



A UAV-based explore-then-exploit system for autonomous indoor facility inspection and scene reconstruction

Chuanxiang Gao, Xinyi Wang, Ruoyu Wang, Zuoquan Zhao, Yu Zhai, Xi Chen^{*}, Ben M. Chen

Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Shatin, NT, Hong Kong, China

ARTICLE INFO

Keywords:

Indoor facility inspection
Explore-then-exploit system
Localization
Coverage path planning
Trajectory generation
Unmanned aerial vehicle

ABSTRACT

Traditional indoor facility inspections on pipelines and boilers are conducted manually and can be logistically challenging, labor-intensive, costly, and dangerous for the inspectors. With the maturity of unmanned technology, the unmanned aerial vehicle (UAV) is becoming a promising alternative to the problematic manual inspection. However, due to the lack of GPS signal indoor, the localization of UAVs is a big challenge to achieve fully autonomous inspection. Moreover, the narrow and complex indoor environment makes it difficult to guarantee flight safety. This paper presents a UAV-based explore-then-exploit system to tackle these problems for autonomous indoor facility data collection and scene reconstruction. The proposed system consists of a hardware description and integration of two UAVs, a two-step simultaneous localization and mapping (SLAM) method for UAV localization and 3D environmental mapping, a safety-guaranteed coverage path planning algorithm for inspection and data collection, as well as an obstacle-aware trajectory generation method. The proposed system is examined in GPS-denied and cluttered indoor environment and 3D scene reconstruction is conducted. The quantitative analysis shows that the positioning accuracy is centimeter-level and the reconstruction error is within 3 cm. The performance analysis demonstrates the robustness and feasibility of our system in reconstructing and inspecting complex indoor environments for high-efficiency and low-cost facility management.

1. Introduction

The pace of urban renewal is accelerating in various cities around the world, given the fact that the current conditions of old buildings and infrastructures cannot be ignored. Regular inspections and repair works are conducive and indispensable for urban safety. Especially for civil infrastructure facility management, such as boiler interiors, air ducts, and pipelines, frequent inspection and fault detection are significant for preventing accidents and minimizing economic losses. Traditionally, these inspections require operators to work in hazardous environments that are difficult and risky to access. Besides, most of these facilities have complex configurations or large surface areas, which increase the time and labor cost of manual defect detection.

To save labor costs for facility management, reduce the time and risk of inspections, and improve the accuracy of facility defect and fault detection, more and more researchers tend to incorporate unmanned aerial/ground vehicles (UAV/UGV) into automatic inspection data collection. Especially, in recent years, UAVs and UGVs are gradually replacing manual inspection due to the breakthrough of unmanned technology. Many research works are focusing on using UAVs and UGVs in inspections and monitoring, including tunnel inspection [1],

bridge inspection [2,3], construction change detection [4], and boiler inspection [5,6]. For UAV applications, because of the limitation of the localization system, most works are conducted outdoors with good GPS signals [7–10]. In Tan et al. [11], the authors proposed an automatic data collection system, based on building models, a path optimization process is executed to collect the data by UAVs following the generated viewpoints. However, BIM models may not be available in many real applications, which means this method cannot be deployed in an unknown environment. To make UAVs usable in indoor scenarios, Hu et al. [12] proposed a UAV-UGV collaborative system for indoor data collection. The data from the UGV can be used by the UAV to navigate in cluttered environments. However, UGV can only move in a 2D plane, which means the constructed map is a 2D map or an incomplete 3D map. For the planning and control of UAV, this map does not meet the requirements for data collection, because many corners in the map are unknown, and the shape of the device is unknown. Moreover, they did not tackle the localization problem which is important for indoor applications. In addition, most of the facilities need to be inspected at different locations and angles, so UAV is much more flexible than UGV.

^{*} Corresponding author.

E-mail address: xichen002@cuhk.edu.hk (X. Chen).

In indoor applications, there is no GPS signal. To achieve autonomous flight, the localization system is needed to provide the position information for the UAV. Generally, UWB and VICON are used to provide the position information. However, these are external positioning devices, it is hard to deploy these devices in new environments. For inspection and reconstruction data collection, the overlap rate and shooting angle are important [13]. To get a better reconstruction result, generally, the overlap rate should be larger than 65%, and the target facility needs to be captured at different locations and angles. To get a high-quality image, the distance between shooting points and the facility should meet the requirements of inspection. Furthermore, flight safety needs to be guaranteed, it is not easy to get the shortest path while ensuring there is no obstacle in the flight path. Moreover, generating a dynamically feasible trajectory to follow a path can be challenging in an obstacle-dense indoor environment. It puts a high requirement for a UAV to adjust the primary path flexibly to avoid obstacles while satisfying the dynamic limits. Motivated by the aforementioned difficulties, this paper presents an efficient UAV-based explore-then-exploit system that can be used in indoor data collection. This system guarantees the coverage of the target facilities, collision avoidance between UAVs and obstacles, and flight safety. The major contributions of this paper are as follows:

- We resolve the indoor inspection and scene reconstruction task in two steps. An efficient and safety-guaranteed explore-then-exploit system is proposed, which can be easily deployed in any scenario without any initial knowledge of the environment.
- A novel two-step SLAM method for the explore-then-exploit system is proposed, in which lidar SLAM and visual SLAM are fused to provide robust localization and 3D mapping information for both UAVs.
- A coverage path planning algorithm for data collection is proposed, in which both the coverage of the target area and the safety of viewpoints are guaranteed, Flight efficiency is also taken into consideration to get the shortest flight path.
- To follow the given viewpoints as quickly as possible, an obstacle-aware trajectory generation algorithm is proposed in which smoothness is considered with collision avoidance to provide a high-quality trajectory.

The remainder of this paper is organized as follows. Section 2 is the literature review and background of corresponding areas. Section 3.1 shows the hardware design and integration of the proposed system. The SLAM system for exploration and exploitation is presented in Section 3.2. Section 3.3 introduces the framework and details of the proposed coverage path planning method. Section 3.4 presents the UAV trajectory generation algorithm. Also, several experiments are conducted to verify the validity of the proposed system, and the details are illustrated in Section 4. Section 5 presents the discussion and quantitative analysis of the reconstruction model. Conclusions and future works are presented in Section 6.

2. Literature review

The proposed system focus on tackling the challenge of flying UAVs in an indoor cluttered environment with no GPS signal and external positioning facilities. SLAM is used to provide the localization information of two UAVs and build a 3D map of the indoor environment. A coverage path planning and trajectory generation algorithm is used to guarantee the coverage of the target area and flight safety. Therefore, this section provides a literature review and background materials for lidar-based SLAM, visual-based SLAM, coverage path planning algorithms, UAV trajectory generation algorithms, and their applications in facility inspection and maintenance.

2.1. SLAM

In indoor GPS-denied environments, many researchers use external positioning facilities like VICON, and UWB to provide localization information to UAVs [14]. However, in the real-world industry environment, it is impractical to arrange external positioning facilities in advance in the flight environment. SLAM plays an important role in estimating the pose of UAVs and building a map during autonomous flight. The SLAM technologies can be divided into two main categories: lidar-based SLAM and visual-based SLAM [15]. In our proposed system, considering the endurance, payload, and capabilities of each UAV, a lidar-based SLAM is implemented in the step of exploration to get the 3D map of the work area and a visual-based SLAM is used to estimate the pose of the UAV in exploitation step. Therefore, this subsection provides the background materials for both lidar-based SLAM and visual-based SLAM.

Lidar-based SLAM is robust and precise in both outdoor and indoor environments because the light detection and ranging sensors have a large scanning radius and high resolution, and can effectively deal with scenes with changing light [16]. In Zhang and Singh [17], the author proposed a real-time lidar odometry method and combine it with a point cloud registration method to achieve real-time lidar odometry and mapping. However, this method has the problem of motion estimation drift. To overcome this problem, Shan and Englot [18] proposed to segment and remove the point cloud of uneven ground before feature extraction to reduce the drift. Zhang and Singh proposed to integrate lidar with camera and IMU to improve the accuracy of pose estimation [19,20]. Some works that focus on improving the computational efficiency of lidar odometry are also conducted [21,22].

Compared to lidar-based SLAM, visual-based SLAM is more suitable for small mobile platforms, because the camera is much smaller and lighter than lidar, which means the UAV can carry more equipment for inspection. There are three main categories of visual-based SLAM (visual-only, visual-inertial, and RGB-D SLAM) [23]. The first visual-only SLAM algorithm is MonoSLAM proposed by Davison et al. [24]. After that, Forster et al. [25] proposed a semi-direct monocular visual odometry and tested it on UAV. Furthermore, ORB-SLAM, ORB-SLAM2, and ORB-SLAM3 [26–28] were proposed and widely used in the current autonomous mobile platform. For visual-inertial SLAM, in Leutenegger et al. [29], the author proposed a keyframe-based visual-Inertial SLAM that combines IMU data and re-projection terms into an objective function and jointly optimizes the odometry. Qin et al. [30] proposed a monocular visual-inertial system, which uses an initial guess for a non-linear optimization process to minimize visual odometry errors. There are also some works that combine the RGB images and depth information together to achieve real-time pose estimation [31–33].

2.2. Coverage path planning algorithm

Coverage path planning (CPP) in inspection data collection is the method of calculating a set of viewpoints that can cover the whole of interesting structures or facilities. The process of CPP can be divided into two parts: viewpoints generation and path generation. Viewpoint defines the 3D location and orientation information and viewpoints generation gives UAVs the position and orientation that need to reach. The results of generated viewpoints significantly influence the quality of collected data. Path generation defines the flight order of viewpoints. Designing a good path can let the UAV move less distance and save time.

Currently, some researchers are focusing on using the method of CPP and UAV to do environment inspection [34–36]. Zhou et al. [8] proposed a one-step method, based on a prior map, it can estimate the height and size of the building. Then the problem of building inspection is formulated into an optimization problem that maximizes reconstruction quality. In Zhang et al. [37], the author combines the sampling-based method with an optimization-based method to improve

both the information quality and the path efficiency. To improve computing efficiency, Cao et al. [14] proposed a hierarchical planning framework. He solved the problem on two levels, one for subspace allocation and one for trajectory generation. For the online CPP problem, Kuang et al. [38] proposed a framework that takes the initial path as input, utilizes SLAM to generate a sparse point cloud, estimates building height, and does path planning based on the prediction results.

