



HAL
open science

Combining Stochastic Optimization and Frontiers for Aerial Multi-Robot Exploration of 3D Terrains

Alessandro Renzaglia, Jilles Dibangoye, Vincent Le Doze, Olivier Simonin

► **To cite this version:**

Alessandro Renzaglia, Jilles Dibangoye, Vincent Le Doze, Olivier Simonin. Combining Stochastic Optimization and Frontiers for Aerial Multi-Robot Exploration of 3D Terrains. IROS 2019 - IEEE/RSJ International Conference on Intelligent Robots and Systems, Nov 2019, Macau, China. hal-02164806v2

HAL Id: hal-02164806

<https://inria.hal.science/hal-02164806v2>

Submitted on 1 Apr 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Combining Stochastic Optimization and Frontiers for Aerial Multi-Robot Exploration of 3D Terrains

Alessandro Renzaglia^{1,2}, Jilles Dibangoye², Vincent Le Doze² and Olivier Simonin²

Abstract—This paper addresses the problem of exploring unknown terrains with a fleet of cooperating aerial vehicles. We present a novel decentralized approach which alternates gradient-free stochastic optimization and a frontier-based approach. Our method allows each robot to generate its trajectory based on the collected data and the local map built integrating the information shared by its teammates. Whenever a local optimum is reached, which corresponds to a location surrounded by already explored areas, the algorithm identifies the closest frontier to get over it and restarts the local optimization. Its low computational cost, the capability to deal with constraints and the decentralized decision-making make it particularly suitable for multi-robot applications in complex 3D environments. Simulation results show that our approach generates feasible trajectories which drive multiple robots to completely explore realistic environments. Furthermore, in terms of exploration time, our algorithm significantly outperforms a standard solution based on closest frontier points while providing similar performances compared to a computationally more expensive centralized greedy solution.

I. INTRODUCTION

Autonomous exploration is one of the most common tasks in which mobile robots can have a crucial impact on helping humans in complex and dangerous scenarios. In particular, the rapid progress in the development of Unmanned Aerial Vehicles (UAVs) has allowed imagining more and more challenging missions where the exploration of large complex terrains, otherwise impossible with standard ground vehicles, is now viable. However, solutions for these scenarios must also face the strong constraints and limitations of nowadays small aerial robotics platforms: limited computational capabilities and reduced batteries lifetimes make the design of solutions with little computational demands of fundamental importance.

The final goal in autonomous exploration tasks is to completely cover an initially unknown environment in a minimum time. To this end, each robot needs to identify its next position with the aim of maximizing the expected gain in the explored area. As in every multi-robot mission, the coordination with the other robots plays a crucial role [1], leading to challenging problems where the goal is to reduce redundant information within teammates to speed up the exploration process. Common solutions for the exploration problem are based on the concept of frontiers, i.e., cells dividing the explored region from the still unknown part [2]. However, most of these solutions do not satisfy the

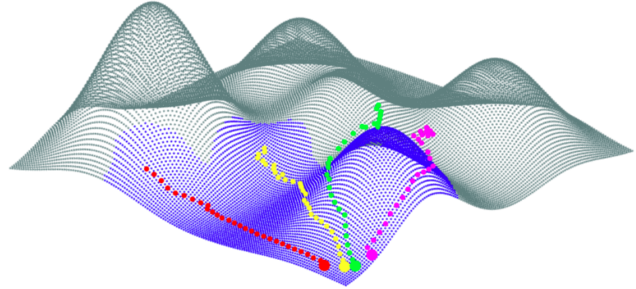


Fig. 1: Four UAVs start their mission to explore an unknown environment.

requirements previously listed for small robotic platforms exploring 3D environments. Some of them need to solve complex centralized optimization problems to assign frontiers to each component of the team and, even considering easier assignation strategies, they all require to regularly identify frontier points and their distances, which for 3D terrains can rapidly become a very complex operation.

To overcome these limitations, we propose a new approach where robots are driven by a local optimizer which exploits on-line collected data to maximize the quantity of new explored environment, and a frontier-based strategy to guaranty the completeness of the exploration. The optimization is carried on by a stochastic optimization algorithm, running independently on each robot and relying on the map information shared within the team. Being the optimization strictly local, robots can, however, get stuck in local optima before completing the task. When they reach such a situation, they exploit the global map to identify a frontier point as next goal before restarting a local exploration. The resulting hybrid method, combining stochastic local optimization and frontier points, overcomes the limits that these approaches have separately while maintaining their advantages and so resulting in a more efficient solution.

