SMaNa: Semantic Mapping and Navigation Architecture for Autonomous Robots

Quentin Serdel[a], Julien Marzat[b] and Julien Moras[c]

DTIS, ONERA, Université Paris-Saclay, 91123 Palaiseau, France
{quentin.serdel, julien.marzat, julien.moras}@onera.fr

Keywords: Field Robots, Mapping, Navigation, Semantics

Abstract: Motivated by recent advances in machine learning applied to semantic segmentation, online 3D mapping is being extended to integrate semantic data. As these developments pave the way to the improvement of many robotic functionalities, the application of semantic mapping for navigation tasks remains to be further explored. In this paper we present an online Semantic Mapping and Navigation ROS architecture (SMaNa), with autonomous exploration as an application example. It is intended to be generic, so as to exploit state-of-the-art semantic mapping methods for unstructured environment and adapt them to perform jointly with a navigation graph builder and a semantic-aware A* path planner. The adequacy of multiple semantic mapping solutions for robot navigation in open environment and the performances of the architecture given the influence of localisation and semantic labelling uncertainty are evaluated in a closed-loop Ignition Gazebo simulation built from the 3DRMS synthetic dataset, and on the outdoor RELLIS-3D dataset.

1 Introduction

Exploration robots in hazardous environments (e.g. Martian ground or post-catastrophe urban area) require a high level of understanding of their surroundings in order to evolve efficiently and safely in an uncharted, potentially unstructured territory. Most existing functional architectures such as Au-Spot (Bouman et al., 2020) rely on geometric and proprioceptive data when performing navigation tasks. The incorporation of semantics into the mapping and navigation process of robots would allow them to reach unprecedented adequacy and autonomy for these tasks. Latest progress in machine learning applied to online semantic segmentation of images and point clouds (He et al., 2020; Lambert et al., 2020; Qi et al., 2017) allowed for the development of semantic mapping methods such as (Grinvald et al., 2019; Grinvald et al., 2021; Xuan and David, 2018; Rosinol et al., 2020) but their application to autonomous robot navigation remains to be pursued. Moving forward into real-world applications of semantic robotic exploration, semantic-aware navigation must be tackled jointly with semantic mapping.

This paper presents a Semantic Mapping and Navigation architecture (SMaNa) for ground exploration robots in unstructured environment, with a focus on the link between these two functionalities and the associated combined performances in particular under localisation and classification uncertainty. SMaNa has been built to be able to integrate semantic mapping solutions in a generic way. We present the application of two online 3D mapping methods: Octomap and TSDF, previously adapted to include semantics respectively in (Xuan and David, 2018) and (Rosinol et al., 2020). A semantic navigation graph builder which can adapt to the output of each mapping process and a weighted A* path planner allowing online exploitation of the scene semantics for navigation tasks are then introduced. [1]

The evaluation of the proposed combined semantic mapping and navigation system has been performed in closed-loop simulation, using a Ignition Gazebo simulated environment generated from the 3DRMS synthetic dataset (Tylecek et al., 2019), as well as on real-world outdoor trajectories in the RELLIS-3D dataset (Jiang et al., 2021). The proposed architecture was used to compare the two aforementioned state-of-the-art mapping solutions in order

[a] https://orcid.org/0009-0006-7612-2663
[b] https://orcid.org/0000-0002-5041-272X
[c] https://orcid.org/0000-0003-2959-7544

---

[1]A video presentation of SMaNA is available at https://tinyurl.com/SMaNaONERA

to select the most suitable one for the purposes of semantically-aware robotic navigation in the presence of noisy data and uncertain estimates and to evaluate their respective computational cost. As an application example, this architecture has also been extended to carry out an autonomous exploration task, and the benefits of using such a semantic-aware mapping and navigation system have been quantified in the case of terrains with different traversability constraints.

2 Related Work

The breakthrough of machine learning applied to semantic segmentation allowed to enrich the previously geometric-only robotic mapping process with the integration of semantic information. The problem of efficient semantic data integration to incrementally build maps has been tackled by many authors (Grinvald et al., 2019; Grinvald et al., 2021; McCormac et al., 2017; Rosinol et al., 2020; Martins et al., 2020; Ewen et al., 2022). It has already been demonstrated that the incorporation of semantic data to robotic mapping provides significant improvement for the application of SLAM (Xuan and David, 2018; Cheng et al., 2022). A first survey of semantic map representations for robot navigation has been presented in (Crespo et al., 2020). Most of the presented method categories use high-level or topological semantic representations, resulting in a significant improvement for robot task performance, human request satisfaction and navigation in indoor applications (Hughes et al., 2022; Sun et al., 2019; Wang et al., 2021). However these approaches hardly apply to outdoor unstructured environments, since the labels should be used at a lower level within a dense semantic representation rather than focusing on particular instances. Most of the developed dense semantic mapping methods have focused on the efficient and precise integration into the dense map, using probabilistic fusion (Rosinol et al., 2020; Xuan and David, 2018) and association between depth and semantic segmentation (Grinvald et al., 2019; Grinvald et al., 2021) for labelling. These methods are claimed to be fitted for robot navigation and exploration however, to the best of our knowledge, the integration of these semantic mapping approaches in a robot planning and navigation pipeline has been little studied so far. Semantic-aware navigation methods in unstructured environment have been proposed lately in (Maturana, 2022; Seymour et al., 2021; Bartolomei et al., 2020). (Achat et al., 2022) presented the development of a robot semantic-aware path planning method relying on a ground truth semantic 3D

grid, so as to perform next-best-view exploration and observation as well as terrain classification for safe navigation. However, few studies have tackled the extension of the navigation process to implement a practical and efficient interaction with semantic mapping. A semantic mapping and navigation method focused on terrain traversability for the safe travelling of a rover on Martian ground has been proposed in (Ono et al., 2015), but this work remains specifically designed for the rover locomotion and part of the process if performed offline due to high computational needs. The update of uncertain semantic maps using noisy observations has been studied in (Kantaros et al., 2022) with a focus on collaborative tasks. In (Maturana, 2022), a joint dense semantic mapping which relies on prior available knowledge associated to a view-point selection method has been presented to carry out the task of car inspection in outdoor environment by a Micro-Aerial Vehicle. In this paper, we extend and evaluate the joint conception of semantic mapping and navigation methods to a wider outdoor unstructured context with a practical focus on online capacities and quantification of robustness to localisation and labelling uncertainty.