There are also some works related to using multi-agent coverage path planning in UAV environment inspection. Jing et al. [39] presented a multi-UAV coverage path planning framework for the inspection of large-scale, complex 3D structures. They combined the set covering problem with the vehicle routing problem and used a modified biased random key genetic algorithm to solve it. In Chen et al. [40], the author proposed a path planning algorithm based on optimal mass transport optimization and assigned the task views to different drones. To inspect post-disaster buildings, Nagasawa et al. [41] first divided the interested area. For each area, a set of camera positions can be generated. This problem was formulated into a multiple traveling salesman problem. In Zheng et al. [42], the author proposed a route planning methodology to get the optimal solution for multi-UAV data capture route planning.

However, most works are conducted outdoors with large free work areas. In the indoor environment, there are fewer free spaces, and the obstacles are more complex. As a result, obstacle avoidance needs to be considered during the process of coverage path planning.

2.3. UAV trajectory generation

Generating a dynamically feasible trajectory to follow a path can be challenging in an obstacle-dense indoor environment. It puts a high requirement for a UAV to adjust the primary path flexibly to avoid obstacles while satisfying the dynamic limits.

The most common approach to solve such a trajectory generation problem is to formulate it as an optimization problem using the gradient-based solver [43,44]. Earlier work on UAV trajectory generation in Mellinger and Kumar [45] uses a polynomial spline to represent the trajectory and formulate obstacles as convex constraints. They solve the QP problem with the goal of minimizing the snap term of the trajectory. In Richter et al. [46], they used a sampling-based approach to search for a collision-free initial reference to warm up the optimization process. Combined with a sequence of polynomial segments, the reference was then converted into convex constraints to generate a smooth trajectory.

Despite the geometric initial reference being collision-free, the final trajectory may encounter obstacles. On the one hand, using convex hulls to formulate the obstacles space may cause a smaller feasible solution space, which could impact the effectiveness of the solving process. On the other hand, in practice, trajectory generation optimization is always a non-convex problem in which gradient-based solvers easily get stuck at a local optimal point. It thus requires a good initial guess to alleviate such a drawback.

Recently, another popular technique is to solve the non-convex optimization problem by combining motion primitives. Motion primitives are a set of continuous trajectories that enable us to encode the dynamic constraints into the planning space. They are conducive to effectively reducing the intractable dimension of optimization variables and alleviating the computational load. One way of pre-computing these primitives is to sample on the vehicle's boundary state constraints and solve a boundary value problem (BVP) [47]. Each primitive connects an initial state to an end state called boundary state constrained primitives (BSCPs). Several work use motion primitives to alleviate their computational load. In Zhou et al. [48], the author proposed a real-time trajectory generation method combined with BSCPs to separate the dynamic and geometric environmental constraints. Similar to the above work, we convert the viewpoints to a time-optimal path for the optimization problem and use a pre-computing motion primitives table

for fast generating a dynamical-feasible and collision-free trajectory to follow the path.

Each of the three above sub-sessions is a broad research area, while they are not sufficiently integrated for real applications in the construction industry. To the best of our knowledge, there is no work that has been published on indoor facility inspection and scene reconstruction using an integrated UAV-based system.

3. Methodology

Fig. 1 shows the architecture of the proposed explore-then-exploit system. The protected UAV is used to explore the unknown environment and build up a 3D map and keyframe graph which can be used in the step of exploitation. The small and light UAV is used to do coverage path planning for the target area and follow the generated trajectory to get detailed information about facilities. This section will introduce detailed information on each module.

3.1. Hardware platform

As shown in Fig. 2, the proposed system consists of two UAVs with different functionality. The advantage of using 2 UAVs compared to using 1 UAV is that it can maximize the capabilities of each UAV, thereby increasing the overall efficiency of task completion. The first UAV is used to explore the whole indoor environment. Because the environment is unknown, the biggest challenge is to guarantee flight safety. As a result, we design a protected UAV with reference to Petris et al. [49]. 3D lidar is selected as the equipment for laser SLAM to achieve robust localization and mapping in an unknown environment. The stereo camera is used to build up the keyframe graph for the step of exploitation.

For the step of exploitation, a small and light UAV is designed, because the endurance needs to be longer to get more images during one flight. The stereo camera and keyframe graph are used to provide the localization information during the flight. DJI ACTION2, a 4 K optical camera, is used to collect detailed features of target facilities.

In the step of exploration, the lidar-based SLAM is performed using Velodyne 16, the keyframe graph is constructed by VINS using the images from Intel RealSense D435i, and the SLAM algorithm is online-calculated using NVIDIA Jetson Xavier NX. After exploration, the created map is processed on a laptop to determine views, paths, and trajectories. NVIDIA Jetson Xavier NX is utilized to control the UAV and perform an online visual-based SLAM computation during the exploitation step.

After a broad comparison, with the consideration of price and weight, we choose Velodyne 16 as the 3D lidar and D435i as the stereo camera. Other lidars like Livox and Ouster and cameras like D455 and ZED are also supported in the proposed framework because it is a general framework. The detailed comparison is listed in the appendix.

3.2. SLAM for explore-then-exploit system

The working space is relatively limited, and the environment is full of obstacles while conducting indoor autonomous inspections, particularly in industrial facilities. As a result, achieving strong localization for UAVs is one of the biggest challenges. Considering the particularities of indoor environments, existing SLAM technology, and the payload of UAVs, we choose Fast-LIO2 [21] as the lidar-based slam for the UAV in the exploration step. Because in Fast-LIO2, the raw points are directly registered in the map, and incremental k-dimensional tree data is used to update and re-balance the map, which makes the lidar-inertial odometry framework accurate, robust, and fast. In the step of exploitation, VINS-MONO [30] is implemented to provide pose estimation for the UAV. VINS-MONO is a robust and versatile monocular visual-inertial state estimator that tightly integrates IMU and optical images together and is widely used on onboard computers with limited

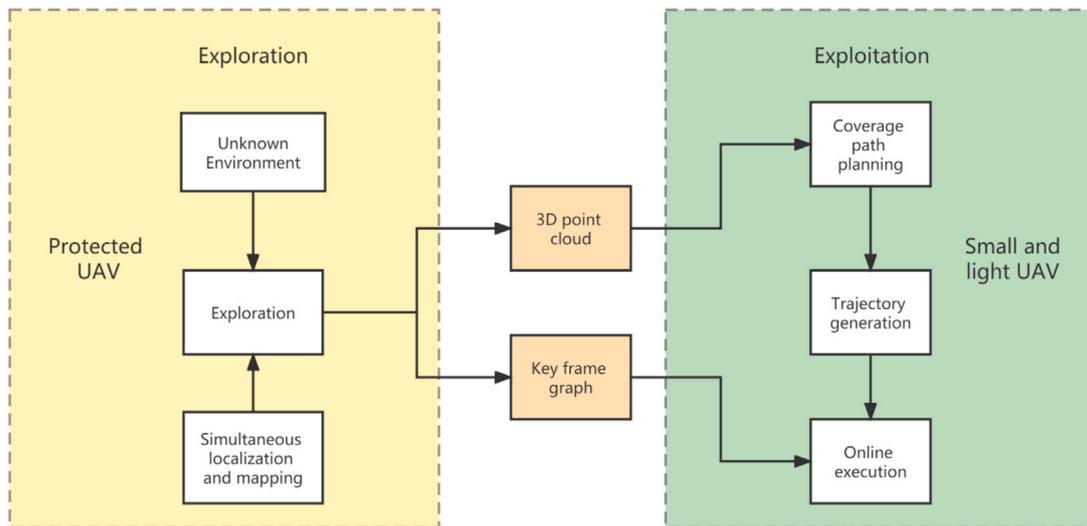


Fig. 1. The overall framework of the explore-then-exploit system.

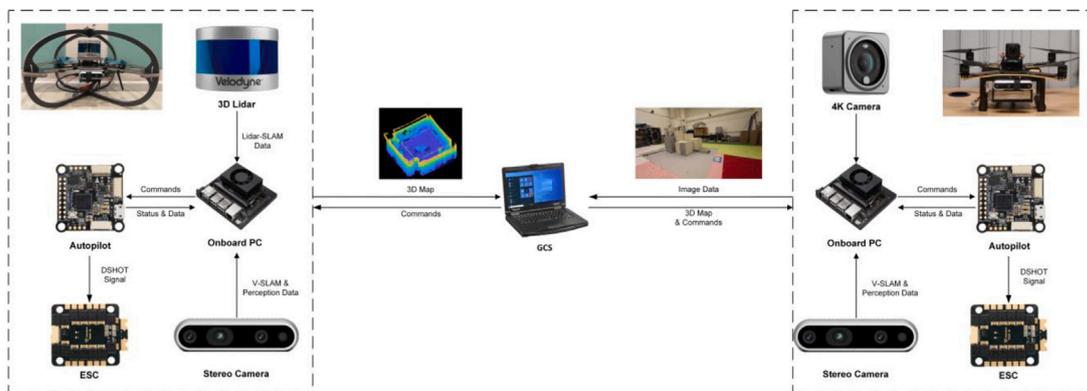


Fig. 2. The hardware design and architecture integrating two designed UAVs.

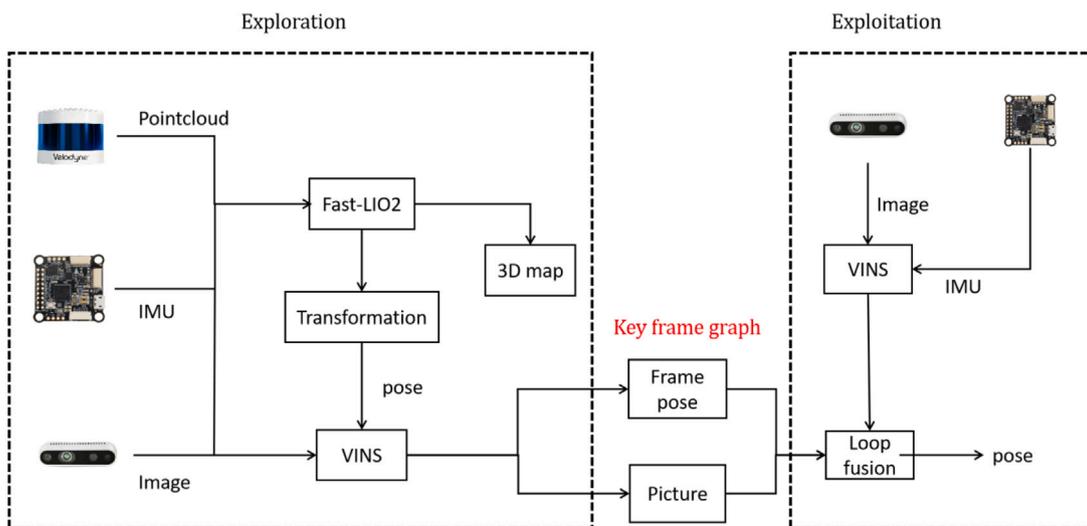


Fig. 3. The framework of SLAM system.

computing resources. We take advantage of these two frameworks and combine them together to provide robust pose estimation for each UAV. The framework of the proposed SLAM method is presented in Fig. 3.