The rest of the paper is organized as follows. The next section briefly reviews the related work, with a special attention to multi-robot exploration of non-planar environments. In Section III, we formalize the problem of multi-robot exploration in terms of an optimization problem. Section IV presents the details of the proposed solution, describing the adopted local optimization and the frontier-based approach. Finally, in Section V, simulation results to test and evaluate the algorithm are provided.

¹ Univ. Grenoble Alpes, Inria, F-38000 Grenoble, France; e-mail: firstname.lastname@inria.fr

² INSA Lyon, CITI Lab, Inria, Chroma, Lyon, France; e-mail: firstname.lastname@inria.fr

II. RELATED WORK

The autonomous exploration problem has been widely studied in the last years and providing a rich overview of the existing literature is not an easy task. With this intent, [3] presents an extensive study and comparison of some of the main proposed solutions. Another interesting survey, with an attention more focused on communication aspects, can found in [4].

Most of the existing methods for autonomous exploration tasks are based on the concept of frontiers, introduced by Yamauchi in his seminal work, firstly for a single robot [5] and then extended to the multi-robot case [2]. Following this simple but efficient idea, numerous variations and improvements, especially considering the exploration of two-dimensional environments, have been proposed in the following years. An efficient but expensive centralized approach to solve the multi-criteria problem of simultaneously considering travelling costs and gain utilities was presented [6]. In [7], the proposed decentralized approach allocates frontier points based on a rank among teammates, in terms of travel distance to each frontier, to obtain a well balanced spatial distribution of robots in the environment. The problem of frontiers assignment in multi-robot exploration is studied also in [8], where the authors propose a multiple traveling salesman problem formulation considering for each robot all current possible goals. Even though this strategy can provide shorter exploration times with respect to alternative solutions, it also implies a very high computational cost, contrary to our proposed solution. Besides the frontier allocation problem, another important aspect in multi-robot applications can be to comply with communication requirements during the task. In [9], the authors explicitly consider this problem presenting a strategy that allows robots to coordinate with teammates in order to form a network that satisfies recurrent connectivity constraints.

Considering three-dimensional scenes, a direct extension of a simple frontier-based method for this case has been presented in [10]. As previously mentioned, in three dimensions, even the simple process of identifying frontier points becomes more challenging - and costly - and has been a subject of research [11]. Also in [12], the authors present a frontier-based approach to explore three-dimensional scenarios using a team of Micro Aerial Vehicles with embedded vision. The novelty of this work is to propose an exchange of just map frontier points between robots instead of the whole grid map in order to save communication bandwidth.

The solution we propose in this paper has the intent of overcoming the constant dependence on frontier points at every decision step, with the goal of reducing the total computational cost. The use of frontiers is, in fact, limited thanks to the combination with a decentralized local optimization algorithm. The resulting solution guides the robots towards a position which maximizes the expected information gain, while maintaining guarantees of completing the exploration task.

III. MULTI-ROBOT EXPLORATION

The multi-robot planning problem considered in this paper aims at autonomously exploring, as fast as possible, an unknown terrain represented as a surface \mathcal{S} in a 3D environment with a fleet of UAVs. Formally, this problem is defined by a tuple $(N, \mathcal{S}, \mathcal{P}, \mathcal{C}, E)$ where:

- N is the number of UAVs $r \in \{1, 2, \dots, N\}$;
- \mathcal{S} is an *unknown* set of 3D points (voxels) denoted $s \in \mathcal{S}$;
- \mathcal{P} is the set of locations \mathbf{P} , with $\mathbf{P}_t^{(r)} \doteq [x_{1,t}^{(r)}, x_{2,t}^{(r)}, x_{3,t}^{(r)}]$ being the position of the UAV r at time step t ;
- $\mathcal{C}(\mathbf{P}^{(r)}) \preceq 0$ defines *partially unknown* constraints each UAV position $\mathbf{P}^{(r)}$ must satisfy so that robot-to-obstacle and robot-to-robot collision avoidance as well as minimum and maximum height of flight constraints are met;
- $E_t(\mathcal{S})$ denotes the portion of the surface \mathcal{S} explored up to time step t . We consider a point s as explored by a UAV if: (1) they are connected by a line-of-sight and (2) they are at a distance smaller than a given sensing range.

