Journal of Intelligent & Robotic Systems (2021) 103: 24
https://doi.org/10.1007/s10846-021-01485-0

REGULAR PAPER

Cooperative Sensor-based Selective Graph Exploration Strategy for a Team of Quadrotors

Jinho Kim[1] · Charles D. Eggleton[1] · Stephen A. Wilkerson[2] · S. Andrew Gadsden[3]

Received: 15 October 2020 / Accepted: 17 August 2021 / Published online: 6 September 2021

Abstract

This paper proposes an exploration strategy in unknown environments for a team of quadrotor Unmanned Aerial Vehicles (UAVs). Based on the frontier information, the proposed strategy builds a roadmap of the explored area in form of a Sensor-based Selective Graph (SSG) using simple data trees of the frontier and the hub node only. In particular, the frontier data tree is utilized to consider the adjacent frontier sectors as one frontier sector, and the next target node is generated maximizing the coverage of frontiers at each movement of quadrotors. In addition, to expand the proposed strategy to the three dimensional (3D) workspace with quadrotors, a Multiple Flight Levels (MFL) approach is proposed to increase the efficiency of the exploration. Moreover, when a quadrotor reaches a dead end where no frontier exists, the efficient backtracking algorithm chooses the best path to backtrack efficiently with a graph map provided by the SSG. With these contributions, we successfully develop the frontier-based exploration strategy for multiple quadrotors, and performance of the overall approach is validated by numerical simulations and experiments.

Keywords Cooperative robot exploration · Frontier-based exploration · Selective target node (STN) · Multiple flight levels (MFL) · Graph · Quadrotor · Unmanned aerial vehicle (UAV)

1 Introduction

Mobile robots have been utilized in various missions such as search and rescue [1–3], reconnaissance [4–6], exploration [7–10] in recent years. Among these applications, an exploration can be defined as the act of moving through an unknown environment while building a map that can be used for subsequent navigation [11]. Interest in exploring using robots further increased after the Fukushima nuclear disaster in 2011. Since it is risky for humans to access disaster area, such as radioactive contaminated locations and earthquake damaged structures, use of robots for exploring such a dangerous area has become more prominent.

To explore in unknown environments, *i.e.,* without any prior map information, mobile robots have to be equipped with some sensors to detect their surroundings. Based on the sensor data, frontier-based exploration strategy is proposed to explore in unknown environments in [11]. In this framework, the mobile robot scans its surrounding using sensors, such as a sonar sensor, a laser range scanner, or a camera, then the robot moves to the frontier which is the boundary between explored area and unexplored area. By iteratively moving toward frontiers, the robot can achieve exploration of the entire area while building a map. This is a frontier-based exploration strategy.

Based on the frontier-based exploration strategy, the Sensor-based Random Tree (SRT) method is proposed and

Jinho Kim is currently a Postdoctoral Research Fellow at the Robotics Research Center at the United States Military Academy, West Point.

✉ Jinho Kim
umbcjhkim@umbc.edu

Charles D. Eggleton
eggleton@umbc.edu

Stephen A. Wilkerson
swilkerson@ycp.edu

S. Andrew Gadsden
gadsden@uoguelph.ca

1 University of Maryland, Baltimore County, 1000 Hilltop Cir, Baltimore, MD 21250, USA

2 York College of Pennsylvania, 441 Country Club Rd, York, PA 17403, USA

3 University of Guelph, 50 Stone Rd E, Guelph, ON N1G 2W1, Canada

developed for a mobile ground robot in [12] and [13]. The proposed method carries out explorations through SRT which represents a roadmap of the explored area, and SRT is incrementally built by the randomized generation of paths towards unexplored areas. In [14], cooperative exploration strategy for a team of mobile ground robots is proposed by building Sensor-based Random Graph (SRG) which is a data structure, and its data structure is constructed from derivations of the Local Safe Region (LSR); the Local Reachable Region (LRR), the Local Frontier (LF), and the Local Informative Region (LIR). Using the SRG, the proposed action planner leads mobile robots to explore an unknown environment by randomly generating the next target nodes. Moreover, in this strategy, an algorithm was designed to minimize the travel distance by suggesting a bridge. This bridge can be generated between the current configuration and a visited configuration when the visited configuration is located within certain range of the current configuration, and they are connected through a safe region without any obstacles between two configurations.

In this paper, we present a strategy developing the SRG exploration strategy so as to efficiently explore unknown environments using a team of quadrotors (Fig. 1). To maximize efficiency of the exploration strategy, it is clear that using multiple robots is better than using a single robot [15, 16]. Moreover, the issue to be considered for the efficiency of the exploration is not only the number of robots, but also the type of robots. Although exploration with mobile ground robots has shown satisfying results [17–22], mobile ground robots have disadvantages compared to aerial robots. First, mobile ground robots occasionally generate longer detours or consume more exploration time by holding its current position in order to avoid collisions with another robot. Second, using mobile ground robots has a limitation in some environments due to topographical factors. For example, mobile ground robots cannot reach the unexplored areas due to collapsed debris or disconnected floors in earthquake damaged structures. Hence, aerial robots need to be considered for playing a key role in an efficient exploration in unknown environments. Among all types of aerial robots, a quadrotor is suitable for exploration missions. Because the quadrotor has advantages over a fixed-wing aircraft in being able to agilely move in any direction, and it is also capable of vertical take-off and landing that requires less space. However, the shortcoming of the quadrotor that the flight time is low due to the high energy consumption of the four motors remains a challenge.

In order to increase efficiency of the exploration, we propose a compact data structure which consists of the frontier data and the hub node data only. In particular, the hub node is defined as a node which contains at least one frontier arc, and the closest hub node is selected to generate backtracking paths when a quadrotor cannot continue exploring at the current position where no more frontier arc exists. Simultaneously, our strategy builds a graph map using the frontier data to generate the shortest backtracking paths during the exploration. The nodes (also called vertices) and edges of the graph represent visited positions and feasible paths, respectively. Straight edges of the graph directly reflect to the straight flight path, which it is beneficial for controlling quadrotors with a simple maneuver. Furthermore, our strategy generates next target positions for covering the most frontier areas with given current information. This strategy is simple and allows quadrotors to cover frontier areas divergently.

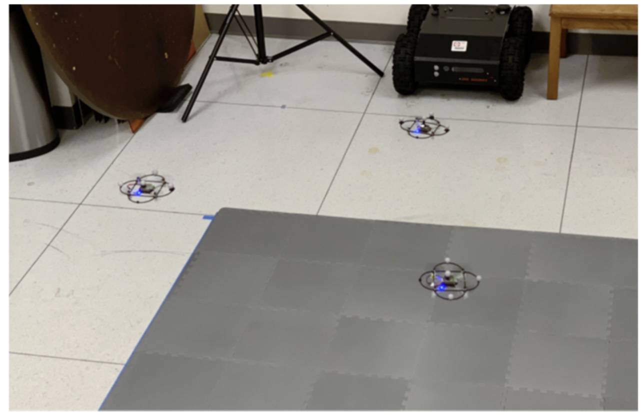

Fig. 1 Three quadrotors are hovering to explore in an unknown environment

Although many algorithms for explorations of quadrotors in the three dimensional (3D) space have already been developed [23–26], the 2D planar map is sufficient to obtain the map information of the single floor indoor environments for exploration missions, and there are advantages in both computational load and practicality. As this paper is considering explorations in the single floor indoor environments, the aforementioned advantage is effective to achieve the goal of this paper.

Hence, we present an exploration strategy on the 2D planar mainly, but we also utilize the 3D space in certain cases to optimize exploration paths of quadrotors for increasing an efficiency of explorations. The topic will be handled in Section 6.2.

We summarize contributions of this paper briefly in Section 2, and the rest of the paper is organized as follows. Section 3 presents the problem setting with assumptions. In Section 4, the data structure used in this paper is described with a frontier data and definition of a hub node, and the pseudocode of the CSSG exploration strategy is presented in Section 5. Then we propose the method for generating selective next targets, the multiple flight levels approach, and the graph-based efficient backtracking method in Section 6. We show numerical simulation results and experimental results of the proposed strategy with a team of quadrotors in Section 7 and Section 8, respectively. Finally, concluding remarks are presented in Section 9.

2 Summary of Contributions

Building on this work, we propose the Cooperative Sensor-based Selective Graph (CSSG) strategy with the following features:

1. *Selective target node*: The proposed method computes the next target node based on the frontier data, not using a random mechanism. To maximize the coverage of the frontier area at each movement, the Selective Target Node (STN) approach defines frontier directions, $\theta_{\mathcal{F}}$, with merged frontier sectors, and then selects the next target node with the most coverage of frontier area in a way that each quadrotor moves in a direction away from each other. The proposed approach also determines how many $\theta_{\mathcal{F}}$ exist at some nodes. With this frontier data, the algorithm assigns the next target node (hub node) to the quadrotor when the quadrotor reaches a dead end where no frontier exists.
2. *Multiple flight levels*: The CSSG provides an algorithm with Multiple Flight Levels (MFL), which enables multiple quadrotors to fly to one certain node at the same time while avoiding collision with each other. With this strategy, a team of quadrotors can continue to explore without any inefficient position holding, waiting for another quadrotor to pass. By doing this, which ground robots cannot implement, the team of quadrotors can achieve better exploration performance.
3. *Graph-based efficient backtracking*: A node can be connected to other nodes in the topology map, and numerous connections can be generated during an exploration. It is referred to as a *graph*. In early path planning methods like Voronoi diagram [27, 28], potential field [29, 30] and A* algorithm [31], prior map information is required to generate a path, so that those methods are not applicable to generate paths toward unexplored areas. By the graph, however, those early path planning methods can be integrated into the backtracking algorithm to generate optimized paths. Hence, the backtracking algorithm, which is proposed in [32], is reinforced, so that the algorithm chooses the best path to backtrack efficiently. In order to employ a graph-based map, this paper defines *node connectivity* and proposes a graph map which is built base on the frontier data during explorations.