In the step of exploration, Velodyne is responsible for providing the point cloud scan and autopilot is responsible for providing the

IMU data. Fast-LIO2 is implemented in the Xavier NX platform to get the pose estimation and 3D map of the indoor environment. In addition, visual-inertial odometry is also implemented based on the IMU data from autopilot and the image from D435i. During the flight, the onboard computer saves the keyframe at 1 Hz. Each keyframe

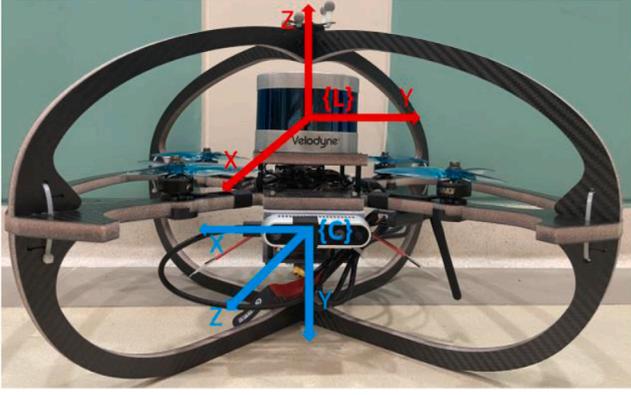


Fig. 4. The coordinate systems of stereo camera and lidar.

contains the global pose and the corresponding image. In addition, the pose of the keyframe is the odometry from Fast-LIO2 because it is more robust than VINS. The corresponding image is from D435i.

However, as illustrated in Fig. 4, the coordinate frame of VINS and Fast-LIO2 is inconsistent. As a result, the odometry in the Fast-LIO2 coordinate needs to be transformed to the VINS coordinate to ensure that the pose saved in the key frame is accurate. Let the coordinate of Fast-LIO2 be L, the coordinate of VINS be C, the pose in L be P_L , the pose in C be P_C and the transformation matrix between L and C be ${}^C T_L$. We can get the following equations:

$${}^C T_L = \begin{bmatrix} {}^C R_L & {}^C P_L \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \quad (1)$$

$$P_C = {}^C T_L P_L \quad (2)$$

where

$${}^C R_L = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}, \quad {}^C P_L = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \quad (3)$$

represent the rotation and the translation of two frames, respectively. The value of $r_{11}, \dots, r_{33}, p_x, p_y, p_z \in \mathbb{R}$ are determined by hardware design. After the coordinate transformation, the key frame graph is successfully built. To increase the matching frequency of loop closure, in the step of exploration, the exploration method is based on the field of view of the camera, so most of the viewing angles in the step of exploitation will be covered.

In the step of exploitation, just relying on visual-inertial odometry is not enough. Because the working time becomes longer, the estimation drift will be larger. Thus, the local pose from VINS and the global pose from the keyframe graph are combined in loop fusion. When the UAV encounters a similar scene, searching and image matching are performed to get the global pose of the UAV and fix the drift from local estimation. After loop fusion, the UAV can get a precise pose estimation with little drift.

3.3. Safety-guaranteed coverage path planning

It is challenging to ensure both target facility coverage and flight safety when collecting data inside. To solve this issue, we propose a safety-guaranteed coverage path planning algorithm, which includes viewpoints generation and path generation. The overall framework of the proposed method is shown in Fig. 5.

In the step of viewpoints generation, several cuboids $B \in [x_l, y_l, z_l, x_r, y_r, z_r]$, with $[x_l, y_l, z_l]$ being the lower left coordinate of B and $[x_r, y_r, z_r]$ being the upper right coordinate of B , are used to describe different objects. The point clouds in different cuboids are treated as regions of interest (ROI) and obstacles. Then the farthest point sampling

algorithm is performed to make the point cloud uniformly distributed in the region of interest. The details are as shown in Algorithm 1. Given the input points set \bar{P} , N is the number of the elements, and N_{new} is the number of the elements of the output points set. We randomly set a start point s as the sampling set \bar{S} . Then iterative updates the distance set \bar{D}_p by calculating the minimum distance between $point_p$ and \bar{S} . In each iteration, a new point that is farthest from \bar{S} is selected and added to \bar{S} . Finally, the distributed points set \bar{S} can be extracted.

Algorithm 1: Implementation of farthest point sampling algorithm

Input: \bar{P} , N , N_{new}

Output: \bar{S}

```

1  $s \leftarrow s \in \bar{P}$ ;
2  $\bar{S} \leftarrow \{s\}$ ;
3 while iterations  $< N_{\text{new}}$  do
4   foreach  $point_p \in \bar{P} \setminus \bar{S}$  do
5     foreach  $point_s \in \bar{S}$  do
6        $\bar{D}_s \leftarrow \bar{D}_s \cup \{\text{distance}(point_p, point_s)\}$ ;
7      $\bar{D}_p \leftarrow \bar{D}_p \cup \{d \in \bar{D}_s \mid \min(d)\}$ ;
8    $point_{\text{new}} \leftarrow \arg \max_{point_p} \bar{D}_p$ ;
9    $\bar{S} \leftarrow \bar{S} \cup \{point_{\text{new}}\}$ ;

```

These distributed point clouds need to be downsampled according to the overlap rate. As shown in Fig. 6, the field of view of the camera is FOV , the overlap rate is R , the aspect ratio of the image is κ , the overlap distance in the horizontal direction is d_H and the overlap distance in the vertical direction is d_V . The effective distance in the horizontal FOV_H and vertical FOV_V directions can be calculated by the following equations:

$$\kappa = \frac{FOV_V}{FOV_H} \quad (4)$$

$$FOV_H = 2D \tan \frac{FOV}{2} \quad (5)$$

$$R = \frac{d_H}{FOV_H} = \frac{d_V}{FOV_V} \quad (6)$$

$$d_H = 2RD \tan \frac{FOV}{2} \quad (7)$$

$$d_V = 2R\kappa D \tan \frac{FOV}{2}. \quad (8)$$

The distributed points set \bar{S} is downsampled to ensure the horizontal and vertical distance between two points is FOV_H and FOV_V . Following point cloud downsampling, the viewpoints are generated at a distance D from the point cloud along the normal direction of the target facility surface. The viewpoints adjustment process is then performed to alter the viewpoints situated in the obstacle area. As shown in Fig. 7, when the generated viewpoint is in the obstacle area, it should be replaced with a new viewpoint by searching in the normal direction until reaching a safe distance. The FOV_H decreases as the distance to the plane decreases. So, to ensure the coverage rate, the new viewpoint has two view angles. Let the angle between the centerline of the FOV and the normal be α , and the distance between the new viewpoint and plane be D_1 . α can be calculated by Eq. (9). Then we can get the best viewpoints within free space.

$$\alpha = -\frac{FOV}{2} + \tan^{-1} \frac{FOV_H}{2D_1} \quad (9)$$

After viewpoints generation, we can get n viewpoints v_i , $i \in \{1, 2, \dots, n\}$. Let

$$\bar{V} = \{v_1, \dots, v_n\} \quad (10)$$

be the set of viewpoints. The next step is to calculate the shortest path through all viewpoints. We formulate it into a safety-guaranteed traveling salesman problem. Let $\bar{N} = \{1, 2, \dots, n\}$, the minimum discrete

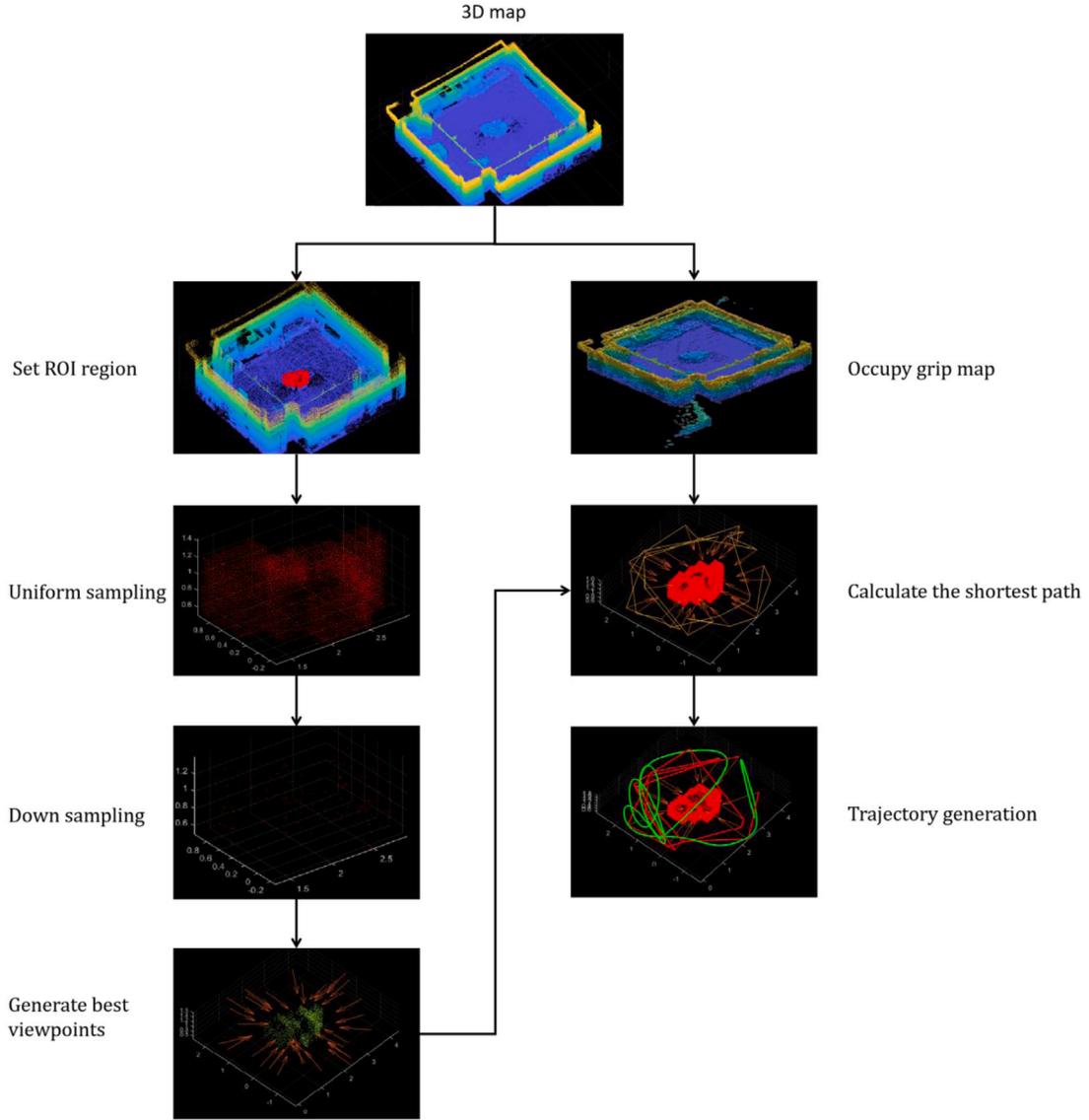


Fig. 5. The entire algorithm procedure of viewpoints generation.

