Communicating Multi-UAV System for Cooperative SLAM-based Exploration

Nesrine Mahdoui, Vincent Frémont, Enrico Natalizio

Noname manuscript No.
(will be inserted by the editor)

Communicating Multi-UAV System for cooperative SLAM-based Exploration

Nesrine Mahdoui · Vincent Frémont · Enrico Natalizio.

Received: date / Accepted: date

Abstract In the context of multi-robot system and more generally for Technological System-of-Systems, this paper proposes a multi-UAV (Unmanned Aerial Vehicle) framework for SLAM-based cooperative exploration under limited communication bandwidth. The exploration strategy, based on RGB-D grid mapping and group leader decision making, uses a new utility function that takes into account each robot distance in the group from the unexplored set of targets, and allows to simultaneously explore the environment and to get a detailed grid map of specific areas in an optimized manner. Compared to state-of-the-art approaches, the main novelty is to exchange only the frontier points of the computed local grid map to reduce the shared data volume, and consequently the memory consumption. Moreover, communications constraints are taken into account within a SLAM-based multi-robot collective exploration. In that way, the proposed strategy is also designed to cope with communications drop-out or failures. The multi-UAV system is implemented into ROS and GAZEBO simulators on multiple computers provided with network facilities. Results show that the proposed cooperative exploration strategy minimizes the global exploration time by 25% for 2 UAVs and by 30% for 3 UAVs, while outperforming state-of-the-art exploration strategies based on both random and closest frontiers, and minimizing the average travelled distance by each UAV by 55% for 2 UAVs and by 62% for 3 UAVs. Furthermore, the system performance is also evaluated in a realistic test-bed comprising an infrastructure-less network, which is used to support limited communications. The results of the test-bed show that the proposed exploration strategy uses 10 times less data than a strategy that makes the robots exchanging their whole local maps.

Keywords Coordinated multi-robot system · UAV · autonomous exploration, frontier-based exploration · SLAM · inter-robot communications.

Nesrine Mahdoui
Sorbonne Universités, Université de Technologie de Compiègne, CNRS, UMR 7253, Heudiasyc, 60 200 Compiègne, France
E-mail: nesrine.mahdoui@hds.utc.fr

Vincent Frémont
Ecole Centrale de Nantes, LS2N, UMR CNRS 6004, Nantes, France
E-mail: vincent.fremont@ec-nantes.fr

Enrico Natalizio
Université de Lorraine, LORIA, UMR CNRS 7503, Vandoeuvre-lès-Nancy, France
E-mail: enrico.natalizio@loria.fr

1 Introduction

In the last decades, the robotic community has shown a growing interest in the development of multi-robot systems. Several research works showed that these systems present several advantages, such as increased efficiency, reduced mission time, robustness to robot failures, and scalability. Hence, these systems are sufficiently flexible to be used in different applications, such as surveillance, infrastructure inspection, search and rescue, etc. Among these applications, the exploration and mapping of unknown environment is still one of the fundamental problem in aerial robotics. The aim is to get the most complete and accurate map of the environment in a relatively short time. Still, when using multiple robots, many challenges arise especially when using robots with limited embedded sensor range, processing capabilities and energy [41]. Indeed, previous multi-robot exploration research focused on motion planning and collision avoidance [21,15,3]. More recently, the emphasis moved to robot coordination and cooperation. Multi-robot systems are mainly composed of three complementary components – *perception* [42], *planning and control* [1], and *communications* [23] – that interact together to get a consistent and robust system. One of the main challenges of the *perception* component is the Simultaneous Localization and Mapping (SLAM) where no global positioning system is used. For the *path planning and control* component, cooperative exploration represents one of the main problems. Thus, in the literature cooperative exploration strategies have been proposed. Usually, these strategies are based on a utility function to assign a robot with an exploration target. The target assignment decision is performed by using specific information exchanged among robots. Therefore, *communicatiosn* are a fundamental component of the multi-robot system and communications issues must be taken into account. In fact, multi-robot systems have to cope with communications failures in order to ensure the mission continuity. Therefore, in this paper we address each of the three components' challenges towards the definition of a system providing robots with precise localization, improving robots area coverage and coordinating the fleet.

1.1 Contributions

In this paper, we present a coordinated Multiple Unmanned Aerial Vehicles (UAVs) system in order to efficiently explore an unknown environment using potentially heterogeneous robots. The main contributions of this work are fourfold: *i)* From a System-of-Systems point of view, we introduce a new fully distributed Multi-UAV system architecture that does not exploit any global information (neither map nor GPS); *ii)* Compared to existing works, we propose a novel utility function that takes into account the distance of each robot in the group from the unexplored set of targets which allows to cover a larger area by effectively spreading the robots in the environment; *iii)* We detail a coordinated exploration strategy, based on an on-line grid-based RGB-D SLAM system and a dynamic-group-*leader* decision making that helps minimizing the global exploration time and the average traveled distance for each robot; *iv)* We design a strategy behavior to deal with communication limitations and failures, which requires only a limited information data exchange.

1.2 Paper structure

The paper is organized as follows. In section 2, we start by introducing a brief state of the art on visual SLAM, UAV to target assignment, utility function, and communications for Multi-UAV systems. In section 3, we present an overview of the proposed Multi-UAV system. Then, in sections 4 and 5, we detail the proposed exploration algorithm and the inter-robots communication module, respectively. We present the results and discussions about the proposed framework in section 6. And finally, we conclude in section 8. The nomenclature and variables, used in this paper, are summarized in Table 1.

Table 1 Nomenclature.

Designation	Description
UAV_i	UAV of index i.
$\mathbf{p}_i$, v_i, ω_i	Pose, linear velocity, and angular velocity of UAV_i.
${}^0W$	Global reference frame.
${}^WF_i$	UAV_i's local reference frame w.r.t ${}^0W$.
${}^W\big[\mathbf{R}\ \mathbf{t}\big]_{F_i}$	Transform of rotation $\mathbf{R}$ and translation $\mathbf{t}$ from reference frame ${}^0W$ to ${}^WF_i$.
$\mathcal{P_S}, \mathcal{O}, \mathcal{L}, \mathcal{F}, \mathcal{G}, \mathcal{T}, \mathcal{C}$	3D points cloud computed by SLAM, 3D voxels, 2D cells, frontier points, candidate frontier points (candidate targets), assigned targets and cluster sets, respectively.
$\mathbf{o}_u$, $\mathbf{o}_f$, $\mathbf{o}_o$	Unknown, free and occupied 3D voxels, respectively.
$\mathbf{l}_u$, $\mathbf{l}_f$, $\mathbf{l}_o$	Unknown, free and occupied 2D cells, respectively.
$\mathbf{f}_{i,j}$	Frontier point j of UAV_i
$\mathbf{t}_i$	Target point i.
$\boldsymbol{I}(\mathbf{t}_i)$	Information gain of $\mathbf{t}_i$.
$\boldsymbol{U}(UAV_i, \mathbf{t}_j)$	Utility of reaching target j by UAV_i.
$\theta(i,j)$	Assignment of UAV_i with target j.
id	Identification number of UAV.
r	Loop rate.
n	Number of UAVs in the fleet.
n_c	Number of UAVs in $\mathcal{C}$.
n_t,	Number of targets in $\mathcal{T}$.
n_g	Number of targets in $\mathcal{G}$.
n_i	Number of frontier points of UAV_i.
s	Sensor maximum range.
λ	Tuning parameter $\in [0, 1]$.
$[r_{min}, r_{max}]$	Range to schedule information gain.
σ_x, σ_y	Parameter to spread the blob in x and y axis, respectively.
d_{tot}	Average distances of other UAVs to the considered target.

2 RELATED WORKS

Cooperative multi-robot systems are playing important roles in today's unknown environment exploration applications. Beside this growing interest for multi-robot systems, several architectures have been proposed to manage the system interoperability. The architecture is often closely related to the decision making process that can be either centralized or decentralized including distributed, hybrid and hierarchical ones [41]. Those architectures present several advantages and disadvantages that are summarized in Table 2. The centralized architecture considers one robot/central server [30] to manage all the computations and tasks assignment. However, these centralized approaches are subject to stranded missions when they do not take into account communications or robot failures. On another side, works such as [43,34], propose to use distributed approaches with fully autonomous robots. These approaches require robots with increased resources to exchange and process an important amount of information in order to synchronize agents and achieve a cooperative mission. Authors in [38] propose a hybrid approach that consists in switching from individual to cooperative exploration behavior when agents are not able to converge to a local minimum at a satisfying rate. Important computation requirements are used for this later approach. Taking into account the advantage of solution optimality from centralized approach, and the advantage of mission reliability and adaptability from decentralized approach; a *leader*-based approach has been proposed in our previous work [22]. The fleet is subdivided into subgroups – called clusters – such that one robot in each subgroup is responsible for targets assignment. In this paper, we propose to improve this work to tackle the problems of relative localization and inter-robots communications.

In this paper, we also focus on the use of visual sensor as the main perception modality. These visual sensors are used to gather information to map the environment and to estimate the robot's trajectory. For that, visual odometry (VO) or SLAM method are dominant. There exist several methods attempting to solve SLAM [7]. Among them, the ORB-SLAM2 vision based framework [25] has shown promising results for pose estimation. It is a lightweight RGB-D SLAM system and a feature based method composed of: a tracking thread for the localization; a local mapping thread for the mapping; and a loop closing thread in charge of detecting loops. These three main threads work in parallel. A place recognition module is used for re-localization and loop detection. Hence, in this work, we make use of the ORB-SLAM2 to construct a model of the environment and to estimate the robot's state within it. The mapping algorithm is performed while a robot attempts to reach a target. So, for an effective environment mapping, the target should be chosen carefully. There is a wide variety of goal assignment strategies to affect one robot to a target.