Let $e(s)$ be the subset of UAVs who explored for the first time the voxel s , i.e.

$$e(s) = \{r \in \mathcal{R} \mid s \in E_t(\mathcal{S}) \setminus E_{t-1}(\mathcal{S}), O(s, \mathbf{P}_t^{(r)}) = 1\}$$

where $O(s, \mathbf{P})$ is true when the point s is observed by a UAV with position \mathbf{P} .

Furthermore, the frontier points are defined as the already explored voxels $s \in E(\mathcal{S})$ that are adjacent with unexplored areas, i.e. with $\mathcal{S} \setminus E(\mathcal{S})$.

The goal of this problem is to find the UAVs' trajectories $(\mathbf{P}_t^{(r)})_{r=1, \dots, N; t=1, \dots, t}$ that explore the entire surface \mathcal{S} in a minimum time T , while satisfying the constraints \mathcal{C} . This problem is impossible to solve off-line being the environment initially unknown and a sub-optimal on-line solution is necessary. To this end, we define for each UAV r an optimization function as follows:

$$J_t^{(r)} = \sum_{s \in E_t(\mathcal{S})} \frac{\delta_s^r}{|e(s)|} \text{ where } \delta_s^r = \begin{cases} 1 & \text{if } r \in e(s) \\ 0 & \text{if } r \notin e(s) \end{cases} \quad (1)$$

Maximizing this function, each robot tries at each iteration to maximize the quantity of newly explored voxels s (delta term), while minimizing the redundant information (i.e. simultaneous exploration of new areas by more than one robot). This is due to the term $|e(s)|$, which represents a penalization to reduce the importance of voxels that have been simultaneously sensed also by other robots. In other words, any voxel has a fixed reward which is divided among all the robots that saw it for the first time. Its maximum value is 1, when only one robot explores a new voxel. This reward attribution forces the robots to spread out, trying to minimize redundancy in new acquired information. Note that the sum over all the robots of the functions $J_t^{(r)}$ is simply the total explored environment up to time t , i.e.,

$$\sum_{r \in N} J_t^{(r)} = \sum_{r \in N} \sum_{s \in E_t(\mathcal{S})} \frac{\delta_s^r}{|e(s)|} = |E_t(\mathcal{S})|. \quad (2)$$

IV. THE FCAO ALGORITHM

In this section, we introduce a novel multi-robot exploration algorithm combining local stochastic approximation for optimization and a frontier-based approach. We will refer to our method to as Frontier-based Cognitive Adaptive Optimization (FCAO) algorithm (Algorithm 1). The resulting decentralized approach has the goal to overcome the weaknesses that these two methods have separately while preserving their strengths.

Algorithm 1: The FCAO Algorithm

```

function FCAO( $w, \alpha, E$ )
  Initialize  $t, \delta t = [t - w : t]$  and  $\mathbf{P}_{\delta t}$ .
  repeat
    CAO( $t, \mathbf{P}_{\delta t}, E, \alpha$ )
     $\mathbf{P}_{t+1} \leftarrow \text{ClosestFrontier}(\mathbf{P}_t, E)$ 
     $t \leftarrow t + 1$ 
  until  $\mathbf{P}_{t+1} = \mathbf{P}_t$ ;

function CAO( $t, \mathbf{P}_{\delta t}, E, \alpha$ )
  repeat
    Update  $J(E)$  with new points.
    Update  $\vartheta_t$  according to Equation (5).
    Generate points  $(\zeta^{(m)})_{m \in \{1, \dots, M\}}$  randomly.
    Select  $m^*$  according to Equation (7)
     $\mathbf{P}_{t+1} \leftarrow \mathbf{P}_t + \alpha \zeta^{(m^*)}$ 
     $t \leftarrow t + 1$ 
  until  $\mathbf{P}_t = \mathbf{P}_{t-1}$ ;

function ClosestFrontier( $\mathbf{P}_t, E$ )
   $F = \text{CandidateFrontiers}(E)$ 
  return SelectClosestFrontier( $F, \mathbf{P}_t$ )

```

A. Local Stochastic Approximation-based Optimization

Finding a real optimum of (1) is in practice unfeasible since the environment is initially unknown and the optimization needs to be carried on on-line during the exploration process. However, any robot can compute the numerical values of (1) at each time step based on collected data. In these scenarios, stochastic optimization algorithms based on local approximation of the original objective function are powerful tools to circumvent the problem. This is, for instance, the case of the CAO algorithm, originally developed and analyzed by Kosmatopoulos in [13], which has been recently proposed, in various modified versions, to obtain solutions in multi-robot problems, such as: multi-robot deployment [14], [15], target detection [16] and multi-robot localization and mapping [17]. In this section we present a modified version of this optimization algorithm to make it suitable to the considered problem.