3 SMaNa Architecture Description

The ROS-based SMaNa architecture (illustrated in Figure 1) is composed of three main building blocks: a 3D semantic mapping process (either Semantic-Octomap or Kimera-TSDF in this work), a 2.5D navigation graph builder and a semantic-aware A* path planner.

3.1 Input Data

The SMaNa architecture requires the robot localisation to be provided, typically given by a SLAM algorithm, along with a labelled point cloud. The latter can be produced using a stereo or RGB-D camera, or with a LiDAR coupled with a RGB camera. The point cloud labelling can be achieved through the application of a semantic segmentation Convolutional Neural Network (CNN) on the camera images as in (He et al., 2020; Lambert et al., 2020; Takikawa et al., 2019), which is then geometrically projected onto the point cloud. Other types of Neural Networks such as PointNet (Qi et al., 2017) directly work on point clouds to produce semantic segmentation. The application targeted is mainly the safe navigation in unstructured and uncharted environments, therefore segmentation networks specialised in terrain classification (Zürn et al., 2020; Ewen et al., 2022) could
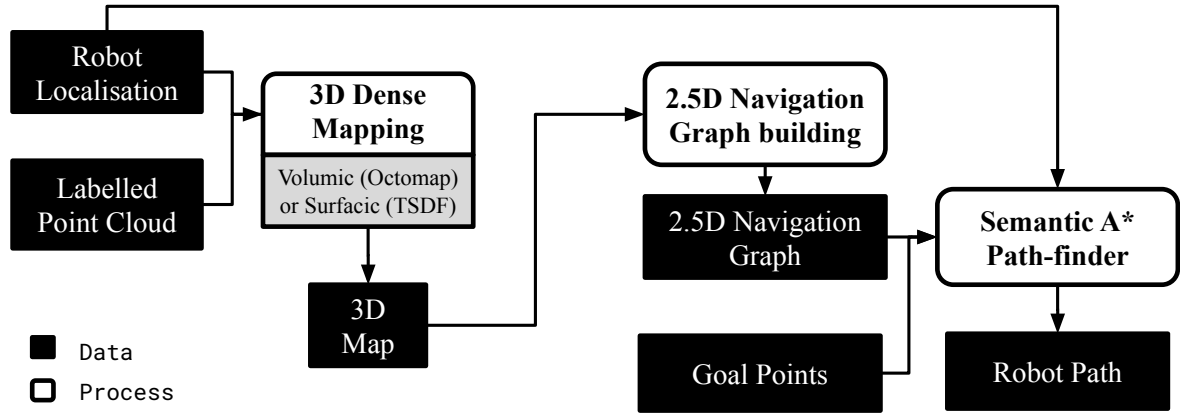
Figure 1: Overview of the SMaNa architecture.

be exploited to produce the required input. Dynamic objects in the robot environment (e.g. pedestrians or moving cars) should be identified using methods such as (Chen et al., 2021) and are assumed to be treated separately from the static scene point cloud before the latter is integrated into the 3D map.

3.2 Semantic Mapping

The architecture can integrate any 3D semantic mapping method compatible with the defined inputs. Two widely used methods, considered as references in the robot mapping domain with available implementations incorporating semantics have been selected for integration in SMaNA and for comparative evaluation. This section briefly describes those methods and their implementations.

3.2.1 Semantic Octomap

Semantic Octomap (Xuan and David, 2018) is an online ROS localisation and 3D semantic mapping architecture. In our context, only the mapping stack is used. The mapping is performed thanks to the use of the well-known Octomap package (Hornung et al., 2013), which provides an octree-based voxel volumic mapping representation. The hierarchical octree structure allows fast queries in $O(log\ n)$ with n being the number of octree nodes, and low memory consumption, making it adequate to online embedded applications such as robot exploration. Each point from the input cloud is associated to a leaf voxel based on its corresponding position. In classical Octomap, probabilistic sensor fusion is used to update the occupancy of voxels from multiple observations. In its semantic extension, additional information related to labels and their confidence is stored and fused. *Semantic Octomap* implements two different strategies

for handling labels, namely *max* and *Bayesian*. The *max* method has been applied in this paper as it requires less complex data transport. It stores a label and an associated confidence for each voxel. The label of a voxel is updated if a point of different label and higher confidence falls into the voxel. Voxel confidence is increased for each point of same label and decreased for each point of a different label with a lower confidence. On the other hand, the *Bayesian* strategy uses the full semantic likelihood input and performs the merge using the Bayesian fusion rule. Since this approach would require to store for each voxel the probability of each class, an approximation is used in *Semantic Octomap* where only the 3 most likely classes are stored and the other classes are assumed to be evenly distributed. For each semantic cloud processed, the Semantic Octomap algorithm has been adapted to produce a sparse point cloud containing the positions and labels of all updated voxels, which is sent to the navigation stack.

3.2.2 Kimera Semantics

Kimera Semantics (Rosinol et al., 2020) also performs online localisation and 3D semantic mapping in a ROS environment, relying on Voxblox (Oleynikova et al., 2017) for mapping. Voxblox is a TSDF surfacic (Nießner et al., 2013) mapping package relying on voxel hashing and marching cubes. When a semantic point cloud is produced, ray casting is performed between the sensor position and each point of the cloud. The distance to the point is computed and allocated to voxels along the ray up to a threshold fixed to 2 times the voxel resolution before and after the point. The point label is allocated to those voxels. Voxels are then gathered in blocks that are hashed and stored with their position in the TSDF map on which queries can be made in $O(1)$. A surface mesh is

computed using marching cubes on the TSDF voxels. The *fast* TSDF mapping method, found to be faster without significant precision loss in preliminary tests is used here due to real-time constraints. It discards all rays from a point cloud which attempt to update a voxel that has already been updated by a ray from the same cloud. In the extended semantic implementation, each TSDF voxel stores a vector of label probabilities, updated according to the number of points with associated labels integrated to the voxel following a Bayesian approach. At a given instant, the label of highest probability is selected as the output voxel label. For each point cloud processed, the TSDF map is produced and sent to the navigation stack.

3.3 Semantic Navigation

The SMaNa navigation stack consists of an online 2.5D semantic navigation graph builder and a weighted A* path-finder. The graph builder collects data from the mapping stack and integrates it into an exploitable structure for navigation. The weighted A* algorithm performs optimal path-finding inside the 2.5D navigation graph layer.

