

Sweep-Your-Map: Efficient Coverage Planning for Aerial Teams in Large-Scale Environments

David Morilla-Cabello¹, Luca Bartolomei², Lucas Teixeira², Eduardo Montijano¹, and Margarita Chli²

Abstract— The efficiency of path-planning in robot navigation is crucial in tasks such as search-and-rescue and disaster surveying, but this is emphasized even more when considering multi-rotor aerial robots due to the limited battery and flight time. In this spirit, this work proposes an efficient, hierarchical planner to achieve comprehensive visual coverage of large-scale outdoor scenarios for small drones. Following an initial reconnaissance flight, a coarse map of the scene gets built in real-time. Then, regions of the map that were not appropriately observed are identified and grouped by a novel perception-aware clustering process that enables the generation of continuous trajectories (*sweeps*) to cover them efficiently. Thanks to this partitioning of the map into a set of tasks, we can generalize the planning to an arbitrary number of drones and perform a well-balanced workload distribution among them. We compare our approach against a state-of-the-art method for exploration and show the advantages of our pipeline in terms of efficiency for obtaining coverage in large environments. *Video* – <https://youtu.be/V2UIrM91oQ8>

Aerial Systems: Perception and Autonomy; Path Planning for Multiple Mobile Robots or Agents; Mapping.

I. INTRODUCTION

Recent advances in robot navigation and perception have enabled the establishment of modern multi-rotor aircraft, i.e., drones, as the best choice for autonomous 3D reconstruction or visual coverage of large-scale outdoor scenarios. Their flexibility allows them to move freely through the environment and observe areas that are not visible from the ground. However, time efficiency is critical for using drones because of their short flight times (due to battery limitations), usually well under 30 minutes. Therefore, the efficiency and effectiveness of the planning algorithms are essential to enable the deployment of drones in large-scale outdoor environments. Similarly, using multiple drones as advocated in this work promises to boost the efficiency of the scene-coverage mission.

Deploying drones for mapping a large area from a high altitude is an effective way to obtain a first estimation, as

Manuscript received: February 24, 2022; Revised: June 10, 2022; Accepted: July 6, 2022.

This work has been supported by the ONR Global grant N62909-19-1-2027, the Spanish projects PGC2018-098817-A-I00 and PGC2018-098719-B-I00 (MCIU/AEI/FEDER, UE), DGA T45-20R, and Spanish grant FPU20-06563, Swiss National Science Foundation (SNSF, Agreement no. PP00P2183720), NCCR Robotics, Amazon and the HILTI group.

¹ David Morilla-Cabello and Eduardo Montijano are with the Instituto de Investigación en Ingeniería de Aragón, Universidad de Zaragoza, Spain {davidmc, emonti}@unizar.es

² Luca Bartolomei, Lucas Teixeira and Margarita Chli are with the Vision for Robotics Lab, Department of Mechanical and Process Engineering, ETH Zurich, Zurich, Switzerland {lbartolomei, pilucas, mchli}@ethz.ch

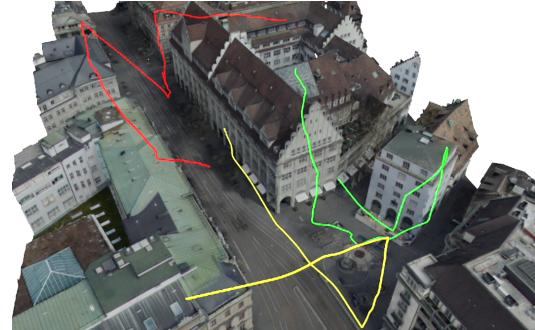


Fig. 1: Team of drones that *sweep* the area of interest by flying paths generated by the proposed planner in order to achieve fast coverage. Using a rough prior map (e.g. captured in a reconnaissance flight) to identify areas that require further observation, this work generates efficient path planning and workload distribution for a team of drones (three in this example) to cover the scene.

collisions with the environment can be more easily avoided. However, this strategy does not provide informative enough viewpoints for scene coverage and impacts the quality of the scene captures. State-of-the-art exploration approaches [1], [2] often lack efficiency because of problems such as over-exploring local regions, and abrupt changes in motion due to constant re-planning or the need for revisiting areas.

To overcome these limitations, this paper presents a hybrid solution that uses the best of both types of strategy in a synergetic way. In this work, we assume a team of drones with cameras, each performing a fast, reconnaissance flight at a high altitude capturing a rough map of the area of interest using a coarse real-time mapping pipeline. Based on this map, the proposed method computes a set of drone trajectories for subsequent flights in order to efficiently cover the area of interest completely. This process aims to maximize the use of *sweep* lines to avoid constant changes in the flight direction, while considering the visibility of surfaces and, at the same time, managing the workload distribution amongst the participating drones to minimize the execution time. The main contribution of this paper is the overall perception-aware global planning that is capable of handling the initial, noisy and coarse map as well as enforcing high-speed trajectories.

II. RELATED WORK

Aerial path planning for efficient exploration has been a topic of extensive research in robotics and computer vision due to its wide applicability.

A. Scene exploration and coverage

With the outlook of practicality, robotics approaches often focus on fast scene exploration, by eliminating the unknown

space as quickly as possible. Frontier exploration methods look for regions, where free and unknown space meet [3]. There are different criteria used to decide which frontier to explore next, such as their proximity to the current field of view [4], following a greedy selection strategy [5] or having global planning dictate their selection [1]. All these methods focus on volumetric representations of the map, whereas our approach considers surfaces and their visibility.

Other works use Active SLAM in 2D environments for indoors ground robot navigation using landmarks [6], [7] or learning methods [8], [9]. In comparison, we consider aerial robots in 3D outdoor environments to obtain comprehensive visual coverage.