The proposed solution represents an efficient way to solve in real-time constrained optimization problems where the exact forms of an objective functions \mathcal{J} is not available, but its numerical values can be retrieved by exploiting the gathered information. The algorithm is based on two main steps: *i*) at each iteration, an approximation function is

built to have a local estimation of the unknown objective function J ; *ii*) the next robot position is selected such that the approximation function is maximized, respecting the problem constraints. These two steps are detailed below.

1. At each time-instant t , J is estimated according to:

$$\hat{J}_t^{(r)}(x_{1,t}^{(r)}, x_{2,t}^{(r)}, x_{3,t}^{(r)}) = \vartheta_t^T \phi(x_{1,t}^{(r)}, x_{2,t}^{(r)}, x_{3,t}^{(r)}) \quad (3)$$

where ϑ_t denotes the vector of parameter estimates calculated at time t and ϕ the nonlinear vector of regression functions. In our case, we chose these functions in ϕ to be polynomials of the state variables up to the third-grade, i.e. the vector components are all the possible combinations given by:

$$\begin{aligned} & (x_1^{(r)})^i (x_2^{(r)})^j (x_3^{(r)})^k \\ & \text{s.t. } i, j, k \in \{0, 1, 2, 3\}, i + j + k \leq 3. \end{aligned} \quad (4)$$

This choice is arbitrary and other solutions can also be adopted. However, it is worth remarking that this approximation is strictly local and it has no ambition to reproduce the original function in the entire exploration space. Its role is exclusively to provide an estimation of J to allow each robot to select its next position. The parameter estimation vector ϑ_t , which has consequently dimension $L = 20$, is then calculated at each time t based on a limited set of past measurements according to:

$$\vartheta_t = \underset{\vartheta}{\operatorname{argmin}} \frac{1}{2} \sum_{\ell=\ell_t}^{t-1} \left(J_\ell^{(r)} - \vartheta^T \phi(\mathbf{P}_\ell^{(r)}) \right)^2 \quad (5)$$

where $\ell_t = \max\{0, t - L - T_h\}$ with T_h being a user-defined non-negative integer. Standard least-squares optimization algorithms can be used to solve (5). Note that using only a limited set of past values has the advantage to significantly reduce the computation time required to obtain the parameter vector ϑ_t but the resulting function is a reliable approximation only locally and cannot be used for global optimization.

2. As soon as the estimator \hat{J}_t is constructed according to (3) and (5), the new robot position is obtained performing a local random search. A set of admissible new positions is generated by randomly perturbing the current state and directly tested on \hat{J}_t . More formally, a set of M possible state configurations is constructed according to:

$$\mathbf{P}_t^{(r,m)} = \mathbf{P}_t^{(r)} + \alpha \zeta_t^{(r,m)}, \forall m \in \{1, \dots, M\}, \quad (6)$$

where $\zeta_t^{(r,m)}$ is a zero-mean, unit random vector with dimension equal to the dimension of $\mathbf{P}_t^{(r)}$ and α represents the exploration step for each iteration. It is worth noticing that this parameter is usually time-dependent in optimization algorithms to ensure the convergence to a local optimum (see e.g. [18]). In our case, we do not have this necessity since in an exploration task there is no interest in moving less than the maximum feasible step and there is no convergence until the total environment is explored.