interval be d , the distance between v_i and v_j be C_{ij} , and the number of sample points be N_{ij} . The set of discrete points \bar{V}_{ij} between v_i and v_j can be calculated by the following equations:

$$N_{ij} = \frac{C_{ij}}{d} \quad (11)$$

$$\bar{V}_{ij} = \{v_{1ij}, v_{2ij}, \dots, v_{N_{ij}}\}. \quad (12)$$

The generated 3D map is converted to occupancy grid map to determine the obstacle area \bar{O} . The formulation is shown in Eq. (13). v_{ij} represents the connection relationship between v_i and v_j . u_i and u_j are any real number. The objective function F of this optimization problem is to minimize the total distance traveling through all these viewpoints. The first two constraints are used to guarantee that each viewpoint can be visited and only visited once, and the third constraint ensures that there are no sub-tours. The last constraint is used to guarantee the generated path is obstacle free. When the points set \bar{V}_{ij} between v_i and v_j has an intersection with obstacle area \bar{O} , the distance C_{ij} is set to positive infinity. A Genetic Algorithm (GA) is used to solve the problem because it has a good global search ability and can get a good solution in a short period. Finally, a path with the shortest travel cost and no collisions can

be obtained.

$$\begin{aligned} \min F &= \sum_{i=1}^n \sum_{j=1}^n c_{ij} v_{ij} \\ \text{s.t.} \quad & \sum_{j \in \bar{N}} v_{ij} = 1, \quad \forall i \in \bar{N} \\ & \sum_{i \in \bar{N}} v_{ij} = 1, \quad \forall j \in \bar{N} \\ & u_i - u_j + n v_{ij} \leq n - 1, \quad \forall i, j \in \bar{N} \setminus \{0\} \\ & v_{ij} \in \{0, 1\}, \quad \forall i, j \in \bar{N} \\ & u_i, u_j \in \mathbb{R}, \quad \forall i, j \in \bar{N} \\ & c_{ij} = \infty, \quad \forall i, j \in \bar{N}, \bar{V}_{ij} \cap \bar{O} \neq \emptyset \end{aligned} \quad (13)$$

3.4. Obstacle-aware trajectory generation

The generated path from the TSP solver cannot be executed by the UAV directly because it does not consider the dynamic constraints of the UAV. As a result, a trajectory generation method is designed to convert the path to a smooth trajectory while avoiding obstacles in an indoor environment.

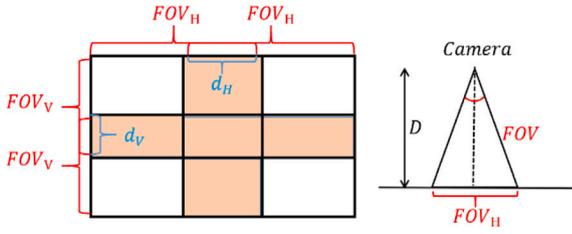


Fig. 6. The illustration of the effective field of view of the camera.

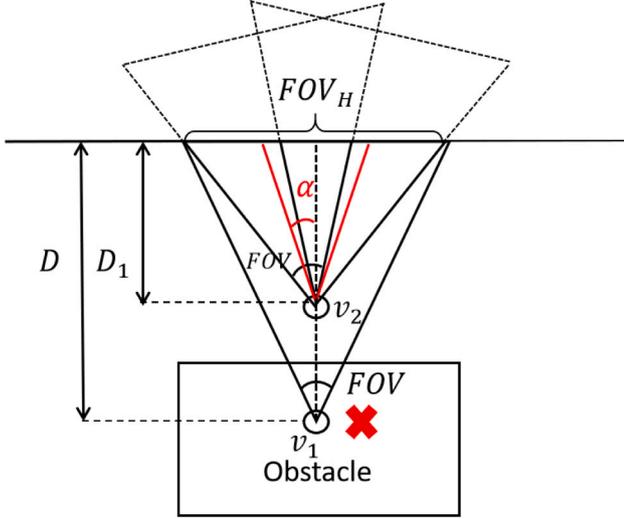


Fig. 7. The illustration of the process of the viewpoint adjustment.

3.4.1. Safe navigation

We formulate the trajectory generation problem as a nonlinear optimization problem minimizing the cost function J that trades off flight safety J_c and path-following performance J_f . This optimization problem can be given as follows:

$$\min_{\mathbf{u}} J = \lambda_c J_c + \lambda_f J_f \quad (14)$$

where λ_c and λ_f represent the weights for each cost term.

To handle the collision of potential threats, we need to penalize the closest distance $d[k]$ between the position on the current executed trajectory of the UAV $\mathbf{p}[k]$ and obstacles in the surrounding environment at time step k . The executed trajectory can be generated through BSCPs, which will be introduced in the next section. To calculate $d[k]$, we represent the environment as a 3D grid map, and the position of the UAV corresponds to the node on the map. For each node, the closest distance value from the node to the obstacles can be obtained efficiently using the Euclidean distance transform (EDT) map [50]. Therefore, the flight safety cost for N time steps in the future is penalized as:

$$J_c = \sum_{k=0}^{N-1} e^{-d[k]}. \quad (15)$$

It can be observed from the cost function that if the UAV moves closer to the obstacle, the cost value will grow rapidly. Therefore, the above evaluation function tends to push the generated trajectories of the UAV to stay away from obstacles to ensure flight safety.

In addition, to follow the path given in Section 3.3 as quickly as possible, we use a jerk-limited trajectory (JLT) method to generate a time-optimal reference trajectory from the UAV's current position to the next closest point on the path [44]. JLT provides a smooth trajectory from arbitrary initial states to a set goal state for the UAV system. It has been proven well suited for UAVs as it could satisfy the maximum

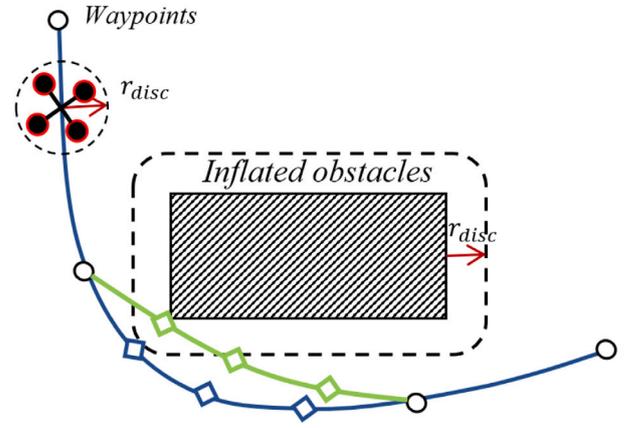


Fig. 8. Representation of following the path using JLT as a reference trajectory. The circles are the viewpoints of the path, the blue squares represent the points on an executed trajectory, and the green squares represent the points on JLT.

thrust and physical limits [51] and can be quickly generated in 3D environments [52]. The deviation of the position between the time-optimal JLT and the current trajectory for N time steps is penalized as

$$J_f = \sum_{k=0}^{N-1} \|\mathbf{p}[k] - \mathbf{p}_{\text{JLT}}[k]\|_2^2, \quad (16)$$

where $\mathbf{p}_{\text{JLT}}[k]$ denotes the position on the JLT at time step k corresponding to $\mathbf{p}[k]$. As shown in Fig. 8, by penalizing the above cost function, the UAV can quickly follow the reference path under the guidance of JLT while avoiding obstacles.

By solving the above problem, a feasible trajectory can be generated to guide the UAV along the reference path while avoiding collisions in indoor environments. However, it can be challenging to solve such general nonlinear optimization problems. Apart from active reactions to obstacles and the following performance, the dynamical feasibility and flight smoothness in cluttered indoor environments should also be considered. Therefore, we construct BSCPs to separate dynamics and smooth constraints from obstacles and path-following constraints to simplify the optimization problem. With the BSCPs, a long feasible trajectory with a large number of state parameters can be encoded in a few parameters (end-state parameter), thus effectively reducing the intractable search space (or dimension) of the optimization problem and alleviating the computational load. Finally, we can use a gradient-free based solver, particle swarm optimization (PSO), to select the high-quality BSCPs by minimizing J while taking the obstacles and the given path into account. PSO is a metaheuristic method that can efficiently find the solution even when the optimization function is not continuous [53]. Compared with the existing results, this gradient-free optimization method has a more stable optimization time and can handle the problem with gradient information.

3.4.2. Construction of BSCPs

In this subsection, we need to pre-compute dynamically feasible motion primitives of the UAV to meet the real flight requirements. A UAV can be modeled by a triple integrator on its x , y , z axes, respectively. Hehn and D'Andrea [54]. Define \mathbf{p} , \mathbf{v} , \mathbf{a} , and \mathbf{j} as position, velocity, acceleration, and jerk, respectively. The smooth constraints can be given as

$$\mathbf{v} \in [\mathbf{v}_{\min}, \mathbf{v}_{\max}], \mathbf{a} \in [\mathbf{a}_{\min}, \mathbf{a}_{\max}], \mathbf{j} \in [\mathbf{j}_{\min}, \mathbf{j}_{\max}]. \quad (17)$$

For each axis, let $\mathbf{x} = [\mathbf{p}, \mathbf{v}, \mathbf{a}]^T$ and $\mathbf{u} = \mathbf{j}$ be the state and control input of the system. Let τ be the discretization time step. The outer loop of the UAV's dynamic model is defined as follows:

$$\mathbf{x}[k+1] = \mathbf{A}\mathbf{x}[k] + \mathbf{B}\mathbf{u}[k], \quad (18)$$