When considering the reconstruction of surfaces, sampling-based approaches propose viewpoints based on their expected information gain. For example, accurate surface reconstructions [10] can be achieved in a Next-Best View fashion [11]. In order to improve the efficiency of the planning, Rapidly-exploring Random Trees are a common approach [10], [11]. To improve the sampling process, [2] applies informed sampling of configurations by reasoning over the available reconstructed model. The method in [12] considers voxels lying on the surface at a frontier. In general, all of these methods use depth cameras that allow for exploration or reconstruction in indoor and small scenarios. The performance in large-scale outdoor scenarios as considered in this work decreases as the sensor range only allows for close observations. In [13], online Multi-View Stereo (MVS) is used to incrementally assess the surface reconstruction. In comparison, the proposed approach executes a fast high-altitude reconnaissance flight to obtain a global coarse map as a prior and provide an insight into the structure of the whole scene at once.

B. Use of a prior map

Other works used priors for improving the view selection for 3D reconstruction and generating a global plan. They analyze a prior map obtained from a previous flight in order to plan views that maximize heuristics for 3D reconstruction as parallax angle [14] or matchability [15]. In [14], the problem is addressed by using submodular optimization to improve the proposed views in the free space and obtain the final trajectory by solving an orienteering problem accounting for a maximum allowed time budget. Submodular optimization is also used by [16] to plan views based on volumetric representations in an anytime optimization.

As discussed by [13], many of the previous methods obtain theirs prior from MVS pipelines, which is time-consuming and might require long waiting times for processing. In this work, we obtain a prior map online using depth completion to extract good estimates of the views to reconstruct the scene. The work in [13] considers individual views without focusing on the trajectory to connect them, which might generate path redundancies. In contrast, we leverage the fact that many of these views can be grouped in a single efficient trajectory in order to cover large parts of the scene, e.g., building facades.

C. Multi-robot extension

All of the aforementioned methods assume a single robot. While they can be extended to multi-robot setups by partitioning the area of interest according to the number of robots, this does not ensure efficient enough collaboration between them. Cooperative frontier-based approaches have also been proposed in a centralized [17] and decentralized [18] way. These methods address the coordination problem in frontier-based approaches but suffer from the aforementioned locality problems. The work in [12] extends to the multi-robot case by greedily assigning the view configurations [19]. The work in [20] distributes the workload through continuous region partitioning based on Voronoi components. By considering the whole map and the set of regions to be covered (tasks) as a Vehicle Routing Problem (VRP), the generalization to multiple drones is straightforward in our pipeline, easily accounting for collaboration between them and minimizing the overall mission time.

III. METHOD

Our goal is the efficient mapping of a bounded 3D outdoor space using a team of drones equipped with one monocular camera each. We achieve this by developing a system that computes smooth and straight flights for the drones to reduce the execution time of a mission. These trajectories are dubbed *sweeps*, as the maneuvers can be executed at higher speeds and do not require changing the flight direction.

In order to follow good practices in MVS reconstruction, we also search for trajectories that yield fronto-parallel views of the scene surfaces to maximize the scene coverage and quality of a posterior reconstruction.

A. System overview

Our planner is illustrated in Figure 3 and the results at different steps of the pipeline are shown in Figure 2. First, an initial down-looking (nadir) flight over the area is performed by the drones (Figure. 2a). The aim of this reconnaissance flight plan is twofold: to capture a large portion of the top view of the area of interest flying at high speeds, and to obtain a global overview of the scene online. This enables better-informed reasoning over the subsequent drone trajectories to complete the coverage due to the detection of missing and poorly observed surfaces on the map (Figure. 2b). These surfaces are then grouped into clusters by a novel perception-aware clustering algorithm (Figure. 2c), favoring the generation of flights that sweep the scene to better capture these surfaces with efficient maneuvers (Figure. 2d). The next step computes global paths for all drones participating in the mission, aiming to minimize the distance traveled and the duration of the mission. This is achieved with a variation of the classical Vehicle Routing Problem (VRP), assigning surface clusters to the drones (Figure. 2e). The processing of the initial map and the global plan is performed by a central server that integrates the measurements obtained in the initial reconnaissance flight. Finally, the flight plans are assigned to the drones and a trajectory planner guides the drones smoothly along the sweeps to obtain new relevant