3.3.1 Navigation Graph

A 2.5D graph layer is built online, integrating new elements from the 3D map to create and update nodes. It is set to receive 3D mapping data, either from Octomap as a point cloud or Kimera as a TSDF map. Each point (representing an updated voxel) of the Octomap cloud message is associated to a node position in the navigation grid. The received TSDF map from Kimera is deserialised and each voxel whose absolute distance to its surface is smaller than the node resolution is associated to a node position in the grid.

After this projection step which is specific to each type of mapping input, the remaining treatment is generic. A fixed maximum size of the graph is given as starting parameter. To ensure efficient memory consumption, nodes inside the graph boundaries are allocated only when data from the 3D map at the same position becomes available. Graph nodes contain a class label, a height and their absolute coordinates as well as pointers to other nodes for the use of a planning algorithm. Each received voxel is associated to the graph node of corresponding (x, y) coordinates. If the node is uninitialised, the label l and height z of the first associated voxel are allocated to it. Else, the node's l and z values are replaced by the label and height of the associated voxel of highest altitude (a specific threshold could be used instead for particular needs). Some non-traversable labels (e.g., those displayed in dark gray in Table 1) are considered as
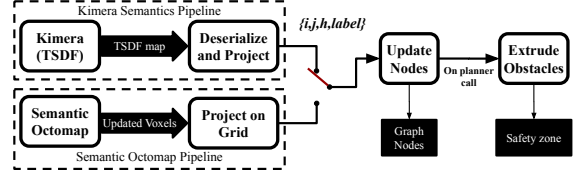

Figure 2: SMaNa navigation graph building. Either Kimera or Semantic Octomap can be used for mapping in the current implementation.

obstacles, as a consequence the nodes associated to these labels are marked as unselectable for the planning algorithm. Each node then contains a *safety zone* Boolean variable which is set to *true* for nodes located around obstacle nodes as far as a given *robot_radius* parameter, marking them as non-traversable for the planning algorithm. Obstacle nodes are stored in a dynamic vector and the calculation of the safety zone is performed for each of them when the path planner is called to use the most recent fused label integration. Figure 2 illustrates the building and update of the graph from either one of the mapping methods. This process has been designed to be extensible to other mapping methods, as long as their output can be handled similarly.

3.3.2 Weighted A* Path planning

The path finding inside the 2.5D navigation graph is solved using a weighted A* algorithm, inspired by the work presented in (Achat et al., 2022). All navigable classes of the environment are associated to a traversability coefficient, which is used to penalise nodes in the A* algorithm. This process allows the robot to choose and follow a safer and more efficient path regarding the nature of its surrounding terrain. The height difference between two neighboring nodes is also included in the A* cost calculation to prevent the robot from crossing too steep slopes. The set of already evaluated nodes is kept in memory as the union of the closed set C and the free set F which contains the non-obstacle nodes neighboring the closed set. It is initialised with a closed set containing the starting node and a free set containing its neighbors. At each iteration, the A* algorithm calculates or updates the weights of the nodes newly added to the free set. The weight of the nodes is not pre-calculated when the navigation graph is built, preventing unnecessary calculation for nodes that might never be visited. The graph holds an 8-connectedness between neighboring nodes. The transition cost for each node n from its parent p is calculated as:

$$c(p \rightarrow n) = d(p,n) + h_f.|h(n) - h(p)| + t_f.T(l) \quad (1)$$

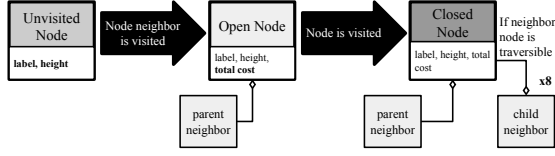with $d(n,m)$ the distance between two nodes, $h(n)$ the node's height, $T(l)$ the cost associated to the node la-

Figure 3: Navigation graph node visit when performing weighted A*

bel l. The weights h_f and t_f should be chosen by the user, in this work they were respectively fixed to 0.1 and 0.8. The total cost $T_c(n)$ of the node for the A* algorithm to select the next best candidate is calculated as follows:

$$T_c(n) = T_c(p) + c(p \rightarrow n) + d(n,g) \qquad (2)$$

with g the goal node of the A* path finder. Algorithm 1 details the process of this weighted A* path-finding, with $\mathcal{N}(n)$ being the set of neighbors of node n and $p(n)$ being the parent of node n. Figure 3 illustrates the processing of a single node during A* path-finding.

Data: 2.5D navigation grid with nodes:
$n\{l,h,p,\mathcal{N}(n)\}$, Start node s, Goal
node g, Set of closed nodes $C\{s\}$, Set
of free nodes $F\{\mathcal{N}(s)\}$, A* path P
while $g \notin C$ *or* $F \neq \emptyset$ **do**
    $C \leftarrow n \in F \mid T_c(n) = \min(T_c(F))$;
    for *each node nb neighboring n* **do**
        if *nb is traversable* **then**
            $\mathcal{N}(n) \leftarrow nb$;
            calculate transition cost
            $c(n \rightarrow nb)$ (1);
            calculate total A* cost $T_c(nb)$ (2);
            $p(nb) = n$;
            $F \leftarrow nb$;
        end
    end
end
if $g \in C$ **then**
    $n = g$;
    while $p(n) \neq s$ **do**
        $P \leftarrow n$;
        $n = p(n)$;
    end
    $P \leftarrow s$;
    return P
end

Algorithm 1: SMaNa weighted A* algorithm

4 Experiments in Simulation

SMaNa has been applied and evaluated in terms of mapping precision under various levels of noise on input localisation and semantic segmentation. The Octomap and TSDF methods precision were both evaluated in a new Ignition Gazebo simulated environment built from the 3DRMS dataset ground truth point cloud (Tylecek et al., 2019) shown in Figure 4. An application to autonomous exploration has been performed within the same environment in order to highlight the interest and applicability of this semantic-aware mapping and navigation system. In order to assess the performances of this architecture in an online embedded context, its RAM and CPU consumptions and the average integration time of a point cloud to the map and graph have been evaluated using the RELLIS-3D outdoor dataset (Jiang et al., 2021). The evaluation was performed on a Intel Xeon(R) W-2123 8 core 3.60GHz CPU with 16 GB of RAM, with no GPU required. Qualitative results of navigation graphs as well as examples of semantic-aware weighted A* paths are also provided.