3 Problem Setting

In this section, we introduce assumptions for the CSSG exploration strategy.

3.1 Assumptions

The CSSG exploration strategy is addressed under the following assumptions.

1. The workspace $\mathcal{W}$ is a subset of an n-dimensional Euclidean space $\mathbb{R}^n$, where $n = 2$ or 3. In this paper, we mainly consider the case $n = 2$, but we also consider the case $n = 3$ in certain cases.
2. Each quadrotor knows configurations of every quadrotor, q_i $(i = 1, 2, \cdots, N)$, where N is the total number of quadrotors. q_i contains positions and angles in the Cartesian coordinates system. In this paper, we call each q a *node*. Specifically, the j-th configuration of q_i is called the j-th node of i, and denoted as q_{ij}.
3. The i-th quadrotor is capable of omnidirectional detection using equipped sensory systems, and it provides information of the surrounding area within a sensor range R_s.
4. From the scanning data, the algorithm computes frontiers and stores it in the Sensor-based Selective Graph (SSG). Each quadrotor can save the SSG in its memory and broadcast it within a communication range R_c at any time. In this paper, we assume that R_c is large enough to cover all the $\mathcal{W}$.

The assumptions are applied to mitigate the complexity of the problem. First, quadrotors are utilized with a forward-looking depth-sensing camera in the form of either a stereo camera or RGB-D camera to detect obstacles, and the camera sensor extracts the 2D planar data of $\mathcal{W} \subset \mathbb{R}^2$ in this paper. However, when the algorithm is activated with the MFL (see Section 6.2), the camera sensor scans the three dimensional (3D) space of $\mathcal{W} \subset \mathbb{R}^3$ to avoid collision with obstacles in the 3D space. Second, even though a quadrotor can rotate during its hovering at q for scanning all directions with a camera, Assumption 3 allows the quadrotor to ignore the rotation procedure. Furthermore, if we use a range sensor in addition to the camera, such as a laser range scanner for 2D scanning, the scanning method can be simplified by omitting the rotation procedure, and relaxing Assumption 3. Third, the CSSG exploration strategy can be fully dedicated to the goal of efficient exploration in unknown environments, not considering the communication range through Assumption 4. A more specific explanation

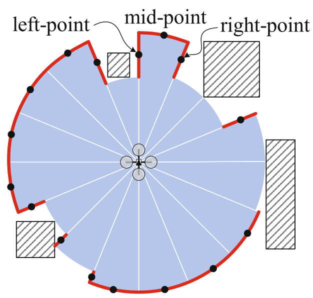

Fig. 2 An example of the SRT-Star. The blue colored star shape depicts a local safe region, $\mathcal{S}$, with frontier arcs which is thick outer line (red), and black dots on frontier arcs represent left, mid and right-points of frontier sectors

and formal definition will be presented in the Sections 5 in regard to the CSSG exploration.

4 Sensor-based Selective Graph

In order to explain the SSG and its data structure, we first introduce the SRT-Star method on which the SSG is based. Then two data structures of the frontier data and hub node data are proposed for the CSSG exploration in this section.

4.1 Background: SRT-Star Method

The SRT-Star is a sensor-based exploration method which enables a robot to explore in unknown environments moving toward a frontier area. The method incrementally generates a data structure, which is called SRT [12]. The SRT-Star method divides the LSR, $\mathcal{S}$, into several cone-shaped sectors, generating an $\mathcal{S}$ that looks like a star as depicted in Fig. 2, as the way to handle the scanning data. Depending on existence or distance of obstacles nearby, each sector can have its own radius, $r \in [0, R_s]$, through which the frontier can be determined.

Each sector can be defined by three points, which indicate that the sector has a frontier. First, the mid-point can be defined as a point which is placed along the middle axis of the sector at R_s. Second, right-point and left-point, can be established when there is a wide gap between adjacent sectors. In Fig. 2, the thick outer line (red) of $\mathcal{S}$ represents frontier arcs while the remaining portions represent either free arcs or obstacle arcs, and left, mid, right-points are defined on the frontier sectors.

Next, Algorithm 1 presents FB-SRT-Star algorithm based on [ug02] for exploration using the frontier data. This algorithm requires an initial configuration of robot q_{init}, maximum iteration number n_{max}, step movement constant α, and minimum step movement d_{min}. In the first step, sensory equipment scans the robot's perimeter. Then using SCAN_LSR and COMPUTE_FRONTIER functions,

Algorithm 1 SRT-Star Method [13].

Require: $q_{init}, n_{max}, \alpha, d_{min}$

1: $q_{curr} = q_{init}$
2: **for** $n = 1$ to n_{max} **do**
3: $\quad \mathcal{S}(q_{curr}) \leftarrow \mathtt{SCAN\_LSR}(q_{curr})$
4: $\quad \mathcal{F}(q_{curr}) \leftarrow \mathtt{COMPUTE\_FRONTIER}(\mathcal{S}(q_{curr}))$
5: $\quad \mathcal{T} \leftarrow \mathtt{ADD}(q_{curr}, \mathcal{S}(q_{curr}), \mathcal{F}(q_{curr}))$
6: $\quad k = 0$
7: $\quad$ **repeat**
8: $\qquad \theta_{rand} \leftarrow \mathtt{RANDOM\_DIR}$
9: $\qquad r \leftarrow \mathtt{RANGE}(\mathcal{S}(q_{curr}), \theta_{rand})$
10: $\qquad q_{cand} \leftarrow \mathtt{TARGET\_CAND}(q_{curr}, \theta_{rand}, \alpha \cdot r)$
11: $\qquad k = k + 1$
12: $\quad$ **until** $\mathtt{VALID}(q_{cand}, d_{min}, \mathcal{T})$ or $k = K_{max}$
13: $\quad$ **if** $\mathtt{VALID}(q_{cand}, d_{min}, \mathcal{T})$ **then**
14: $\qquad \mathtt{MOVE}(q_{cand})$
15: $\qquad q_{curr} \leftarrow q_{cand}$
16: $\quad$ **else**
17: $\qquad \mathtt{MOVE}(q_{cand,parent})$
18: $\qquad q_{curr} \leftarrow q_{cand,parent}$
19: $\quad$ **end if**
20: $\quad$ **return** $\mathcal{T}$
21: **end for**

the algorithm collects $\mathcal{S}$, and frontiers, $\mathcal{F}$, at the current position, q_{curr} (Line 3-4). These data are stored in the SRT structure, $\mathcal{T}$ (Line 5). The next step involves using data from $\mathcal{F}$ and generating a set of candidate directions, θ_{cand}, to find the next target node, q_{cand}. The algorithm selects a random direction, θ_{rand}, using RANDOM_DIR function (Line 8) and computes the radius, r, of $\mathcal{S}(q_{curr})$ along θ_{rand} (Line 9). Finally, this algorithm generates q_{cand} by taking α multiplied by r along θ_{rand} (Line 10). After the next target node is validated by VALID function, the robot moves to q_{cand}, and q_{curr} turns into q_{cand} (Line 12-15). However, if q_{cand} is not validated or generated, the robot moves to a point $q_{cand,parent}$, which has a frontier arc (Line 17-19). The algorithm repeats all these steps until n reaches n_{max}. The detailed description is presented in [13].

4.2 Structure of Sensor-based Selective Graph (SSG)

The SSG is an essential data structure for efficient cooperative exploration in unknown environments. Basically, the SSG provides quadrotors with two data structures which are directly derived from the LSR; the *Frontier Data* and the *Hub Node Data*. During the exploration with the SSG, each quadrotor builds its own SSG$_i$ ($i = 1, 2, \cdots, N$) and extends its exploration to frontier directions.

The SSG contributes to generating the next target node maximizing the coverage of frontier area at each movement. The STN is the method which is based on the SSG and selects the next target node with a defined objective function. In addition, the additional contribution of the SSG is the generation of a graph which is used to provide quadrotors with optimized paths. Especially, in the case when the quadrotor reaches a dead end, the algorithm generates an optimized path for the quadrotor to move to the nearest hub node that has frontiers. The STN and graph-based backtracking algorithm will be described in Section 6.

Here, the frontier data and the hub node data are presented to understand the SSG.