In [20] a comparison of some assignment strategy used for multi-robot system is done. The compared strategies include Hungarian-based method [28], Greedy [40], Broadcast of Local Eligibility (BLE) method [37] and K-means clustering [36]. Results show that Hungarian-based methods outperform other approaches in majority of cases. Unlike the Hungarian-based method, the iterative assignment can be implemented in a distributed environment. Also the Hungarian-based approach are computationally heavy compared to the simple greedy algorithm which is preferred in applicable scenario.

Most of the target to robot assignments are based on an utility function that defines the advantage that a robot have to reach this target according to the mission's aim [5]. The work proposed in [2] presents a new utility function that takes into account the traveling cost to the target and the connectivity utility. This allows a trade off between minimizing the amount of exploration time and connectivity. To speed up velocity, authors in [8] propose a rapid frontier selection technique to select goals from the robot's field of view. This approach minimizes the overall mission time by minimizing the change in velocity of the robot. Nonetheless, it increases the total path length traveled. In [17], maximizing the reconstructed model is favored over the mission time. Further, the proposed approach solves simultaneously exploration and coverage problems in order to maximize the completeness of reconstructed model. Whereas in [35], the aim is to maximize the utility of targets that minimizes the potential for overlap in information gain amongst members of the fleet. The utility of reaching a target depends basically on the aim of the mission while taking into account some additional constraint such as time, completeness of the map, limited sensor and communication range, and number of robots.

For multi-robot exploration, one of the most critical point is linked to inter-robot communications. The challenge is to maintain reliable communications during the mission, in order to make the robots cooperate [31,16]. The strategy used for exploration affects the data exchange among robots including type, destination and frequency. The exchanged information may be composed of key-frames and map points shared between a robot and a server [32], or only features of selected key-frames and relative-pose estimates shared among robots and ground station [13]. But mostly, robots exchange their local copies of the map and their poses [4, 33,14]. The amount of exchanged data may rapidly increase in size, which may cause network congestion and data loss. In order to reduce the bandwidth require-

Table 2 Multi-robot system decision making architecture.

Approach	Centralized	Decentralized (distributed or hierarchical)
Advantages	Optimal solution. Simple and lightweight processing on-board robots.	Robustness in dynamic environment. Reliability in case of other robots failure. Adaptability and flexibility. Decision making autonomy.
Disadvantages	Weakness in dynamic environment. Important network requirement. System vulnerability in front of central control agent. Additional computational requirements. Unsuitable for large scale systems.	Suboptimal solution. Complex on-board processing. Important amount of exchanged information.

ments, authors in [24] propose to send only compressed key-frames and updated key-frame poses. Authors in [10] propose a Decentralized Data Fusion-Smoothing And Mapping (DDF-SAM) approach, where each robot propagates towards other robots, its condensed local graph in order to achieve scalability and robustness to node failure. Most works deal with the communication problem while assuming ideal communications or aim to keep team members within range of one another in order to focus their attention in higher level problem [33,6]. But considering communication losses and/or limited bandwidth help to prevent from mission failure ensuring a more realistic scenario. Indeed, in real scenarios, many issues can arise such as having distance among robots that exceeds the communication range, losing major information in a broken communication link, losing precious time in sending information due to limited bandwidth. The exploration strategy have to take into account the mentioned issues to avoid mission failure in real world scenario. This topic is understudied, yet, some works began to tackle the exploration problem while considering communication limitations [9,32]. In [11], the aim is to sense a geometrically complex environment by assigning targets to robots while satisfying spatial and temporal resolutions. This approach uses a min-max energy path planning algorithm that obeys to a deadline time. Other works [16,29] propose to use protocol and routing solutions to overcome the robots' communication issues. In our work, we make the choice to let UAVs exchange with each other only frontier points, robot poses, and assigned targets. This exchange happens at each iteration while considering UAVs' role, which are adapted according to the network topology. This adaptation allows also to cope with communication limitations.

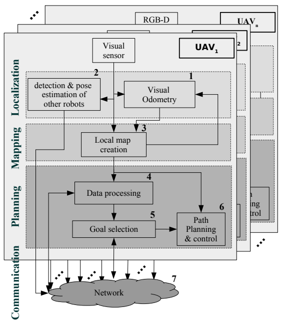


Fig. 1 Architecture block diagram.

3 Multi-UAV system overview

Using potentially heterogeneous UAVs, the main objective of cooperative exploration is to achieve a full coverage of an unknown environment in minimum time.

3.1 System architecture

The proposed framework in Fig. 1 is an overview of the software architecture used for Multi-UAV system. It presents the different modules and data flows among them. This block diagram is distributed and embedded over all fleet members composed of n UAVs. We suppose that each UAV is equipped with an embedded RGB-D visual sensor. To maintain an accurate estimate of the UAV's pose in the environment, a simultaneous localization (block **1**) and mapping (block **3**) are performed. Block **3** in the mapping layer is responsible for constructing a detailed grid map of the explored regions and keeping track of them. In the data processing block (block **4**), some specific information are picked out and exchanged using the communication layer where the network (block **7**) is in charge of maintaining data flow among UAVs. The collected data are then locally processed in the same block **4** to get exploitable information for exploration. Thereafter, block **5** performs targets selection. Planning the path and reaching it are the roles of block **6**. Block **2** is used to visually detect other UAVs in the environment then estimate their relative transform using visual fiducial markers or tags such as WhyCon [26] or AprilTag [27].

3.2 System coordinate frames

In this paper, we assume that the UAV fleet explore a 3D bounded unknown environment with a global reference frame ${}^0W$ (See Fig. 2). Each robot (UAV$_i$, with $i \in \mathbb{N}^*$), maintains a relative motion matrix ${}^{F_i}\left[\mathbf{R}\ \mathbf{t}\right]_{UAV_i}$ w.r.t. its corresponding local reference frame ${}^{W}F_i$, and a global transform ${}^{W}\left[\mathbf{R}\ \mathbf{t}\right]_{UAV_i}$ w.r.t. the global reference frame ${}^0W$. During the mission, the information

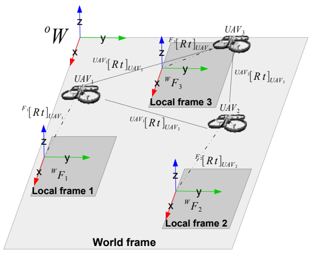


Fig. 2 Multi-UAV global and local coordinate systems.

computed within local frames ${}^{W}F_i$ of all the UAVs are processed in parallel. Before that, however, those information need to be converted in ${}^0W$ by knowing the UAV's local reference frame transform w.r.t. ${}^0W$ (${}^{W}\left[\mathbf{R}\ \mathbf{t}\right]_{F_i}$). Thereby, UAV's initial pose in ${}^0W$ needs to be known. To do that, the UAV, with the lowest *id* number in the fleet (UAV$_1$ in Fig. 2), is considered as a landmark. The global frame is defined such that it coincides with the UAV marker's local frame where ${}^{W}\left[\mathbf{R}\ \mathbf{t}\right]_{F_1} = \left[\mathbb{I}_3\ 0\right]$. Using block **2** in Fig. 1, the relative transform ${}^{UAV_1}\left[\mathbf{R}\ \mathbf{t}\right]_{UAV_i}$ is estimated, from which, transform ${}^{W}\left[\mathbf{R}\ \mathbf{t}\right]_{UAV_i}$ and thus ${}^{W}\left[\mathbf{R}\ \mathbf{t}\right]_{F_i}$ are computed.

3.3 Roles in the fleet

Cooperation in Multi-UAV systems often goes through the exchange of data [41]. In a limited communication ability, the data sharing link cannot always be correctly established due to limited communication range, data loss, obstacles, and traffic congestion. In the proposed work, each group of robots that may communicate with one another, form a cluster $\mathcal{C}$. The fleet is composed of, at least, one cluster (if $n = n_c$). In each $\mathcal{C}$, one robot takes the role of *leader* and is in charge of making cooperative decision, based on some specific shared information, for all the other robots in $\mathcal{C}$ that have the role of *explorers*. The decision making process relies completely on the *leader*, which can lead to mission interruption; especially when the *leader* to *explorer* communication link is lost, or the *leader* is out of order. To overcome these problems, the roles are constantly updated, in order to select a *leader* if the current one experienced any issue. The roles are not previously defined but are adapted depending on the fleet topology changes. All UAVs' role are initialized to *leader*. Then, as soon as UAVs start to exchange their identification number *id*, clusters are formed and each UAV chooses its appropriate role. The *leader* role is taken by the UAV with the lowest *id* number in the $\mathcal{C}$.

4 Multi-UAV exploration and coordination

4.1 Simultaneous Localization And Mapping

The exploration task requires the UAV to implicitly maintain an accurate estimate of its pose in addition to a map of the observed environment. Fig. 3 shows the outline of the localization and mapping layers used in the framework.

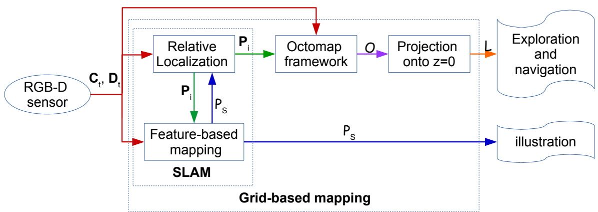


Fig. 3 Localization and Mapping layers outline. $\mathbf{p}_i$ is the robot's estimated pose, $\mathcal{P}_\mathcal{S}$ represents the 3D point cloud computed by the SLAM system, and $\mathcal{O}$ and $\mathcal{L}$ contain the 3D and 2D grid map, respectively.