4.1 Simulation description

Our main objective here is to evaluate the performance of the SMaNa architecture on the semantic mapping capability but also on its ability for mobile robot navigation in a representative unstructured environment. This requires what we call a *closed-loop simulation* environment where a semantic sensor view can be generated from the current robot pose, and a ground truth semantic 3D model (mesh or point cloud) can be available to compute relevant metrics.

On the one hand, there are many semantic datasets available for different purposes, e.g. RELLIS-3D (Jiang et al., 2021) or the 3DRMS challenge (Tylecek et al., 2019) in off-road environments, SemanticKitti (Behley et al., 2019) for urban navigation, SceneNN (Hua et al., 2016) in an indoor office environment. However, there is usually no 3D ground truth provided with these datasets, which strongly limits their usage for our purpose. An exception is the 3DRMS-challenge synthetic dataset which contains a ground truth semantic point cloud, it has moreover been exploited to evaluate semantic-aware navigation in (Achat et al., 2022). On the other hand, there are also few simulation environments with built-in semantic-world representations for the targeted autonomous mobile robot application with traversability considerations and mapping evaluation. The TESSE simulator (Ravichandran et al., 2020) is probably the closest related work including a com-

bination of metric-semantic data and physics-based simulation, however only an indoor office environment with object-oriented tasks has been made available in the ICRA 2020 GOSEEK challenge[2] without direct access to the 3D ground truth. The CARLA simulator (Dosovitskiy et al., 2017) is dedicated to the evaluation of autonomous driving systems in urban environments and can generate semantized sensor views but no 3D ground truth. The Micro-Air vehicle Flightmare simulator (Song et al., 2021) is also able to simulate RGB cameras with ground truth depth and semantic segmentation, however the 3D ground truth model only provides occupancy and not semantics. Moreover, neither of these simulators are natively dedicated to field robotic navigation, and so they are not easily applicable to consider traversability constraints.

We therefore propose a new simulation setup with all the required features, which will be contributed to the community. It is based on the openly available Ignition Gazebo engine[3], which offers the possibility to render RGB, depth and semantic images, natively integrates a physics engine, and can interact easily with the ROS-based infrastructure. Following the above dataset and simulator review, we selected the 3DRMS dataset to build the test environment. This dataset was initially proposed to evaluate semantic 3D reconstruction in the context of robot navigation in an outdoor environment of around $12\ m \times 12\ m$. The semantic is described using 8 classes, given in Table 1. The ground truth semantic point cloud was used to reconstruct meshes to populate the world (and it is thus also available to evaluate the mapping). Points of each class were split to build one mesh per class and then added to a SDF world file with a specific label. Ground and objects have been processed using two different pipelines. The objects have been reconstructed using voxelization with a resolution of 5 cm whereas the ground has been reconstructed using the Poisson reconstruction algorithm (Kazhdan et al., 2006) to obtain a smooth surface compatible with robot wheel motion. To perform this reconstruction, a pre-filtering was applied on the point cloud to remove superimposed ground layers. Finally, boundary walls were added to maintain the robot into the world. The robot model considered is the Wifibot wheeled differential-drive robot equipped with a front sensor providing RGB, depth and semantic segmentation in undistorted, rectified images with a resolution of 640x640 at a rate of 30 Hz with a field-of-view of 57°. The simulator also provides the ground truth pose of the robot base link.

Table 1: Colors and traversability costs associated to semantic classes simulated in Ignition Gazebo from the 3DRMS dataset

class	unknown	grass	terrain	hedge	topiary	rose	obstacle	tree
color								
cost		2	1					

4.2 Mapping Robustness to Noisy Inputs

4.2.1 Production of Degraded data

Semantic information produced by segmentation neural networks are not perfect and generate some classification errors. This must be considered when evaluating mapping algorithms to be realistic. However, using NN-based algorithms in simulation is challenging and presents a high risk of giving unrealistic results. Different segmentation approaches may also behave very differently so the result could be dependent on one particular algorithm. This is why we proposed instead to simulate the semantic segmentation errors to overcome these biases.

The ground truth robot localisation and semantic segmentation of the robot camera images have been artificially degraded in order to evaluate the robustness of the mapping algorithms to noisy inputs. Gaussian noises of parameterised variance have been added to the ground truth robot position and yaw angle. To degrade the ground truth semantic segmentation images, a superpixel segmentation using the SLIC method (Achanta et al., 2012) has first been applied to each of them. Each superpixel is associated to the most frequent semantic label of the pixels within its borders. To simulate the misclassification that can be typically produced by a semantic segmentation network, the label of each superpixel has a probability of being changed according to a confusion matrix (similar in spirit to (Krstinić et al., 2020)) defined in Table 2. Four different levels of misclassification probabilities, chosen to be comparable to the RELLIS-3D semantic segmentation benchmark results, have been set for evaluation. A random classification confidence between 0.33 and 1.0 was associated to each pixel (see Figure 5).