4.2.1 Frontier Data

The frontier data structure, $\mathcal{F}$, is a data structure which shows how many frontier arcs and free arcs exist at a node q. Specifically, the frontier data structure of the i-th quadrotor is denoted by $\mathcal{F}_i$ which contains the frontier data of every q_i. Mathematically, $\mathcal{F}$ of q_{ij} can be written for the SSG as follows:

$$\mathcal{F}_{q_{ij}}(k) = \begin{cases} (-1, r_k) & \text{if } obstacle\ arc, \\ (0, r_k) & \text{if } free\ arc, \\ (1, r_k) & \text{if } frontier\ arc, \end{cases} \tag{1}$$

where i is the quadrotor number, j is the node number, k is the sector number, and r_k is the radius of the k-th sector.

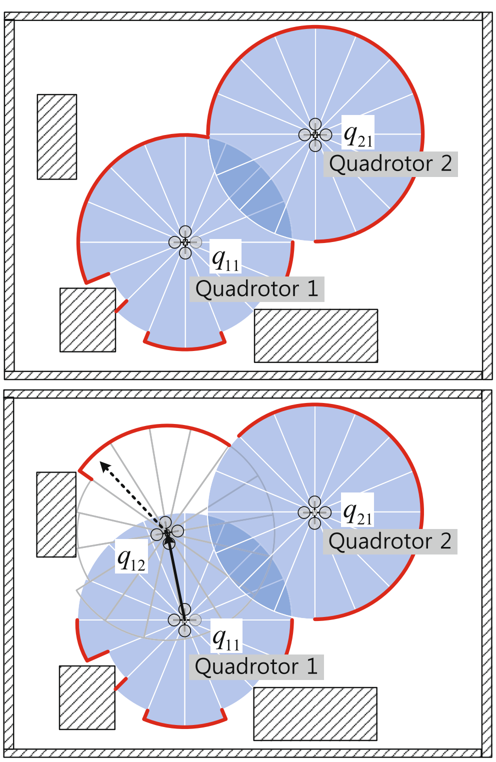

Fig. 3 An example of SRT-Star exploration

Figure 3 illustrates an example of $\mathcal{F}$ with two quadrotors. In this figure, `Quadrotor 1` and `2` have $\mathcal{S}$ which is lightly colored (blue area), and the frontier arc is depicted as thick outer lines (red). $\mathcal{F}$ of `Quadrotor 1` at q_{12}, $\mathcal{F}_{q_{12}}$, can be expressed by Eq. 3. It can also be seen that $\mathcal{F}_{q_{11}}$ is changed from Eq. 4 to 5 as the `Quadrotor 1` flies to q_{12}. Note that $\mathcal{F}_{q_{21}}$ is also updated as the `Quadrotor 1` flies to q_{12}.

$$\mathcal{F}_1 = \{\mathcal{F}_{q_{11}}, \mathcal{F}_{q_{12}}\}, \tag{2}$$

$$\mathcal{F}_{q_{12}} = \left[(1, R_s), (1, R_s), (0, R_s), \cdots, (-1, r_{13}), \cdots\right], \tag{3}$$

$$\mathcal{F}_{q_{11-1}} = \left[(0, R_s), \cdots, (1, R_s), (1, R_s), (1, R_s)\right], \tag{4}$$

$$\mathcal{F}_{q_{11-2}} = \left[(0, R_s), \cdots, (0, R_s), (0, R_s), (0, R_s)\right]. \tag{5}$$

4.2.2 Hub Node Data

The hub node is the node which has at least one frontier, and it can be defined by satisfying the following conditions:

$$q_{hub} = \{q \mid \max_k \left(\mathcal{F}_q(k)(1)\right) = 1,\ q \in \mathcal{W}\}. \tag{6}$$

The hub node data structure, $\mathcal{H}$, is built to store the hub nodes information during explorations, and the structure of $\mathcal{H}$ consists of three parts: the node number of its $\mathcal{F}$, the number of frontier sectors, and the number of merged frontier sectors (See Section 6.1.1). Note that the number of $\theta_{\mathcal{F}}$ is always less than or equal to the number of frontier sectors.

For example, in Fig. 3, the node q_{11} can be saved in $\mathcal{H}$ of `Quadrotor 1`, $\mathcal{H}_1$, as the hub node when the `Quadrotor 1` moves to q_{12}. Hence, $\mathcal{H}_1$ can be written as $\{(1, 5, 3)\}$. This indicates that the first node of $\mathcal{F}_1$ is a hub node having five frontier sectors and three merged frontier sectors. Note that the third value is computed by the merged frontier identification technique introduced in the next section. As the quadrotor continues exploring, other hub nodes can be added, and $\mathcal{H}_1$ might be updated to $\{(1, 5, 3), (3, 3, 2), \cdots, (5, 2, 1)\}$ as an example.

When the i-th quadrotor can no longer continue to explore at the current position where there is no frontier, the algorithm assigns the closest hub node in $\mathcal{H}_i$ as the next target node. If $\mathcal{H}_i = \emptyset$, then the closest hub node in $\mathcal{H}_j$ $(j \neq i)$ is allotted as the next target node.

5 Cooperative Sensor-based Selective Graph (CSSG) Exploration

In this section, the CSSG exploration is described with the pseudocode to understand its process as shown in Algorithm 2. At the beginning, quadrotors build the SSG by scanning their surroundings and perceiving $\mathcal{S}$ (Line 1). Then the algorithm selects the next target node by using the STN if $\mathcal{F}$ is not empty at q_{curr} (Line 2-3). However, if there is

Algorithm 2 CSSG exploration.

1: Perceive $\mathcal{S}$ and build `SSG`
2: **if** $\mathcal{F}_i$ at q_{curr} is non-empty **then**
3: Compute q_{cand} using `STN`
4: **else**
5: Find the closest q_{hub}
6: **if** $\mathcal{H}_i$ is non-empty **then**
7: Assign $q_{hub} \in \mathcal{H}_i$ as a target node
8: **else**
9: Find q_{hub} in $\mathcal{H}_j$
10: **if** $\mathcal{H}_j$ is non-empty **then**
11: Assign $q_{hub} \in \mathcal{H}_j$ as a target node
12: **else**
13: Homing mode: Assign q_{init} as a target node
14: **end if**
15: **end if**
16: Compute q_{cand} using `graph` to backtrack to the target node
17: **end if**
18: **if** Assigned q_{cand} the same with q_{cand} of another robot or crossed paths **then**
19: Activate `MFL`
20: **end if**
21: `MOVE`(q_{cand})
22: $q_{curr} \leftarrow q_{cand}$
23: Update `SSG`

no more frontier, the algorithm tries to find the closet q_{hub} from its $\mathcal{H}$ and the others' $\mathcal{H}$ (Line 5-11) and computes the optimized path by using the graph (Line 16). If there is no more remaining frontier in all $\mathcal{H}$, all quadrotors return to the initial positions using the graph as well (Line 12-16). Before moving to the next target node, the algorithm decides whether to activate the MFL to assign the next target node to upper or lower flight altitude (Line 18-20). Finally, quadrotors move to the next nodes and the algorithm updates the SSG (Line 21-23), and iterate this process until all $\mathcal{F}$ turn to be empty. This is a summarization of the CSSG exploration.

6 Action Planner

In this section, the basic concepts and policy to operate the CSSG exploration strategy are introduced. The Selective Target Node (STN) method is proposed with the merged frontier identification (SSG-Merged) technique, and the Multiple Flight Levels (MFL) approach and graph-based backtracking algorithm are proposed to increase efficiency of the CSSG exploration strategy.

6.1 Selective Target Node (STN) Method

The work of [14] proposed the randomized selection method for the next moving points. Since, there is a possibility that each robot can move to the same direction with the randomized selection method, it might be not efficient when there are multiple robots used to explore unknown environments. Hence, in order to maximize coverage of the frontier area at each movement, we propose a new method of generating candidates of $\theta_{\mathcal{F}}$ and selecting the next target node which covers a larger frontier area. In this section, we introduce a novel frontier identification technique, the SSG-Merged, by merging frontier sectors. Then the STN method is proposed to find a optimal next target node whose coverage of the frontier area is maximized.

6.1.1 Merged Frontier Identification Technique (SSG-Merged)

Figure 4 shows an example of the star shape of $\mathcal{S}$, in which thick outer lines (red) represent frontier arcs, and numbers on sectors indicate the numbers of each sector. The SSG-Merged considers the adjacent frontier sectors as one frontier sector. In Fig. 4, the frontier sectors 4-5 are merged as one frontier sector with size of two, and the frontier sectors 7-10 are also considered as one frontier sector with size four, and so on. Hence, a quadrotor can have four $\theta_{\mathcal{F}}$ with different conditions, so that the quadrotor can select the best $\theta_{\mathcal{F}}$ which covers the most frontier area at q_{curr}.

The radius of the merged frontier sector, r_m, is defined as $\mathcal{F}_q(k)(2)$ if the mid-point of the merged frontier sector is located in k-th sector, which is one of the separated sectors in the merged frontier sector. However, if the midpoint of the merged frontier sectors is located between two separated sectors, k and $k+1$, then r_m has a value from $min[\mathcal{F}_q(k)(2),\ \mathcal{F}_q(k+1)(2)]$.