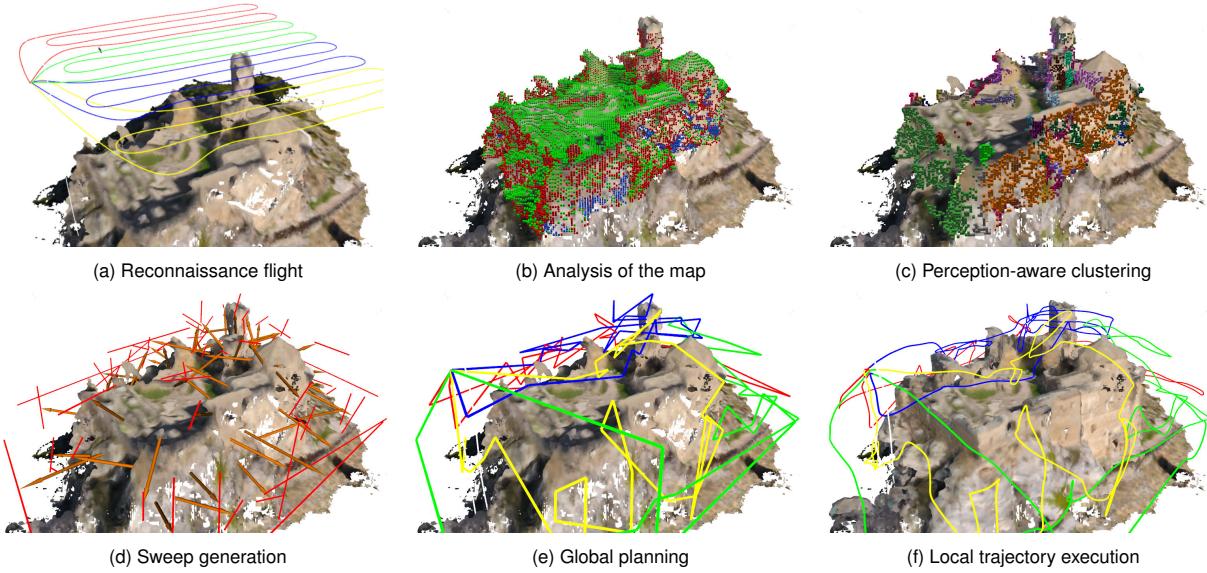


Fig. 2: The drones perform a down-looking flight to compute online a coarse initial map shown in (a), which is used to detect poorly observed or missing areas visualized in (b); red voxels correspond to surfaces seen from an oblique point of view (i.e., poorly observed) and blue voxels represent missing areas. Using perception-aware clustering these missing areas get clustered, shown in different colors in (c). The clusters are used to compute *sweeps*, visualized in (d), to observe them efficiently. The orange arrows represent the surface normals and red lines, the computed sweeps. The global paths of each drone are shown in (e), as computed by a VRP aiming to minimize the mission time and favor longer sweeps. These get smoothed out by a local planner to result in the final drone trajectories seen in (f).

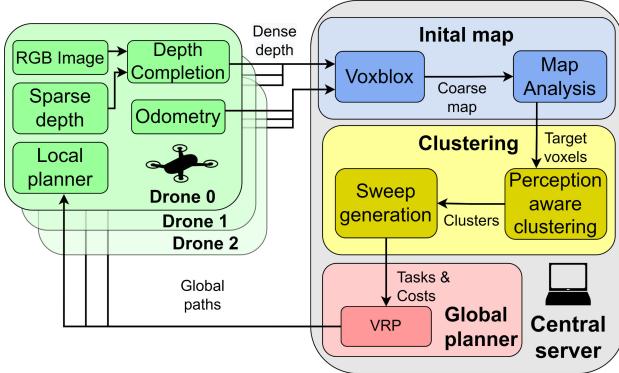


Fig. 3: Proposed pipeline. The drones send measurements for the initial map integration to a central server. This processes the information to generate an efficient plan for the team of drones, which is communicated back to the drones.

views of the scene (Fig. 2f). This execution is carried out without the need of exchanging information with the server or between the drones, favoring the deployment of small and low-powered platforms. In practice, one run of the pipeline is enough to cover most of the scene. Only complex concave surfaces, galleries, and narrow passages could remain unexplored as they are not detected from the top of the scene. A possible way to explore them would be to integrate the local plans observations into the initial map to repeat the process until the whole scene is covered.

B. Initial map

The reconnaissance flight captures top views of the scene to obtain a first approximation of the map quickly. However, the high altitude, together with the use of monocular cameras onboard the drones render the generation of this map challenging without the use of MVS expensive reconstruction methods. To compute it online, we use a depth completion

system [21] onboard the drones that provides dense depth measurements from a sparse input, e.g., SLAM.

The depth measurements are integrated into a common voxel-based Truncated Signed Distance Field (TSDF) map, that incrementally builds a Euclidean Signed Distance Field (ESDF) map [22], \mathcal{M} . Voxels are organized in a uniform grid, where each voxel, $m \in \mathcal{M}$, contains a distance, d_m , to the closest surface and a weight, w_m , that contains the confidence about the depth measurement of that voxel. Moreover, we denote by \mathbf{p}_m the centroid of the voxel and \mathbf{n}_m its normal vector. Voxels that do not have any measurement have an associated weight equal to w_0 .

The initial map is analyzed in order to detect voxels that require additional observations. In particular, voxels that belong to a poorly observed surface, \mathcal{M}_s , and voxels without measurements (i.e., are unobserved), \mathcal{M}_u .

Surfaces are identified locating the voxels that satisfy

$$w_m > w_0 \text{ and } |d_m| < d_v , \quad (1)$$

where d_v is the voxel size.

Aligning the sensor's depth direction with the surface normal, as shown in Figure 4, is key in enabling accurate and high-quality scene reconstructions. With this in mind, we identify poorly observed surface voxels, \mathcal{M}_s , as

$$-\mathbf{o}_m \cdot \mathbf{n}_m > \cos(\theta_t) , \quad (2)$$

where \mathbf{o}_m is the observation direction of the camera for the voxel and θ_t is the threshold angle to consider the observation of the surface valid. We consider $\theta_t = 45^\circ$ as a good indication that the visibility of a surface is poor. During the initial flight, the cameras are looking downward (i.e., $-Z$ axis). Thus, vertical and oblique surfaces are considered

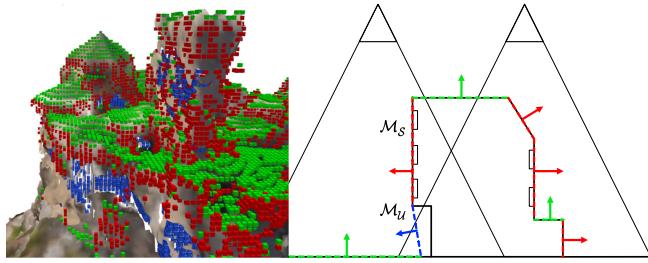


Fig. 4: The analysis of the initial map, visualized from a side view on the right with two down-looking cameras, indicates the quality of the views of a building. An example on a map obtained in the reconnaissance flight is shown on the left. Voxels on the left are visualized as dashed lines on the right, with arrows indicating the estimated surface normals. Red and green indicate poorly and well-captured surfaces, respectively, while blue indicates accessible unknown areas, whose normals are estimated to point towards free space.

poorly observed, while horizontal or low tilted surfaces are considered as correctly observed.