4.2.2 Mapping Precision Evaluation

The precision of both Semantic Octomap and fast Semantic TSDF mapping methods from noisy inputs has been evaluated with the robot following an arbitrary 44 m long trajectory in the simulated environment. The mapping resolution for both methods has been fixed to 5 cm. Four noise profiles were added to the
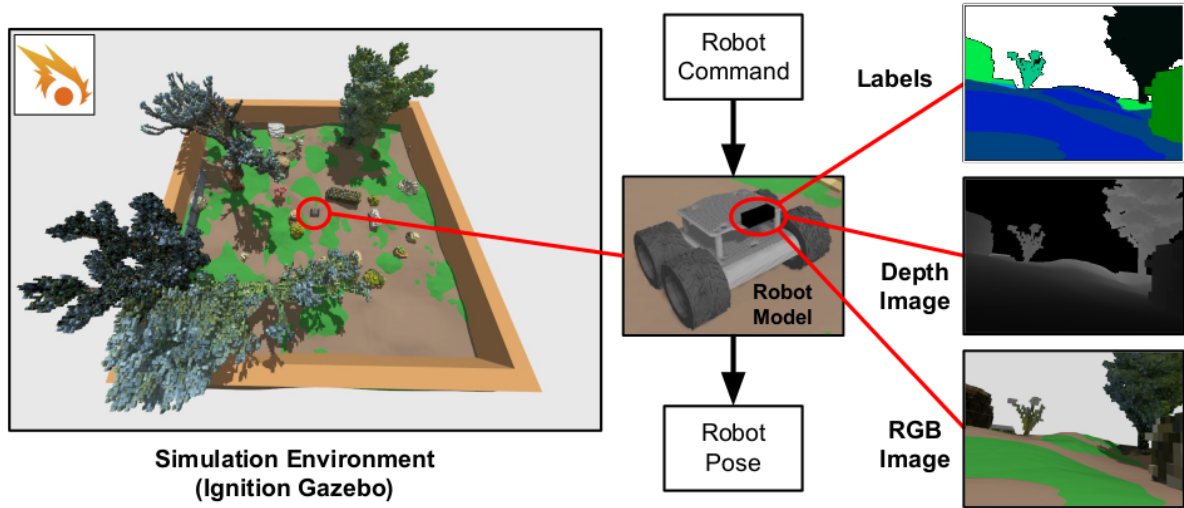
Figure 4: Ignition Gazebo simulation environment based on a ground truth point cloud from the 3DRMS dataset and a ground robot model. The robot is equipped with a differential drive module taking a velocity command as input and outputting the robot pose along with a simulated camera producing RGB images, depth maps and semantic segmentation images.
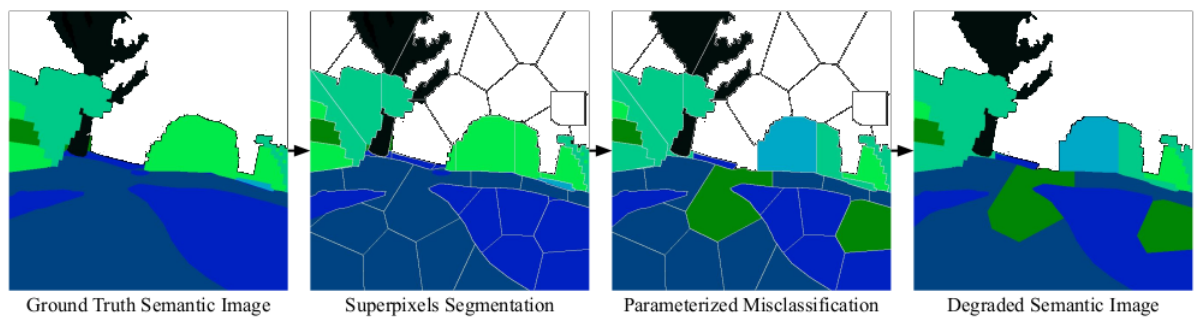

Figure 5: Example of the degradation of a ground truth semantic image taken from the simulation using the proposed SLIC superpixel confusion method.

Table 2: Confusion matrix for superpixels misclassification probabilities, the values of p for the 4 level of noise (*none, low, medium, high*) are respectively 1.0, 0.9, 0.8 and 0.7.

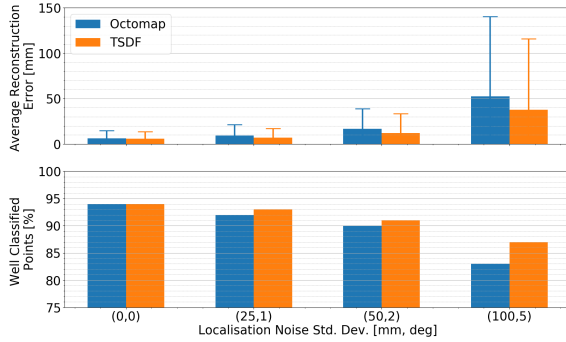|	0	1	2	4	5	6	7	8
0	1	0	0	0	0	0	0	0
1	0	**p**	(1-p)/2	(1-p)/2	0	0	0	0
2	0	(1-p)/2	**p**	(1-p)/2	0	0	0	0
4	0	0	0	**p**	(1-p)/2	(1-p)/2	0	0
5	0	0	0	(1-p)/2	**p**	(1-p)/2	0	0
6	0	0	0	0	(1-p)/2	**p**	(1-p)/2	0
7	0	0	0	0	0	(1-p)/2	**p**	(1-p)/2
8	0	0	0	0	0	(1-p)/2	(1-p)/2	**p**

Figure 6: Average reconstruction error (with std dev) and classification ratio with both mapping methods and different levels of localisation degradation.
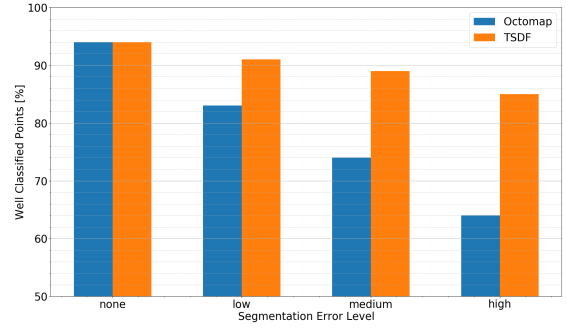

Figure 7: Classification ratio of maps produced by both mapping methods with different levels of image semantic segmentation degradation.
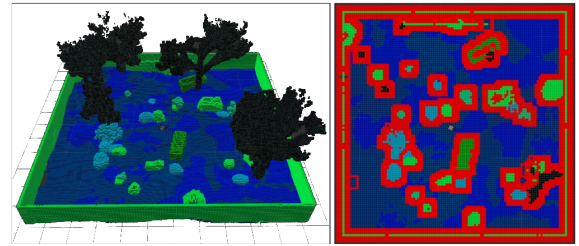

Figure 8: Complete reconstructed semantic Octomap of the simulation environment with a voxel resolution of 5 cm (left) and associated navigation graph of 10 cm resolution (right).

robot localisation and the average absolute distance between the points of the produced maps and a ground truth point cloud generated from the simulation environment 3D mesh have been computed. The good classification ratio has been computed as the number of points from each produced map of same semantic label as the closest point of the ground truth point cloud over the total number of points (see Figure 6). Four different confusion matrices with increasing misclassification probabilities were also defined (see Table 2). Figure 7 presents the influence of misclassification of input images to the classification ratio for both methods. These results demonstrate that the *fast* TSDF mapping method is more robust to robot localisation error than the Octomap. Moreover, the TSDF semantic label integration method also shows significantly better robustness to misclassification than the *max* label integration method of the semantic Octomap (11 % loss between the lowest and highest levels for TSDF, against 30 % for Octomap).

4.3 Application to Autonomous Exploration

As an illustrative real-time use-case for the SMaNa architecture in the proposed simulator, the architecture has been connected to an exploration stack, presented in Figure 9. The latter incorporates a mobile robot controller and the *next-best-view* frontier exploration strategy from (Achat et al., 2022). It samples views from the graph border of traversable space and grants them a score based on the intersection of the unexplored area and a mask representing the sensor field of view. Here, it has been adapted to penalise view scores by the cost of the A* path between the robot position and each candidate view.