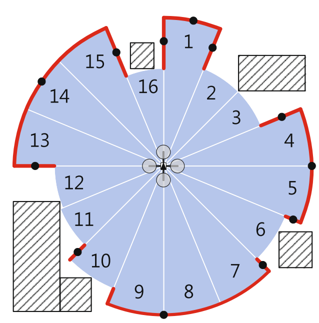

Fig. 4 An example of the SSG-Merged. The number on each sector indicates the sector number (k). The frontier sectors 4-5, 7-10 and 13-15 can be merged into each one frontier sector with the SSG-Merged. Hence, the quadrotor has four $\theta_{\mathcal{F}}$ to move next

6.1.2 Maximization of Frontier Area Coverage

To compute a candidate of the next target node, the STN method defines $\theta_{\mathcal{F}}$ as the angle from the positive x-axis of the inertial frame to the midpoint of the corresponding merged frontier sectors.

Let $q_{cand}(\theta_{\mathcal{F}(a)})$ be the candidate of next target node of a-th merged frontier sector at q_i. Then the objective function based on distances, J, can be formulated as Eq. 7.

$$J = \sum_{i=1}^{N} c_i \|\mathbf{x}_i - \mathbf{x}_{cand}\|, \tag{7}$$

where c_i denotes weighting factors, and $\|\mathbf{X}\|$ denotes the norm of the vector $\mathbf{X}$, and $\mathbf{x}_i$ and $\mathbf{x}_{cand}$ are the current position of i-th quadrotor, $[x_i, y_i, z_i]$, and the position of $q_{cand}(\theta_{\mathcal{F}(a)})$, $[x_{cand}(\theta_{\mathcal{F}(a)}), y_{cand}(\theta_{\mathcal{F}(a)}), z_{cand}(\theta_{\mathcal{F}(a)})]$, respectively. Note that candidates for the next target node can be calculated by taking α multiplied by r_m along $\theta_{\mathcal{F}(a)}$.

Once J of each candidate is calculated, the STN method chooses q_{cand} which has the largest value of J.

6.2 Multiple Flight Levels (MFL) Approach

The MFL approach is an approach to leverage an efficiency of the CSSG exploration by integrating the 3D space into the 2D workspace. Unlike an exploration of ground mobile robots, the advantage of quadrotors is the capability to utilize the 3D space. During the exploration with mobile ground robots, one robot has to hold its position before moving to the next target node when sharing the next target node with another robot. Quadrotors, however, can fly to the same node in the 2D plane at the same time activating the MFL.

Once multiple quadrotors are assigned the same next target node or generated paths are crossed, one quadrotor moves to upper level of the next target node and the other quadrotor flies to the lower level. If there is another quadrotor with the same next target position, additional levels are added in the MFL. The height of each layer is a user parameter, the user can set the parameter value taking into account the workspace characteristics and quadrotor size.

As a result, quadrotors can fly to any next target node without having to wait for other quadrotors to pass. Furthermore, the STN efficiently assigns all the different next target nodes after the quadrotors moved to the same node with the MFL. See Fig. 5 for the illustration of the MFL.

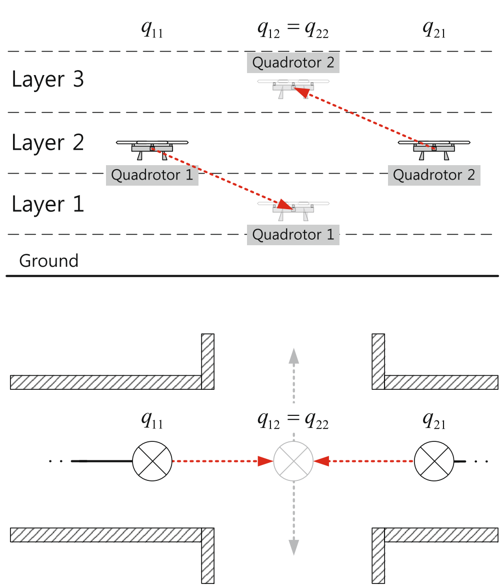

Fig. 5 (Top) An example of the multiple flight layers. The `Quadrotor 1` is assigned `Layer 1` to move to the next node, q_{12}, while the `Quadrotor 2` is assigned `Layer 3`. (Bottom) Nodes on the 2D plane map

6.3 Graph-Based Backtracking Algorithm

In this section, the graph is proposed to generate an optimized backtracking path with an efficient backtracking algorithm. The graph is a topological map in which nodes are connected each other. Based on the SSG, the graph can be built at during explorations, and it is updated at each movement.

Let $\mathcal{C}_{q_a}^{q_b}$ denote a *node connectivity* of q_a and q_b, where $q_a, q_b \in \mathcal{W}$ and $d(q_a, q_b) < 2R_s$. Here, $d(q_a, q_b)$ is the Euclidean distance between nodes q_a and q_b. Then $\mathcal{C}_{q_a}^{q_b}$ can be defined *connected* if the frontier data of q_a and q_b satisfy Eq. 8.

$$\begin{aligned} &\mathcal{F}_{q_a}(k_{q_b})(1) = \mathcal{F}_{q_b}(k_{q_a})(1) = 0, \\ &\mathcal{F}_{q_a}(k_{q_b})(2) > d(q_a, q_b), \\ &\mathcal{F}_{q_b}(k_{q_a})(2) > d(q_a, q_b), \end{aligned} \tag{8}$$

where k_{q_a} and k_{q_b} denote the sector number which faces q_a and q_b, respectively. Naturally, the frontier data of q_{curr} and the previous node, q_{prev}, satisfy (8), so that $\mathcal{C}_{q_{curr}}^{q_{prev}}$ is always

Algorithm 3 `Graph` map building.

Load `SSG`
Update $\mathcal{C}_{q_{curr}}^{q_{prev}}$ as *connected*
if $\mathcal{F}_{q_a}(k_{q_b})(1) = \mathcal{F}_{q_b}(k_{q_a})(1) = 0$ **then**
  Compute $d(q_a, q_b)$
  if $d(q_a, q_b) < \mathcal{F}_{q_a}(k_{q_b})(2)$ and $d(q_a, q_b) < \mathcal{F}_{q_b}(k_{q_a})(2)$ **then**
    Update $\mathcal{C}_{q_a}^{q_b}$ as *connected*
  end if
end if
Build `graph` Map

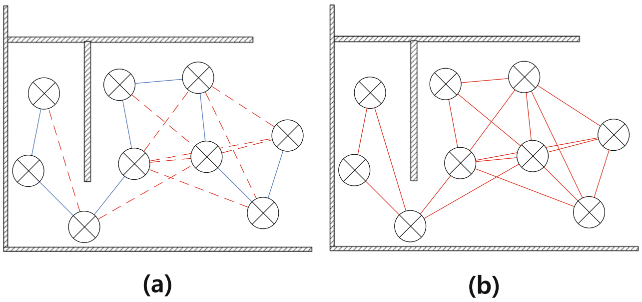

Fig. 6 (a) An example of an exploration in which the solid lines (blue) depict the exploration path, and the dashed lines (red) represent the *connected* lines. (b) The graph of this exploration by using the SSG

called *connected*. Algorithm 3 describes how to build a graph map.

When the quadrotor cannot find any frontier direction at q_{curr}, the quadrotor has to move back to the q_{hub} where frontiers exist or to the initial position in case that no more frontier remains in every $\mathcal{H}$. In order to generate the backtracking path, first, the efficient backtracking algorithm finds out the closest q_{hub} in its own $\mathcal{H}$. If no q_{hub} is in its $\mathcal{H}$, then the algorithm chooses the closest q_{hub} in the others' $\mathcal{H}$ as the target node. Once the target node is selected, the algorithm generates the shortest path on the graph map.

Each *connection* in the graph map has the distance data between two nodes, so that the graph provides quadrotors with the shortest path on the graph to reach to the assigned q_{hub}. The path generation based on the real time graph is processed by Dijkstra's algorithm [33] in this paper, and its efficiency is presented in [34, 35].

Figure 6 shows an example of exploration and its graph. In Fig. 6(a), the solid lines (blue) represent the exploration path, the dashed lines (red) are the *connected* lines except connections between nodes and previous nodes. As a result, the graph of this exploration can be illustrated as shown in Fig. 6(b).

7 Simulation Results

This section introduces a quadrotor model used in the simulations, and the result of MATALB simulations in different two environments is presented to analyze the performance of the proposed strategy.

7.1 Quadrotor Dynamics and PD Controller

Since this paper mainly focuses on exploration algorithms, we apply a simple proportional-derivative (PD) controller

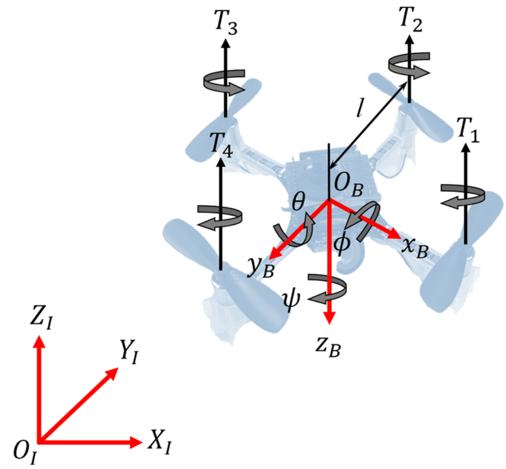

Fig. 7 A configuration of a quadrotor. $[X_I, Y_I, Z_i]$ denotes the inertial coordinate frame, and $[x_B, y_B, z_B]$ is the body coordinate frame. $[x, y, z]$ denotes the position of the quadrotor in the inertial frame, and $[\phi, \theta, \psi]$ represent roll, pitch, and yaw angles, respectively defined in the body frame. T_i $(i = 1, 2, 3, 4)$ is the thrust force of the i-th rotor, l denotes the length between each rotor and the geometric center of the quadrotor. Also, M is the mass of the quadrotor

for movement to the target positions. Please refer to Fig. 7 and [36] for further understanding the quadrotor dynamics and PD controller used in this paper. In addition, we tuned proportional and derivative gains of the PD controller using gradient methods which are presented in [37]; $k_{p,*}$ and $k_{d,*}$ denotes the propositional and derivative gains of $*$, respectively. In this work, we do not solve time-related optimization problems, but a simple go-to mission is only used for the quadrotor control.