The second step is the analysis of the unobserved voxels. Out of all the unobserved voxels in the map, with weight equal to w_0 , we find those that are accessible (i.e., can be observed). Unobserved voxels are accessible if they are surrounded by free space voxels, m_f , defined by

$$w_m > w_0 \text{ and } d_m > d_v. \quad (3)$$

The accessible unobserved voxels, \mathcal{M}_u , are then formalized as the voxels, such that

$$\exists m_f \in \mathcal{N}_{26}(m), \quad (4)$$

where $\mathcal{N}_{26}(m)$ is the set of 26-connected neighbors, around the voxel m . Finally, the set of voxels that need further observations is defined as

$$\mathcal{M}_t = \mathcal{M}_s \cup \mathcal{M}_u. \quad (5)$$

C. Perception-aware clustering

This step performs a novel perception-aware clustering over \mathcal{M}_t . In particular, voxels get grouped together, such that can be observed by a drone in a single efficient *sweep* trajectory by considering the distribution of their normals in the cluster. This clustering also aims at generating a natural partition of the scene into a set of *tasks* that can be assigned to a team of drones. In the following, we explain how the clustering works and how sweep paths are generated from them.

The proposed clustering is based on the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) method [23]. The basic method groups voxels¹ that are close together in space, and identifies isolated voxels in low-density regions as noise. It works by iteratively expanding clusters, \mathcal{C}_i , to neighboring voxels that fulfill the following *density* condition:

$$|\mathcal{N}_\sigma(\mathbf{p}_m)| > \epsilon, \quad (6)$$

where $|\mathcal{N}_\sigma(\mathbf{p}_m)|$ is the number of neighboring voxels in a radius σ of the voxel's center, \mathbf{p}_m , and ϵ is the minimum number of neighbors to include the voxel in that cluster.

¹The original method refers to points.

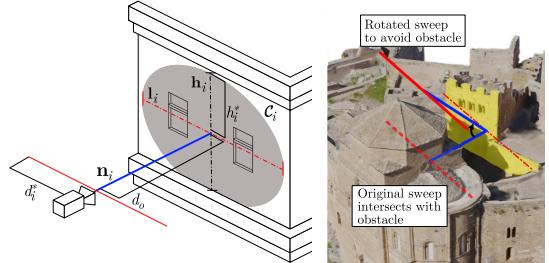


Fig. 5: Sweep definition and refinement scheme (left). The gray area represents a surface cluster. The dashed red line is the major eigenvector that will be covered by the sweep (red solid line). The blue vector is the normal. An example in a real map is shown on the right where the observation direction, \mathbf{n}_i , was adjusted to avoid an obstacle.

Our goal is to group regions observable from a similar point of view (i.e., surfaces). Thus, we extend DBSCAN by adding a second condition for expansion. This condition checks if the normal of a candidate voxel, \mathbf{n}_m , lies within the distribution of normals in the cluster. The normals in \mathcal{M}_s are estimated from the gradient of distances in the ESDF initial map. The normals of unobserved voxels are computed as the average of all the directions that lead from \mathbf{p}_m to free space voxels in $\mathcal{N}_{26}(m)$ (Figure 4). We also smooth the estimated normals using neighboring values to filter noise.

In particular, we focus on the distribution of the cosine distance with respect to the mean normal of the cluster, \mathbf{n}_c ,

$$d_\alpha(\mathbf{n}_m, \mathbf{n}_c) = 1 - \frac{\mathbf{n}_m \cdot \mathbf{n}_c}{\|\mathbf{n}_m\| \|\mathbf{n}_c\|}. \quad (7)$$

We then compute the average $\mu_d(\mathcal{C}_i)$, and standard deviation $\sigma_d(\mathcal{C}_i)$ of the distances from all the normals of the voxels in the cluster to \mathbf{n}_c . The *normal direction* condition checks that the distance of the normal between the candidate voxel and the cluster's distribution is sufficiently small,

$$d_\alpha(\mathbf{n}_m, \mathbf{n}_c) < \min(\mu_d(\mathcal{C}_i) + 2\sigma_d(\mathcal{C}_i), \tau). \quad (8)$$

where τ is a fixed value.

We identify $\mu_d(\mathcal{C}_i) + 2\sigma_d(\mathcal{C}_i)$ as the relative tolerance to the cluster's distribution and τ as the absolute tolerance. The aim of the relative tolerance is to adapt the expansion of the cluster to the surface in question, e.g., allowing soft curvatures. On the other hand, the absolute tolerance avoids the cluster to expand through discontinuities such as edges.

Finally, we perform a merging step that fuses small clusters with the most similar neighbor. If no neighbor is found, these voxels are discarded.

Considering that each voxel cluster resembles a surface, a *sweep* is defined as a linear trajectory that is orthogonal to the normal of the cluster (Figure 5). Among all the possible sweeps, we find the longest one through the inertia moments of the cluster, \mathbf{l}_i . Then, for each voxel in the cluster, we compute the longest distance from the center, projected on this axis,

$$d_i^* = \max_{m \in \mathcal{C}_i} |\mathbf{l}_i^T (\mathbf{p}_m - \bar{\mathbf{c}}_i)|, \quad (9)$$

where \mathbf{p}_m is the centroid of the voxel and $\bar{\mathbf{c}}_i$ the centroid of the cluster. The extension of this distance from the centroid of the cluster in both directions of \mathbf{l}_i generates the path that

traverses the cluster through its length. We name both ends of this path, the *entrance* points of the cluster.

In order to guarantee that the whole surface is visible with a single sweep, we compute its height in the direction of the axis perpendicular to the sweep direction

$$\mathbf{h}_i = \mathbf{l}_i \times \mathbf{n}_i. \quad (10)$$

The value of the height is computed in the same way as (9) using the axis \mathbf{h}_i instead:

$$h_i^* = \max_{m \in \mathcal{C}_i} |\mathbf{h}_i^T (\mathbf{p}_m - \bar{\mathbf{c}}_i)|, \quad (11)$$

where h_i^* is the half height of the cluster. Then, we use the relationship between the field of view (FoV) angle of the camera and h_i^* to compute the distance that is able to cover the height of the cluster. The observation distance, d_o , along the normal is computed as

$$d_o = \frac{h_i^*}{\tan(\frac{\text{FoV}}{2})} \quad (12)$$

Finally, if the sweep intersects an obstacle we perform a rotation of the observation direction to refine it (Figure 5).