In this paper, the SLAM module provides an estimate of the robot's 3D coordinates $\mathbf{p}_i = [x, y, z, q_x, q_y,$

$q_z, q_w]^\top$ w.r.t. the local reference frame ${}^W F_i$.
Using the estimated motion and the 3D point cloud from the RGB-D sensor, a 3D occupancy grid is built during the grid-based mapping process. For that, the Octomap framework [19] is used. The environment is approximated to a 3D voxels grid $\mathcal{O}$ where each voxel $\mathbf{o} \in \mathcal{O}$ is represented by its centroid (See Fig. 4). Using the sensor measurements, voxels are labeled to unknown $\mathbf{o}_u$, free $\mathbf{o}_f$ or occupied $\mathbf{o}_o$. This 3D occupancy grid $\mathcal{O}$ is down-projected onto the plane $z = 0$ of the local frame ${}^W F_i$ to get a 2D cell grid $\mathcal{L} = \text{proj}_{(z=0)}(\mathcal{O})$. Cells are occupied as soon as there is an occupied voxel in the z cell range. And, they are free if all voxels in the z cell range are so. During the exploration, each robot chooses a target from $\mathcal{L}$ and moves to it while maintaining a fixed z altitude which leads to a 2D exploration and navigation problem.

4.2 Proposed Exploration Strategy

In a Multi-UAV system, the exploration strategy needs to be cooperative in order to be efficient. The proposed strategy is described in Algorithm 1. The main objective is to cooperatively choose specific regions to be simultaneously explored using a frontier-based approach. This is done by selecting candidate targets and assigning them to each robot in an optimized manner.

Algorithm 1 Exploration strategy for coordinated Multi-UAV.

1: From cells $\mathbf{l}_l \in \mathcal{L}$, select frontier points $\mathbf{f}_{i,j} \in \mathcal{F}$ and compute their respective information gain $\boldsymbol{I}(\mathbf{f}_{i,j})$.
2: Process frontier points $\mathbf{f}_{i,j}$ to get candidate goals $\mathbf{t}_k \in \mathcal{G}$ (See Algorithm 2).
3: Assign UAV$_i$ with target k (See Algorithm 3).
4: Send targets to the corresponding robots.

4.2.1 Frontier selection and information gain

The frontier selection process is used to define the frontiers of regions bounded by obstacles or unknown spaces. In this work, the frontier cells $\mathbf{f}_{i,j} \in \mathcal{F}$ are selected from the set of cells $\mathcal{L}$ ($\mathcal{F} \subset \mathcal{L}$) such that they are either *i)* free $\mathbf{l}_f$ and adjacent to unknown, or *ii)* labeled as occupied $\mathbf{l}_o$. Occupied cells $\mathbf{l}_o$ are considered as frontier cells to be able to perform frontier processing in the next step. They could not be chosen as target and will be discarded later. For example, in Fig. 4, the frontier cells are: $\mathbf{l}_f(2,1)$, $\mathbf{l}_f(2,2)$, $\mathbf{l}_o(2,3)$, and $\mathbf{l}_o(3,3)$. Thus, for a cluster $\mathcal{C}$ containing UAV$_i$, the frontiers are $\mathcal{F}$=$\{\mathbf{f}_{i,1}(2,1), \mathbf{f}_{i,2}(2,2), \mathbf{f}_{i,3}(2,3), \mathbf{f}_{i,4}(3,3)\}$.

In frontier-based exploration approaches, only cells adjacent to unknown ones may be defined as candidate frontier points and are likely to be chosen as target. Thereby, the information gain is associated to each of them in order to estimate the utility of reaching each frontier. This corresponding information gain can be defined in different manner depending on the mission purpose. Authors in [6] propose to use a probability function to reduce an assigned constant value thanks to the relative distance to the UAV's pose. This strategy is general and does not take into account the updated explored cells. The approach proposed in [17], affects to the information gain the number of unknown and not occluded cells in the view frustum of the target. This method depends on the real estimate of information gained when visiting the considered pose. However, it requires more computation. In the proposed strategy, the information gain are associated in a way that they define the number of unknown cells $\mathbf{l}_u$ around the target.

4.2.2 Frontier points processing

All frontier points $\mathbf{f}_{i,j} \in \mathcal{F}$ of UAV$_i$ in the cluster $\mathcal{C}$ with $i \in [1...n_c]$, are collected. Points in $\mathcal{F}$ are then processed using Algo. 2 to get candidate frontier points considered as candidate targets $\mathbf{t}_k \in \mathcal{G}$ with $k \in [1...n_g]$ (See Fig. 4). The operators $\bigcup$ and $\bigcap$ are defined such that, for example, if we have two UAVs with the frontier points $\mathbf{f}_{1,1} = [1,0]$ and $\mathbf{f}_{1,2} = [2,3]$ for UAV$_1$, and $\mathbf{f}_{2,1} = [1,1]$ and $\mathbf{f}_{2,2} = [2,3]$ for UAV$_2$; we would have $p_u = \{[1,0],[1,1],[2,3]\}$ and $p_i = \{[2,3]\}$.

Algorithm 2 Frontier processing algorithm.

Input: Frontier points $\mathbf{f}_{i,j} \in \mathcal{F}$ of UAV$_i$ with $i \in [1...n_c]$.
Output: Candidate targets $\mathcal{G}$.
1: $p_u = \bigcup_{i=1}^{n_c} \mathbf{f}_{i,j}$.
2: $p_i = \bigcap_{i=1}^{n_c} \mathbf{f}_{i,j}$.
3: $\mathcal{G} = p_u \setminus p_i$.
4: Delete the obstacle frontier points $\mathbf{f}_{i,j}(x,y) = \mathbf{l}_o(x,y)$ from $\mathcal{G}$.
5: **return** $\mathcal{G}$.

The obstacle frontier points – labeled as occupied – are only kept to compute the intersection of frontier points. Only the free frontier cells $\mathbf{l}_f$ can be considered as candidate target. When using local frontier points instead of local maps, the frontier process replaces the map matching process where the aim is to clear overlapping areas. Indeed, in the frontier processing step, the points that belong to the overlapping areas are cleared. Therefore, using frontier points allows important memory saving.

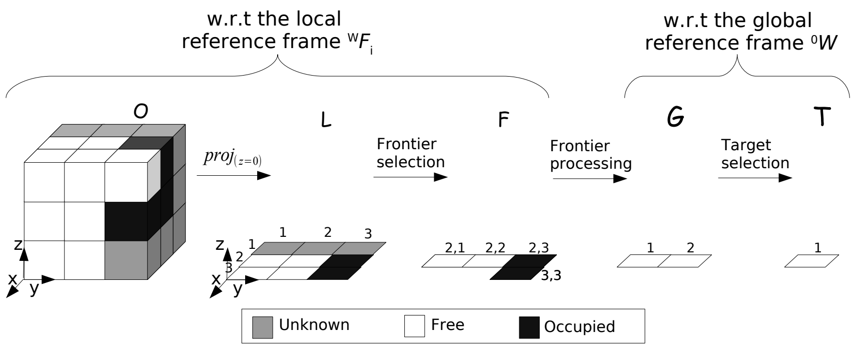


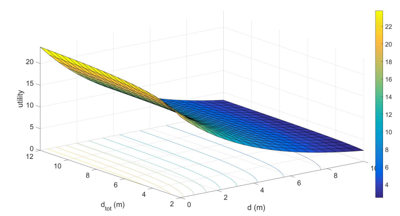
Fig. 4 Grid map structure evolution: from 3D voxels $\mathcal{O}$ to 2D cells $\mathcal{L}$ to 2D frontier cells $\mathcal{F}$ to candidate frontier cells – points – $\mathcal{G}$ (candidate targets) to 2D target cells – points – $\mathcal{T}$.

4.2.3 Utility function

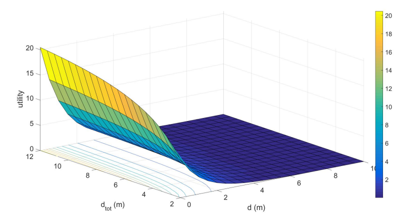
The proposed utility function (See Eq. 1) aims to simultaneously increase the explored area rate and to reduce the distance of each UAV to its corresponding target. It also considers the average distances to each robot in the group to this target in order to maximize distances among robots.

$$\boldsymbol{U}(UAV_i, \mathbf{t}_j) = \boldsymbol{I}(\mathbf{t}_j)\exp(-\lambda.(dmin(\mathbf{p}_i, \mathbf{t}_j) + \\ + \frac{n_c - 1}{\sum_{k=1,k\neq i}^{n_c}(dmin(\mathbf{p}_k, \mathbf{t}_j))})), \quad (1)$$

where UAV$_i$ is the considered robot, $\mathbf{t}_j \in \mathcal{G}$ and $\boldsymbol{I}(\mathbf{t}_j)$ are respectively the candidate target and its corresponding information gain, λ is a trade-off parameter, n_c is the number of UAVs in the cluster $\mathcal{C}$, and $dmin(\mathbf{p}_i, \mathbf{t}_j)$ is the minimum distance from UAV$_i$'s pose to the candidate target j. The proposed utility function is inspired from [17] and it has been presented in our previous work [22]. In the case of a single UAV, the utility function tends to choose the closest target with the maximum of information gain. Regarding the Multi-UAV case, the utility function is based on the average neighbors distances. As shown in Fig. 5, with an information gain of $\boldsymbol{I}(\mathbf{t}_j) = 25$ and three UAVs in the cluster ($n_c = 3$); an increasing distance of UAV to the target will reduce the utility function. Whereas, the more average distance of other UAVs w.r.t. the target, the more the utility. So the function tends to chose the closest target to the considered UAV but at the same time, the farthest one from the others. The proposed utility function performs a trade-off between rapid exploration and filling in details the map using a tuning parameter λ. From Fig. 5a and Fig. 5b, it is noticed that the bigger λ, the less important the distance $dmin(\mathbf{p}_i, \mathbf{t}_j)$ and thus filling in details is favored over rapid exploration and vice versa.