7.2 Simulations

The proposed CSSG exploration strategy for a team of quadrotors is numerically simulated in this section. To evaluate the performance of the CSSG strategy, three different strategies were performed to verify the efficiency of the proposed strategy: 1) CSSG strategy, 2) CSSG strategy without STN method (the random mechanism) and 3) SRG strategy with the graph-based backtracking algorithm. By comparing the first and second approaches, performance of the STN method can be evaluated. Also, we can evaluate the performance of MFL approach from the comparison of last two approaches.

Simulations were carried out to explore two different environments with various numbers of quadrotors. The environments used in this simulation are an apartment-like environment with multiple rooms and a corridor-like

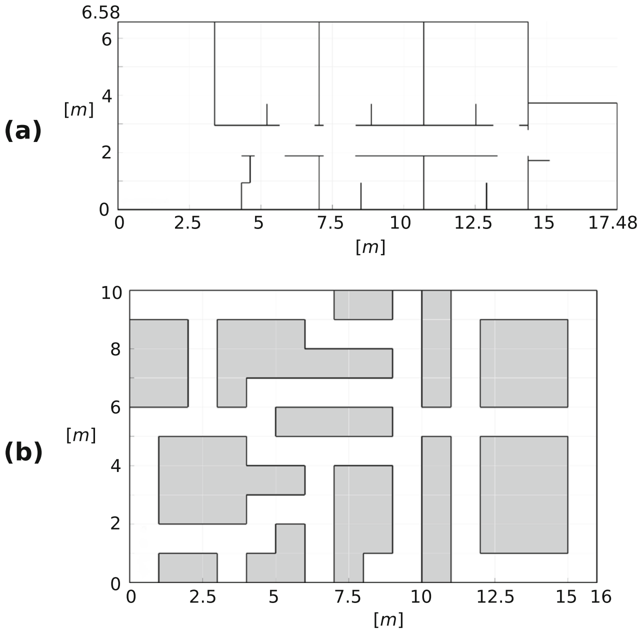

Fig. 8 Two environments used in the simulation. ((a) Apartment-like environment, (b) Corridor-like environment)

environment depicted in Fig. 8. The quadrotor model parameters for this simulation are:

$$
\begin{aligned}
M &= 0.3\,kg,\ l = 0.2\,m,\\
k_{p,\phi} &= 50.7,\ k_{d,\phi} = 20.7,\\
k_{p,\theta} &= 42.1,\ k_{d,\theta} = 16.4,\\
k_{p,z} &= 15.0,\ k_{d,z} = 7.2,\\
k_{p,\psi} &= 18.8,\ k_{d,\psi} = 9.9.
\end{aligned}
$$

Also, exploration parameter settings for this simulation are:

$$
\begin{aligned}
I_{max} &= 18,\\
R_s &= 1.0\,m,\\
\alpha &= 0.8,\\
MFL &= [0.50\,m, 1.00\,m,\ 1.50\,m]
\end{aligned}
$$

The mission of the team of quadrotors is to explore the simulation environments from initial positions and return to the initial positions after exploring the entire area. For each environment and each number of robots, 20 different simulations were performed for each strategy. We assume that each quadrotor is equipped with a 360 degree laser-range scanner, with R_s of 1.00 m, and a RGB-D camera for the MFL approach. Note that R_s is set as 1.00 m due to flight safety.

An example of the CSSG exploration with three quadrotors in the apartment-like environment and its graph map are depicted in Fig. 9. To compare three strategies, we evaluated the average number of iterations and distances traveled per quadrotor, and the results are shown in Figs. 10–11. Since we only consider the minimization of the total distance quadrotors fly as performance indexes, low performance indexes can guarantee low energy consumption of the quadrotors as well.

As can be seen from Fig. 10, the result of explorations in the apartment-like environment using the CSSG exploration strategy shows the outperformed result with respect to the performance indexes. The CSSG exploration strategy efficiently explored frontiers reducing both the number of iterations and distance traveled per quadrotor by 5.41-11.71% and 4.16-13.65% compared to utilizing the

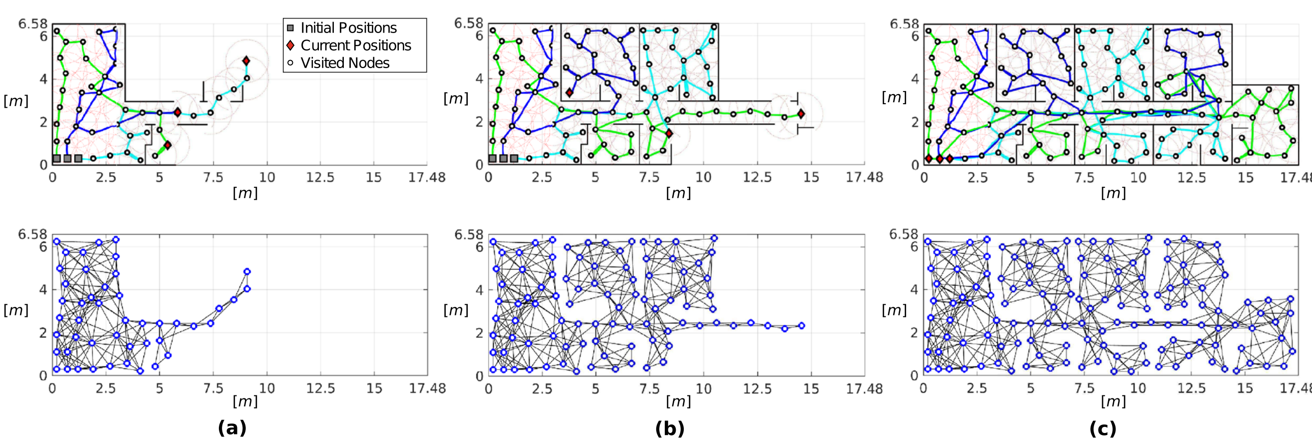

Fig. 9 Pathes of three quadrotors on 2D plane (above) and its graph maps (below) with the CSSG exploration strategy. In the plots of pathes, the black thin lines are walls, the red dotted circles around nodes represent the sensor range areas. In the plots of graph maps, the circles with blue lines are nodes and black lines represent the *connected* condition between two nodes. (Iteration: (a) 20, (b) 40, (c) 79)

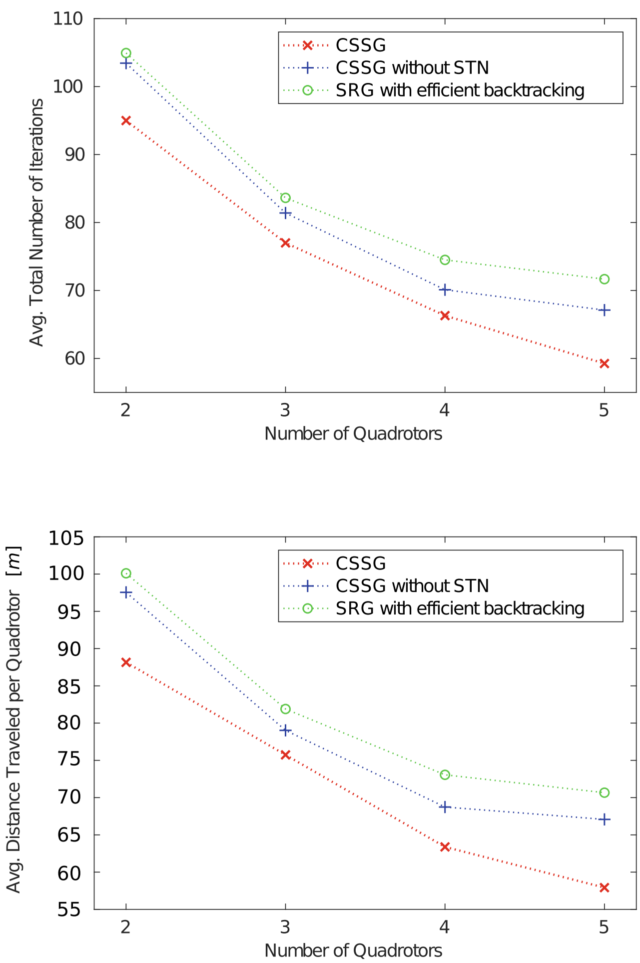

Fig. 10 Performances of three strategies for the aprtment-like environment. The plot above shows the result of the average total number of iterations, and the plot below shows the average distance traveled per quadrotor