A robot exploring an unstructured environment might face different issues when travelling in some types of terrain (e.g., generating noisy data due to vibrations, getting a wheel stuck, etc.), which is the main motivation here for semantic-aware exploration. Given the choice of traversable classes (Table 1), only 48 % of the simulated environment surface is free, which represents a very cluttered and challenging setting for autonomous navigation. 55 % of the traversable surface is composed of *terrain* nodes and the other 45 % of *grass*, which is the class to be avoided by the robot here. To evaluate the impact of the proposed strategy, we calculated the percentage of *grass* nodes over the total of nodes travelled by the robot during the environment exploration with and without taking into account the traversability cost. This evaluation shows that the robot travels at 60 % on *terrain* when the traversability of labels is ignored against 70 % when taking it into account with a dou-
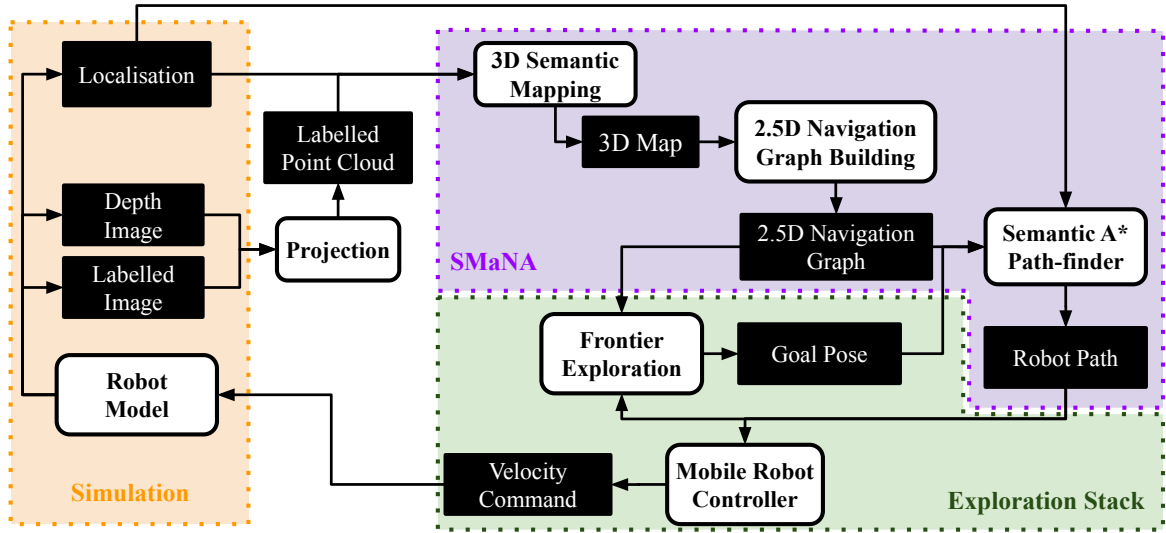
Figure 9: Overview of the SMaNa architecture integration in an exploration scenario, connected to the inputs and outputs of the contributed Ignition Gazebo simulation.

bled cost on the *grass* class and 88 % with a quadrupled cost. The traversability cost on the different classes can thus be tuned to set a trade-off between exploration speed and safety. Figure 8 displays the 3D semantic map and 2.5D navigation graph resulting from the complete exploration of the simulation environment with SMaNa and the added exploration module.

5 Experiments with real data

In order to progress toward the application of the SMaNa architecture to real-world scenarios, we measured its computational and memory footprint and verified that the generated navigation graph allows safe navigation. The RELLIS-3D dataset was selected for this purpose, as it contains semantically labelled data taken from a LiDAR and cameras onboard of a ground robot in an outdoor unstructured environment.

The mapping and navigation graph building processes consumption were evaluated during the application of SMaNa data integration on LiDAR scans produced at 10 Hz during the complete 176 s RELLIS-3D *00000* sequence and augmented with their ground truth semantic labels. The total RAM and average CPU consumptions were measured as well as the average integration time per point of a Li-DAR scan in the 3D map and in the navigation graph. The maximum integration distance for all mapping methods was set to 10 m. The Octomap ray-casting distance was set to 1 m and the *voxel_per_side* TSDF parameter was set to 16. Figure 10 shows quali-

