2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Finding Things in the Unknown: Semantic Object-Centric Exploration with an MAV

Sotiris Papatheodorou[1,2,3], Nils Funk[1], Dimos Tzoumanikas[1], Christopher Choi[1], Binbin Xu[1,4] and Stefan Leutenegger[1,2,3]

***Abstract*— Exploration of unknown space with an autonomous mobile robot is a well-studied problem. In this work we broaden the scope of exploration, moving beyond the pure geometric goal of uncovering as much free space as possible. We believe that for many practical applications, exploration should be contextualised with semantic and object-level understanding of the environment for task-specific exploration. Here, we study the task of both *finding* specific objects in unknown space as well as *reconstructing* them to a target level of detail. We therefore extend our environment reconstruction to not only consist of a background map, but also object-level and semantically fused submaps. Importantly, we adapt our previous objective function of uncovering as much free space as possible in as little time as possible with two additional elements: first, we require a maximum observation distance of background surfaces to ensure target objects are not missed by image-based detectors because they are too small to be detected. Second, we require an even smaller maximum distance to the found objects in order to reconstruct them with the desired accuracy. We further created a Micro Aerial Vehicle (MAV) semantic exploration simulator based on Habitat in order to quantitatively demonstrate how our framework can be used to efficiently find specific objects as part of exploration. Finally, we showcase this capability can be deployed in real-world scenes involving our drone equipped with an Intel RealSense D455 RGB-D camera.**

***Index Terms*— Aerial Systems: Perception and Autonomy, Visual-Based Navigation**

I. Introduction

Exploring an unknown environment using an autonomous robot has been well-researched. The objective is typically formulated as uncovering as much free space as possible in a given timeframe, or uncovering all the somehow limited free space as fast as possible [1], [2], [3], [4]. Hereby, a tight interaction between free space mapping, planning, and control is crucial. MAVs are ideal platforms for autonomous exploration due to their ability to quickly move in any direction, allowing them to explore complex 3D environments.

However, exploration isn't often a goal in and of itself, it is a necessary step in solving a larger task. We believe that many robotics tasks that require exploration also require object-level and semantic understanding of the environment. Although there has been work aimed at exploring an unknown environments while detecting objects of interest, their goal is often finding a single object instance [5], [6] or isn't focused on object reconstruction [7].

Fig. 1. Objects reconstructed using our method [Top left] and their corresponding ground-truth meshes provided by the Matterport3D dataset [9] [Bottom left]. Top-down view of 3D reconstruction after exploration, and MAV path in yellow [Right].

In this paper we propose a task-specific exploration method aimed at finding all objects of interest in an unknown environment and reconstructing them with high accuracy. To the best of our knowledge, this is the first work using object-level maps for active object reconstruction and exploration. In summary, we propose the following contributions:

- A mapping pipeline suitable for exploration, path planning and high quality object reconstructions.
- An exploration utility function tailored to discovering objects and creating high quality object reconstructions.
- A scheme that accounts for incomplete depth maps preventing the exploration algorithm from getting stuck in regions of the scene where depth measurements consistently cannot be obtained.
- An open source MAV exploration simulator based on Habitat [8] and the Robot Operating System (ROS) [4].

II. Related Work

The concept of using frontiers for exploration was introduced in [1]. Frontiers are the boundaries between known free and unknown space and indicate regions that will expand the map of known space when observed. In [1] the closest

This work was funded in part by the Imperial College President's PhD Scholarship, SLAMcore Ltd., EPSRC grant ORCA Stream B - Towards Resident Robots, and EPSRC grant Aerial ABM EP/N018494/1.

[1]Smart Robotics Lab, Department of Computing, Imperial College London. E-mail addresses: {s.papatheodorou18, nils.funk13, dimosthenis.tzoumanikas14, christopher.choi, b.xu17, s.leutenegger}@ic.ac.uk

[2]Smart Robotics Lab, Department of Informatics, Technical University of Munich. E-mail address: stefan.leutenegger@tum.de

[3] Munich Institute of Robotics and Machine Intelligence (MIRMI)

[4] University of Toronto Robotics Institute, University of Toronto. E-mail address: binbin.xu@utoronto.ca

[4]https://github.com/smartroboticslab/semantic-exploration-icra-2023

frontier was selected as the robot's next goal but various methods of selecting the best frontier to visit next have been proposed [10], [11], [12] since.

A more recent family of exploration methods, pioneered by [2], are those based on next-best-view selection. In [2] candidate next views are sampled in free space and a utility computed for each one based on the unknown volume observed from the candidate and the length of the path to the candidate. The candidate view with the highest utility is then selected as the next robot goal. Informed sampling, e.g. close to frontiers [3], [13], [14], [4] can greatly increase the sample-efficiency of these methods. The choice of candidate view utility function is another important design consideration for an exploration algorithm [15], [16], [17], [18].