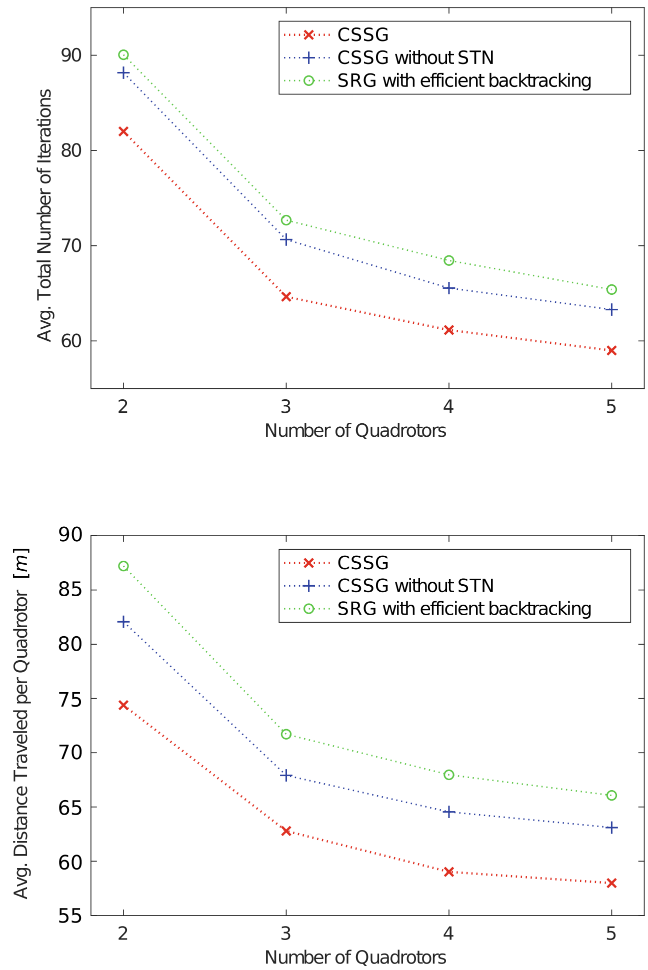

Fig. 11 Performances of three strategies for the corridor-like environment. The plot above shows the result of the average total number of iterations, and the plot below shows the average distance traveled per quadrotor

same strategy without STN method, respectively. Also, performance of the MFL approach is determined by comparing results of the CSSG strategy without STN method and the SRG strategy in Fig. 10. The number of iterations and total distance traveled per quadrotor are reduced by 1.42-6.36% and 2.55-5.92%, respectively, when the MFL approach is applied. The result shows that the efficiency of the MFL approach tends to increase as the number of quadrotors increases. This result is natural because the probability of encountering each other increases as the number of quadrotors increases.

The result of exploration in the corridor-like environment is shown in Fig. 11, where there is a similar improvement with the explorations as in the apartment-like environment. In this case, the number of iterations and distance traveled per quadrotor with the CSSG strategy are reduced by 8.94-11.05% and 12.23-14.71% compared to applying the SRG strategy, respectively. Since there were many diverging roads in this environment and each quadrotor was assigned to different directions in the crossroads via the STN method and MFL approach, the averaged distance traveled per quadrotor with the CSSG strategy is more reduced, compared to the results of the apartment-like environment.

In the simulation, since the random mechanism strategy generated zigzag movements in many directions which were randomly selected, we determined that the CSSG exploration strategy generated shorter exploration distance than the random mechanism strategy. In addition, the STN method with the SSG-Merged technique helped quadrotors decrease the number of total iterations by providing positions which cover the most frontier areas. Furthermore, each quadrotor was able to keep moving to the frontier or backtracking with the MFL approach whereas mobile robots were required to hold their positions waiting for the other robots to pass without the MFL approach. As a result, the CSSG exploration strategy generated optimized next target nodes and continued to explore without any inefficient position holding, so that the proposed strategy for the team of quadrotors could explore more efficiently than other strategies.

8 Experimental Results

The experimental setting is described and the proposed CSSG exploration strategy for the team of quadrotors are experimentally validated in this section.

8.1 Experimental Setting

In order to carry out the experiment, we decreased the system complexity to focus on the CSSG strategy and its algorithms by employing a position capture system. The position capture system feedbacks current positions of quadrotors in real time, and the algorithm computes the frontier data considering obstacle data which are virtually located within in R_s from the current positions. By using this localization and sensing methods, we were able not only to decrease the system complexity, but also to avoid installation of a sensory system and building and experimental physical environment, so that the experiment focused on validating the CSSG exploration strategy.

The experimental setting is illustrated in Fig. 12. In this experimental setting, four of OptiTrack Prime 13W cameras [38] were used as the position capture system. The OptiTrack Prime 13W camera captures 1.3 megapixels images at 120 frames per second which are sent to the PC via GigE/PoE. Also, we used two small-sized quadrotors, the Crazyflie 2.0, developed by Bitcraze [39], and Crazyswarm ROS package [40] for control of the Crazyflie 2.0. Our Crazyflie 2.0 has 0.1 m of l and 38 grams

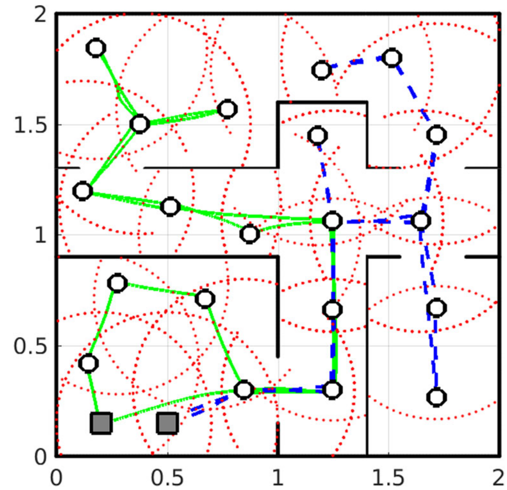

Fig. 13 The 2D pathes of quadrotors in the experiment. The black lines represent walls, and the initial positions are depicted as gray squares. The solid and dashed lines show 2D pathes of the quadrotor 1 (green) and 2 (blue), respectively. White circles are created nodes during the explorations and red dotted circles around nodes delineate scanning range areas

of the total mass including a battery, four reflective markers and its frame. The Crazyradio PA [41], a 2.4GHz USB radio, is used to communicate with Crazyflie 2.0.

The size of the workspace is $2m \times 2m$ and consists of three rooms and hallways as shown in Fig. 13. We set the remaining experimental parameters as follows:

$$
\begin{aligned}
I_{max} &= 18, \\
R_s &= 0.5\,m, \\
\alpha &= 0.8, \\
MFL &= [0.5\,m, 1.0\,m,\ 1.5\,m], \\
q_{11} &= [0.20\,m,\ 0.15\,m,\ 1.00\,m,\ 0\,deg], \\
q_{21} &= [0.50\,m,\ 0.15\,m,\ 1.00\,m,\ 0\,deg].
\end{aligned}
$$