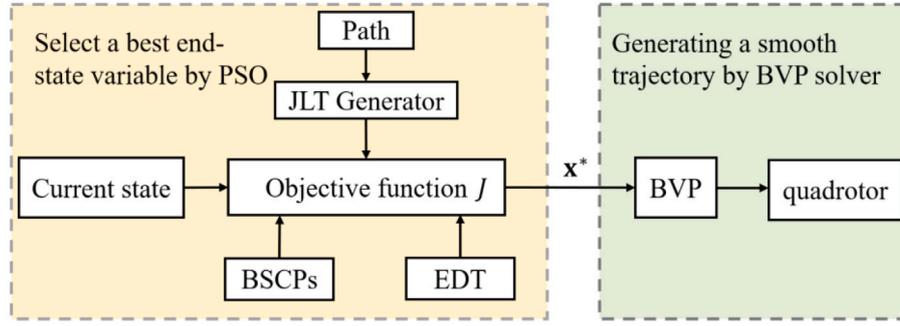


Fig. 9. The overall structure of trajectory generation.

where

$$\mathbf{A} = \begin{bmatrix} 1 & \tau & \tau^2/2 \\ 0 & 1 & \tau \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \tau^3/6 \\ \tau^2/2 \\ \tau \end{bmatrix}. \quad (19)$$

Given the above dynamic model of the UAV, its motions are formulated as BSCPs which can be obtained by solving BVP selected by PSO. The purpose of constructing BSCPs is to control the system from any initial state to the desired goal state $\mathbf{x}_d = [\mathbf{p}_\theta, 0, 0]^T$. The relative error state is defined as:

$$\mathbf{s} = \mathbf{x} - \mathbf{x}_d = \begin{bmatrix} \mathbf{p} - \mathbf{p}_\theta \\ \mathbf{v} \\ \mathbf{a} \end{bmatrix}. \quad (20)$$

To reach \mathbf{x}_d , a BVP solver needs to be designed to drive \mathbf{s} to zero. For this purpose, an offline model-based dynamic programming approach is employed to search for the optimal action u^* of the system. The approximation of value function $Q(\mathbf{s})$ from \mathbf{s} to the origin can be defined as:

$$Q(\mathbf{s}[k]) = r(\mathbf{s}[k], u[k]) + Q(\mathbf{s}[k+1]), \quad (21)$$

where $r(\mathbf{s}, u)$ is the expected reward of action u from relative state \mathbf{s} . To take the desired goal state, input penalty, and invariant constraints into account, the reward function is defined as follows:

$$r(\mathbf{s}, \mathbf{p}) = \mathbf{s}^T \mathbf{W} \mathbf{s} + \lambda \mathbf{p}^2 + J_s(\mathbf{s}, \mathbf{u}), \quad (22)$$

where \mathbf{W} represents the corresponding weight matrix associated with relative error state derivation $\mathbf{s}^T \mathbf{W} \mathbf{s}$, λ is the weight of penalizing the control input, and $J_s(\mathbf{s}, \mathbf{u})$ penalizes the violation of dynamic constraints in Eq. (17) to prevent the UAV making aggressive movements.

The $J_s(\mathbf{s}, \mathbf{p})$ can be defined as:

$$\begin{aligned} J_s(\mathbf{s}, \mathbf{p}) &= \lambda_v (\max(\|\mathbf{v}_{\min} - \mathbf{v}\|, 0) + \max(\|\mathbf{v} - \mathbf{v}_{\max}\|, 0))^2 \\ &= \lambda_a (\max(\|\mathbf{a}_{\min} - \mathbf{a}\|, 0) + \max(\|\mathbf{a} - \mathbf{a}_{\max}\|, 0))^2 \\ &= \lambda_j (\max(\|\mathbf{j}_{\min} - \mathbf{j}\|, 0) + \max(\|\mathbf{j} - \mathbf{j}_{\max}\|, 0))^2, \end{aligned} \quad (23)$$

where $\lambda_v, \lambda_a, \lambda_j$ are the trade-off between \mathbf{v} , \mathbf{a} and \mathbf{j} , respectively and $\|\cdot\|$ denotes the l_2 norm. Through the iterative process, we can get the minimum value function for each state:

$$Q^*(\mathbf{s}[k]) = \min_{\mathbf{p}[k] \in \mathcal{J}} r(\mathbf{s}[k], \mathbf{u}[k]) + Q^*(\mathbf{s}[k+1]), \quad (24)$$

where \mathbf{u} is chosen from a set of admissible control inputs \mathcal{J} . Once the above value iteration has converged, a look-up table consisting of an optimal action for each state is constructed. As a result, for any relative state \mathbf{s} , we can obtain a unique dynamically feasible trajectory \mathbf{p} of the UAV determined by \mathbf{u}^* .

The overall framework of our method is shown in Fig. 9 and Algorithm 2. The collision information has been stored in the EDT map (line 1). For each viewpoint that needs to be strictly passed, We use JLT method to generate a time-optimal reference trajectory \bar{T}_i^r between two viewpoints for quickly following (line 3–4). According to the objective function J , which accounts for the obstacle constraints

Algorithm 2: Trajectory generation with collision avoidance using particle swarm optimization

Input: $\bar{V}, \bar{O}, \mathbf{x}$
Output: \bar{T}

- 1 $\mathcal{M} \leftarrow EDTmap(\bar{O})$;
- 2 randomize m particles;
- 3 **for each** $v_i, i = 1, 2, \dots, n$ **do**
- 4 $\bar{T}_i^r \leftarrow genJLT(\mathbf{x}, v_i)$;
- 5 **for each** $\mathbf{x}_j, j = 1, 2, \dots, m$ **do**
- 6 $\mathbf{v}_j \leftarrow \mathbf{v}_j + \omega_1(\mathbf{x}_j^l - \mathbf{x}_j) + \omega_2(\mathbf{x}^g - \mathbf{x}_j)$;
- 7 $\mathbf{x}_j \leftarrow \mathbf{x}_j + \mathbf{v}_j$;
- 8 $\bar{T}_j \leftarrow BSCP(\mathbf{x}, \mathbf{x}_j)$;
- 9 $cost_j \leftarrow J(\bar{T}_j, \mathcal{M}, \bar{T}_i^r)$
- 10 **if** $cost_j < cost_j^l$ **then**
- 11 $cost_j^l \leftarrow cost_j$
- 12 $\mathbf{x}_j^l \leftarrow \mathbf{x}_j$
- 13 **if** $cost_j < cost_j^g$ **then**
- 14 $cost_j^g \leftarrow cost_j$
- 15 $\mathbf{x}_j^g \leftarrow \mathbf{x}_j$
- 16 $\mathbf{x}^* \leftarrow \mathbf{x}^g$
- 17 $\bar{T}_i \leftarrow BVP(\mathbf{x}, \mathbf{x}^*)$;
- 18 $\bar{T} = [\bar{T}_1, \dots, \bar{T}_n]$

and path-following performance, we can assess all potential end states of the UAV that have been transformed into BSCPs. To minimize J at each time step, the PSO method can choose the end-state constraints \mathbf{x}^* from these potential candidates (line 5–15). After obtaining \mathbf{x}^* , by using the BVP solver, the smooth trajectory \bar{T}_i following the path can be generated with collision avoidance (line 16–17).

4. Experiments

To elaborate on the proposed explore-then-exploit system, we conduct several field experiments in indoor environments without external localization devices. The accuracy and stability of the proposed SLAM system, the robustness of the proposed trajectory generation method, and the performance of the whole system are tested separately. The experiments are conducted in an 8.4 meters long, 6.0 meters wide, and 6.0 meters-high environment. The maximum and minimum dynamic feasibility limits of the velocity, acceleration, and jerk are set as $v = [2.5, -0.5]$, $a = [2.0, -0.5]$, and $j = [5.0, -2.0]$, respectively. VICON is used to get the ground truth of flight trajectory. For the performance test of the whole system, both the step of exploration and exploitation is performed in each scenario, and the 3D map and keyframe graph from the SLAM system is built by the onboard computer Xavier NX. The coverage path planning and trajectory generation module is processed by a laptop with an Intel i7-9700 CPU and 16 GB RAM. The overlap

Table 1
The comparison of the accuracy and stability of different algorithms (vins loop is our method).

Method	MeanError _x (m)	MeanError _y (m)	MeanError _z (m)	MaxError _x (m)	MaxError _y (m)	MaxError _z (m)	VarError _x	VarError _y	VarError _z
Lidar	0.04304	0.03180	0.28513	0.26068	0.17975	0.44477	0.00149	0.00102	0.00515
Vins only	0.11788	0.08165	0.14776	0.39805	0.36375	0.64746	0.01165	0.00749	0.01858
Vins loop	0.08058	0.07666	0.15278	0.27833	0.33723	0.55962	0.00460	0.00427	0.01197

Table 2
The performance comparison of trajectory optimization.

Method		Mellinger [45]	Richter [46]	Our
Simple facility without obstacles	Trajectory length (m)	90.53	100.07	71.55
	Collision distance (m)	0.25	0.27	0.38
Simple facility with obstacles	Trajectory length (m)	95.85	96.71	83.15
	Collision distance (m)	0.06	0.28	0.36
Complex facility with obstacles	Trajectory length (m)	140.74	147.25	114.30
	Collision distance (m)	–	0.07	0.18

rate is set at 65%, and the distance to the plane is set at 2 m. Finally, the collected images are used to reconstruct the facility. We use DJI TERRA in a desktop with an Intel i9-10920X CPU and NVIDIA RTX 4000 GPU to reconstruct the model. The reconstruction result shows our method can collect high-quality data for indoor facility inspection. Fig. 10 shows the detailed information of the laboratory environment, 3D map from lidar SLAM, generated trajectory, work scene, and the overall and detailed feature of the reconstruction result.

4.1. Performance test of proposed SLAM system

In this experiment, Fast-LIO2 and VINS are performed to get the pose of the UAV, and VICON is used to provide the ground truth. Fig. 11 shows the flight trajectory and error of different methods. Table 1 shows the quantitative comparison including the mean error, maximum error, and error variance. Compared to VINS, our method achieves better performance in both mean error, maximum error, and measurement stability and is close to Fast-LIO2.

4.2. Performance test of proposed trajectory generation method

In this experiment, the proposed trajectory generation method is compared with the two most widely used methods: minimum trajectory generation [45] and polynomial trajectory generation [46] under the three different scenarios. The UAV is required to pass through the viewpoints in 3D indoor environments as quickly as possible while avoiding obstacles. The number of viewpoints increases with the complexity of the environment. Each method is tested 20 times for the same environment. For fairness, each method is tested 20 times for the same environment. Table 2 displays the statistical average data to evaluate the performance of the trajectory generation of 20 trails. It can be observed that our method takes a shorter time than others to follow the reference viewpoints while maintaining safe distances from obstacles.

Collision distance measures the closest distance with obstacles for the whole trajectory. The method proposed by Mellinger generates a dynamically feasible trajectory to strictly follow the viewpoints, but it does not consider the complex obstacles. To restrict the trajectory in the safe region, Richter uses RRT to generate collision-free paths as the initial guess. Although it slightly sacrificed the flight time for safety, this kind of optimization method with hard constraints is difficult to meet the requirements of collision avoidance in a narrow space. In contrast, our method penalizes the closest distance with obstacles to push the trajectory away from collisions to achieve better performance.