The SLAM community has recently moved its attention from pure geometric representations to semantically annotated reconstructions [19]. One approach is to build object-centric maps for each detected object in the scene. SLAM++ [20] is an early object-level mapping system that matches observed objects to pre-scanned shapes. Later, works such as Fusion++ [21] and Kimera [22] were proposed to incrementally build object-level maps inside a dense SLAM system. Object-level representations in a SLAM system have shown to provide more reliable loop closure detection [21] and also robustness to dynamic objects [23].

Powered by these advancements, several recent works propose the inclusion of semantic information for autonomous exploration in unknown environments. In [7], [6], the authors propose a method to create a semantically-annotated mapping system and include the re-observation of detected objects in the utility for the proposed sampling-based path planning algorithm. [24] further takes inter-object spatial relationships into consideration by maintaining semantic linking maps for the next-best-view selection. In terms of information-based exploration, [25] develops a Bayesian multi-class semantic mapping system, where a closed-form lower bound for Shannon mutual information is computed to evaluate an optimal trajectory. Instead of using local goal detection and heuristic utility functions, learning-based approaches have also been proposed to train a navigation policy network in the semantic map via reinforcement learning [5]. The goal of these semantically-guided exploration methods is either finding one object of a specific class or finding all objects in the environment as a result of classic exploration. They do not focus on reconstructing the found objects in detail nor employ an exploration algorithm to facilitate the detection of all objects in the environment. It is these gaps that our proposed method aims to fill.

III. Problem Formulation

The goal of this work is to use a sensor-equipped MAV to explore and map an unknown space and, at the same time, find and reconstruct in detail an unknown number of objects. Thus, the MAV must be able to map its environment, plan safe exploration paths, detect objects and create separate object reconstructions.

A. Environment Model

We model the static environment as a bounded volume $V \subset \mathbb{R}^3$ whose points $\mathbf{v} \in V$ have an associated occupancy probability $P_o(\mathbf{v})$. If no prior information about the environment is available, the occupancy of all points $\mathbf{v} \in V$ is initially *unknown*, defined as $P_o(\mathbf{v}) = 0.5$. Due to the geometry of the environment and the MAV, as well as the sensor's mounting pose, there can be points $V_{\text{unob}} \subset V$ that cannot be observed by the sensor. The goal of exploration is to create a map M of the observable part of the environment $V_{\text{obs}} = V \setminus V_{\text{unob}}$ by updating the occupancy probability of all $\mathbf{v} \in V_{\text{obs}}$ to either *free* or *occupied*. Frontiers [1], the boundaries between *free* and *unknown* space, identify regions that will extend the map when observed. The map M can be used to plan collision-free paths for the MAV to follow.

It is assumed that the environment contains $N_o \in \mathbb{N}$ static objects of interest, each one assigned a semantic class from the set of semantic classes $\mathscr{C}$. Each semantic class has an associated target map resolution $r_c, c \in \mathscr{C}$ selected by the user. The goal of object-centric exploration is to find all objects and create a high-quality map $M_i^c, i \in \{1 \dots N_o\}, c \in \mathscr{C}$ with a resolution at least as fine as r_c for each one.

B. MAV Model

For the purposes of exploration and path planning, the MAV's state $\mathbf{x}$ consists of its position vector $\mathbf{r} = [x, y, z]^T \in V$ in the world frame $\underrightarrow{\mathscr{F}}_W$ and a yaw angle scalar, $\psi \in [-\pi, \pi)$ with respect to the origin of the world coordinate, thus $\mathbf{x} = [x, y, z, \psi]^T \in V \times [-\pi, \pi)$. We assume that the MAV has a maximum linear velocity $v_{\max} \in \mathbb{R}^+$ and a maximum yaw rate $\omega_{\max} \in \mathbb{R}^+$ and is enclosed in a sphere of radius R centred at $\mathbf{r}$. While our planning does not consider the MAV's dynamics or roll and pitch angles, the full 6 degree of freedom pose is needed for mapping. It is expressed as a rigid body transformation matrix $\boldsymbol{T}_{WC}$ from the MAV camera frame $\underrightarrow{\mathscr{F}}_C$ to the world frame $\underrightarrow{\mathscr{F}}_W$ and can be estimated using an onboard SLAM system or an external motion capture system.

The MAV is equipped with an RGB-D sensor of resolution $W \times H$, focal lengths f_x and f_y, and depth range $[d_{\min}, d_{\max}] \subset \mathbb{R}^+$ inclusive. The sensor produces synchronised pairs of colour and depth images $\mathbf{C}$ and $\mathbf{D}$ respectively.

Each colour image $\mathbf{C}$ has a number of corresponding object instance segmentation masks $\mathbf{S}_k$, $k \in \{1 \dots N_d\}$ where $N_d \in \mathbb{N}$ is the number of objects detected in $\mathbf{C}$.