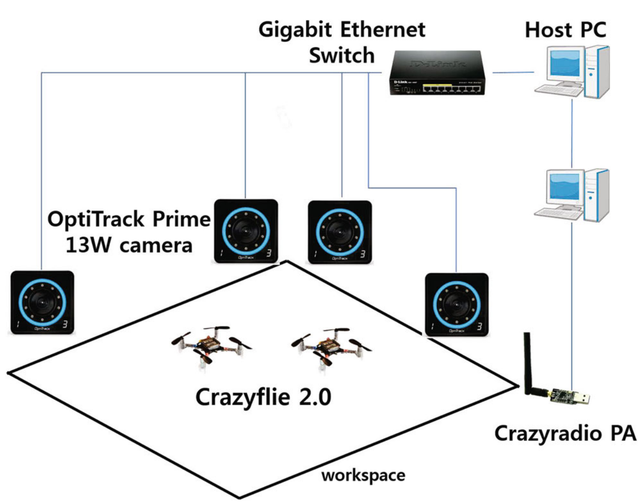

Fig. 12 Experimental setting

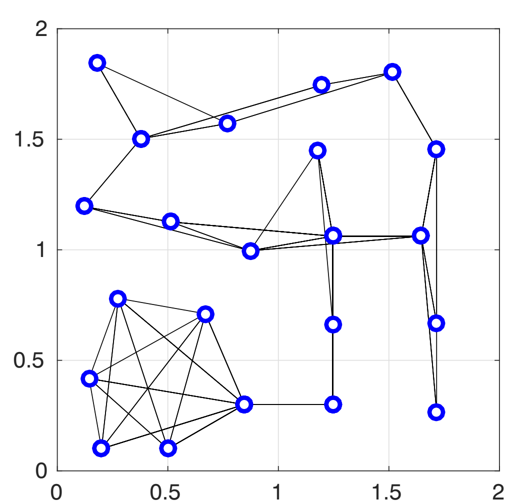

Fig. 14 The graph map built after completion of exploration

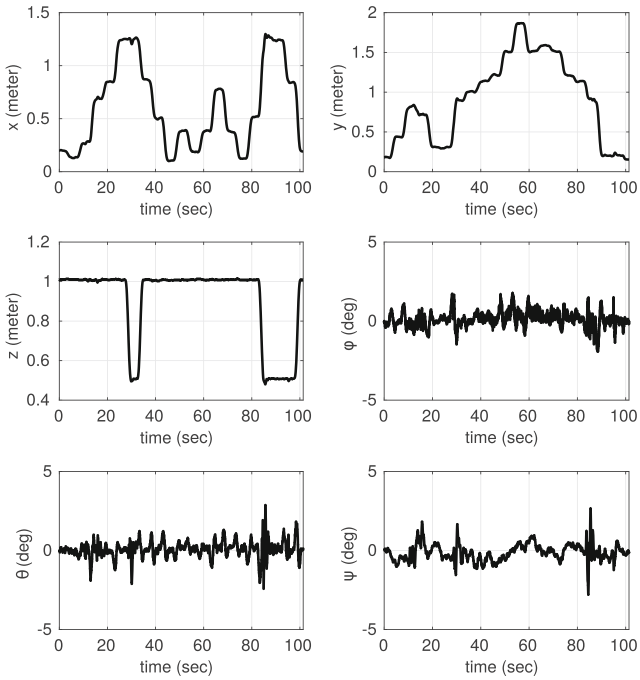

Fig. 15 History of state variable of the quadrotor 1

8.2 Experimental Results

The proposed CSSG exploration strategy is validated by the experiment. The 2D pathes of quadrotors and the graph are shown in Figs. 13 and 14, respectively. In addition, history of state variable of each quadrotor are shown in Figs. 15 and 16. From Fig. 17, we can see a team of quadrotors has explored by activating the MFL at about 27-35 seconds and 82-100 seconds. The numerical results of the experiment are described in Table 1. The first two rows of Table 1 quantify the number of nodes visited and total distance traveled of each quadrotor, and the last two rows show the total exploration time and the flight time using MFL. From this experiment, we confirm that the proposed CSSG exploration strategy is successfully performed for the team of quadrotors using the STN method, MFL approach, and graph.

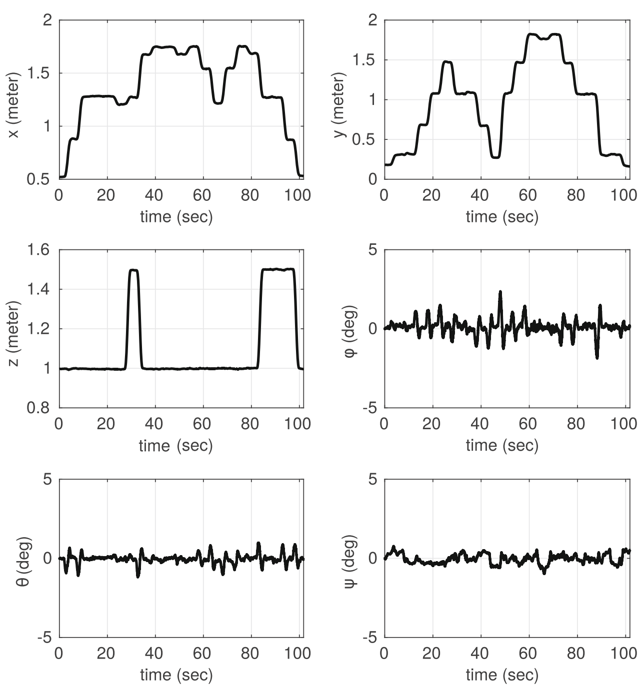

Fig. 16 History of state variable of the quadrotor 2

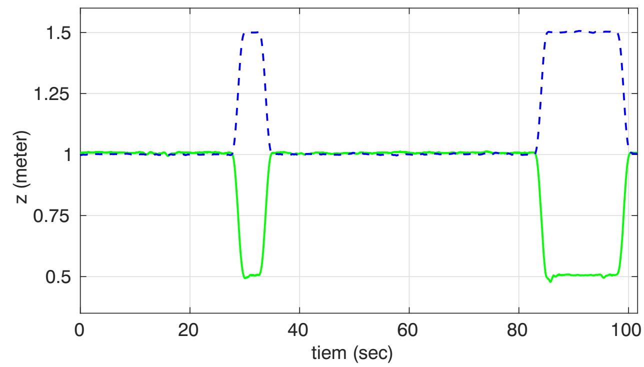

Fig. 17 History of z of two quadrotors. The solid and dashed lines indicate z of the quadrotor 1 and 2, respectively

8.3 Limitation of the Experiment

A dilemma exists to go beyond the validation of the proposed algorithm and apply the proposed strategy to the real time testing and real world environment. The proposed strategy requires scanning a 3D space (2D overall, 3D for Multiple Flight Levels (MFL)). To do this, a sensor and onboard computer heavier than the weight of the Micro UAV quadrotor used in the proposed experiment must be mounted. Hence, the scale of the quadrotor must be increased to mount the sensor, which requires a high altitude to avoid downwash from the upper quadrotor for the MFL strategy. This is the biggest limitation in the indoor environment. Therefore, in order to be more applicable in a real environment, a small sensor and a small computing chip applicable to the Micro UAV quadrotor used in this experiment are required. As technology advances, these limitations are expected to be resolved in a short time.

Table 1 Numerical results of the experiment

	Quadrotor 1	*Quadrotor* 2
The No. of nodes visited	21	21
Distance traveled (m)	10.454	9.936
Exploration time (sec.)	101.66	
Flight time using MFL (sec.)	25.10	

9 Conclusion

In this paper, the Cooperative Sensor-based Selective Graph (CSSG) exploration strategy for a team of quadrotor Unmanned Aerial Vehicles (UAVs) is proposed. The strategy combines the Select Target Node (STN) method with a novel frontier identification technique, which helps to select the next target node for covering the most frontier area using the objective function. Also, the Multiple Flight Levels (MFL) is proposed to utilize the three dimensional (3D) space. The MFL allows quadrotors to move to the next target node in the two dimensional plane without holding thier positions when sharing the next target node with another quadrotor. Furthermore, this paper applies a graph to build a topological map base on the frontier data during explorations, which enables to generate an optimized backtracking path. The numerical simulations and experimental results show satisfactory performance with the proposed exploration strategy for the team of quadrotors in unknown environments.

In theory, the proposed strategy can be applied to many different setups beyond the experimental setup presented and discussed in the paper. The robustness of the strategy itself was shown to be satisfactory as the team of quadrotors is able to successfully explore an unknown environment. Regarding the robustness, the proposed method was found to be stable to the unknown environments through simulation and experimentation. It did not fail during our experimentation. To further study the stability, we will consider a number of different environments, and run a comparative study. Also, this work will be extended to develop a distributed system for multiple quadrotor UAVs as future work.

Acknowledgment The authors would like to thank Dr. George E. Piper, U.S. Naval Academy, Annapolis, MD, for his helpful advice for this research. Also, the authors would like to thank Esther H. Kim, Johns Hopkins Medicine, Baltimore, MD, for her assistance in editing. This work was partially supported by NACA #58-8042-6-033 from the USDA National Institute of Food and Agriculture.