Trajectory length implied the overall flight time. In any scenario, trajectories from Mellinger and Richter are longer than ours. This is because their algorithm does not optimize the time of path-following. In addition, the overshoot of the trajectories is unavoidable because they solve the optimization problem with strict constraints. In contrast, our

method generates time-optimal JLT as references based on the given viewpoints to pass through these points as quickly as possible. In particular, in a complex environment, our trajectory length increases more slowly than others proving that our method can perform satisfying results even in obstacle-dense environments.

4.3. Performance test of the whole system

4.3.1. Simple facility with no obstacle

We use some polystyrene boxes in the first experiment to represent the target facility. In this experiment, 43 images are collected, and all surfaces of the target facility have been collected successfully, which means our coverage path planning can ensure the complete coverage of the target object. The result is shown in the sixth and seventh row of Fig. 10, a very detailed 3D reconstruction model can be built, which represents the collected images are of very good quality.

4.3.2. Simple facility with obstacles

In the third experiment, we add some obstacles in the environment to test whether our proposed method can generate a safe path and trajectory. The viewpoints adjustment process is performed to guarantee the generated viewpoints maintain a safe distance to the obstacle. In this experiment, the UAV can successfully follow generated trajectory, avoid obstacles, and collect the data. 47 images are collected and used to reconstruct the target facility. The reconstruction result is also used to measure the quality of collected data. As illustrated in the seventh row of Fig. 10, even in an environment with obstacles, the detailed texture of the 3D model is also clear.

4.3.3. Complex facility with obstacle

In the last experiment, we change the shape of the target facility to an irregular polyhedron and added many obstacles in the environment to test the acquisition quality of the proposed system for a complex facility in a cluttered environment. The distance between the obstacle and the target facility is only 1.5 m, and some area is only 0.8 m. Working in such a challenging environment needs the SLAM system to provide a precise pose estimation, otherwise, the UAV would collide with facilities and obstacles. Also, the trajectory generation algorithm needs to generate a feasible path for the UAV to follow. In this experiment, our coverage path planning algorithm can generate viewpoints covering the target facility in a limited space. The trajectory generation algorithm can generate a trajectory considering the dynamic constraints of the UAV to ensure the UAV can fly safely in a narrow environment.

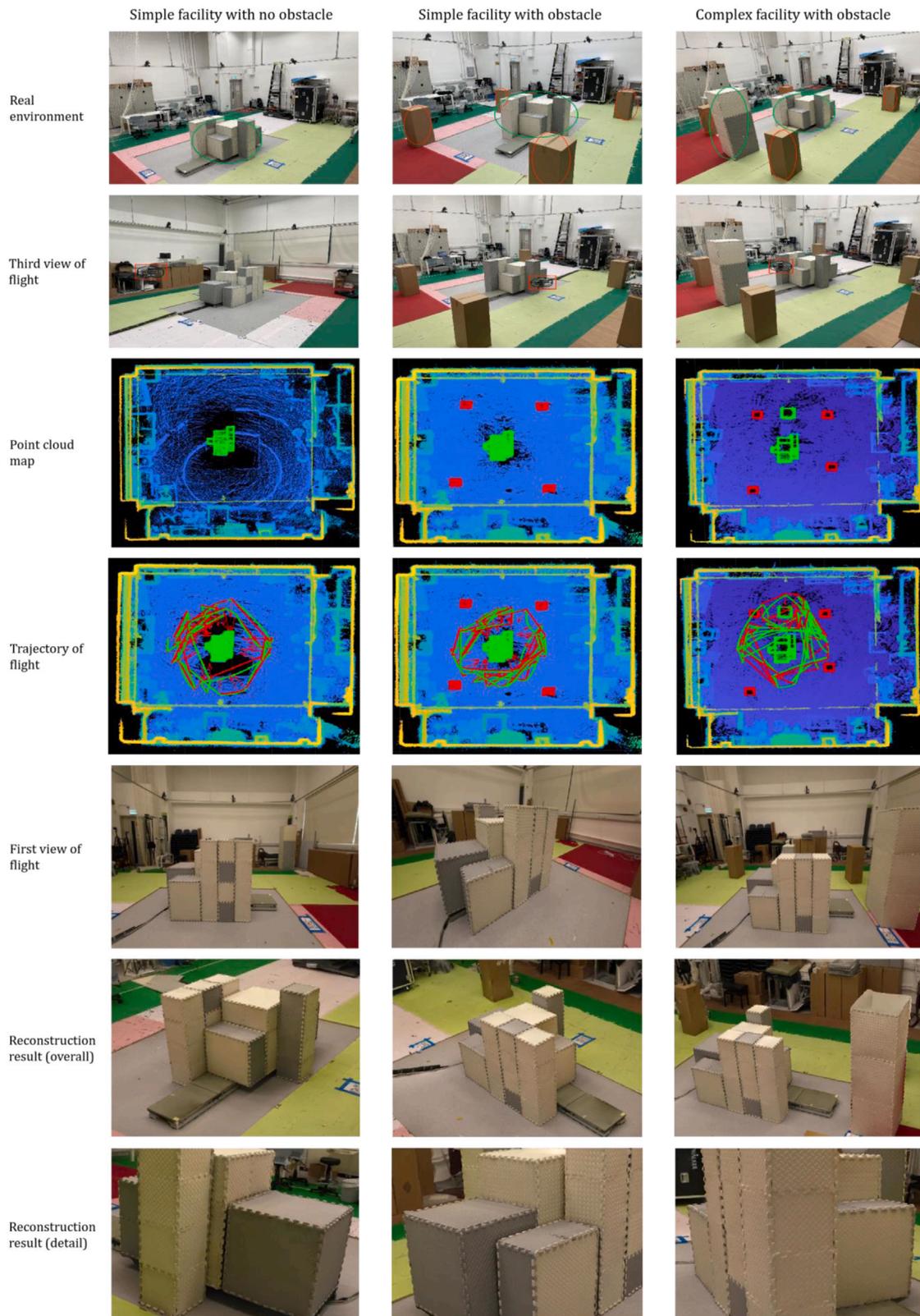


Fig. 10. The experiments environment, setup, process, and results.

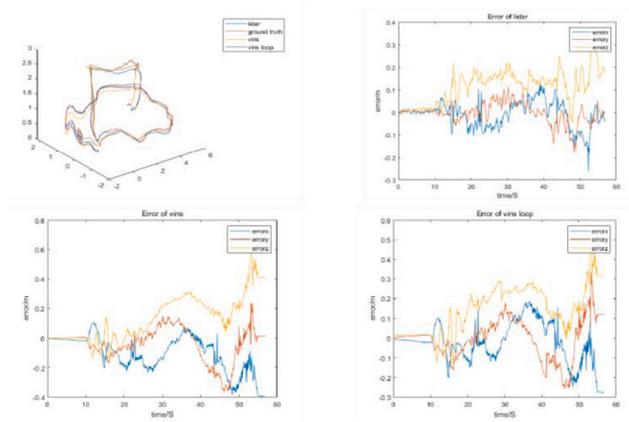


Fig. 11. The trajectory and measurement errors of different methods.

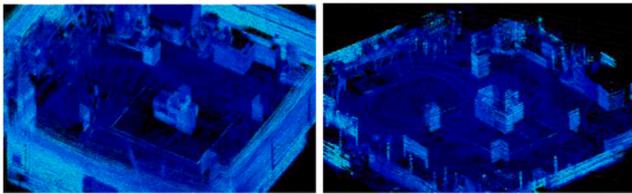


Fig. 12. 3D map built by UAV and UGV respectively. The left map is built by UAV. The right map is built by UGV.

5. Discussion

In indoor facility inspections, there are two main concerns: what scenarios the proposed system is suitable for and whether the system outputs can satisfy the requirements of inspection. For the first question, our system does not rely on external positioning facilities, as mentioned in Section 4.1, the measurement error of our SLAM system is within 10 cm. The distance between the two facilities is 0.8 m, and the tip-to-tip dimension of our UAV is 41 cm, meaning our trajectory generation method can find a safe trajectory within a 0.4 m distance. These advantages allow our system to apply to most indoor scenarios. In addition, compared to BIM and UAV-based methods, our method does not require the initial knowledge of the target environment, the first UAV with a 3D Lidar can explore the unknown environment and construct a 3D map making the proposed method applicable to any scenario. Moreover, compared to the UGV-UAV cooperative method, the proposed method is more robust in 3D environments. As shown in Fig. 12, we use an unmanned ground vehicle and drone carrying 3D Lidar Velodyne 16 respectively to model indoor scenes. As shown in the right image, only the bottom environmental information is shown on the map constructed by UGV. In contrast, the map built by UAV is a complete 3D map with all the necessary information on facilities and obstacles.

For the second question, to quantitatively evaluate our system output, we calculate the mean absolute error (MAE), root means square error (RMSE), and mean absolute percentage error (MAPE) of the reconstructed model in the three experiments. As shown in Fig. 13, we put some markers on the facility and measure the distance between each marker. Finally, we get 45 measurement distances and compare them with the corresponding distance in the reconstructed model. The results illustrated in Table 3 show that the accuracy of the reconstruction model is within 3 cm, indicating the high quality of collected images and reconstruction.

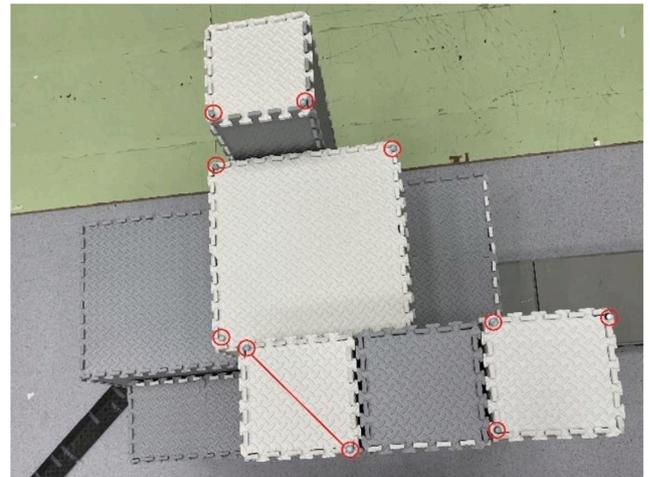


Fig. 13. The markers and measurement distances of the target facility.

Table 3
The quantitative analysis of reconstruction model.