tative examples of Semantic-Octomap and Kimera-TSDF maps, associated navigation graphs and A* paths. The results of the RAM and CPU consumption evaluation for both methods with different map and graph resolutions are shown in Figure 11 and the integration time evaluation for the same methods and resolutions is presented in Figure 12. Results of this evaluation show that the semantic Octomap mapping method and its associated graph building seem to be better suited for the integration of such data with the chosen parameters. Indeed, as each LiDAR cloud of the used sequence contains approximately 131000 points, the average integration time of a single scan to the navigation graph using the TSDF method at a map voxel resolution of 0.1 m and a graph node resolution of 0.2 m would be of 7.2 s, against 0.56 s for the Octomap method. The total integration time of a LiDAR scan for the Octomap method at this resolution allows for a real-time application in this setting, creating a dense graph in which paths can be drawn from one end of the trajectory to the other. The TSDF integration would introduce too much delay for the robot to move at the same speed without travelling on nodes not yet allocated or would necessitate to discard a great number of scans, thus creating holes and making the graph non navigable. For a map voxel resolution of 0.3 m and a graph node resolution of 0.6 m, these integration times fall respectively to 1.4 s an 0.10 s. Even with this resolution, the TSDF method still struggles to update the graph at a satisfying rate whereas the Octomap method is able to operate at data production rate. In order to apply the TSDF method to real-world embedded applications, smaller clouds
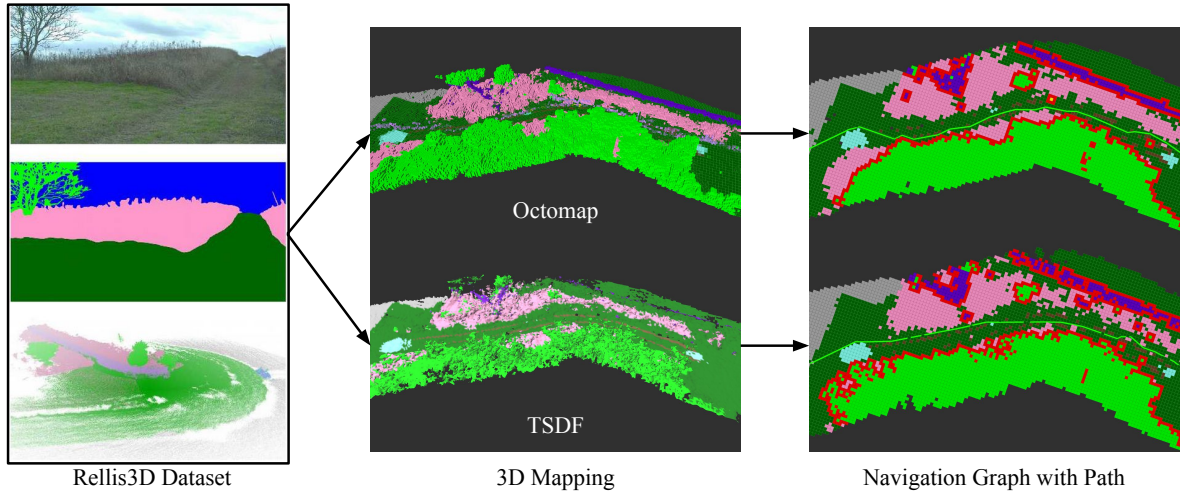
Figure 10: Qualitative examples of the maps, navigation graphs and paths produced by SMaNa from the RELLIS-3D sequence *00000*. Here, both mapping methods were used at a voxel resolution of 25 cm to produce graphs at 50 cm resolutions. Paths are displayed as light green lines on top of the graphs and *safety zone* nodes around obstacles are in red. The label color table is available in the RELLIS-3D description paper (Jiang et al., 2021).
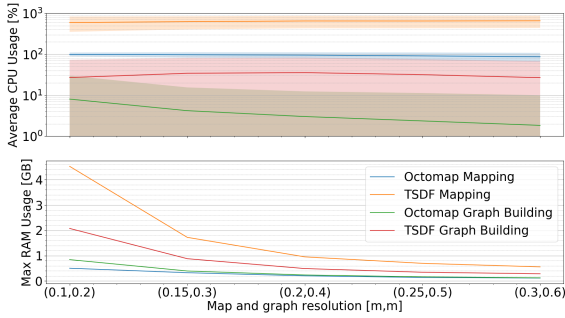

Figure 11: Average CPU usage (for 8 cores, full usage corresponds to 800 %) with standard deviation (log scale) and max RAM usage for map and graph generation during the entirety of the RELLIS-3D sequence *00000*.
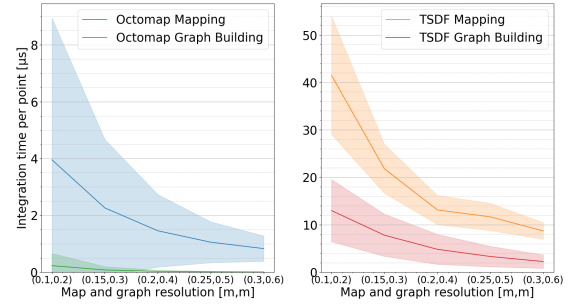

Figure 12: Average integration time per point with standard deviation for map and graph generation with both mapping methods from input LiDAR scans of the RELLIS-3D sequence *00000*.

should be used in input, by reducing the maximum integration distance or introducing a cut-out angle on the LiDAR scans. A RGB-D camera or stereo rig with a small image resolution could also be better suited as it would produce more dense and spatially constrained data than a LiDAR, less costly to integrate in a 3D map and a graph, with less voxels and nodes to allocate or update at each scan.

6 Conclusions and Perspectives

This paper has presented an original system to jointly optimize the process of online navigation using dense semantic mapping. Two state-of-the-art semantic mapping methods have been implemented and evaluated in this specific context, and the genericity of the navigation algorithm allows for other semantic map-

ping and planning methods to be used. A *closed-loop* simulation environment based on Ignition Gazebo has been developed to be able to evaluate the mapping as well as the navigation capabilities for autonomous mobile robots in outdoor unstructured environments. It relies on the 3DRMS-challenge dataset which presented the adequate features, but other datasets could be processed in the same way to build a collection of worlds in this contributed simulation setup with the same semantic rendering and evaluation characteristics.

The architecture has been evaluated in this setup under the addition of various controlled levels of noise on both localisation and semantic segmentation to demonstrate the accuracy of the reconstructed map and the ability to navigate using the underlying navigation graph. As an application of this architecture in a more advanced task, an exploration application has

been performed in closed-loop simulation to highlight the capability of SMaNa to navigate avoiding specific terrain types that could be hazardous. Finally, an experiment on real data using the RELLIS-3D dataset has been conducted to validate the usability of SMaNa in a real scenario. The capacity to run in real-time has also been evaluated and this shows that the Octomap mapping method is better suited for large outdoor navigation than the TSDF, however the above-mentioned uncertainty study suggests that the TSDF method evaluated is more robust to localisation and segmentation noise than the Octomap one.

This first development and evaluation of a joint online semantic mapping and navigation architecture for unstructured environment opens many new possibilities for real-world deployment. In particular, the proposed SMaNa architecture could be extended with a compatible visual-Lidar SLAM for unstructured environment. In order to deploy the architecture in a real-world scenario, the training and implementation of a semantic segmentation network from relevant sensor data should be carried out to generate semantic clouds. Moreover, the evaluation results of the discrete explicit mapping methods applied to robot navigation presented in this paper highlight some of their limitations, which could be mitigated by the use of a dedicated continuous implicit representation. Simulation environments based on semantic generation capabilities such as the OAISYS architecture (Müller et al., 2021) can also be considered for more extensive evaluations in critical environments such as those encountered during extra-planetary missions.

REFERENCES

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Susstrunk, S. (2012). Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34.

Achat, S., Marzat, J., and Moras, J. (2022). Path planning incorporating semantic information for autonomous robot navigation. In *19th International Conference on Informatics in Control, Automation and Robotics (ICINCO), Lisbon, Portugal*.

Bartolomei, L., Teixeira, L., and Chli, M. (2020). Perception-aware path planning for uavs using semantic segmentation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5808–5815.

Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S., Stachniss, C., and Gall, J. (2019). SemanticKITTI: A Dataset for Semantic Scene Understanding of Li-DAR Sequences. In *IEEE/CVF International Conf. on Computer Vision (ICCV)*.