D. Global planner

In the next step, the objective is to compute high-level paths for the drones to cover all the clusters. We propose to solve this problem with an adaptation of the min-max Vehicle Routing Problem (VRP).

Originally, this algorithm looks for optimal routes for a set of agents, K , that visit once all the locations of a given set, V . Denote by c_{ij} the cost to go from location i to location j , which we consider is the same for all the agents, and define $\mathcal{X} = \{x_{ij}^k\}$, for $i, j \in V$, and $k \in K$, the set of binary variables that indicate whether agent k has traverse the route from i to j or not. Then, the min-max VRP solves

$$\min_{\mathcal{X}} \max_{k \in K} \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij}^k, \quad s.t. \quad (13a)$$

$$\sum_{k \in K} \sum_{i \in V} x_{ij}^k = 1 \quad \forall j \in V \setminus \{0\} \quad (13b)$$

$$\sum_{k \in K} \sum_{j \in V} x_{ij}^k = 1 \quad \forall i \in V \setminus \{0\} \quad (13c)$$

$$\sum_{k \in K} \sum_{i \in V} x_{i0}^k = \sum_{j \in V} \sum_{k \in K} x_{0j}^k = |K| \quad (13d)$$

$$\sum_{i,j \in S} x_{i,j}^k \leq |S| - 1, \quad \forall S \subset V \setminus \{0\}, S \neq \emptyset \quad (13e)$$

$$x_{ij}^k \in \{0, 1\} \quad \forall i, j \in V \quad (13f)$$

where (13a) is the cost function, which denotes the largest cost among all the agents for a given assignment, constraints (13b) and (13c) indicate that drones only visit each location once. Constraints in (13d) impose the drones to start and end at the initial point. Constraints (13e) are the sub-tour elimination constraints. Finally, conditions (13f) impose binary conditions on the decision variables.

In order to adapt the VRP to the clusters and their *sweeps*, we propose a definition of the costs, c_{ij} , that considers them. Given two clusters, i and j , we compute the path between them, as the line that joins their closest entrance points with distance, d_{ij} , if there are no obstacles. In case there are obstacles, we consider the same path, but flying over the top of the scene. This way we guarantee that all the clusters are reachable from each other, but we favor assignments of the nearby ones. Additionally, to account for the cost of covering each cluster, we add the distance of the sweep to all the costs with it as the destination. The distance of the sweep generated for C_i is $l_i^* = 2d_i^*$, with d_i^* defined in (9). Therefore, the cost c_{ij} is defined as

$$c_{ij} = d_{ij} + l_j^*. \quad (14)$$

Lastly, to compute the solution of (13), we consider an implementation with limited capacities. We simplify the objective to minimize the total cost traveled by all the drones

$$\min_{\mathcal{X}} \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij}^k, \quad (15)$$

and we add a capacity constraint for each of them,

$$\sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij}^k < c_{\max} \quad \forall k \in K. \quad (16)$$

Our solution searches for the minimum value of c_{\max}^* that solves the problem using the bisection method.

E. Local planner

For the last step of the proposed pipeline, the local planner by Zhou et al. [24] is used to plan in two stages: an initial kinodynamic A* path search based on motion primitives finds a safe, feasible and minimum-time initial path, and a B-spline optimization generates smooth and collision-free trajectories that use gradient information from the ESDF and dynamic constraints.

In order to cover a surface efficiently and effectively, the sweep direction needs to be orthogonal to the observation vector. To enable safe and efficient navigation, while obtaining high-quality scene observations, we decouple the problems of navigation and observation. We assume that the observation camera is mounted on an actuated gimbal, which is able to set the yaw and pitch directions. A second sensor, such as a laser ranger or a depth camera is used for navigation.

IV. EXPERIMENTS AND RESULTS

To assess the performance of the proposed method, the pipeline is run on photo-realistic outdoor scenarios of varying sizes and difficulty, namely on the *Bunker*, *Wood Bridge*, *Loarre Castle*, and *Zurich*² models visible in Figure 8. The transfer of this simulation setup to real-world cases was proved in previous work [21] [25]. The Gazebo RotorS simulator is used with ground-truth odometry of the drones. During the initial map construction, flying at a high altitude

²Model provided by wingtra.com

TABLE I: Execution times to complete a scene coverage mission. The reconnaissance flight time, in parenthesis, is included in the total time. *Ours (single)* refers to our pipeline using one drone, while *Ours (multi)* indicates the time taken by the longest flight of any drone in a team (four in this case), indicating the end of the mission. For [2], we report the time to reach the same extent of coverage achieved by each of our methods (Table III). In larger maps, [2] is not able to achieve our coverage after one hour of execution, so the total coverage by that time is reported. The '*' indicates that the global planner only assigned two drones to this map, as introducing more would not reduce the total time.

Method	Bunker	Wood Bridge	Loarre Castle	Zurich
Kompis et al. [2] (single)	859.71 s	897.08 s	>3600 s [67.08%]	>3600 s [13.91%]
Ours (single)	491.91 s (183.06 s)	331.16 s (122.54 s)	1474.95 s (329.88 s)	2027.88 s (588.56 s)
Kompis et al. [2] (multi)	214.43 s	405.98 s	1440.59 s	>3600 s [60.49%]
Ours (multi)	126.26 s (42.04 s)	172.09* s (53.31 s)	433.74 s (117.11 s)	741.07 s (269.53 s)

enables the use of accurate RTK GPS systems with small odometry errors. The uncertainty in the successive flights can be alleviated by overestimating the observation distance and safety radius. As we target our application to consumer platforms, problems such as aerodynamics or other electrical and mechanical delays are assumed to be solved by their system. The drones are equipped with a monocular camera mounted on an actuated gimbal that can rotate independently of the orientation of the drone. Its resolution is 752×480 and FoV is $80^\circ \times 55^\circ$. The drones' linear and angular maximum velocity and acceleration are set to 2 m s^{-1} and 0.9 m s^{-2} , respectively, for fairness with the compared system and to ensure safety at all times. During the reconnaissance flight, the drones fly at a fixed height over the model in a grid pattern with their cameras looking downward. The voxel size used for the initial map and planning is 0.2 (*Bunker* and *Wood Bridge*), 0.5 (*Loarre*) and 0.7 (*Zurich*).