Method	MAE (cm)	RMSE	MAPE
Experiment1	1.7111	1.9149	2.85%
Experiment2	2.0667	2.3898	3.78%
Experiment3	2.7333	3.1376	5.04%

6. Conclusion and future works

This paper proposes an explore-then-exploit system that can work in a cluttered environment for complex facility inspection and reconstruction data collection. The proposed system consists of the hardware setup and integration of each UAV, the SLAM system for the step of exploration and exploitation, the coverage path planning algorithm in a cluttered environment, and the trajectory generation algorithm for safe flight in a narrow environment. The hardware and software connection of each UAV is proposed. Lidar-based SLAM and Visual-based SLAM is used and integrated to provide robust and precise pose estimation in indoor inspection. The coverage path planning method that considers both the flight safety and coverage of the target facility is proposed and verified in different indoor environments. A trajectory generation algorithm is presented to show the UAV can follow the generated trajectory in a cluttered and narrow environment. To verify the proposed system can be used in indoor facility inspection data collection, we build up three scenarios and use the proposed system to collect data. The 3D reconstruction results in different experiments

Table 4
Comparison among different types of lidar.

Model	Product Image	Price	Parameters
Ouster OS0-32		\$4000	<ul style="list-style-type: none"> • Field of View (Vertical): 90° • Field of View (Horizontal): 360° • Measurement Range: 35 m • Range Accuracy: Up to ±5 cm • Vertical Resolution: 32 channels • Weight: ~447 g
Velodyne VLP-16		\$3600	<ul style="list-style-type: none"> • Field of View (Vertical): 30° • Field of View (Horizontal): 360° • Measurement Range: 100 m • Range Accuracy: Up to ±3 cm • Vertical Resolution: 16 channels • Weight: ~830 g
Livox MID-40		\$599	<ul style="list-style-type: none"> • Field of View (Circular): 38.4° • Measurement Range: 260 m • Range Accuracy: Up to ±2 cm • Vertical Resolution: N/A • Weight: ~760 g

Table 5
Comparison among different types of lidar.

Model	Product Image	Price	Parameters
RealSense D435i		\$345	<ul style="list-style-type: none"> • Depth Field of View (H × V): 87° × 58° • RGB sensor Field of View (H × V): 69° × 42° • Measurement Range: 0.3 m to 3 m • Depth Accuracy: Up to ±4 cm @ 2 m • Weight: ~75 g
RealSense D455		\$419	<ul style="list-style-type: none"> • Depth Field of View (H × V): 87° × 58° • RGB sensor Field of View (H × V): 90° × 65° • Measurement Range: 0.6 m to 6 m • Depth Accuracy: Up to ±8 cm @ 4 m • Weight: ~116 g
STEREOLABS ZED 2		\$449	<ul style="list-style-type: none"> • Depth Field of View (H × V): 110° × 70° • RGB sensor Field of View (H × V): 110° × 70° • Measurement Range: 0.2 m to 20 m • Depth Accuracy: Up to ±3 cm @ 3 m • Weight: ~124 g

show our system can collect high-quality data in a cluttered indoor environment safely.

This paper presents a new idea for indoor facility inspection which greatly reduces labor costs. Taking boiler inspection as an example, normally, it costs two people and one day to inspect the boiler interior, but with our system, it only takes one hour and one staff to complete the entire inspection task. With a centimeter-level SLAM system, UAVs can fly in a narrow space as small as 0.8 m width, and collect data at any location and angle, meaning that our system can also be widely used in indoor scenarios. Compared with 3D scanning equipment like FARO, which has higher accuracy and resolution, UAV is more flexible so that it can move to any location and build up a complete 3D map of the whole environment without human aid. Furthermore, deploying FARO in an indoor environment is very expensive. When the work area is difficult for humans to enter like the air ducts and boiler interior, UAV is a good choice for the exploration stage.

Compared with UGV, UAV cannot carry many devices and the duration of UAV is much shorter. In addition, with the consideration of safety and the dynamic constraints of UAVs, they cannot fly very fast in the environment. As a result, for large-scale indoor environments, like the entire power plant, the battery needs to be changed frequently, and the total inspection time will be extended. In the future, we will extend our system to a multi-agent collaborative system to save inspection time. New algorithms need to be developed, including but not limited to collaborative exploration, collaborative coverage path planning, and obstacle avoidance between UAVs.

CRediT authorship contribution statement

Chuanxiang Gao: Conceptualization, Investigation, Formal analysis, Writing – original draft. **Xinyi Wang:** Conceptualization, Investigation, Methodology, Formal analysis, Writing – original draft. **Ruoyu Wang:** Investigation, Formal analysis, Writing – original draft. **Zuoquan Zhao:** Investigation, Formal analysis, Writing – original draft. **Yu Zhai:** Investigation, Formal analysis, Writing – original draft. **Xi Chen:** Conceptualization, Resources, Supervision, Writing – review & editing, Project administration. **Ben M. Chen:** Conceptualization, Funding acquisition, Resources, Supervision, Writing – review & editing, Project administration.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Chuanxiang Gao, Xinyi Wang, Ruoyu Wang, Zuoquan Zhao, Yu Zhai, Xi Chen and Ben M. Chen reports financial support was provided by Hong Kong Centre for Logistics Robotics.

Data availability

Data will be made available on request.

Acknowledgments

This project is supported in part by the Research Grants Council of Hong Kong SAR (Grant No: 14209020 and Grant No: 14206821) and in part by the Hong Kong Centre for Logistics Robotics (HKCLR).

Appendix

There are a lot of new 3D lidars recently, like Ouster and Livox. They have the same or better performance than Velodyne. We have summarized the detailed parameters of three representative lidars in Table 4. For RGB sensors, there are also many alternatives such as D455 and ZED cameras. We have summarized the detailed parameters in Table 5.