IV. Proposed Approach

Our approach consists of a mapping module and a planning module. The mapping module receives the current camera pose $\boldsymbol{T}_{WC}$, depth and colour images and object instance masks and produces a background map and associated frontiers as well as individual object maps. The planning module receives the current pose, the background map and its frontiers and the individual object maps and produces the path to the MAV's next goal pose. The mapping module is run as often as possible using the latest measurements while the planning module is run each time the MAV reaches

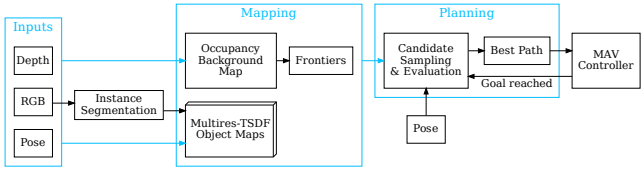

Fig. 2. Diagram of the proposed approach. The mapping module receives depth and colour image pairs and the corresponding instance segmentation and pose to update the background and individual object maps and, the set of frontiers. The planning module receives the background and object maps, the set of frontiers and, the current pose to produce the next goal path. Planning happens again when the MAV has completed the goal path.

its goal pose. Thus, the background and object maps are being continuously updated as the MAV moves to its next goal pose. The required initial information consists of the semantic classes of interest, their associated resolutions and enough *free* space around the MAV's initial position to allow planning paths. Figure 2 shows a diagram of the pipeline.

A. Background Mapping

The background map with voxel resolution $r \in \mathbb{R}^+$ is created using a modified version of the multi-resolution occupancy mapping pipeline from [26] with some extra information stored in the map compared to [3]. In [26] an octree-based volumetric occupancy map is created in real-time from posed depth images. Path planning can be performed directly on the map due to its explicit free space representation and the fact that occupancy data is propagated to higher levels of the octree, allowing for efficient occupancy queries.

1) Frontier Detection: As in [3], the mapping pipeline is modified to maintain a set of frontiers $\mathscr{F}$. After each map update, a set of candidate frontiers is generated by forming the union of any previously known frontiers with the set of updated octree nodes. These candidate frontiers are then checked for frontiers at the finest allocated level to produce the new set of frontiers $\mathscr{F}$.

2) Observed Distance Fusion: An addition compared to [3] is storing the minimum distance each octree leaf node has been observed from. This allows finding map regions that haven't been observed from a small enough distance as described in Section IV-D.3.

3) Colour Fusion: The final addition to the map compared to [3] is integrating colour information in each voxel. The colour update is performed as a weighted average using the same weight as for the occupancy probability.

4) Map Update: The background map M is updated as quickly as possible using pairs of depth and colour images and their corresponding poses without filtering out any detected objects. This allows performing path planning without having to consider the individual object maps since the background map already contains this information.

B. Object Mapping

Object mapping was performed using a similar method to [23], with changes aimed at improving performance, memory usage and reconstruction quality. We changed the underlying map representation to a modified version of the adaptive-resolution octree-based TSDF mapping from [27] which offers improved performance and reduces aliasing artefacts. A TSDF representation was chosen for objects because of its reduced computational requirements compared to occupancy representations.

1) Object Matching: Similar to [23], objects detected in the colour image $\mathbf{C}$ are matched to known objects in the map, after instance masks smaller than a threshold are discarded to prevent unrealistic detections in synthetic datasets. Each known object is first raycasted from the current camera pose $\boldsymbol{T}_{WC}$ to create a per-object instance mask, taking occlusions from the background and other objects into account. These raycasted instance masks represent the parts of known objects visible from the current pose. For each detected object instance mask $\mathbf{S}_k$, its intersection over union (IoU) with each per-object raycasted instance mask is computed. The detected object is associated with the known object for which the highest mask IoU above a certain threshold was achieved. If there is no IoU above the threshold then a new map is created for the detected object. While in [23] the object map resolution was constant, in our method it depends on the object's semantic class, allowing efficient mapping of objects of a wide range of sizes.

2) Object Map Update: As in [23], we integrate colour and foreground probability information in each object map voxel using a weighted average update. In the case of visible known objects that were undetected in the current colour image, depth and colour information is integrated into them using their raycasted instance masks while omitting the update of foreground probabilities. Compared to [23], we also keep track of the minimum observed distance as in the background map.

3) Memory Usage Reduction: We reduce the memory footprint of each voxel by using 16-bit and 8-bit fixed-point numbers for the TSDF value and foreground probability, respectively. This reduces the required memory per voxel by 39% with negligible degradation of reconstruction accuracy.

C. History of Incomplete Depth Maps

Depth images often contain invalid or missing data due to e.g. reflective surfaces, occlusions or missing geometry in the case of synthetic datasets. A depth image with a large amount of missing data can cause a naive exploration algorithm to get stuck expecting a reduction in map entropy that never happens due the amount of missing data. To tackle this issue, we store the history of invalid depth measurements as a low-resolution 3D grid over V containing binary 360° images. All images are initialised to 1 denoting valid depth. For each depth image $\mathbf{D}$ integrated into the background map, the history image $\mathbf{H}$ corresponding to pose $\boldsymbol{T}_{WC}$ is fetched. The 2D coordinates of each invalid measurement in $\mathbf{D}$ are back-projected in 3D and then re-projected onto $\mathbf{H}$, setting the corresponding pixel to 0.