(a) $\lambda = 0.2$



(b) $\lambda = 0.8$

Fig. 5 Utility function behavior: $\boldsymbol{I}(t_j) = 25$, $n_c = 3$, $d_{tot} = \sum_{k=1,k\neq i}^{n_c}(dmin(\mathbf{p}_k, \mathbf{t}_j))$. The average distance of other UAVs d_{tot} has a minimum value different from zero since n_c is different from zero too.

4.2.4 Goal assignment process

In order to make appropriate target-to-UAV assignment, the utility of reaching each candidate frontier is considered. The goal assignment process is described in Algo. 3. For each UAV$_i$, the utilities of reaching all

Algorithm 3 Goal assignment algorithm.

Input: Candidate targets $\mathbf{t}_k \in \mathcal{G}$, $k \in [1...n_g]$ and their respective information gain $\boldsymbol{U}(\mathbf{t}_k)$, position $\mathbf{p}_i$ of all robots in the considered cluster $\mathcal{C}$.
Output: $\theta(UAV_i, \mathbf{t}_g)$ assignment of UAV$_i$ with target g.

```
T = ∅.
while no goal for UAV_i do
    Compute its corresponding utility of reaching each remaining candidate goal U(UAV_i, t_k) with t_k ∈ G \ T.
    t_g = argmax_{t_k ∈ G∩T} U(UAV_i, t_k).
    Schedule the information gain of the remaining candidates t_k ∈ G \ T.
    T = T ∪ {t_g}.
end while
return θ(UAV_i, t_g) assignment.
```

the candidate targets are computed. Then, the target $\mathbf{t}_g$ that maximizes the utility is assigned to UAV$_i$. After that, $\mathbf{t}_g$ is removed and the remaining candidate targets are scheduled in order to avoid to select the same target or another one close to it. This assignment process is performed for the available UAVs in $\mathcal{C}$ in a sequential manner until getting all assigned targets $\mathbf{t}_g \in \mathcal{T}$ with $g \in [1...n_t]$ (See Fig. 4). The goal selection process is realized by each cluster/group *leader* (if $n > n_c$) or the Fleet *leader* (if $n = n_c$). This assignment aims to distribute the robots in the environment in a cooperative way to explore simultaneously different unknown regions. As long as candidate frontier points are still available, the *leader* continues to assign targets to *explorers* and they attempt to reach their assigned goals. When the *leader* notices that no candidate targets are left, that means that all the environment has been explored successfully and the mission is accomplished. Thus, it has to send back to the *explorers* an acknowledgment to prevent them assuming a communication loss.

The target assignment process is performed in each loop. The frequency of assigning targets is important since it defines when a target should be assigned to the UAV. This frequency impacts the duration and the efficiency of the mission. In a distributed approach, as soon as the UAV reaches its current target, it selects a new one without consulting the others. In a centralized approach, the first UAV to reach its current target has to wait until the others reach their respective targets. This can be a problem as soon as one of them fails or leaves the mission. Another possibility is to begin to assign targets once one UAV reaches its target. But this may generate incomplete tasks. In the proposed strategy, the frequency of assignment or loop rate r is predefined depending on the average time to reach a target such that $r = \{\frac{s}{v_{i,max}}, \frac{s}{v_{i,min}}\}$ where s is the maximum sensor range and v_i is the UAV's velocity.

The information gain of each remaining candidate target $\boldsymbol{I}(\mathbf{t}_k)_t$ with $\mathbf{t}_k \in \mathcal{G} \setminus \{t_g\}$ at time $t-1$, that belongs to the threshold range $[r_{min}, r_{max}]$, is scheduled at time t depending on its distance w.r.t. the target t_g, using Eq. 2. This function is Gaussian with an amplitude that corresponds to the information gain maximum value; a center that corresponds to the target position $(\mathbf{t}_g(x), \mathbf{t}_g(y))$; and σ_x and σ_y that spread the blob in x and y axis, respectively.

$$\boldsymbol{I}(\mathbf{t}_k)_t = \boldsymbol{I}(\mathbf{t}_k)_{t-1}(1 - \exp(-(\frac{(\mathbf{t}_k(x) - \mathbf{t}_g(x))^2}{2.\sigma_x^2} + \frac{(\mathbf{t}_k(y) - \mathbf{t}_g(y))^2}{2.\sigma_y^2}))), \quad (2)$$

where $\mathbf{t}_k(x)$ and $\mathbf{t}_k(y)$ are the remaining candidate target coordinates; $\mathbf{t}_g(x)$ and $\mathbf{t}_g(y)$ are the target coordinates; and σ_x and σ_y are the spreads of the blob. The less the distance of the frontier point $\mathbf{t}_k$ w.r.t. the target $\mathbf{t}_g$, the less the information gain. When reducing the information gain, the candidate targets are less likely to be chosen and thus, robots ensure a certain distance among their future targets.

4.2.5 Path planning and control

As explained in Sect.4.1, UAVs are assumed to navigate in a simplified 2D environment with a fixed z value. Block **6** in Fig. 1 is responsible for planning a path to the selected target and attempting to reach it. For the navigation task, each UAV maintains a local and a global planner along with a local and a global costmap, respectively. The costmap is a 2D cell grid $\mathcal{L}$ with additional inflation that consists on propagating cost values out from occupied cells and decreasing them with distance. The global costmap has the size of the UAV's map whereas, the local costmap has a fixed size moving window. Given a starting point – the current pose – and an endpoint – the assigned target – in the global costmap, the global planner produces a plan using a navigation function computed with Dijkstra algorithm [12]. It consists on following the adjacent free cells until reaching the goal. Then taking into account the local costmap, the local planner generates velocity commands for the UAV's mobile base. A recovery rotational behavior is also performed when needed in order

to clear the robot's field of view. The described tasks above are ensured by the *move base*[1] package.
The target assigned by the *leader* is ensured to belong to an unknown area using the exploration strategy. The trajectory planning process is performed locally on each robot. And since the UAVs do not exchange their local maps nor fuse them, they are likely to revisit already explored areas while following the planned path. To minimize these overlapped regions during navigation, a priority is given to frontier points $\mathbf{f}_{i,j}$ to be a target for UAV_i over UAV_k with $k \neq i$. This helps the UAV to maintain the same direction during exploration.

5 Inter-UAV communications

Interactions among members of the fleet are useful for the exploration strategy to prevent UAVs to explore the same regions, and allow them to cooperatively discover the unknown areas more rapidly and in an optimized manner. However, inter-UAV communications is a challenging issue that requires to answer some practical questions: Which type of data nodes must exchange? If so, how to identify the end-points of the data exchange? How to cope with communication limitations and how often data or control information must be exchanged? In this Section, we will try to give a quick summary of the answers to these questions.

Concerning the type of exchanged data and the end-points of the communication flow, in the proposed cooperative strategy, the UAVs must exchange data and control information in the following way:

- *Data*: Only local frontier points $\mathbf{f}_{i,j} \in \mathcal{F}$, current pose $\mathbf{p}_i$, and current target point $\mathbf{t}_g$ are exchanged among the UAVs, instead of the whole copy of the local map. This is expected to produce a considerable reduction of exchanged data volume, and, consequently, memory consumption.
- *Control*: UAV_i with $i \in [1...n_c]$ forward its *id* number and current pose $\mathbf{p}_i$. All the *explorers* UAV_k with $k \in [1...n_c]$ and $k > i$, send their local frontier points $\mathbf{f}_{k,l} \in \mathcal{F}$ to the selected *leader*. Then, the *leader* performs the goal assignment process and sends back to each *explorer* in $\mathcal{C}$ a selected target point $\mathbf{t}_g$ to reach.

Concerning the communication limitations, it is important to ensure the mission continuity. In case of losing contact with the *leader* and before another one is selected, *explorers* let a timer τ expire while waiting for target assignment. If no target is received, the *explorer* selects its own target according to local information. Using this strategy, as long as – at least – one UAV exists in the fleet, the mission will continue until all the bounded environment is explored (no candidate frontier points are left).

In the proposed strategy, data flow exchange is repeated at each iteration while taking into account network topology changes to define clusters. The starting points and endpoints are defined according to these roles. The UAV's role also specifies the type of exchanged data. In addition to the exchanged current pose $\mathbf{p}_i$ and *id* number i, if the UAV_i is an *explorer*, it would passively share information about itself and its surrounding environment with the *leader* (frontier points $\mathbf{f}_{i,j} \in \mathcal{F}$); else, its role would be to send targets to visit to the *explorers* (target points $\mathbf{t}_k \in \mathcal{G}$).

6 Results

Simulations have been performed to evaluate the proposed exploration strategy. Additional tests while using relative localization have been done to measure the system performances. Furthermore, experiments using an infrastructure-less network have been conducted to get as close as possible to a real UAVs deployment scenario. The obtained results point out the performance of the proposed multi-robot system under real network limitations.
The number of robots used for evaluation is limited to three, however, the proposed system architecture is not constrained to a fixed number of robots.