References

- [1] E. Menendez, J.G. Victores, R. Montero, S. Martínez, C. Balaguer, Tunnel structural inspection and assessment using an autonomous robotic system, *Autom. Constr.* 87 (2018) 117–126, <http://dx.doi.org/10.1016/j.autcon.2017.12.001>.
- [2] J. Seo, L. Duque, J. Wacker, Drone-enabled bridge inspection methodology and application, *Autom. Constr.* 94 (2018) 112–126, <http://dx.doi.org/10.1016/j.autcon.2018.06.006>.
- [3] N. Bolourian, A. Hammad, Lidar-equipped UAV path planning considering potential locations of defects for bridge inspection, *Autom. Constr.* 117 (2020) 103250, <http://dx.doi.org/10.1016/j.autcon.2020.103250>.
- [4] R. Huang, Y. Xu, L. Hoegner, U. Stilla, Semantics-aided 3d change detection on construction sites using UAV-based photogrammetric point clouds, *Autom. Constr.* 134 (2022) 104057, <http://dx.doi.org/10.1016/j.autcon.2021.104057>.
- [5] A. Ortega, J. Reyes Muñoz, M. McGee, A.R. Choudhuri, A. Flores-Abad, Drone inspection flight path generation from 3d cad models: Power plant boiler case study, in: *AIAA Scitech 2020 Forum*, 2020, p. 1091, <http://dx.doi.org/10.2514/6.2020-1091>.
- [6] H. Ali, S. Batai, A. Akynov, Vision-based robot for boiler tube inspection, in: *International Conference on Computational Vision and Bio Inspired Computing*, Springer, 2018, pp. 475–482, http://dx.doi.org/10.1007/978-3-030-41862-5_45.
- [7] W. Jing, D. Deng, Y. Wu, K. Shimada, Multi-UAV coverage path planning for the inspection of large and complex structures, in: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, IEEE, <http://dx.doi.org/10.1109/iros45743.2020.9341089>.
- [8] M.D. Phung, C.H. Quach, T.H. Dinh, Q.P. Ha, Enhanced discrete particle swarm optimization path planning for UAV vision-based surface inspection, *Autom. Constr.* 81 (2017) 25–33, <http://dx.doi.org/10.1016/j.autcon.2017.04.013>.
- [9] K. Xie, H. Yang, S. Huang, D. Lischinski, M. Christie, K. Xu, M. Gong, D. Cohen-Or, H. Huang, Creating and chaining camera moves for quadrotor videography, *ACM Trans. Graph.* 37 (4) (2018) 1–13, <http://dx.doi.org/10.1145/3197517.3201284>.
- [10] C. Peng, V. Isler, Adaptive view planning for aerial 3d reconstruction, in: *2019 International Conference on Robotics and Automation, ICRA*, IEEE, <http://dx.doi.org/10.1109/icra.2019.8793532>.
- [11] Y. Tan, S. Li, H. Liu, P. Chen, Z. Zhou, Automatic inspection data collection of building surface based on BIM and UAV, *Autom. Constr.* 131 (2021) 103881, <http://dx.doi.org/10.1016/j.autcon.2021.103881>.
- [12] D. Hu, V.J. Gan, T. Wang, L. Ma, Multi-agent robotic system (mars) for UAV-UGV path planning and automatic sensory data collection in cluttered environments, *Build. Environ.* 221 (2022) 109349, <http://dx.doi.org/10.1016/j.buildenv.2022.109349>.
- [13] Y. Furukawa, B. Curless, S.M. Seitz, R. Szeliski, Towards internet-scale multi-view stereo, in: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 1434–1441, <http://dx.doi.org/10.1109/CVPR.2010.5539802>.
- [14] C. Cao, J. Zhang, M. Travers, H. Choset, Hierarchical coverage path planning in complex 3D environments, in: *2020 IEEE International Conference on Robotics and Automation*, 2020, pp. 3206–3212, <http://dx.doi.org/10.1109/ICRA40945.2020.9196575>.
- [15] M. Kuzmin, Classification and comparison of the existing SLAM methods for groups of robots, in: *2018 22nd Conference of Open Innovations Association, FRUCT*, IEEE, 2018, pp. 115–120, <http://dx.doi.org/10.23919/FRUCT.2018.8468281>.
- [16] M.U. Khan, S.A.A. Zaidi, A. Ishtiaq, S.U.R. Bukhari, S. Samer, A. Farman, A comparative survey of lidar-SLAM and lidar based sensor technologies, in: *2021 Mohammad Ali Jinnah University International Conference on Computing, MAJICC*, 2021, pp. 1–8, <http://dx.doi.org/10.1109/MAJICC53071.2021.9526266>.
- [17] J. Zhang, S. Singh, Loam: Lidar odometry and mapping in real-time, in: *Robotics: Science and Systems*, Vol. 2, Berkeley, CA, 2014, pp. 1–9, <http://dx.doi.org/10.15607/RSS.2014.X.007>.
- [18] T. Shan, B. Englot, Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain, in: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, 2018, pp. 4758–4765, <http://dx.doi.org/10.1109/IROS.2018.8594299>.
- [19] J. Zhang, S. Singh, Laser-visual-inertial odometry and mapping with high robustness and low drift, *J. Field Robotics* 35 (8) (2018) 1242–1264, <http://dx.doi.org/10.1002/rob.21809>.
- [20] J. Zhang, S. Singh, Visual-lidar odometry and mapping: Low-drift, robust, and fast, in: *2015 IEEE International Conference on Robotics and Automation, ICRA*, IEEE, 2015, pp. 2174–2181, <http://dx.doi.org/10.1109/ICRA.2015.7139486>.
- [21] W. Xu, Y. Cai, D. He, J. Lin, F. Zhang, Fast-lio2: Fast direct lidar-inertial odometry, *IEEE Trans. Robot.* (2022) 1–21, <http://dx.doi.org/10.1109/TRO.2022.3141876>.
- [22] H. Wang, C. Wang, C.-L. Chen, L. Xie, F-loam : Fast lidar odometry and mapping, in: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, 2021, pp. 4390–4396, <http://dx.doi.org/10.1109/IROS51168.2021.9636655>.
- [23] A. Macario Barros, M. Michel, Y. Moline, G. Corre, F. Carrel, A comprehensive survey of visual SLAM algorithms, *Robotics* 11 (1) <http://dx.doi.org/10.3390/robotics11010024>.
- [24] A.J. Davison, I.D. Reid, N.D. Molton, O. Stasse, Monoslam: Real-time single camera SLAM, *IEEE Trans. Pattern Anal. Mach. Intell.* 29 (6) (2007) 1052–1067, <http://dx.doi.org/10.1109/TPAMI.2007.1049>.
- [25] C. Forster, M. Pizzoli, D. Scaramuzza, SVO: Fast semi-direct monocular visual odometry, in: *2014 IEEE International Conference on Robotics and Automation, ICRA*, 2014, pp. 15–22, <http://dx.doi.org/10.1109/ICRA.2014.6906584>.
- [26] R. Mur-Artal, J.M.M. Montiel, J.D. Tardós, ORB-SLAM: A versatile and accurate monocular SLAM system, *IEEE Trans. Robot.* 31 (5) (2015) 1147–1163, <http://dx.doi.org/10.1109/TRO.2015.2463671>.
- [27] R. Mur-Artal, J.D. Tardós, Orb-slam2: An open-source SLAM system for monocular, Stereo, and RGB-D cameras, *IEEE Trans. Robot.* 33 (5) (2017) 1255–1262, <http://dx.doi.org/10.1109/TRO.2017.2705103>.
- [28] C. Campos, R. Elvira, J.J.G. Rodríguez, J.M.M. Montiel, J.D. Tardós, Orb-slam3: An accurate open-source library for visual, visual-inertial, and multimap SLAM, *IEEE Trans. Robot.* 37 (6) (2021) 1874–1890, <http://dx.doi.org/10.1109/TRO.2021.3075644>.
- [29] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, P. Furgale, Keyframe-based visual-inertial odometry using nonlinear optimization, *Int. J. Robot. Res.* 34 (3) (2015) 314–334, <http://dx.doi.org/10.1177/0278364914554813>.
- [30] T. Qin, P. Li, S. Shen, Vins-mono: A robust and versatile monocular visual-inertial state estimator, *IEEE Trans. Robot.* 34 (4) (2018) 1004–1020, <http://dx.doi.org/10.1109/TRO.2018.2853729>.
- [31] P. Kim, B. Coltin, H.J. Kim, Linear RGB-D SLAM for planar environments, in: *Proceedings of the European Conference on Computer Vision, ECCV*, 2018, pp. 333–348, http://dx.doi.org/10.1007/978-3-030-01225-0_21.
- [32] T. Schops, T. Sattler, M. Pollefeys, BAD SLAM: Bundle adjusted direct rgb-d slam, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, 2019, <http://dx.doi.org/10.1109/CVPR.2019.00022>.
- [33] M. Hsiao, E. Westman, M. Kaess, Dense planar-inertial SLAM with structural constraints, in: *2018 IEEE International Conference on Robotics and Automation, ICRA*, 2018, pp. 6521–6528, <http://dx.doi.org/10.1109/ICRA.2018.8461094>.
- [34] A. Bouras, Y. Bouzid, M. Guiatni, Multi-UAV coverage path planning for gas distribution map applications, *Unmanned Syst.* 10 (03) (2022) 289–306, <http://dx.doi.org/10.1142/S2301385022500170>.
- [35] K. Schmid, H. Hirschmüller, A. Dömel, I. Grixia, M. Suppa, G. Hirzinger, View planning for multi-view stereo 3D reconstruction using an autonomous multicopter, *J. Intell. Robot. Syst.* 65 (1) (2012) 309–323, <http://dx.doi.org/10.1007/s10846-011-9576-2>.
- [36] Y. Liu, R. Cui, K. Xie, M. Gong, H. Huang, Aerial path planning for online real-time exploration and offline high-quality reconstruction of large-scale urban scenes, *ACM Trans. Graph.* 40 (6) (2021) 1–16, <http://dx.doi.org/10.1145/3478513.3480491>.
- [37] H. Zhang, Y. Yao, K. Xie, C.-W. Fu, H. Zhang, H. Huang, Continuous aerial path planning for 3D urban scene reconstruction, *ACM Trans. Graph.* 40 (6) (2021) 1–15, <http://dx.doi.org/10.1145/3478513.3480483>.
- [38] Q. Kuang, J. Wu, J. Pan, B. Zhou, Real-time UAV path planning for autonomous urban scene reconstruction, in: *2020 IEEE International Conference on Robotics and Automation, ICRA*, IEEE, 2020, pp. 1156–1162, <http://dx.doi.org/10.1109/ICRA40945.2020.9196558>.
- [39] W. Jing, D. Deng, Y. Wu, K. Shimada, Multi-UAV coverage path planning for the inspection of large and complex structures, in: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, IEEE, 2020, pp. 1480–1486, <http://dx.doi.org/10.1109/IROS45743.2020.9341089>.
- [40] F. Chen, Y. Lu, B. Cai, X. Xie, Multi-drone collaborative trajectory optimization for large-scale aerial 3D scanning, in: *2021 IEEE International Symposium on Mixed and Augmented Reality Adjunct, ISMAR-Adjunct*, IEEE, 2021, pp. 121–126, <http://dx.doi.org/10.1109/ISMAR-Adjunct54149.2021.00034>.
- [41] R. Nagasawa, E. Mas, L. Moya, S. Koshimura, Model-based analysis of multi-UAV path planning for surveying postdisaster building damage, *Sci. Rep.* 11 (1) (2021) 1–14, <http://dx.doi.org/10.1038/s41598-021-97804-4>.

- [42] X. Zheng, F. Wang, Z. Li, A multi-UAV cooperative route planning methodology for 3D fine-resolution building model reconstruction, *ISPRS J. Photogramm. Remote Sens.* 146 (2018) 483–494, <http://dx.doi.org/10.1016/j.isprsjprs.2018.11.004>.
- [43] M. Roberts, D. Dey, A. Truong, S. Sinha, S. Shah, A. Kapoor, P. Hanrahan, N. Joshi, Submodular trajectory optimization for aerial 3D scanning, in: *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5324–5333, <http://dx.doi.org/10.1109/ICCV.2017.569>.
- [44] L. Xi, X. Wang, L. Jiao, S. Lai, Z. Peng, B.M. Chen, GTO-MPC-based target chasing using a quadrotor in cluttered environments, *IEEE Trans. Ind. Electron.* 69 (6) (2021) 6026–6035, <http://dx.doi.org/10.1109/TIE.2021.3090700>.
- [45] D. Mellinger, V. Kumar, Minimum snap trajectory generation and control for quadrotors, in: *2011 IEEE International Conference on Robotics and Automation*, IEEE, 2011, pp. 2520–2525, <http://dx.doi.org/10.1109/ICRA.2011.5980409>.
- [46] C. Richter, A. Bry, N. Roy, Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments, in: *Robotics Research*, Springer, 2016, pp. 649–666, http://dx.doi.org/10.1007/978-3-319-28872-7_37.
- [47] S. Lai, M. Lan, B.M. Chen, Model predictive local motion planning with boundary state constrained primitives, *IEEE Robot. Autom. Lett.* 4 (4) (2019) 3577–3584, <http://dx.doi.org/10.1109/LRA.2019.2928255>.
- [48] Y. Zhou, S. Lai, H. Cheng, A.H.M. Redhwan, P. Wang, J. Zhu, Z. Gao, Z. Ma, Y. Bi, F. Lin, et al., Toward autonomy of micro aerial vehicles in unknown and global positioning system denied environments, *IEEE Trans. Ind. Electron.* 68 (8) (2020) 7642–7651, <http://dx.doi.org/10.1109/TIE.2020.3008378>.
- [49] P.D. Petris, H. Nguyen, M. Dharmadhikari, M. Kulkarni, N. Khedekar, F. Mascari, K. Alexis, Rmf-owl: A collision-tolerant flying robot for autonomous subterranean exploration, in: *2022 International Conference on Unmanned Aircraft Systems, ICUAS*, 2022, pp. 536–543, <http://dx.doi.org/10.1109/ICUAS54217.2022.9836115>.
- [50] Y. Chen, S. Lai, J. Cui, B. Wang, B.M. Chen, GPU-accelerated incremental euclidean distance transform for online motion planning of mobile robots, *IEEE Robot. Autom. Lett.* <http://dx.doi.org/10.1109/LRA.2022.3177852>.
- [51] R. Haschke, E. Weitnauer, H. Ritter, On-line planning of time-optimal, jerk-limited trajectories, in: *2008 IEEE International Conference on Intelligent Robots and Systems*, 2008, pp. 3248–3253, <http://dx.doi.org/10.1109/IROS.2008.4650924>.
- [52] S. Lai, M. Lan, Y. Li, B.M. Chen, Safe navigation of quadrotors with jerk limited trajectory, *Front. Inform. Technol. Electron. Eng.* 20 (1) (2019) 107–119, <http://dx.doi.org/10.1631/FITEE.1800719>.
- [53] V.T. Hoang, M.D. Phung, T.H. Dinh, Q.P. Ha, QAngle-encoded swarm optimization for UAV formation path planning, in: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, pp. 5239–5244, <http://dx.doi.org/10.1109/IROS.2018.8593930>.
- [54] M. Hehn, R. D'Andrea, Quadrocopter trajectory generation and control, *IFAC Proc. Vol.* 44 (1) (2011) 1485–1491, <http://dx.doi.org/10.3182/20110828-6-IT-1002.03178>.