Among all M candidate new configurations $\mathbf{P}_t^{(r,m)}$, the ones that correspond to non-feasible positions, i.e. that

violate the constraints \mathcal{C} , are neglected and the new robot position is $\mathbf{P}_{t+1}^{(r)} = \mathbf{P}_t^{(r,m^*)}$, where m^* is calculated as follows:

$$m^* = \underset{m \in \{1, \dots, M\}}{\operatorname{argmax}} \hat{J}_t^{(r)}(\mathbf{P}_t^{(r,m)}). \quad (7)$$

The random generation of the candidates is essential and crucial for the efficiency of the algorithm, because this choice guarantees that \hat{J}_t is a reliable and accurate estimate for the unknown function J . Finally, we remark that in this paper we focus only on the high-level planning problem without explicitly considering any dynamics constraints for the robots. This choice can also be justified assuming that the new way points are sufficiently distanced to be always accessible by the robot. However, more restricting assumptions, including the robot's dynamics, and so reducing the space of feasible next positions would affect only the definition of \mathcal{C} but not the algorithm.

B. Closest frontier

As previously stated, the proposed optimization is strictly local and, even though can very efficiently find a feasible solution to a constrained trajectory generation problem, it needs variations in the measured values of the optimization function to provide better solutions. In other words, as every local optimization method, it cannot overcome local optima without an external input. In the exploration task, this means that while the robot movements bring new explored areas the algorithm provides a clear direction to follow, but when it remains surrounded by already explored regions, the optimization method would provide only random movements. Only in these cases, the set of frontiers remaining in the environment is considered and the robot is driven towards the closest one. This requires to compute each UAV-to-frontier distance - in 3D -, then to select the shortest (adaptation of the standard [5] approach to 3D spaces). Such a frontier assignation allows the robot to efficiently overcome the deadlocks and restart the local optimization from it. More globally, alternating local explorations with frontier assignations leads to guarantee the completeness of the exploration as in any frontier-based method.

This step makes use of global map which each robot builds integrating the information received by other teammates. We refer to [2] for a more detailed description of this strategy.

The important aspect of this approach is that, while the information is shared, the decision remains decentralized. In particular, this ensures robustness with respect to robots failure since, if a robot interrupt the new information transmission, the remaining robots will still continue to explore the entire environment. Furthermore, being the system completely asynchronous, robots do not need to wait for other robots, and the loss of one or more robots does not produce any interruption in the algorithm.

The counter effect of this approach is the possibility to have redundancy in the exploration, i.e. multiple robots might be assigned to the same frontier. This drawback remains also in our approach but it is largely limited by the fact that

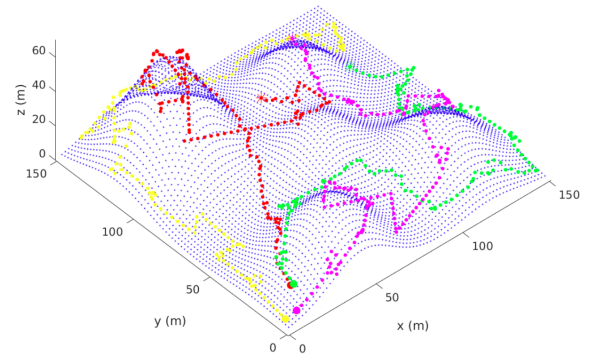


Fig. 2: Trajectories of four robots exploring an unknown terrain.

frontiers are employed only rarely and in an asynchronous way among the agents.

V. SIMULATION RESULTS

The proposed approach has been tested in simulations and compared with common alternative solutions. The terrains to explore are arbitrary and, in a first study, generated as random mixtures of Gaussians with varying height and variance. The robots must maintain a minimum height of flight from the terrain and minimum distances from other robots to avoid any collision. In the simulation we assume omni-directional sensing capabilities limited by a limited range. In Fig. 2, an illustrative example of an exploration mission carried out by 4 robots, showing the trajectories for each one of them, is presented. As for the other simulation results reported in this section, the initial positions of the robots can vary slightly but they are always considered close one each other starting from a corner of the region to explore.

To quantitatively evaluate its performance, the proposed approach has been compared with two standard alternative solutions: *i*) a decentralized closest-frontier algorithm, where each robot selects independently from the other teammates the frontier point closest to him; *ii*) a centralized greedy algorithm where to each robot is - sequentially - assigned its closest frontier point, which is then removed from the list. For the latter, the frontier points are firstly divided into clusters to further reduce the overlap in the robots trajectories and clusters' centroids are considered. A first result, with 8 robots starting from the same set of initial positions, is presented in Fig. 3. As a first remark, we can see how our approach clearly outperforms the closest-frontier algorithm. This can be justified by the high level of redundancy in the exploration only considering closest frontiers selection, while the local optimizer efficiently spreads out the team avoiding overlap in the acquired information. Then, it is interesting to note that even if the decision is decentralized this method can obtain a performance very close to the centralized greedy approach.