References

1. Doroodgar, B., Ficocelli, M., Mobedi, B., Nejat, G.: The search for survivors: Cooperative human-robot interaction in search and rescue environments using semi-autonomous robots. In: Robotics and Automation (ICRA), 2010 IEEE International Conference on, pp. 2858-2863. IEEE (2010)
2. Nagatani, K., Kiribayashi, S., Okada, Y., Tadokoro, S., Nishimura, T., Yoshida, T., Koyanagi, E., Hada, Y.: Redesign of rescue mobile robot quince. IEEE (2011)
3. Sugiyama, H., Tsujioka, T., Murata, M.: Real-time exploration of a multi-robot rescue system in disaster areas. Adv. Robot. **27**(17), 1313–1323 (2013)
4. Olson, E., Strom, J., Morton, R., Richardson, A., Ranganathan, P., Goeddel, R., Bulic, M., Crossman, J., Marinier, B.: Progress toward multi-robot reconnaissance and the magic 2010 competition. J. Field Robot. **29**(5), 762–792 (2012)
5. Korpela, C., Root, P., Kim, J., Wilkerson, S., Gadsden, S.A.: A framework for autonomous and continuous aerial intelligence, surveillance, and reconnaissance operations. In: Airborne Intelligence, Surveillance, Reconnaissance (ISR) Systems and Applications XIII, vol. 9828, International Society for Optics and Photonics, p. 982803 (2016)
6. Quann, M., Ojeda, L., Smith, W., Rizzo, D., Castanier, M., Barton, K.: An energy-efficient method for multi-robot reconnaissance in an unknown environment. In: American Control Conference (ACC), 2017, IEEE, pp. 2279–2284 (2017)
7. Amigoni, F., Caglioti, V.: An information-based exploration strategy for environment mapping with mobile robots. Robot. Auton. Syst. **58**(5), 684–699 (2010)
8. Butzke, J., Likhachev, M.: Planning for multi-robot exploration with multiple objective utility functions. In: Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on, pp. 3254-3259. IEEE (2011)
9. Senthilkumar, K., Bharadwaj, K.: Multi-robot exploration and terrain coverage in an unknown environment. Robot. Auton. Syst. **60**(1), 123–132 (2012)
10. Keidar, M., Kaminka, G.A.: Efficient frontier detection for robot exploration. Int. J. Robot. Res. **33**(2), 215–236 (2014)
11. Yamauchi, B.: A frontier-based approach for autonomous exploration. In: Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA), pp. 146–151. IEEE (1997)
12. Oriolo, G., Vendittelli, M., Freda, L., Troso, G.: The srt method: Randomized strategies for exploration. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), vol. 5, IEEE, pp. 4688–4694 (2004)
13. Freda, L., Oriolo, G.: Frontier-based probabilistic strategies for sensor-based exploration. In: Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on, pp. 3881–3887. IEEE (2005)
14. Franchi, A., Freda, L., Oriolo, G., Vendittelli, M.: The sensor-based random graph method for cooperative robot exploration. IEEE/ASME Trans. Mechatron. **14**(2), 163–175 (2009)
15. Cao, Y.U., Fukunaga, A.S., Kahng, A.B., Meng, F.: Cooperative mobile robotics: Antecedents and directions. In: Intelligent Robots and Systems 95.'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on, vol. 1, IEEE, pp. 226–234 (1995)
16. Khamis, A., Hussein, A., Elmogy, A.: Multi-robot task allocation: A review of the state-of-the-art. In: Cooperative Robots and Sensor Networks 2015, Springer, pp. 31–51 (2015)
17. Ström, D.P., Bogoslavskyi, I., Stachniss, C.: Robust exploration and homing for autonomous robots. Robot. Auton. Syst. **90**, 125–135 (2017)
18. Nelson, E., Corah, M., Michael, N.: Environment model adaptation for mobile robot exploration. Auton. Robot. **42**(2), 257–272 (2018)
19. Francis, G., Ott, L., Marchant, R., Ramos, F.: Occupancy map building through bayesian exploration. Int. J. Robot. Res. **38**(7), 769–792 (2019)
20. Shrestha, R., Tian, F.-P., Feng, W., Tan, P., Vaughan, R.: Learned map prediction for enhanced mobile robot exploration. IEEE (2019)
21. Kim, D., Carballo, D., Di Carlo, J., Katz, B., Bledt, G., Lim, B., Kim, S.: Vision aided dynamic exploration of unstructured terrain with a small-scale quadruped robot. IEEE (2020)

22. Zeng, T., Si, B.: Mobile robot exploration based on rapidly-exploring random trees and dynamic window approach. IEEE, ICCAR (2019)
23. Zhu, C., Ding, R., Lin, M., Wu, Y.: A 3d frontier-based exploration tool for mavs. IEEE (2015)
24. Aguilar, W.G., Rodríguez, G.A., Álvarez, L., Sandoval, S., Quisaguano, F., Limaico, A.: Visual slam with a rgb-d camera on a quadrotor uav using on-board processing. In: International Work-Conference on Artificial Neural Networks, Springer, pp. 596–606 (2017)
25. Aguilar, W.G., Rodríguez, G.A., Álvarez, L., Sandoval, S., Quisaguano, F., Limaico, A.: Real-time 3d modeling with a rgb-d camera and on-board processing. In: International Conference on Augmented Reality, Virtual Reality and Computer Graphics, Springer, pp. 410–419 (2017)
26. Kaufman, E., Takami, K., Ai, Z., Lee, T.: Autonomous quadrotor 3d mapping and exploration using exact occupancy probabilities. In: 2018 Second IEEE International Conference on Robotic Computing (IRC), IEEE, pp. 49–55 (2018)
27. Sugihara, K.: Approximation of generalized voronoi diagrams by ordinary voronoi diagrams. CVGIP: Graph. Models Image Process. **55**(6), 522–531 (1993)
28. Choset, H., Burdick, J.: Sensor-based exploration: The hierarchical generalized voronoi graph. Int. J. Rob. Res. **19**(2), 96–125 (2000)
29. Warren, C.W.: Global path planning using artificial potential fields. In: Robotics and Automation, 1989. Proceedings. 1989 IEEE International Conference on, pp. 316-321. IEEE (1989)
30. Koren, Y., Borenstein, J.: Potential field methods and their inherent limitations for mobile robot navigation. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), IEEE, pp. 1398–1404 (1991)
31. Hart, P.E., Nilsson, N.J., Raphael, B.: A formal basis for the heuristic determination of minimum cost paths. IEEE Trans. Syst. Sci. Cybern. **4**(2), 100–107 (1968)
32. Kim, J., Seong, K.J., Kim, H.J.: An efficient backtracking strategy for frontier method in sensor-based random tree. In: Control, Automation and Systems (ICCAS), 2012 12th International Conference on, IEEE, pp. 970–974 (2012)
33. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numerische mathematik **1**(1), 269–271 (1959)
34. Wang, H., Yu, Y., Yuan, Q.: Application of dijkstra algorithm in robot path-planning. In: 2011 Second International Conference on Mechanic Automation and Control Engineering, IEEE, pp. 1067–1069 (2011)
35. Kim, J., Bonadies, S., Lee, A., Gadsden, S.A.: A cooperative exploration strategy with efficient backtracking for mobile robots. In: IEEE International Symposium on Robotics and Intelligent Sensors (IRIS), pp. 104-110. IEEE (2017)
36. Goodman, J.M., Kim, J., Gadsden, S.A., Wilkerson, S.A.: System and mathematical modeling of quadrotor dynamics. In: Unmanned Systems Technology XVII, Vol. 9468, International Society for Optics and Photonics, p. 94680R (2015)
37. Kim, J., Wilkerson, S.A., Gadsden, S.A.: Comparison of gradient methods for gain tuning of a pd controller applied on a quadrotor system. In: Unmanned Systems Technology XVIII, Vol. 9837, International Society for Optics and Photonics, p. 98370V (2016)
38. Optitrack: Prime 13W (2021). https://optitrack.com/products/prime-13w/
39. Bitcraze: Crazyflie 2.0. https://www.bitcraze.io/crazyflie-2/ (2021)
40. Preiss, J.A., Honig, W., Sukhatme, G.S., Ayanian, N.: Crazyswarm: A large nano-quadcopter swarm. In: Robotics and Automation (ICRA), 2017 IEEE International Conference on, pp. 3299–3304. IEEE (2017)
41. Bitcraze: Crazyradio PA. https://www.bitcraze.io/crazyradio-pa/ (2021)

Jinho Kim received the Bachelors in Aerospace Engineering from Chungnam National University, Daejeon, South Korea, in 2011, and the M.S. and Ph.D. degrees in Mechanical Engineering from the University of Maryland, Baltimore County, Baltimore, MD, USA, in 2017 and 2018, respectively. He is currently a Postdoctoral Research Fellow of the Robotics Research Center at the United States Military Academy at West Point. His current research interests include the application of nonlinear control theory, robotics, multi-robot system, and vision-based autonomous control and navigation.

Charles D. Eggleton received his B.S. in Naval Architecture and his M.S. and Ph.D in the Aeronautics and Astronautics from Stanford, in 1989 and 1994, respectively. He subsequently did a Post-Doctoral position in the Department of Biomedical Engineering at The Johns Hopkins University and then joined the the Department of Mechanical Engineering at University of Maryland, Baltimore County in 1998. He is currently a Professor and continues to conduct research and mentor students.

Stephen A. Wilkerson received the B.S. degree in Mechanical Engineering from The John Hopkins University, Baltimore, MD, USA, in 1982, the M.S. degree in Structural Engineering from The George Washington University, Washington, DC, USA, in 1985, and the Ph.D. degree in Mechanical Engineering from The Johns Hopkins University, in 1990. His Ph.D. thesis was titled "A Boundary Integral Approach to Three-Dimensional Underwater Explosion Bubble Dynamics." He continues his research and mentoring of students with the York College of Pennsylvania, York, PA, USA.

S. Andrew Gadsden completed his Bachelors in Mechanical Engineering and Management (Business) at McMaster University in 2006. In 2011, he completed his Ph.D. in Mechanical Engineering at McMaster in the area of estimation theory. Andrew worked as a postdoctoral researcher at the Centre for Mechatronics and Hybrid Technology (Hamilton, Ontario, Canada). He also worked as a Project Manager in the pharmaceutical industry (Apotex Inc.) for three years. Before joining the University of Guelph in 2016, he was an Assistant Professor in the Department of Mechanical Engineering at the University of Maryland, Baltimore County. Andrew worked with a number of colleagues in NASA, the US Army Research Laboratory (ARL), US Department of Agriculture (USDA), National Institute of Standards and Technology (NIST), and the Maryland Department of the Environment (MDE). He is an elected Fellow of ASME, has been nominated for the rank of IEEE Senior Member, and is a Professional Engineer of Ontario. Andrew is an Associate Editor for the International Journal of Robotics and Automation and is a reviewer for a number of ASME and IEEE journals and international conferences. Andrew is a 2019 SPIE Rising Researcher award winner based on his work in intelligent estimation theory, and is also a 2018 Ontario Early Researcher (ERA) award winner based on his work in intelligent condition monitoring strategies. He was also awarded the 2019 University of Guelph Faculty Association (UGFA) Distinguished Professor Award for Excellence in Teaching in the College of Engineering and Physical Sciences.