The parameters for the clustering step depend on the resolution of the prior map (i.e., voxel size v_s). We set $\epsilon = 10v_s$, $\sigma = 6$ (Eq. (6)) and $\tau = 0.4$. We also apply an inflation factor over the coarse map of $20v_s$ to the observation and safety distances for the sweep generation. Due to computational resources required to simulate several drones, the local paths are executed by a single drone sequentially, which starts from and comes back to the same initial point. The simulation runs until all the local trajectories have been executed.

We run the experiments considering three different algorithms. We name *Ours (single)* and *Ours (multi)* the solutions obtained running our pipeline with one and four drones respectively. In the *multi* version, we perform an ablation study to show the difference in visual coverage obtained after the reconnaissance flight and the successive flights resulting from our pipeline. Even when our pipeline is not directly comparable in terms of the sensor setup with other exploration methods that use stereo pairs, the third method uses the planning approach of Kompis et al. [2] for single and multiple drones, which is among the state-of-the-art planners with available implementation. In the version with multiple drones, the environment is segmented equally among the

TABLE II: Computation time for the different map processing steps: analysis, clustering and global planner. Mean and standard deviation for 10 runs.

		Bunker	Zurich
Analysis	-	2.22 ± 0.082 s	6.96 ± 1.05 s
Clustering	-	1.83 ± 0.045 s	7.12 ± 0.297 s
Global planner	Ours (single)	0.99 ± 0.053 s	15.43 ± 0.689 s
	Ours (multi)	0.94 ± 0.064 s	17.10 ± 0.737 s

drones. This comparison is not intended to rank the two methods but to showcase the potential advantages of the proposed planning approach in terms of efficiency.

A. Planning efficiency

The times for the execution of the plan are shown in Table I. The times for the reconnaissance flight and initial map construction with Voxblox are included in the total and shown below. For the method of Kompis et al., we report the times necessary to achieve the same coverage as our system.

The results for *Ours (single)* and *Ours (multi)* validate that our setup can generalize to an arbitrary number of drones. When using several drones instead of one, times are a fraction of the number of drones with little overhead. In the case of *Wood bridge*, the global planner assigned the tasks to only two drones even if four were available. Due to the scene structure, adding more drones would not reduce the time of the mission as drones would have to return to the initial point. Compared to Kompis et al., our method is able to completely cover the maps faster in every case. For large maps (i.e., *Loarre Castle* and *Zurich*), [2] is not able to cover the environment after one hour of execution and we report the amount of coverage obtained at that time.

There are two main reasons for this difference. Firstly, the different approaches to drone dynamics in the planners. Stop-and-go motions are necessary as the exploration process is incremental. This limits the planning horizon of the system to a local region. In their approach, the drone has to stop in order to acquire each individual view and plan the next (see Figure 6). In our case, the drone is able to keep moving while observing a whole surface in a sweep. Notice that our system could potentially use higher velocities and accelerations for large trajectories in free space, as in the case of the reconnaissance flight, further improving the planning efficiency. The second reason for the time difference is that their planner revisits areas in order to obtain thorough coverage, committing resources to small regions with difficult accessibility. The reason for their low coverage results in the *Zurich* map is explained by their viewpoint proposal method, which leads to larger re-planning times when the scale of the map grows.

We also report the time for the initial map processing. The times for the analysis and clustering steps depend on the size and resolution of the map. The global planning step depends on the number of generated clusters and the number of agents. We show the results in Table II in the smallest and biggest maps: *Bunker* and *Zurich*. The time is always

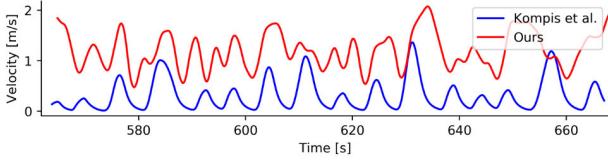


Fig. 6: Extract of the moving average for the velocity during the simulation in *Loarre Castle* for *Ours* and [2]. While traditional methods stop to capture a view and plan the next goal, our method is able to keep flying at higher speed.

TABLE III: For each method, we report the RMSE of the reconstructions and the extent of the coverage for a threshold of 0.1 meters at the completion time of the experiment as reported in Table I. Recon. indicates the metrics from a reconstruction using only the images captured during the reconnaissance flights.

Method	Bunker	Wood Bridge	Loarre Castle	Zurich
Kompis et al. [2] (single)	0.085 m 75.6 %	0.068 m 55.63 %	0.049 m 42.58 %	0.074 m 9.13 %
Ours (single)	0.027 m 97.35 %	0.043 m 93.23 %	0.048 m 97.92 %	0.09 m 95.96 %
Kompis et al. [2] (multi)	0.076 m 89.10 %	0.074 m 60.29 %	0.059 m 50.20 %	0.087 m 20.90 %
Recon. (ablation)	0.04 m 84.75 %	0.039 m 62.93 %	0.063 m 88.54 %	0.147 m 75.34 %
Ours (multi)	0.026 m 96.36 %	0.039 m 92.37 %	0.043 m 98.64 %	0.086 m 97.34 %

below one minute which is negligible for the total time of the mission.

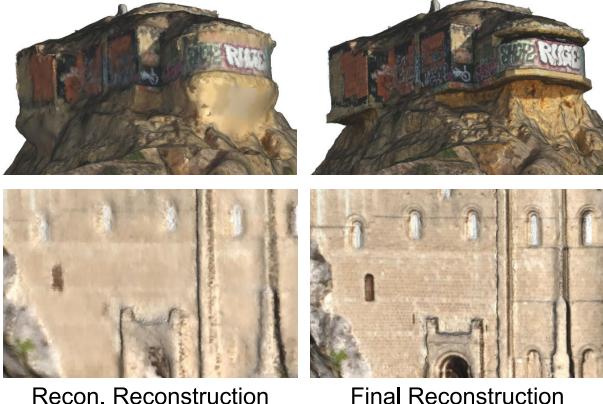


Fig. 7: Comparison of the coverage quality after the reconnaissance flight (left) and the successive flights (right). Occluded regions under the *Bunker* are not reconstructed (up). In addition, even though vertical surfaces such as *Loarre*'s walls are covered, their observation yields poor scene reconstructions (down).