6.1 Simulation results

The simulations are performed using Robot Operating System (ROS) running on a 2.60GHz i7 Linux machine. For the quad-rotor simulation, the AR-drone model[2] equipped with an RGB-D camera in a forward-looking configuration, is used. A bounded unknown environment is generated using Gazebo simulator.

6.1.1 Parameters tuning

The utility function (See Eq. 1) used in the exploration strategy can be tuned, using a trade off parameter λ, between fast exploration and filling in details the map.

[1] http://wiki.ros.org/move_base

[2] http://wiki.ros.org/ardrone_autonomy

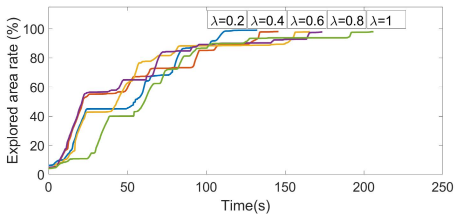


Fig. 6 The impact of variating the trade off parameter λ over exploration time.

Table 3 Common parameters.

Parameter	Value
RGB-D maximum range s (m)	4
loop rate l (s)	20
Trade-off parameter λ	0.2
RGB-D horizontal FoV	$\pi/3$
Occupancy grid resolution (m)	0.05
Range to schedule Ig $[\sigma_x, \sigma_y]$ (m)	$[3, 3]$
Environment dimension (m^2)	8×8
Linear velocity v_i $(m.s^{-1})$	$[0.1, 0.3]$
Angular velocity ω_i $(rad.s^{-1})$	$[0.1, 0.3]$

Fig. 6 shows different runs while varying this parameter. By increasing λ, the information gained when reaching the goal is favored over the distance and thus the cost to it, and vice versa. So, when λ is small, the traveled distance is small and so the exploration time. Though, some times during the mission, high values of λ are noticed to reach higher exploration rate than smaller ones.
The frequency or loop rate r of target assignment may also affect exploration time performances. The values of r are variated to take into account the robot velocity $\mathbf{v}_i$ and the sensor's maximum range s. The impact of varying the loop rate is evaluated in Fig. 7.

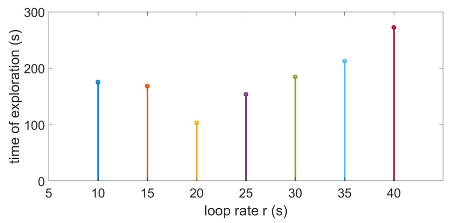


Fig. 7 Exploration time while varying the loop rate r.

Given a robot velocity $\mathbf{v}_i = [0.1, 0.3]$ $m.s^{-1}$ and a maximum sensor range of $s = 4$ m, loop rate ranges in $r = [10, 40]$. The loop rate should not be too small to allow the robot to reach its target; nor too big to prevent long waits for the next goal assignment.
Depending on results in Fig. 6 and Fig. 7, respectively, λ is set to 0.2 and r to $20s$. The simulation parameters are summarized in Table 3.

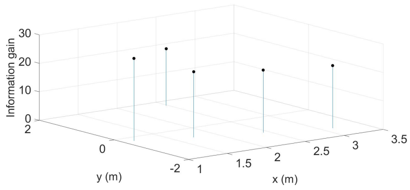
6.1.2 Performance of the exploration strategy

The proposed exploration strategy has been evaluated in terms of distribution of the robots in the environment, overlap rate, exploration time, and total traveled distance by each robot. Furthermore, several validations have been done to select the parameters configuration.
The goal assignment process is performed according to the algorithm described in Sect. 4.2.4. Nevertheless, after assigning a target to the first robot in the list, the same target or another one close to it may be assigned to the second robot in the list. To overcome these issues, the information gain of the remaining candidate targets is scheduled. This allows to discard an already assigned target and to keep a certain distance between the new target and the previous one assigned.
Suppose that a target is assigned to the first robot in the cluster list. Fig. 8 shows the goal selected for the second robot when a sequential assignment is performed:

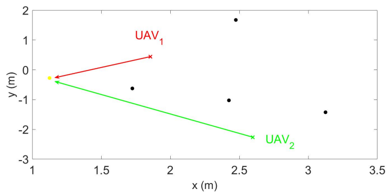
- without further frontier points processing (See Fig. 8b) which results in assigning the same target to two different robots;
- while removing the assigned target from the remaining candidate frontier points (See Fig. 8c) which results in too close assigned targets.
- while scheduling information gain after each target assignment (See Fig. 8d) which allows to space out assigned targets. The information gain is scheduled following Eq. 2. The information gain value increases with distance to the candidate target $\mathbf{t}_g$.

The goal assignment process may sometimes be not optimal since it depends on the robots' order in the list. For example, suppose that robots UAV$_i$ and UAV$_j$ have the same best target assignment $\mathbf{t}_k$ such that it offers the maximum utility over candidate frontier points: $\mathbf{t}_k = argmax_{\mathbf{t}_m} \boldsymbol{U}(\text{UAV}_i, \mathbf{t}_m)$ with $\mathbf{t}_m \in \mathcal{G}$ and $\mathbf{t}_k = argmax_{\mathbf{t}_n} \boldsymbol{U}(\text{UAV}_j, \mathbf{t}_n)$ with $\mathbf{t}_n \in \mathcal{G}$. Robot UAV$_i$ have another candidate frontier point $\mathbf{t}_l$ with $\boldsymbol{U}(\text{UAV}_i, \mathbf{t}_k) > \boldsymbol{U}(\text{UAV}_i, \mathbf{t}_l) > \boldsymbol{U}(\text{UAV}_j, \mathbf{t}_k)$. So the optimal solution would be to assign $\mathbf{t}_l$ to UAV$_i$ and $\mathbf{t}_k$ to UAV$_j$. But, if UAV$_i$ is the first in the list, $\mathbf{t}_k$ is assigned to it and another candidate frontier point with less utility than $\mathbf{t}_k$, is assigned to UAV$_j$. Thus the solution with sequential goal assignment is not always optimal.

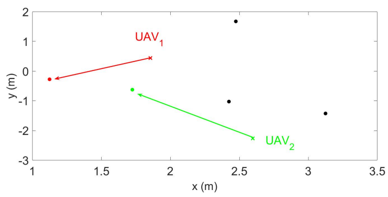
To overcome this problem, all the number of possible combination $\frac{n_g!}{n_g!\,(n_g-n_c)!}$ with n_g the number of candidate targets and n_c the number of robots, needs to be considered. This considerably increases the computation time when the robots number increases. There-



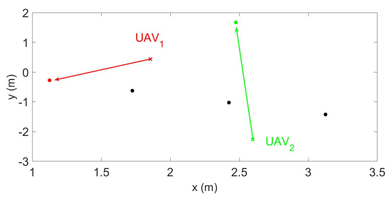
(a) Candidate targets.



(b) Targets assignment when no further process is performed for the remaining candidate targets.



(c) Targets assignment when UAV_1's target is removed from the remaining candidate targets.



(d) Targets assignment when the information gain of the remained candidate targets is scheduled.

Fig. 8 Goal assignment: after assigning a target to UAV_1, a target is assigned in a sequential manner to UAV_2. (a) represents the candidate frontier points with their respective information gain. (b), (c), and (d) represent respectively, the targets assignment when: no further process is performed for the remaining candidate targets; UAV_1's target is removed from the remaining candidate targets and; the information gain of the remained candidate targets is scheduled.

fore, in the proposed algorithm, sequential assignment is favored over computing all possible permutations.
The use of an effective goal assignment process should limit the generated overlap. In Fig. 9, the time evolution of overlap is evaluated using two cooperative robots. The overlap undergoes a significant increase at the end of the exploration to reach 33%. This is explained by the closeness of the local maps at the end of the mission to fully fill in details the global grid map.

While reaching their respective assigned goals, each robot is in charge of creating a detailed grid map of

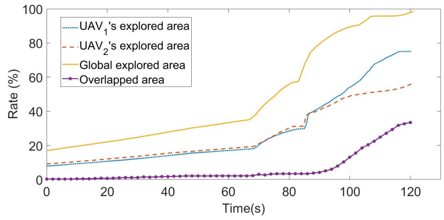


Fig. 9 Explored and overlapped area rate using two cooperative UAVs.

the visited area in order to get a global map of the environment.

Fig. 10 shows the evolution of the respective projected 2D local grid map of two robots during a cooperative exploration mission. The global projected 2D grid map is also created and represented for evaluation.

The robots' initial positions are (1,0,0) for UAV_1 and (1,-3,0) for UAV_2. Despite a relatively close initial positions, the proposed strategy effectively spread the robots so that UAV_1 is in charge of the left side of the environment and UAV_2 of the right one.
To effectively evaluate the exploration strategy performance in terms of time and traveled distance, different runs with 1, 2 and 3 UAVs have been conducted until the explored area rate reaches almost 99%. A traveled distance evaluation is represented in Fig. 11. This distance effectively decreases with the number of UAVs. The average distance traveled by each UAV is reduced by 55% for 2 UAVs and by 62% for 3 UAVs. The error of the traveled distance is slightly reduced from 1 to 2

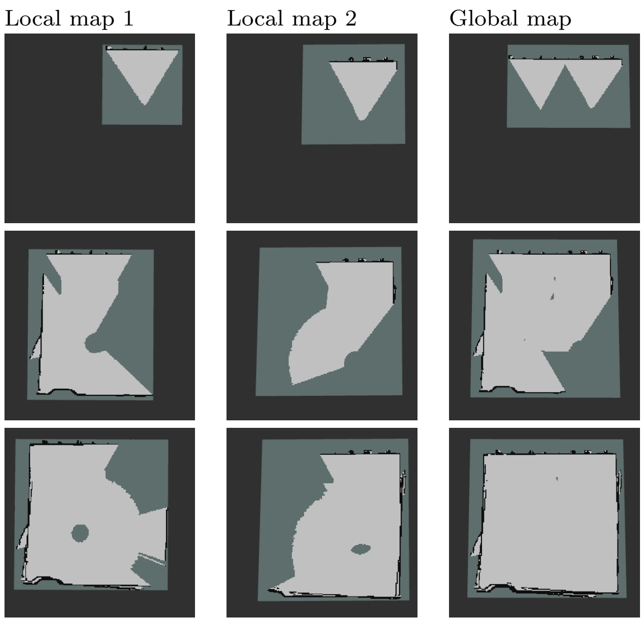


Fig. 10 Coordinated exploration using two robots. Columns 1, 2 and 3 show the evolution of the local grid map of UAV_1, UAV_2 and the global grid map over time, respectively.