D. Exploration Planning

Our algorithm is a hybrid between sampling-based and frontier-based exploration. Candidate next poses are sampled

close to the frontiers and the known objects, ranked based on the proposed utility function, and the highest ranked pose is selected as the next goal. Compared to [3], in our approach

- candidate next positions are also sampled close to objects instead of only close to frontiers,
- the exploration utility is computed using two more raycasts in addition to the entropy raycast in [3], and
- the invalid depth history described in Section IV-C is used to prevent the exploration from becoming stuck.

1) Candidate Next Position Sampling: Candidate next positions $\hat{\mathbf{r}}_j \in V,\ j \in \{1 \dots n\},\ n \in \mathbb{N}^+$ are sampled among the set of frontiers and the known objects without replacement until n candidates have been sampled or no more frontiers or objects are left to sample, as shown in Figure 3 [Left]. Sampling at frontiers favours exploring unknown space while sampling at objects allows obtaining higher quality object observations. Sampling can be biased towards frontiers or objects using the frontier sampling probability P_{fr} to strike a balance between object reconstruction quality and exploration.

2) Path Planning to Candidate Next Positions: For each sampled candidate position, a path $P_j(\mathbf{r}, \hat{\mathbf{r}}_j)$ to it is planned from the MAV's current position, as shown in Figure 3 [Left]. The path is planned on the background map using the Informed RRT* [28] implementation from the Open Motion Planning Library [29]. Since sampled positions are at or near non-*free* space, paths are planned as close to them as possible with no requirement for a complete solution.

3) Raycasting at Candidate Next Positions: In order to evaluate each candidate next position, three low-resolution, $w \times h$, 360° raycasts are performed from each candidate position $\hat{\mathbf{r}}_j$. The raycasting resolution $w \times h$ is independent of the input image resolution $W \times H$ and typically much smaller to reduce the computational requirements [30]. This results in three gain images with values in the interval $[0, 1]$ inclusive. The 360° raycasting rays from a single candidate are shown in Figure 3 [Middle].

An entropy raycast of the background map M is performed, resulting in an entropy gain image $\mathbf{G}_{ent}$. Each pixel of $\mathbf{G}_{ent}$ contains the normalised sum of the Shannon entropy of each voxel $\mathbf{v}$ along the corresponding ray. Entropy is accumulated along each ray until $d_{\max}$ or an *occupied* voxel is reached and the sum is normalised with its maximum possible value $\frac{(d_{\max} - d_{\min})}{r}$. The entropy gain image guides the exploration towards observing *unknown* space resulting in expansion of the map.

A background gain image $\mathbf{G}_{\text{bg}}$ is created by raycasting the minimum observed distance information stored in the background map. Each pixel of $\mathbf{G}_{\text{bg}}$ contains the distance gain of the octree leaf node $\mathbf{n}$ containing the first *occupied* voxel hit by the corresponding ray. The distance gain of $\mathbf{n}$ is

$$G(\mathbf{n}) = \begin{cases} 0, \text{ if } d_{\text{node}}(\mathbf{n}) \leq d_{\text{bg}} \vee d_{\text{node}}(\mathbf{n}) \leq d_{\text{exp}}(\mathbf{n}) \\ \frac{d_{\text{node}}(\mathbf{n}) - \max\left(d_{\text{exp}}(\mathbf{n}),\ d_{\text{bg}}\right)}{d_{\max}}, \text{ otherwise} \end{cases}, \quad (1)$$

where $d_{\text{node}}(\mathbf{n}) \in \mathbb{R}^+$ is the minimum distance $\mathbf{n}$ has been observed from, $d_{\text{exp}}(\mathbf{n}) \in \mathbb{R}^+$ is the expected observed distance of $\mathbf{n}$ from $\hat{\mathbf{r}}_j$ and $d_{\text{bg}} \in \mathbb{R}^+$ is the desired observed distance for the background map. The desired observed distance d_{bg} is set to a value that ensures observed objects will appear large enough in a colour image taken from d_{bg} or closer that they can be detected. The background gain image helps ensure all of the environment has been observed from close enough that all objects have been detected.

Similarly to the background gain image, an object gain image $\mathbf{G}_{\text{obj}}$ is computed by raycasting the minimum observed distance stored in all object maps while taking occlusions into account. The per-pixel gain is computed in the same manner as for the background except that the object desired observed distance $d_{\text{obj}} \in \mathbb{R}^+$ is used instead of d_{bg} in Equation (1). The desired observed distance $d_{\text{obj}} < d_{\text{bg}}$ for objects of all semantic classes is set to a value that minimises the distance-based depth sensor noise while taking $d_{\min}$ and the MAV dimensions into account. The object gain image results in the MAV getting closer, higher-quality observations of existing objects.