B. Coverage and surface quality

Besides the efficiency of our planner, we have also assessed that the coverage and the quality of the views are correct. The images captured from the monocular cameras of the drones have been used to generate 3D reconstructions of the scenes using COLMAP. The reconstructed models are compared with the ground-truth (GT) virtual models. We consider that a point in the GT surface is covered if the

closest distance to a point from the reconstructed mesh is below a threshold of 0.1m. Our metric is the percentage of covered points in the ground-truth mesh. We also measure the accuracy of the reconstruction as the RMSE of the distances from the reconstructed model to the ground truth mesh. While we are mainly interested in the first two metrics, the accuracy indicates that our method can be used to obtain accurate 3D reconstructions of the environment. We also report the coverage, its quality, and the reconstruction accuracy from the Voxblox generated mesh of the pipeline in [2]. The voxel size for their reconstruction is the same they use in their experiments, 0.1, which is the threshold used for considering a point covered in our setup. The results are reported in Table III. For Kompis et al., the reported value is the coverage achieved by the completion time of our plan.

The reconnaissance flight (*Recon.*) is able to cover a large amount of surface. However, the coverage quality is low, yielding poor scene reconstructions (Figure 7). After the execution of our pipeline, we obtain images that ensure good observation of surfaces. We can see similar coverage for the case of single and multi-drone approaches as the drones traverse similar sweeps. Compared to Kompis et al., our system is able to achieve more coverage in less time. Notice how the coverage difference is increased with the size of the map. Qualitative results are shown in Figure 8 for all the maps. It might be seen that our pipeline misses some areas with difficult accessibility. In return, it is able to cover the overall scene in a fraction of the time. This demonstrates that a substantial amount of information can be extracted from the map by planning more efficiently and shows the advantage of using prior knowledge about the scene structure for planning.

V. CONCLUSION

In order to improve the efficiency in large-scale deployments of drones for visual coverage, this article proposes a multi-stage planner that generates long linear trajectories (*sweeps*) that observe a large amount of surface in a continuous motion. We accomplish this by leveraging a prior coarse map to cluster these surfaces and improve the posterior coverage trajectories. This approach is generalized to an arbitrary number of drones, managing the workload distribution between them in order to minimize the completion time of the mission. Comparison with alternative approaches to the exploration of scenes shows the advantages of our pipeline for large scenarios, where the overall coverage of the scene in a minimal amount of time is necessary. We show that a single run of our pipeline is able to obtain coverage of scenes faster and with great accuracy.

Future work will explore the integration of the proposed pipeline in a real platform, including a mapping framework to ensure safe local navigation and additional coordination systems to deploy a team of autonomous drones in large-scale environments. Besides, exploring the extension to a team of heterogeneous aerial drones (i.e., fixed-wing UAVs for the nadir flight) could improve even further the efficiency of the system by allocating each to different task modalities.

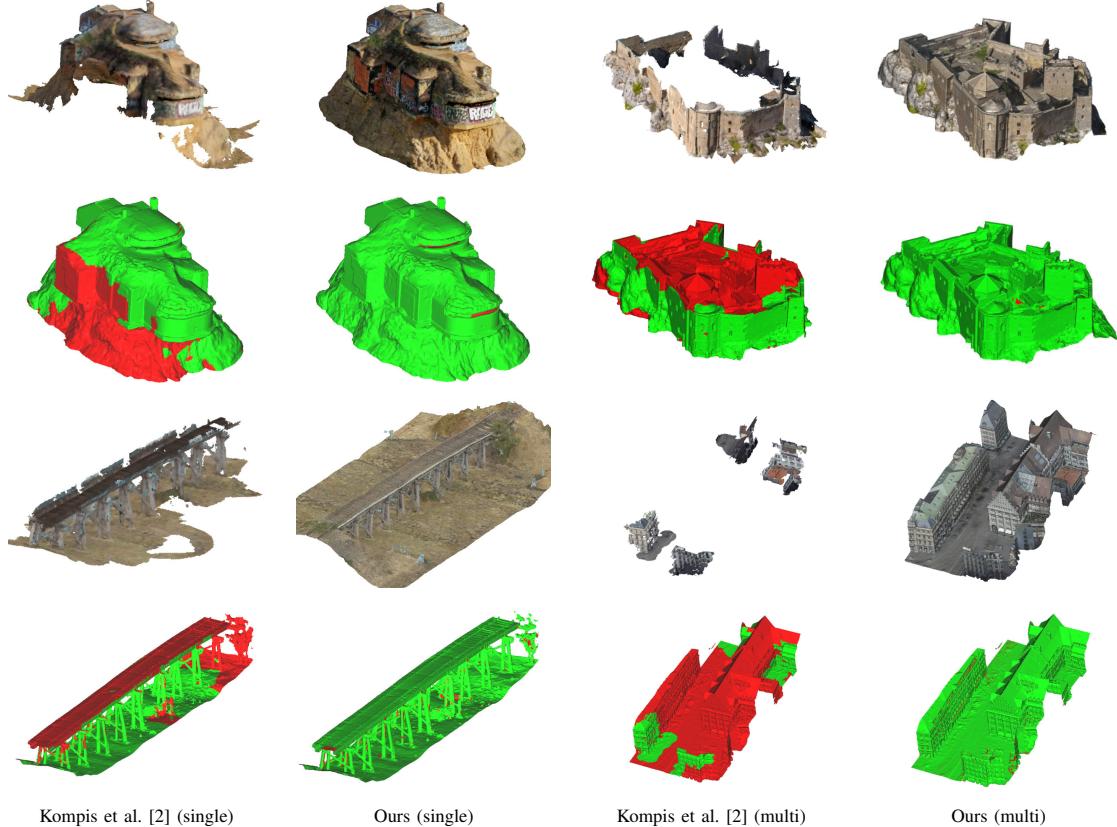


Fig. 8: Qualitative comparison of the coverage obtained for all the maps considering a fixed time. In green are points in the ground truth (GT) mesh that have been covered during the mission, while red indicates the opposite. Our planner is able to obtain more coverage of the overall scene, despite missing some some details in inaccessible/non-directly visible surfaces. Detailed numbers of the coverage and accuracy of the reconstructions are provided in Table III.