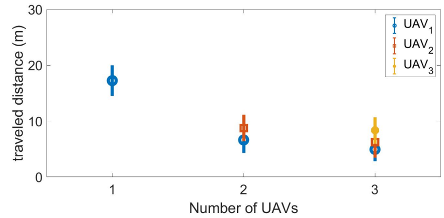


Fig. 11 Traveled distance evaluation.

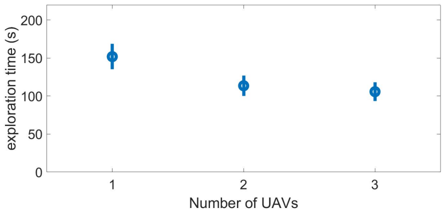


Fig. 12 Average exploration time.

and 3 UAVs.
Fig. 12 shows that the average exploration time decreases when the number of robots in the fleet increases. The computed error is decreased as well. The exploration time is reduced by 25% for 2 UAVs and by 30% for 3 UAVs. The exploration time and distance are not directly divided by 2 or 3 when multiplying by 2 or 3 the number of robots, respectively. During these simulations, the robots' initial positions are: (1,0,0) for 1 UAV; (1,0,0) and (1,-3,0) for 2 UAVs; and (1,1,0), (1,-1,0) and (1,-3,0) for 3 UAVs. The results presented above (See Sec.6.1.2) were evaluated without a relative localization. So, for a more challenging realistic scenario, runs with relative localization algorithm have been performed to evaluate both SLAM and system performances along relative localization.

6.1.3 Comparison with existing exploration strategies

In order to evaluate our proposed exploration strategy, we compare it with some state-of-the-art well known strategies based on both random and closest frontiers. We present the results obtained with a Random Frontier (RF) selection strategy, as a baseline for comparison [18]. Also, we compared our proposed strategy to a simple but efficient frontier-based exploration strategy based on exploring the Closest Frontier (CF) [39].

Fig. 13 shows this comparison using 1, 2 and 3 UAVs. Our proposed strategy clearly outperforms the two other methods in the three cases. By increasing the number of UAVs, our strategy gives better results compared to the others. Moreover, our strategy presents a relatively small and almost constant variation against the traveled distance range.

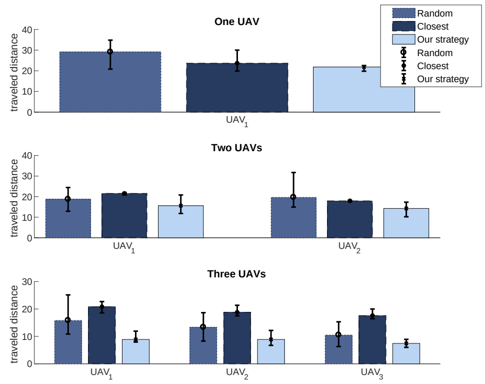


Fig. 13 Exploration strategies comparison.

6.1.4 System performance with the relative localization

The ORB-SLAM2 approach [25] have been implemented[3] and evaluated along with the proposed exploration strategy. Each robot runs its own ORB-SLAM2 within its local reference frame ${}^{W}F_i$. But, exchanged information among robots have to be expressed in a common reference frame. So information such as the pose $\mathbf{p}_i$ and the frontier points $\mathbf{f}_{i,j}$ are necessarily transformed into the global reference frame ${}^{0}W$ before being exchanged. Therefore the *leader* makes all the needed computation and sends back to the *explorers* the targets in ${}^{0}W$. When a robot receives its assigned goal, it transforms it into ${}^{W}F_i$ in order to plan a path to it. To perform a transformation from local reference ${}^{W}F_i$ to global one ${}^{0}W$, the robot has to know – at least – its initial pose w.r.t. ${}^{0}W$. So, as explained in Section 3.2, the global reference frame of the environment is initialized such that it coincides with the local reference frame of the first group *leader* in the fleet: ${}^{0}W \equiv {}^{W}F_1$ in this example.

Then, by detecting this robot using tags mounted on it, the other robots are able to estimate their respective transform to it ${}^{F_1}\left[\mathbf{R}\ \mathbf{t}\right]_{F_j}$, $j \in [2 ... n_c]$. For simulation evaluations, the information of transform – computed while detecting the tag – are assumed to be known.

The SLAM approach has been implemented along with our exploration strategy and evaluated using the Gazebo virtual environment. Despite the need of structure and texture, the ORB-SLAM2 has been able to perform localization in a simulated area. Table 4 resumes tests to set some of the required parameters (linear velocity v_i and angular velocity ω_i) to perform visual SLAM without tracking loss or – at least – a fast re-localization.

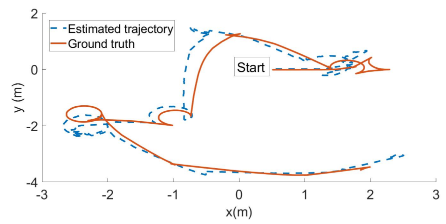
[3] *https://github.com/raulmur/ORB_SLAM2*

Table 4 SLAM behavior while modifying linear and angular velocity. T.L: Tracking Loss; RL: Re-Localization.v_i and ω_i are in $m.s^{-1}$ and $rad.s^{-1}$, respectively.

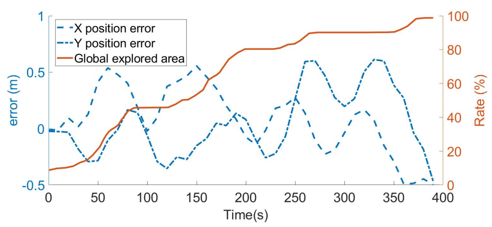
ω_i \ v_i	0.1	0.2	0.3
0.1	No T.L.	No T.L.	T.L., R.L.
0.2	No T.L.	T.L., R.L.	T.L.
0.3	T.L., R.L.	T.L.	T.L.

Fig. 14 shows the SLAM system performances during one robot exploration. The exploration time is relatively important compared to the exploration without SLAM since the velocity have been considerably reduced. The drift and so the trajectory error are limited thanks to the loop-closure algorithm performed within ORB-SLAM2.

Results for exploration using two UAVs running each one the SLAM algorithm are presented in Fig. 15. As expected, the exploration time using 1 UAV is greater than using 2 UAVs. An important drift occurs at the end of the UAV$_2$'s trajectory because it did not visit a known place and therefore it could not rectify its trajectory with a loop closure optimization.

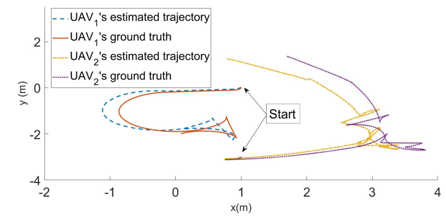


(a) The estimated trajectory versus the Ground truth.

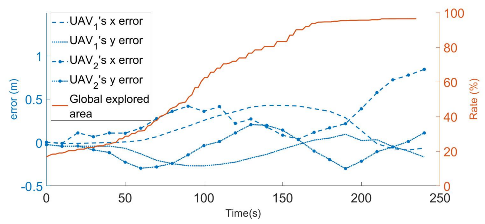


(b) The exploration rate and the estimated position errors during the mission.

Fig. 14 One UAV exploration using SLAM algorithm. (the kinect's maximum range is 4m, the UAV's linear velocity $v_i \in [0.1, 0.2]$ $m.s^{-1}$ and angular velocity $\omega_i = 0.1$ $rad.s^{-1}$). The RMSE(x)=0.2968 and RMSE(y)=0.2944.



(a) The estimated trajectory versus the Ground truth.



(b) The exploration rate and the estimated position errors during the mission.

Fig. 15 Two UAVs cooperative exploration performing each one SLAM algorithm. (the kinect's maximum range is 4m, the UAV's linear velocity $v_i \in [0.1, 0.2]$ $m.s^{-1}$ and angular velocity $\omega_i = 0.1$ $rad.s^{-1}$). For UAV$_1$, the RMSE(x)=0.2419 and RMSE(y)=0.1415 ;for UAV$_2$ the RMSE(x)=0.3649 and RMSE(y)=0.1614.