4) Candidate Next Pose Evaluation: A single gain image $\mathbf{G}$ is produced from a weighted sum of the individual gain images and masking by the invalid depth history

$$\mathbf{G} = \left(\alpha_{\text{ent}}\mathbf{G}_{\text{ent}} + \alpha_{\text{bg}}\mathbf{G}_{\text{bg}} + \alpha_{\text{obj}}\mathbf{G}_{\text{obj}}\right) \circ \hat{\mathbf{H}}, \quad (2)$$

where $\alpha_{\text{ent}}, \alpha_{\text{bg}}, \alpha_{\text{obj}} \in [0, 1]$, $\alpha_{\text{ent}} + \alpha_{\text{bg}} + \alpha_{\text{obj}} = 1$ are weighting factors, $\hat{\mathbf{H}}$ is the invalid depth history image corresponding to position $\hat{\mathbf{r}}_j$ and $\circ$ denotes the element-wise matrix product. The weighting factors are used to solve the object-level exploration-exploitation dilemma by balancing between exploring, detecting new objects and improving the quality of existing objects. A sliding window on the gain image $\mathbf{G}$ is used to compute the yaw angle ψ_j at which the maximum gain g_j is achieved, similarly to [31]. Some example gain images and the corresponding optimal yaw angle are shown in Figure 3 [Right]. The utility u_j of each candidate j is computed to maximise the gain over time as $u_j = g_j / t_j$, where the time t_j required for the MAV to complete the path to candidate j is estimated based on the assumption that it always flies at speed $v_{\max}$ and rotates at speed $\omega_{\max}$.

5) Next Goal View: The candidate $G \in \{1 \dots n\}$ with the highest utility is selected as the next goal. A yaw angle needs to be assigned to each vertex of the goal path P_G. The first vertex, which coincides with the current MAV position, gets assigned the current yaw while the final vertex gets assigned the optimal yaw ψ_G. The yaw of intermediate path vertices is computed similarly to the final vertex yaw, by performing 360° raycasts and computing the optimal yaw angle.

6) Termination Condition: The exploration stops when $\mathscr{F} = \varnothing$, and all of the background and objects have been observed from a distance at least d_{bg} and d_{obj}, respectively.

V. Experimental Evaluation and Discussion

A. MAV Simulator

To quantitatively evaluate the proposed method and make our results reproducible, we created an MAV simulator suitable for object-centric exploration. The MAV dynamics

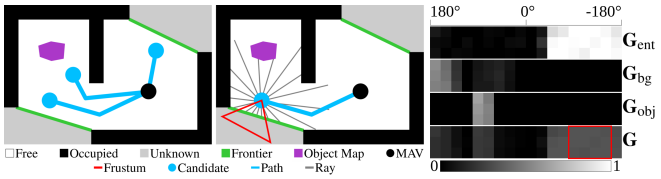

Fig. 3. [Left] Candidate view sampling near frontiers and objects, and path planning to candidates from the current pose. [Middle] Sparse 360° raycasting from one of the candidate views and optimal yaw frustum in red. [Right] Entropy, background, object and combined gain images from the raycast shown in [Middle] with the optimal yaw field-of-view in red.

are simulated using the prediction step of the MAV Model Predictive Controller from [32]. The simulator uses Habitat [8] to provide colour and depth images as well as ground-truth object segmentation masks rendered from the MAV poses. We included additive Gaussian, zero-mean noise with a standard deviation based on distance in the synthetic depth images to make them more closely resemble those produced by a RealSense D455 sensor. The noise standard deviation for a depth measurement d was computed as $\sigma(d) = \max(\sigma_{\min}, \min(sd, \sigma_{\max}))$.

B. Simulated Experiments

All simulated experiments were run on a computer with an Intel Core i7-6700K CPU, 16 GB of memory and an NVIDIA RTX 3080 GPU. They were run on Ubuntu 20.04 using ROS Noetic and compiled with GCC 9.4.0 using the O3 optimisation level. The parameters used for the simulated experiments can be found in Table I.

TABLE I
SIMULATED EXPERIMENT PARAMETERS

Parameter	Value	Parameter	Value
$v_{\max}$	1.5 m/s	r, r_{chair}	0.04 m, 0.02 m
$\omega_{\max}$	0.75 rad/s	$w \times h$	36×10
R	0.125 m	P_{fr}	0.5
$W \times H$	320×240	n	20
f_x, f_y	262.5	d_{bg}, d_{obj}	3 m, 1 m
$\sigma_{\min}$, $\sigma_{\max}$	0.005 m, 0.2 m	$d_{\min}$, $d_{\max}$	0.1 m, 10 m
s	0.002	α_{ent}, α_{bg}, α_{obj}	0.34, 0.33, 0.33

The proposed semantic exploration method is compared against a classic exploration method, similar to the one from [3] with the addition of object mapping and the invalid depth history described in Section IV-C. The classic exploration method is obtained by setting $\alpha_{\text{ent}} = 1$, $\alpha_{\text{bg}} = \alpha_{\text{obj}} = 0$ and $P_{fr} = 1$. This is an ablation study of the importance of sampling candidates near objects and the two distance gain images for efficient object-centric exploration. It also helps showcase that the proposed method retains competitive exploration performance.