Bouman, A., Ginting, M. F., Alatur, N., Palieri, M., Fan,

D. D., Touma, T., Pailevanian, T., Kim, S.-K., Otsu, K., Burdick, J., and Agha-Mohammadi, A.-a. (2020). Autonomous Spot: Long-range autonomous exploration of extreme environments with legged locomotion. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2518–2525.

Chen, X., Li, S., Mersch, B., Wiesmann, L., Gall, J., Behley, J., and Stachniss, C. (2021). Moving Object Segmentation in 3D LiDAR Data: A Learning-based Approach Exploiting Sequential Data. *IEEE Robotics and Automation Letters (RA-L)*, 6:6529–6536.

Cheng, Q., Zeller, N., and Cremers, D. (2022). Vision-based large-scale 3D semantic mapping for autonomous driving applications. *arXiv preprint arXiv:2203.01087*.

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

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., and Koltun, V. (2017). CARLA: An open urban driving simulator. In *Conference on Robot Learning*, pages 1–16.

Ewen, P., Li, A., Chen, Y., Hong, S., and Vasudevan, R. (2022). These maps are made for walking: Real-time terrain property estimation for mobile robots. *IEEE Robotics and Automation Letters*.

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

Grinvald, M., Tombari, F., Siegwart, R., and Nieto, J. (2021). TSDF++: A Multi-Object Formulation for Dynamic Object Tracking and Reconstruction. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 14192–14198.

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

Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C., and Burgard, W. (2013). OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*. Software available at http://octomap.github.com.

Hua, B.-S., Pham, Q.-H., Nguyen, D. T., Tran, M.-K., Yu, L.-F., and Yeung, S.-K. (2016). SceneNN: A scene meshes dataset with aNNotations. In *Fourth International Conference on 3D Vision (3DV)*, pages 92–101.

Hughes, N., Chang, Y., and Carlone, L. (2022). Hydra: A real-time spatial perception system for 3D scene graph construction and optimization. In *Robotics: Science and Systems (RSS)*.

Jiang, P., Osteen, P., Wigness, M., and Saripalli, S. (2021). RELLIS-3D dataset: Data, benchmarks and analysis. In *IEEE international Conference on Robotics and Automation (ICRA)*, pages 1110–1116.

Kantaros, Y., Kalluraya, S., Jin, Q., and Pappas, G. J. (2022). Perception-based temporal logic planning in uncertain semantic maps. *IEEE Transactions on Robotics*, 38(4):2536–2556.

Kazhdan, M., Bolitho, M., and Hoppe, H. (2006). Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7, page 0.

Krstinić, D., Braović, M., Šerić, L., and Božić-Štulić, D. (2020). Multi-label classifier performance evaluation with confusion matrix. *Computer Science & Information Technology*, 1.

Lambert, J., Liu, Z., Sener, O., Hays, J., and Koltun, V. (2020). MSeg: A composite dataset for multi-domain semantic segmentation. In *Computer Vision and Pattern Recognition (CVPR)*.

Martins, R., Bersan, D., Campos, M. F., and Nascimento, E. R. (2020). Extending maps with semantic and contextual object information for robot navigation: a learning-based framework using visual and depth cues. *Journal of Intelligent & Robotic Systems*, 99(3):555–569.

Maturana, D. (2022). *Semantic Mapping for Autonomous Navigation and Exploration*. PhD thesis, Carnegie Mellon University.

McCormac, J., Handa, A., Davison, A., and Leutenegger, S. (2017). SemanticFusion: Dense 3D semantic mapping with convolutional neural networks. In *IEEE International Conference on Robotics and automation (ICRA)*, pages 4628–4635.

Müller, M. G., Durner, M., Gawel, A., Stürzl, W., Triebel, R., and Siegwart, R. (2021). A Photorealistic Terrain Simulation Pipeline for Unstructured Outdoor Environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Nießner, M., Zollhöfer, M., Izadi, S., and Stamminger, M. (2013). Real-time 3D reconstruction at scale using voxel hashing. *ACM Transactions on Graphics (TOG)*, 32.

Oleynikova, H., Taylor, Z., Fehr, M., Siegwart, R., and Nieto, J. (2017). Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Ono, M., Fuchs, T. J., Steffy, A., Maimone, M., and Yen, J. (2015). Risk-aware planetary rover operation: Autonomous terrain classification and path planning. In *2015 IEEE Aerospace Conference*, pages 1–10.

Qi, C. R., Su, H., Mo, K., and Guibas, L. J. (2017). Pointnet: Deep learning on point sets for 3D classification and segmentation. In *IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 652–660.

Ravichandran, Z., Griffith, J. D., Smith, B., and Frost, C. (2020). Bridging scene understanding and task execution with flexible simulation environments. *arXiv preprint arXiv:2011.10452*.

Rosinol, A., Abate, M., Chang, Y., and Carlone, L. (2020). Kimera: an open-source library for real-time metric-semantic localization and mapping. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*.

Seymour, Z., Thopalli, K., Mithun, N., Chiu, H.-P., Samarasekera, S., and Kumar, R. (2021). Maast: Map attention with semantic transformers for efficient visual navigation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 13223–13230.

Song, Y., Naji, S., Kaufmann, E., Loquercio, A., and Scaramuzza, D. (2021). Flightmare: A flexible quadrotor simulator. In *Conference on Robot Learning*, pages 1147–1157.

Sun, N., Yang, E., Corney, J., and Chen, Y. (2019). Semantic path planning for indoor navigation and household tasks. In Althoefer, K., Konstantinova, J., and Zhang, K., editors, *Towards Autonomous Robotic Systems*, pages 191–201, Cham. Springer International Publishing.

Takikawa, T., Acuna, D., Jampani, V., and Fidler, S. (2019). Gated-SCNN: Gated shape CNNs for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5229–5238.

Tylecek, R., Sattler, T., Le, H.-A., Brox, T., Pollefeys, M., Fisher, R. B., and Gevers, T. (2019). The second workshop on 3d reconstruction meets semantics: Challenge results discussion. In Leal-Taixé, L. and Roth, S., editors, *ECCV 2018 Workshops*, pages 631–644, Cham. Springer International Publishing.

Wang, C., Cheng, J., Chi, W., Yan, T., and Meng, M. Q.-H. (2021). Semantic-aware informative path planning for efficient object search using mobile robot. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 51(8):5230–5243.

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

Zürn, J., Burgard, W., and Valada, A. (2020). Self-supervised visual terrain classification from unsupervised acoustic feature learning. *IEEE Transactions on Robotics*, 37(2):466–481.