For the same scenario, another important aspect has been studied: the computational time required to find the new robot position at each algorithm iteration. As we already

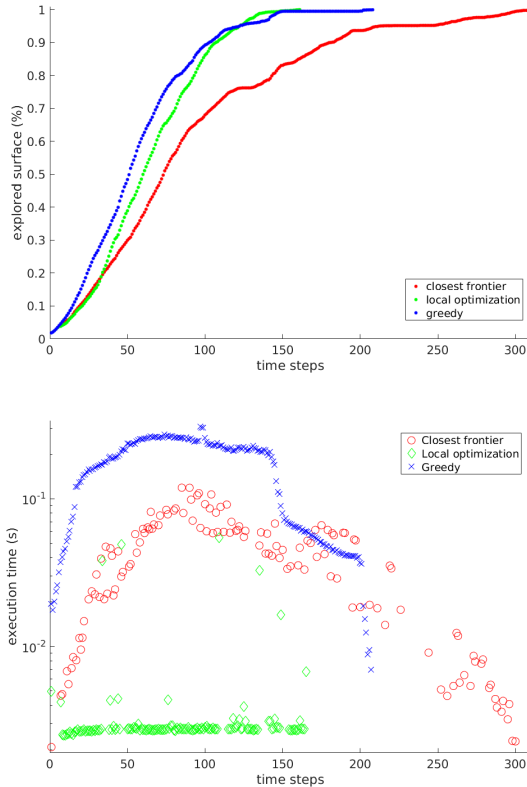


Fig. 3: The proposed approach is here compared with two alternatives: a standard decentralized closest frontier algorithm and a centralized greedy selection of frontier points. The results correspond to a scenario with 8 robots in an environment similar to the one shown in Fig. 2. Top: Portion of explored environment as a function of time steps. Bottom: Time required to select the next state for each iteration.

stated, one of the strong points of our approach is the low computational cost. This is clearly shown in Fig. 3(bottom), where this factor is not only significantly below the time required by the greedy algorithm but even faster than the very simple closest frontier selection. The main difference here is the need for frontier-based approaches to compute each time the distances between a robot and all the frontier points, which in 3D becomes an expensive operation. In our case, this effect is very significant even if we do not compute the exact geodesic distance between two points, which would be even more expensive, but instead we just consider the straight line connecting the two points and adapting the altitude to pass above the surface, given a certain minimum height of flight from the terrain. The cost of our approach is so independent of the number of frontier points, as can be seen in Fig. 3, where the run time of the local optimizer is constant (except for the very few steps where frontiers are exploited) while the other two are varying during time since depending on the number of frontiers present in the map at each time step. Furthermore, being decentralized, its cost is also independent of the number of robots, contrary to the centralized greedy algorithm. The only significant cost of

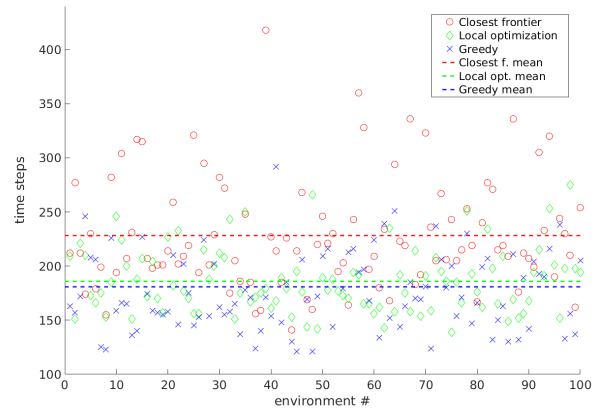


Fig. 4: Results obtained with the three algorithms over 100 different randomly generated environments. Dashed lined represent the means for each approach.

the local optimizer comes from solving the problem in (5), which exclusively depends on the dimension of the function ϕ and the number of past considered measurements, which are both constant along the mission.