Both methods are evaluated on the Matterport3D [9] dataset which consists of semantically annotated 3D scans of real world interior spaces. Since this dataset is collected from real-world data, there is noise and holes in the ground-truth depth and segmentation masks. The evaluation consists of 5 runs of each method, classic and semantic, on sequences `1LXtFkjw3qL`, `29hnd4uzFmX`, `2azQ1b91cZZ`, `2n8kARJN3HM`, `2t7WUuJeko7`, and `8WUmhLawc2A`. These sequences were chosen because they have relatively little missing data while offering interesting spaces to explore. Chairs were chosen as the semantic object class of interest due to their abundance in the used sequences. Some reconstructed objects and their corresponding ground-truth meshes are shown in Figure 4.

Fig. 4. Objects reconstructed using the proposed method [Top] and their corresponding ground-truth meshes [Bottom]. Notice some artefacts due to erroneous segmentation masks.

Figure 5 [Left] shows the explored volume over time. It can be observed that the performance of the proposed method is on-par with classic exploration.

Figure 5 [Right] shows the percentage of objects detected over time. The proposed method finds more objects overall, faster and more consistently than the classic approach, indicating the importance of the background distance gain and sampling candidates near objects. Sampling candidates near known objects can help detect new objects quickly because objects are often close to each other, e.g. chairs tend to be close to other chairs. One reason for objects not being detected is partial observations, which lead to high uncertainty in the object detection network. If an object does not cover a large enough part of the image, it cannot be detected. Thus, observing small parts of an object from different perspectives in different images can result in full exploration while the object remains undetected. Another problem arises from the imperfect instance segmentation masks in the Matterport3D dataset, where the mask of one object might bleed onto another object. This can cause the algorithm to erroneously merge the two distinct objects into one, only one of which will be considered matched to a ground-truth object.

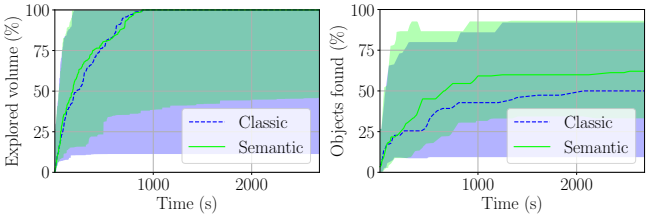

Fig. 5. Median, 10^{th} and 90^{th} percentiles of explored volume [Left] and percentage of objects found [Right].

We use the (pseudo) ground-truth meshes from Matterport3D to evaluate the meshes generated by our method in terms of accuracy and completeness. Accuracy is computed as the root-mean-square error while completeness is computed as the percentage of the ground-truth mesh vertices for which there is a reconstructed mesh vertex within 5 cm. Figures 6 and 7 show the background and object reconstruction

accuracy over time and completeness over time, respectively, aggregated over all runs. It can be seen that both methods achieve a similar level of reconstruction accuracy. There is little variation over time in the background and object accuracy because they are observed from a close enough distance even during classic exploration as the Matterport3D consists of interior spaces. The apparent discrepancy between the explored volume and background completeness is due to the fact that the former is a volume metric while the latter is a surface metric. This means that the resulting background mesh often has small holes due to occlusions or insufficient data causing a lower completeness score.

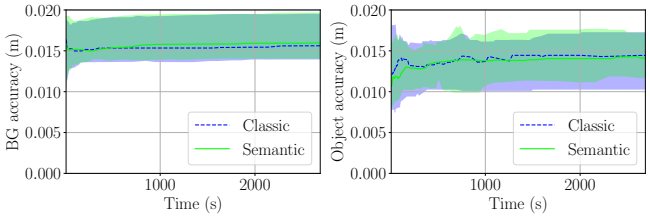

Fig. 6. Median, 10^{th} and 90^{th} percentiles of background [Left] and object [Right] accuracy.

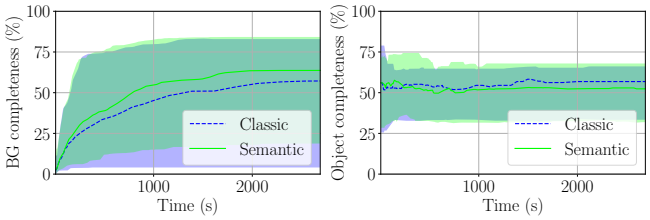

Fig. 7. Median, 10^{th} and 90^{th} percentiles of background [Left] and object [Right] completeness.

We also evaluate the percentage of the background and objects that were observed from at most the desired distance, d_{bg} and d_{obj}, respectively, as shown in Figure 8. We observe that the proposed method observes a larger part of the environment and objects from the desired distance and in the case of objects, it does so more consistently. The inaccurate semantic segmentation masks in the Matterport3D dataset can result in artefacts in the object reconstructions, e.g. including part of the background. These artefacts can be occluded, preventing them from being observed and reducing the percentage of objects observed from at most d_{obj}.