REFERENCES

- [1] B. Zhou, Y. Zhang, X. Chen, and S. Shen, "Fuel: Fast uav exploration using incremental frontier structure and hierarchical planning," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 779–786, 2021.
- [2] 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, 2021.
- [3] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *IEEE Int. Symp. on Computational Intelligence in Robotics and Automation*, 1997, pp. 146–151.
- [4] T. Cieslewski, E. Kaufmann, and D. Scaramuzza, "Rapid exploration with multi-rotors: A frontier selection method for high speed flight," in *IEEE Int. Conf. on Intelligent Robots and Systems*, 2017, pp. 2135–2142.
- [5] 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, 2022.
- [6] H. Carrillo, I. Reid, and J. A. Castellanos, "On the comparison of uncertainty criteria for active slam," in *IEEE Int. Conf. on Robotics and Automation*, 2012, pp. 2080–2087.
- [7] L. Carlone, J. Du, M. Kaouk Ng, B. Bona, and M. Indri, "Active slam and exploration with particle filters using kullback-leibler divergence," *J. of Intelligent & Robotic Systems*, vol. 75, no. 2, pp. 291–311, 2014.
- [8] D. S. Chaplot, D. Gandhi, S. Gupta, A. Gupta, and R. Salakhutdinov, "Learning to explore using active neural slam," in *IEEE Int. Conf. on Learning Representations (ICLR)*, 2020.
- [9] G. Georgakis, B. Bucher, A. Arapin, K. Schmeckpeper, N. Matni, and K. Daniilidis, "Uncertainty-driven planner for exploration and navigation," in *IEEE Int. Conf. on Robotics and Automation*, 2022, pp. 11295–11302.
- [10] L. Schmid, M. Pantic, R. Khanna, L. Ott, R. Siegwart, and J. Nieto, "An efficient sampling-based method for online informative path planning in unknown environments," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1500–1507, 2020.
- [11] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, "Receding horizon "next-best-view" planner for 3d exploration," in *IEEE Int. Conf. on Robotics and Automation*, 2016, pp. 1462–1468.
- [12] G. Hardouin, F. Morbidi, J. Moras, J. Marzat, and E. M. Mouaddib, "Surface-driven Next-Best-View planning for exploration of large-scale 3D environments," in *IFAC World Congress*, Jul. 2020.
- [13] S. Song, D. Kim, and S. Choi, "View path planning via online multiview stereo for 3-d modeling of large-scale structures," *IEEE Transactions on Robotics*, vol. 38, no. 1, pp. 372–390, 2022.
- [14] M. Roberts, D. Dey, A. Truong, S. Sinha, S. Shah, A. Kapoor, P. Hanrahan, and N. Joshi, "Submodular trajectory optimization for aerial 3d scanning," in *Int. Conf. on Computer Vision*, 2017, pp. 5324–5333.
- [15] N. Smith, N. Moehrle, M. Goesele, and W. Heidrich, "Aerial path planning for urban scene reconstruction: A continuous optimization method and benchmark," *ACM Trans. Graph.*, vol. 37, no. 6, pp. 183/1–15, 2018.
- [16] B. Hepp, M. Nießner, and O. Hilliges, "Plan3d: Viewpoint and trajectory optimization for aerial multi-view stereo reconstruction," *ACM Trans. Graph.*, vol. 38, no. 1, 2018.
- [17] A. Mannucci, S. Nardi, and L. Pallottino, "Autonomous 3d exploration of large areas: A cooperative frontier-based approach," in *Modelling and Simulation for Autonomous Systems*, 2018, pp. 18–39.
- [18] R. G. Colares and L. Chaimowicz, "The next frontier: Combining information gain and distance cost for decentralized multi-robot exploration," in *ACM Symposium on Applied Computing*, 2016, p. 268–274.
- [19] G. Hardouin, J. Moras, F. Morbidi, J. Marzat, and E. M. Mouaddib, "Next-best-view planning for surface reconstruction of large-scale 3d environments with multiple uavs," in *IEEE Int. Conf. on Intelligent Robots and Systems*, 2020, pp. 1567–1574.
- [20] A. Dutta, A. Bhattacharya, O. P. Kreidl, A. Ghosh, and P. Dasgupta, "Multi-robot informative path planning in unknown environments through continuous region partitioning," *Int. Journ. of Advanced Robotic Systems*, vol. 17, no. 6, 2020.
- [21] L. Teixeira, M. R. Oswald, M. Pollefeys, and M. Chli, "Aerial single-view depth completion with image-guided uncertainty estimation,"

IEEE Robotics and Automation Letters, vol. 5, no. 2, pp. 1055–1062, 2020.

[22] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto, “Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning,” in *Int. Conf. on Intelligent Robots and Systems*, 2017, pp. 1366–1373.

[23] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in *Int. Conf. on Knowledge Discovery and Data Mining*, 1996, p. 226–231.

[24] B. Zhou, F. Gao, L. Wang, C. Liu, and S. Shen, “Robust and efficient quadrotor trajectory generation for fast autonomous flight,” *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3529–3536, 2019.

[25] F. Maffra, L. Teixeira, Z. Chen, and M. Chli, “Real-time wide-baseline place recognition using depth completion,” *IEEE Rob. Autom. Lett.*, vol. 4, no. 2, pp. 1525–1532, 2019.