–671 (2017)

[26] Bartolomei, L., Teixeira, L., Chli, M.: Perception-aware path planning for UAVs using semantic segmentation. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5808–5815 (2020)

[27] Roggeman, H., Marzat, J., Bernard-Brunel, A., Le Besnerais, G.: Autonomous exploration with prediction of the quality of vision-based localization. IFAC-PapersOnLine **50**(1), 10274–10279 (2017)

[28] Ryll, M., Ware, J., Carter, J., Roy, N.: Semantic trajectory planning for long-distant unmanned aerial vehicle navigation in urban environments. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1551–1558 (2020)

[29] Sadat, A., Casas, S., Ren, M., Wu, X., Dhawan, P., Urtasun, R.: Perceive, predict, and plan: Safe motion planning through interpretable semantic representations. In: European Conference on Computer Vision (ECMR), pp. 414–430 (2020)

[30] Mozart, A., Moraes, G., Guidolini, R., Cardoso, V.B., Oliveira-Santos, T., de Souza, A.F., Badue, C.S.: Path planning in unstructured urban environments for self-driving cars. In: International Conference on Informatics in Control, Automation and Robotics (ICINCO) (2021)

[31] Maturana, D., Chou, P.-W., Uenoyama, M., Scherer, S.: Real-time semantic mapping for autonomous off-road navigation. In: Field and Service Robotics, pp. 335–350 (2018)

[32] Jaillet, L., Cortés, J., Siméon, T.: Sampling-based path planning on configuration-space costmaps. IEEE Transactions on Robotics **26**(4), 635–646 (2010)

[33] Lu, D.V., Hershberger, D., Smart, W.D.: Layered costmaps for context-sensitive navigation. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 709–715 (2014)

[34] Galindo, C., Saffiotti, A., Coradeschi, S., Buschka, P., Fernandez-Madrigal, J.-A., González, J.: Multi-hierarchical semantic maps for mobile robotics. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2278–2283 (2005)

[35] Grinvald, M., Furrer, F., Novkovic, T., Chung, J.J., Cadena, C., Siegwart, R., Nieto, J.: Volumetric Instance-Aware Semantic Mapping and 3D Object Discovery. IEEE Robotics and Automation Letters **4**(3), 3037–3044 (2019)

[36] Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional networks for biomedical image segmentation. Medical Image Computing and Computer-Assisted Intervention (MICCAI) (2015)

[37] Thomas, H., Qi, C.R., Deschaud, J.-E., Marcotegui, B., Goulette, F., Guibas, L.J.: Kpconv: Flexible and deformable convolution for point clouds. Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2019)

[38] Landrieu, L., Simonovsky, M.: Large-scale point cloud semantic segmentation with superpoint graphs. IEEE/CVF Conference on Computer Vision and Pattern Recognition, 4558–4567 (2018)

[39] Guiotte, F., Lefèvre, S., Corpetti, T.: Attribute filtering of urban point clouds using max-tree on voxel data. In: International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing, pp. 391–402 (2019)

[40] Ebendt, R., Drechsler, R.: Weighted A* search–unifying view and application. Artificial Intelligence **173**(14), 1310–1342 (2009)

[41] Dechter, R., Pearl, J.: Generalized best-first search strategies and the optimality of A*. Journal of the ACM **32**(3), 505–536 (1985)

[42] Campana, M., Lamiraux, F., Laumond, J.-P.: A gradient-based path optimization method for motion planning. Advanced Robotics **30**(17-18), 1126–1144 (2016)

[43] González-Banos, H.H., Latombe, J.-C.: Navigation strategies for exploring indoor environments. The International Journal of Robotics Research **21**(10-11), 829–848 (2002)

[44] Darmanin, R., Bugeja, M.: Autonomous exploration and mapping using a mobile robot running ROS. In: International Conference on Informatics in Control, Automation and Robotics (ICINCO), pp. 208–215 (2016)

[45] Okada, Y., Miura, J.: Exploration and observation planning for 3D indoor mapping. In: IEEE/SICE International Symposium on System Integration (SII), pp. 599–604 (2015)

Notation

q	Configuration of the robot in a given bounded space $\mathbb{S}$
ξ	Position state components of q in 2D or 3D
P	Path as an ordered list of configurations
Ω	Set of semantic labels l_i
$\mathcal{V}$	Subset of Ω to be observed during exploration
$\mathcal{M}$	Multi-layer map, with n_m aligned layers
$\mathcal{S}$	Semantic grid aligned with the map
$\mathcal{M}_i$	Map layer
c	Cost function
u	Utility function
$v_i^{\max}$	Maximum velocity in a cell with traversable label l_i