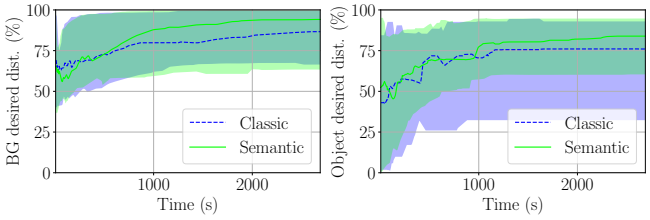

Fig. 8. Median, 10^{th} and 90^{th} percentiles of percentage of background [Left] and objects [Right] observed from at least d_{bg} and d_{obj} respectively.

C. Real World Experiments

A real world experiment was further conducted in order to showcase the feasibility of using the proposed approach on-board an MAV. The MAV used for the experiments was a DJI F550 hexacopter equipped with an Intel RealSense D455 RGB-D camera. The MAV pose was provided by a Vicon motion capture system. All processing was performed on-board the MAV on an NVIDIA Jetson Xavier NX computer. The experiments were conducted in a 7 m $\times$ 5.5 m $\times$ 5 m room with a 3.6 m $\times$ 3 m $\times$ 1.3 m volume available to the MAV for safety reasons. The semantic classes used for object reconstruction were limited to backpacks and keyboards. The background map resolution was 4 cm and the target object resolution was 2 cm for all semantic classes. The pre-trained Mask R-CNN [33] model from [34] was ported to TensorRT to obtain object segmentation masks. Segmentation was run on the latest colour image not currently being integrated, as fast as possible, on a background thread.

The experiment lasted for 405 s during which 81 frames were integrated and 31 planning iterations took place. The average frame integration time was 1.835 s, the average segmentation time was 1.167 s and the average planning time was 2.803 s. Segmentation information was available for 71 frames. This rather high percentage was achieved because the segmentation time was typically smaller than the integration time so that the segmentation information for the next frame was most often available before the integration of the current frame has finished. During exploration 2 out of 3 backpacks and 2 out of 3 keyboards placed in the environment were detected even though all of them were observed by the camera. This is due to the fact that the deployed Mask R-CNN wasn't fine-tuned on this particular environment and didn't detect the objects even though they were visible.

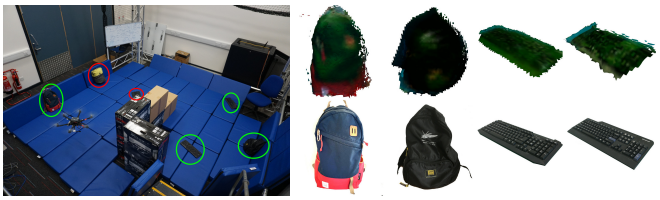

Fig. 9. [Left] The real world experimental setup and the MAV. Detected and undetected objects are circled in green and red respectively. [Top right] Meshes of the 2 backpacks and 2 keyboards reconstructed during the experiment. [Bottom right] Photos of the corresponding real world objects.

VI. Conclusions and Future Work

In this paper, we have presented a task-specific exploration pipeline for finding and creating high-quality reconstructions of objects in an unknown environment without sacrificing exploration performance. We have evaluated its effectiveness in extensive simulation studies and have demonstrated that it can be run on-board an MAV in real-world scenes.

We have identified several directions for future work: more robust object matching, dynamic tracking objects, object-centric exploration by multiple collaborating robots, and using machine learning to estimate where undiscovered objects are located given the currently known environment.

REFERENCES

[1] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation*. IEEE Computer Society, 1997, p. 146.

[2] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, "Receding horizon path planning for 3D exploration and surface inspection," *Autonomous Robots*, vol. 42, no. 2, pp. 291–306, Feb. 2018.

[3] A. Dai, S. Papatheodorou, N. Funk, D. Tzoumanikas, and S. Leutenegger, "Fast frontier-based information-driven autonomous exploration with an MAV," in *International Conference on Robotics and Automation*, Paris, France, May 2020, pp. 9570–9576.

[4] D. Duberg and P. Jensfelt, "UFOExplorer: Fast and scalable sampling-based exploration with a graph-based planning structure," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2487–2494, April 2022.

[5] D. S. Chaplot, D. P. Gandhi, A. Gupta, and R. R. Salakhutdinov, "Object goal navigation using goal-oriented semantic exploration," in *Advances in Neural Information Processing Systems (NeurIPS)*, December 2020.

[6] Z. Zeng, A. Röfer, and O. C. Jenkins, "Semantic linking maps for active visual object search," in *IEEE International Conference on Robotics and Automation (ICRA)*, Paris, France, May 2020, pp. 1984–1990.