6.2 System performance using an infrastructureless network

As the UAVs are equipped with IEEE 802.11b,g wireless card, we setup an infrastructureless network within the set of robots to quantify the data exchange among members of the fleet, as well as, to determine the performance of the robot network. Runs with 2 and 3 UAVs are performed. The network is composed of two 2.60GHz i7 Linux machines and one 2.50GHz i7 Linux machine. A multi-master system – also called multi-core system – is used. Each UAV manages its own master to avoid losing contact with ROS master in case of unstable network connection. In this type of system, a synchronization among ROS cores is required to allow robot to robot communication. Hence, the multi-master fkie package[4], where cores synchronization is ensured by UDP protocol, is used. The TCP protocol is responsible for exchanging data among ROS topics through the network. For an effective evaluation, especially concerning the time, clock synchronization needs to be ensured. Network time protocol (NTP) is used to synchronize laptops within a few milliseconds of Coordinated Universal Time (UTC).
The first evaluation aims to point out the amount of the exchanged data (See Fig. 16). The size of data to

[4] $http://wiki.ros.org/multimaster\_fkie$

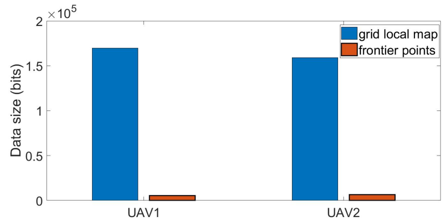


Fig. 16 Data size when UAVs exchange a whole copy of their local grid map versus frontier points of it. Values represent the average exchanged data during the mission.

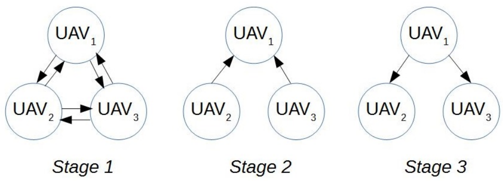
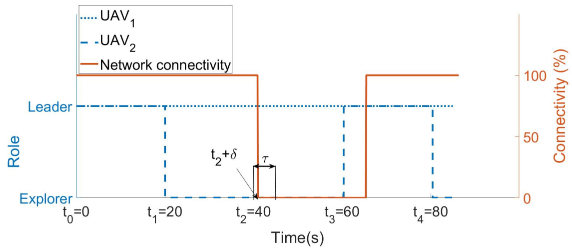
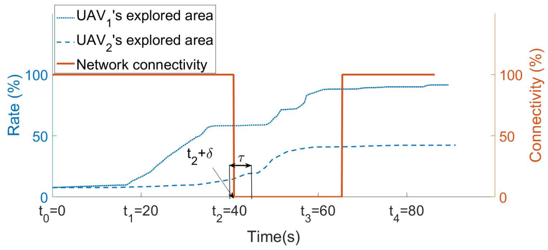


Fig. 17 Network topology evolution during a loop r with three cooperative UAVs.



(a) Robot role along with network connectivity.



(b) Exploration rate along with network connectivity.

Fig. 18 Two cooperative robots exploration along with network connectivity.

be exchanged is almost divided by 10 in the proposed strategy that shares only frontier points compared to a strategy that makes robots to exchange a whole copy of their local map. Depending on the size and frequency of the exchanged data, the allocated time for communications may increase with the increasing number of UAVs. Thus, evaluations of timing behavior and its potential impact on the exploration performances have been conducted. Fig. 17 shows the network topology evolution during data exchange. Threes stages are noticed where the exchanged information are the following:

- *Stage 1:* *id* number and poses $\mathbf{p}_i$, $i \in [1...n_c]$.
- *Stage 2:* frontier points $\mathbf{f}_{i,j}$, $i \in [1...n_c]$, $j \in [1...n_i]$.
- *Stage 3:* target points assignment $\theta(\text{UAV}_i, \mathbf{t}_k)$, $i \in [1...n_c]$, $k \in [1...n_g]$.

Table 5 shows the average data exchange time during exploration. A slight increase in the computed average time occurs when increasing the number of UAVs. This time spent in communication is negligible compared to the total time of exploration.

To evaluate the system behavior during communication failure, the network connectivity has been voluntarily interrupted during the exploration. Fig. 18 shows the UAV's role and the exploration rate performance when the network connectivity is interrupted and then recovered.

The system performs neighbor discovering, role selection and target assignment at each loop of $t_{i+1} = t_i + i.r$ with $t_0 = 0s$ and $r = 20s$. In Fig. 18a, at $t = t_0$, both robots begin with a *leader* role. When discovering each other (at $t = t_1$), UAV_1 selects itself as *leader* and UAV_2 as *explorer*. Consequently, UAV_1 assigns a target to UAV_2. UAV_2 receives the target and attempts to reach it. At $t = t_2 + \delta$, the connectivity is voluntarily interrupted, just after the role selection but before the target information is assigned to UAV_2. After a time period τ, UAV_2 selects a target taking into account its own local data. In the next loop (at $t = t_3$), since the connectivity is still interrupted, UAV_2 finds no neighbors and selects itself as a *leader*. Both robots perform exploration independently, that is, without cooperation. Shortly after $t = t_3$, the connectivity is re-established. Thus, robots are able to cooperate again and UAV_2 takes over the role of *explorer*.

In Fig. 18b, even after the network connectivity is interrupted, the exploration continues to be performed by both UAVs. When the connectivity is re-established, the *leader* collects the frontier points and performs frontier processing where it finds that no candidates targets are remaining, that is, all the environment is now explored. Therefore, the mission is accomplished.

7 Discussion

The aim of this work is to propose a complete system-of-systems framework used for multi-robot environment exploration. In this work, we propose both novelty and exploitation of state-of-the-art methods to make effective this scalable framework. As a property of system-of-systems, each of the component/block presented in Fig. 1 can be replaced using different approaches than the ones we used here.

Table 5 Communication module timings.

# UAVs		**Time spent in *stage 1* (s)**			**Time spent in *stage 2* (s)**			**Time spent in *stage 3* (s)**			**Time for exploration (s)**
		UAV_1	UAV_2	UAV_3	UAV_1	UAV_2	UAV_3	UAV_1	UAV_2	UAV_3	
Two UAVs	UAV_1	$\emptyset$	0.136± 0.139		$\emptyset$	$\emptyset$		$\emptyset$	0.022± 0.017		120, 1
	UAV_2	0.065± 0.068	$\emptyset$		0.026± 0.008	$\emptyset$		$\emptyset$	$\emptyset$		
Three UAVs	UAV_1	$\emptyset$	0.056± 0.065	0.575± 0.769	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	0.335± 0.407	0.765± 0.678	86
	UAV_2	0.107± 0.111	$\emptyset$	0.483± 0.678	0.185± 0.244	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	
	UAV_3	0.267± 0.165	0.616± 0.549	$\emptyset$	0.251± 0.109	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	

Moreover, the proposed exploration strategy ensures a mission continuity in the case of communication loss. Nevertheless, the robots may explore regions already explored by the other ones, since no local maps are exchanged nor fused to keep track of visited areas. Thus, in case of communication loss, the mission accomplishment is favored over consumption minimization of resources, such as time and energy.

In this paper, one limitation is that the explored environment is free of obstacles. This is due to the concave shape assumption of the local maps (without obstacles inside) in the frontier points processing step. This algorithm needs to be adapted in order to take into account obstacles. Consequently, the utility function will use the robot path instead of the travelled distance.

8 Conclusions and future works

In this paper, we introduced a new distributed multi-UAV system architecture for SLAM-based cooperative exploration under limited communication bandwidth, designed as a System-of-Systems. Based on their embedded visual sensor, and using state-of-the-art visual-SLAM algorithms, UAVs are individually able to localize themselves, discover their neighbors and create a 3D grid map of their environment. To explicitly handle the connectivity limitations, the proposed cooperative exploration strategy is based on a new utility function that takes into account the distance of each UAV in the group from the unexplored set of targets, and makes a trade-off between fast exploration and detailed grid map using limited network resources. Using the group *leader* decision making, targets are assigned to UAVs in order to simultaneously explore different regions of the environment in an optimized manner. Simulation results shows that the strategy adopted minimizes the mission time by 25% for 2 UAVs and by 30% for 3 UAVs. It also decreases the average traveled distance by each UAV by 55% for 2 UAVs and by 62% for 3 UAVs. In addition and in contrary to recent approaches, by scheduling exchanged information, UAVs are efficiently spread out into the environment while avoiding to select the same target or another one close to it. Moreover, test-bed results with network communication, show that by exchanging frontier points, local poses and assigned targets, the adopted strategy reduces the data needed up to 10 times compared to a strategy that makes the robots to exchange the whole local maps. Furthermore, the group *leader* decision making allows to take into account the communication drop-out or failure by adapting the UAV's role according to the network topology changes.
When the communication is interrupted, ending the mission is favored over the time spent to do it. Thus, in the future work, under the proposed exploration strategy, we aim to investigate the possibility of keeping track of other UAVs' explored area using the frontier points. Also, we plan to implement the framework on real quad-rotor fleet; and integrate obstacles for more complex environment exploration.

Acknowledgements This work has been carried out in the framework of the Labex MS2T and DIVINA challenge team, which are funded by the French Government, through the program "Investments for the Future", managed by the French National Research Agency (Reference ANR-11-IDEX-0004-02).