Finally, to have more statistically meaningful results, we compared these three methods in 100 randomly generated environments (8 peaks with equal variance and centered in randomly selected positions uniformly distributed over the environment). Results, reporting the final values for each trial and the means for each algorithm, are presented in Fig. 4. Similarly to the previous comparison, the proposed optimization-based approach has a performance very close to the centralized greedy algorithm, and they both clearly outperform the closest frontier method. We can also note that the latter, besides having an average worse behavior, presents also a greater variance in the results, while the other two are significantly more robust with respect to changes in the environment.

A. Realistic Urban-like Environment

We present here the results obtained considering more realistic models for both the environment and the system to show the feasibility and employability of our approach in real applications. The simulations have been carried out using ROS and Gazebo simulator, considering a rich and realistic urban-like environment including several buildings, vehicles, etc. (see Fig. 5). The drones have been modeled after the specifics of the Intel Aero quadrotor. At each position, the UAVs obtain a point cloud representation of the surrounding environment included in a down-looking half-sphere with 20m radius. The map of the explored surface and a list of frontier points, whenever necessary, are generated based on the point cloud by using the ROS package Octomap_server [19]. The result of the exploration conducted by two UAVs is presented in Fig. 5, where the initial environment and the final 3D map are shown, and in Fig. 6, where the number of octree nodes explored in time by the two UAVs is provided.

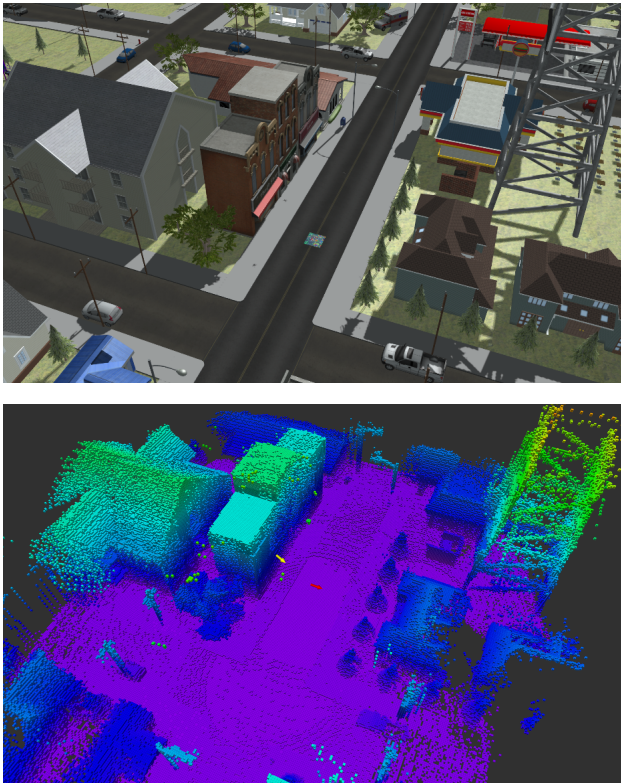


Fig. 5: Environment to explore (top) and final 3D map (bottom) obtained by 2 UAVs performing FCAO algorithm.

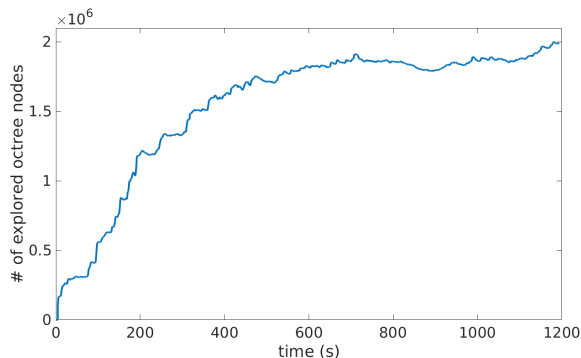


Fig. 6: Number of octree nodes explored across time corresponding to the scenario shown in Fig. 5.

VI. CONCLUSIONS

This paper presented a new decentralized optimization-based approach for the exploration of arbitrary surfaces in 3D environments with a team of aerial vehicles. The local optimization is carried out adopting a stochastic derivative-free optimization algorithm with a very low computational cost. This approach has been coupled with a frontier-based method to allow restarting the optimization-based exploration whenever a robot gets stuck in local optima. Results in simulation proved the efficiency of this approach and a comparison with standard frontier-based approaches has been carried out. This analysis showed that our approach significantly improves the performance of a decentralized closest frontier algorithm,

while has a similar performance with respect to a much more expensive centralized greedy assignation of frontiers to the team. A final simulated experiment in Gazebo also shows the applicability of our approach in a more realistic scenario.