[7] T. Dang, C. Papachristos, and K. Alexis, "Autonomous exploration and simultaneous object search using aerial robots," in *2018 IEEE Aerospace Conference*, Mar. 2018, pp. 1–7.

[8] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, D. Parikh, and D. Batra, "Habitat: A platform for embodied AI research," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea, October 2019, pp. 9339–9347.

[9] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang, "Matterport3D: Learning from RGB-D data in indoor environments," in *IEEE International Conference on 3D Vision (3DV)*, Qingdao, China, October 2017, pp. 667–676.

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

[11] W. Gao, M. Booker, A. H. Adiwahono, M. Yuan, J. Wang, and W.-Y. Yau, "An improved frontier-based approach for autonomous exploration," *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pp. 292–297, 2018.

[12] M. Faria, I. Maza, and A. Viguria, "Applying frontier cells based exploration and lazy theta* path planning over single grid-based world representation for autonomous inspection of large 3D structures with an UAS," *Journal of Intelligent & Robotic Systems*, vol. 93, no. 1, pp. 113–133, Feb 2019.

[13] Y. Kompis, L. Bartolomei, R. Mascaro, L. Teixeira, and M. Chli, "Informed sampling exploration path planner for 3D reconstruction of large scenes," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7893–7900, October 2021.

[14] P. Zhong, B. Chen, S. Lu, X. Meng, and Y. Liang, "Information-driven fast marching autonomous exploration with aerial robots," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 810–817, April 2022.

[15] A. Akbari and S. Bernardini, "Informed autonomous exploration of subterranean environments," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7957–7964, October 2021.

[16] L. Schmid, V. Reijgwart, L. Ott, J. Nieto, R. Siegwart, and C. Cadena, "A unified approach for autonomous volumetric exploration of large scale environments under severe odometry drift," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4504–4511, July 2021.

[17] A. Batinovic, A. Ivanovic, T. Petrovic, and S. Bogdan, "A shadowcasting-based next-best-view planner for autonomous 3D exploration," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2969–2976, April 2022.

[18] Z. Li, T. Li, J. Wang, and M. Q.-H. Meng, "Learning robot exploration strategy with 4D point-clouds-like information as observations," *IEEE Robotics and Automation Letters*, vol. 7, no. 1, pp. 1–8, January 2022.

[19] J. McCormac, A. Handa, A. J. Davison, and S. Leutenegger, "SemanticFusion: Dense 3D semantic mapping with convolutional neural networks," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[20] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. J. Kelly, and A. J. Davison, "SLAM++: Simultaneous localisation and mapping at the level of objects," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

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

[22] A. Rosinol, M. Abate, Y. Chang, and L. Carlone, "Kimera: an open-source library for real-time metric-semantic localization and mapping," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[23] B. Xu, W. Li, D. Tzoumanikas, M. Bloesch, A. Davison, and S. Leutenegger, "MID-Fusion: Octree-based object-level multi-instance dynamic SLAM," in *2019 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2019.

[24] R. Ashour, T. Taha, J. M. M. Dias, L. Seneviratne, and N. Almoosa, "Exploration for object mapping guided by environmental semantics using UAVs," *Remote Sensing*, vol. 12, no. 5, March 2020.

[25] A. Asgharivaskasi and N. Atanasov, "Active Bayesian multi-class mapping from range and semantic segmentation observations," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[26] N. Funk, J. Tarrio, S. Papatheodorou, M. Popović, P. F. Alcantarilla, and S. Leutenegger, "Multi-resolution 3D mapping with explicit free space representation for fast and accurate mobile robot motion planning," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3553–3560, April 2021.

[27] E. Vespa, N. Funk, P. H. Kelly, and S. Leutenegger, "Adaptive-resolution octree-based volumetric SLAM," in *IEEE International Conference on 3D Vision (3DV)*, Québec, Canada, September 2019, pp. 654–662.

[28] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Informed RRT*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014.

[29] I. A. Sucan, M. Moll, and L. E. Kavraki, "The open motion planning library," *IEEE Robotics Automation Magazine*, vol. 19, no. 4, pp. 72–82, Dec 2012.

[30] H. Oleynikova, Z. Taylor, R. Siegwart, and J. Nieto, "Safe local exploration for replanning in cluttered unknown environments for microaerial vehicles," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1474–1481, 2018.

[31] M. Selin, M. Tiger, D. Duberg, F. Heintz, and P. Jensfelt, "Efficient autonomous exploration planning of large-scale 3-D environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1699–1706, 2019.

[32] D. Tzoumanikas, Q. Yan, and S. Leutenegger, "Nonlinear MPC with motor failure identification and recovery for safe and aggressive multicopter flight," in *IEEE International Conference on Robotics and Automation (ICRA)*, Paris, France, May 2020, pp. 8538–8544.

[33] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Computer Vision (ICCV), 2017 IEEE International Conference on*. IEEE, Oct. 2017, pp. 2980–2988.

[34] W. Abdulla, "Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow," https://github.com/matterport/Mask_RCNN, 2017.