References

1. Beard, R.W., McLain, T.W.: Multiple uav cooperative search under collision avoidance and limited range communication constraints. In: 42nd IEEE International Conference on Decision and Control, vol. 1, pp. 25–30 Vol.1 (2003)
2. Benavides, F., Monzón, P., Chanel, C.P.C., Grampín, E.: Multi-robot cooperative systems for exploration: Advances in dealing with constrained communication environments. In: Robotics Symposium and IV Brazilian Robotics Symposium (LARS/SBR), 2016 XIII Latin American, pp. 181–186. IEEE (2016)

3. Bennewitz, M., Burgard, W.: An experimental comparison of path planning techniques for teams of mobile robots. In: Autonome Mobile Systeme 2000, pp. 175–182. Springer (2000)
4. Bresson, G., Aufrre, R., Chapuis, R.: A general consistent decentralized simultaneous localization and mapping solution. Robotics and Autonomous Systems **74**, 128 – 147 (2015)
5. Burgard, W., Moors, M., Fox, D., Simmons, R., Thrun, S.: Collaborative multi-robot exploration. In: Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on, vol. 1, pp. 476–481. IEEE (2000)
6. Burgard, W., Moors, M., Stachniss, C., Schneider, F.E.: Coordinated multi-robot exploration. IEEE Transactions on robotics **21**(3), 376–386 (2005)
7. Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., Leonard, J.J.: Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. IEEE Transactions on Robotics **32**(6), 1309–1332 (2016)
8. Cieslewski, T., Kaufmann, E., Scaramuzza, D.: Rapid exploration with multi-rotors: A frontier selection method for high speed flight. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2017)
9. Couceiro, M.S., Figueiredo, C.M., Rocha, R.P., Ferreira, N.M.: Darwinian swarm exploration under communication constraints: Initial deployment and fault-tolerance assessment. Robotics and Autonomous Systems **62**(4), 528 – 544 (2014)
10. Cunningham, A., Paluri, M., Dellaert, F.: Ddf-sam: Fully distributed slam using constrained factor graphs. In: Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference. IEEE (2010)
11. Dai, R., Fotedar, S., Radmanesh, M., Kumar, M.: Quality-aware uav coverage and path planning in geometrically complex environments. Ad Hoc Networks (2018)
12. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numerische mathematik **1**(1), 269–271 (1959)
13. Forster, C., Lynen, S., Kneip, L., Scaramuzza, D.: Collaborative monocular slam with multiple micro aerial vehicles. In: Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference. IEEE (2013)
14. Fox, D., Ko, J., Konolige, K., Limketkai, B., Schulz, D., Stewart, B.: Distributed multirobot exploration and mapping. Proceedings of the IEEE **94**(7), 1325–1339 (2006)
15. Fujimura, K., Singh, K.: Planning cooperative motion for distributed mobile agents. Journal of Robotics and Mechatronics **8**, 75–80 (1996)
16. Gupta, L., Jain, R., Vaszkun, G.: Survey of important issues in uav communication networks. IEEE Communications Surveys Tutorials **18**(2), 1123–1152 (2016). DOI 10.1109/COMST.2015.2495297
17. Heng, L., Gotovos, A., Krause, A., Pollefeys, M.: Efficient visual exploration and coverage with a micro aerial vehicle in unknown environments. In: 2015 IEEE International Conference on Robotics and Automation (ICRA), pp. 1071–1078. IEEE (2015)
18. Holz, D., Basilico, N., Amigoni, F., Behnke, S.: Evaluating the efficiency of frontier-based exploration strategies. In: Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK), pp. 1–8. VDE (2010)
19. Hornung, A., Wurm, K.M., Bennewitz, M., Stachniss, C., Burgard, W.: Octomap: An efficient probabilistic 3d mapping framework based on octrees. Autonomous Robots **34**(3), 189–206 (2013)
20. Kulich, M., Juchelka, T., Přeučil, L.: Comparison of exploration strategies for multi-robot search. Acta Polytechnica **55**(3), 162–168 (2015)
21. Latombe, J.C.: Robot Motion Planning. Kluwer Academic Publishers, Norwell, MA, USA (1991)
22. Mahdoui, N., Fremont, V., Natalizio, E.: Cooperative exploration strategy for micro-aerial vehicles fleet. In: 2017 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2017), pp. 1–6. IEEE, Daegu, Korea (2017)
23. Min, B.C., Parasuraman, R., Lee, S., Jung, J.W., Matson, E.T.: A directional antenna based leaderfollower relay system for end-to-end robot communications. Robotics and Autonomous Systems **101**, 57 – 73 (2018)
24. Mohanarajah, G., Usenko, V., Singh, M., D'Andrea, R., Waibel, M.: Cloud-based collaborative 3d mapping in real-time with low-cost robots. IEEE Transactions on Automation Science and Engineering **12**(2), 423–431 (2015)
25. Mur-Artal, R., Tardós, J.D.: ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras. IEEE Transactions on Robotics **33**(5), 1255–1262 (2017)
26. Nitsche, M., Krajník, T., Čížek, P., Mejail, M., Duckett, T.: Whycon: An efficent, marker-based localization system. In: IROS Workshop on Open Source Aerial Robotics (2015)
27. Olson, E.: AprilTag: A robust and flexible visual fiducial system. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pp. 3400–3407. IEEE (2011)
28. Pal, A., Tiwari, R., Shukla, A.: Multi robot exploration using a modified a* algorithm. In: Asian Conference on Intelligent Information and Database Systems, pp. 506–516. Springer (2011)
29. Perkins, C.E., Royer, E.M.: Ad-hoc on-demand distance vector routing. In: Proceedings WMCSA'99. Second IEEE Workshop on Mobile Computing Systems and Applications, pp. 90–100 (1999). DOI 10.1109/MCSA.1999.749281
30. Rocha, R., Dias, J., Carvalho, A.: Cooperative multi-robot systems: A study of vision-based 3-d mapping using information theory. Robotics and Autonomous Systems **53**(3), 282 – 311 (2005)
31. Rooker, M.N., Birk, A.: Multi-robot exploration under the constraints of wireless networking. Control Engineering Practice **15**(4), 435–445 (2007)
32. Schmuck, P.: Multi-uav collaborative monocular slam. In: Robotics and Automation (ICRA), 2017 IEEE International Conference on, pp. 3863–3870. IEEE (2017)
33. Schuster, M.J., Brand, C., Hirschmller, H., Suppa, M., Beetz, M.: Multi-robot 6d graph slam connecting decoupled local reference filters. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5093–5100 (2015)
34. Sheng, W., Yang, Q., Tan, J., Xi, N.: Distributed multi-robot coordination in area exploration. Robotics and Autonomous Systems **54**(12), 945–955 (2006)
35. Simmons, R., Apfelbaum, D., Burgard, W., Fox, D., Moors, M., Thrun, S., Younes, H.: Coordination for multi-robot exploration and mapping. In: AAAI/IAAI, pp. 852–858 (2000)
36. Solanas, A., Garcia, M.A.: Coordinated multi-robot exploration through unsupervised clustering of unknown space. In: Intelligent Robots and Systems, 2004.(IROS

2004). Proceedings. 2004 IEEE/RSJ International Conference on, vol. 1, pp. 717–721. IEEE (2004)
37. Werger, B.B., Matarić, M.J.: Broadcast of local eligibility for multi-target observation. In: Distributed autonomous robotic systems 4, pp. 347–356. Springer (2000)
38. Wu, W., Zhang, F.: Robust cooperative exploration with a switching strategy. IEEE Transactions on Robotics **28**(4), 828–839 (2012)
39. Yamauchi, B.: A frontier-based approach for autonomous exploration. In: Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97. 'Towards New Computational Principles for Robotics and Automation', pp. 146–151 (1997). DOI 10.1109/CIRA.1997.613851
40. Yamauchi, B.: Frontier-based exploration using multiple robots. In: Proceedings of the second international conference on Autonomous agents, pp. 47–53. ACM (1998)
41. Yan, Z., Jouandeau, N., Cherif, A.A.: A survey and analysis of multi-robot coordination. International Journal of Advanced Robotic Systems **10**(12), 399 (2013)
42. Yang, S., Scherer, S.A., Yi, X., Zell, A.: Multi-camera visual slam for autonomous navigation of micro aerial vehicles. Robotics and Autonomous Systems **93**, 116 – 134 (2017)
43. Yuan, J., Huang, Y., Tao, T., Sun, F.: A cooperative approach for multi-robot area exploration. In: Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, pp. 1390–1395. IEEE (2010)

Nesrine Mahdoui graduated from Ecole Nationale d'Ingénieur en Génie électrique, Tunisia, and she received an M.Sc. degree in Complex Systems in Interaction from the Université de Technologie de Compiègne (UTC), France, in 2015. She is actually finishing her PhD in Systems of Systems Engineering at the UTC. Her research interests are related to multi-robot's perception and communication for cooperative SLAM applications.
Vincent Frémont received the M.S. degree in automatic control and computer science from the Ecole Centrale de Nantes, France, in 2000 and the Ph.D. degree in automatic control and computer science from the Ecole Centrale de Nantes, France, in 2003. He joined the Institut de Recherche en Communications et Cybernetique de Nantes in 2000, and the Laboratoire de Vision et Robotique de Bourges, in 2003. From 2005 to 2018, he was an Associate Professor at the Université de Technologie de Compiègne (UTC). Since 2018, he is a Full Professor at Ecole Centrale de Nantes within the LS2N Lab. His research interests belong to perception for autonomous mobile robotics with an emphasis on computer vision, machine learning and multi-sensor fusion.
Enrico Natalizio is a Full Professor at Université de Lorraine within the LORIA Lab. Until 2018, he was an Associate Professor at the Université de Technologie de Compiègne (UTC), France, in the Network and Optimization Group within the Heudiasyc Lab. He obtained his Ph.D. from the Universita della Calabria, Italy and he was a visiting researcher at the Broadband Wireless Networking Laboratory at Georgia Tech in Atlanta, USA. Between 2005 and 2010, he worked as a research fellow and a contract professor at the Universita della Calabria, Italy. From 2010 to 2012 he worked at INRIA Lille as a postdoctoral researcher. His current research interests include group communication in wireless robot and sensor networks and coordination and cooperation among swarm networked devices.