Future work will focus on a more thorough validation of our algorithm in richer environments as well as with a more realistic model for the UAVs in prevision of experiments in real scenarios.

REFERENCES

- [1] Z. Yan, N. Jouandeau, and A. A. Cherif, "A survey and analysis of multi-robot coordination," *International Journal of Advanced Robotic Systems*, vol. 10, no. 12, p. 399, 2013.
- [2] B. Yamauchi, "Frontier-based exploration using multiple robots," in *Proceedings of the second international conference on Autonomous agents*. ACM, 1998, pp. 47–53.
- [3] M. Juliá, A. Gil, and O. Reinoso, "A comparison of path planning strategies for autonomous exploration and mapping of unknown environments," *Autonomous Robots*, vol. 33, no. 4, pp. 427–444, 2012.
- [4] F. Amigoni, J. Banfi, and N. Basilico, "Multirobot exploration of communication-restricted environments: A survey," *IEEE Intelligent Systems*, vol. 32, no. 6, pp. 48–57, 2017.
- [5] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Computational Intelligence in Robotics and Automation, 1997. CIRA'97., Proceedings., 1997 IEEE International Symposium on*. IEEE, 1997, pp. 146–151.
- [6] W. Burgard, M. Moors, C. Stachniss, and F. E. Schneider, "Coordinated multi-robot exploration," *IEEE Transactions on robotics*, vol. 21, no. 3, pp. 376–386, 2005.
- [7] A. Bautin, O. Simonin, and F. Charpillet, "Minpos: A novel frontier allocation algorithm for multi-robot exploration," in *International conference on intelligent robotics and applications*. Springer, 2012, pp. 496–508.
- [8] J. Faigl, M. Kulich, and L. Přeučil, "Goal assignment using distance cost in multi-robot exploration," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 3741–3746.
- [9] J. Banfi, A. Q. Li, I. Rekleitis, F. Amigoni, and N. Basilico, "Strategies for coordinated multirobot exploration with recurrent connectivity constraints," *Autonomous Robots*, vol. 42, no. 4, pp. 875–894, 2018.
- [10] C. Dornhege and A. Kleiner, "A frontier-void-based approach for autonomous exploration in 3d," *Advanced Robotics*, vol. 27, no. 6, pp. 459–468, 2013.
- [11] C. Zhu, R. Ding, M. Lin, and Y. Wu, "A 3d frontier-based exploration tool for mavs," in *2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, 2015, pp. 348–352.
- [12] N. Mahdouli, V. Frémont, and E. Natalizio, "Cooperative frontier-based exploration strategy for multi-robot system," in *2018 13th Annual Conference on System of Systems Engineering (SoSE)*. IEEE, 2018, pp. 203–210.
- [13] E. B. Kosmatopoulos, "An adaptive optimization scheme with satisfactory transient performance," *Automatica*, vol. 45, no. 3, 2009.
- [14] A. Renzaglia, L. Doitsidis, A. Martinelli, and E. B. Kosmatopoulos, "Multi-robot three-dimensional coverage of unknown areas," *The International Journal of Robotics Research*, vol. 31, no. 6, 2012.
- [15] A. Renzaglia, L. Doitsidis, S. A. Chatzichristofis, A. Martinelli, and E. B. Kosmatopoulos, "Distributed multi-robot coverage using micro aerial vehicles," in *IEEE 21st Mediterranean Conference on Control and Automation*, 2013, pp. 963–968.
- [16] K.-S. Tseng and B. Mettler, "Near-optimal probabilistic search via submodularity and sparse regression," *Autonomous Robots*, vol. 41, no. 1, 2017.
- [17] A. Kapoutsis, S. Chatzichristofis, L. Doitsidis, J. B. de Sousa, J. Pinto, J. Braga, and E. B. Kosmatopoulos, "Real-time adaptive multi-robot exploration with application to underwater map construction," *Autonomous robots*, vol. 40, no. 6, pp. 987–1015, 2016.
- [18] D. Bertsekas and J. Tsitsiklis, "Gradient convergence in gradient methods with errors," *Journal on Optimization*, vol. 10, no. 3, 2000.
- [19] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: An efficient probabilistic 3d mapping framework based on octrees," *Auton. Robots*, vol. 34, no. 3, pp. 189–206, 2013.