2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Online Whole-body Motion Planning for Quadrotor using Multi-resolution Search

Yunfan Ren*, Siqi Liang*, Fangcheng Zhu, Guozheng Lu, and Fu Zhang

***Abstract*— In this paper, we address the problem of online quadrotor whole-body motion planning (SE(3) planning) in unknown and unstructured environments. We propose a novel multi-resolution search method, which discovers narrow areas requiring full pose planning and normal areas requiring only position planning. As a consequence, a quadrotor planning problem is decomposed into several SE(3) (if necessary) and $\mathbb{R}^3$ sub-problems. To fly through the discovered narrow areas, a carefully designed corridor generation strategy for narrow areas is proposed, which significantly increases the planning success rate. The overall problem decomposition and hierarchical planning framework substantially accelerate the planning process, making it possible to work online with fully onboard sensing and computation in unknown environments. Extensive simulation benchmark comparisons show that the proposed method is one to several orders of magnitude faster than the state-of-the-art methods in computation time while maintaining high planning success rate. The proposed method is finally integrated into a LiDAR-based autonomous quadrotor, and various real-world experiments in unknown and unstructured environments are conducted to demonstrate the outstanding performance of the proposed method.**

I. Introduction

Quadrotors with a large thrust-to-weight ratio can perform extremely aggressive maneuvers, enabling applications like searching and rescuing in highly complex unstructured environments. With the whole-body motion planning (SE(3) planning), which simultaneously considers the position and attitude of a drone, quadrotors can fly through areas that are smaller than the robot's body in normal flights (*e.g.*, narrow gap shown in Fig. 1).

Traditional motion planning for quadrotor like [1]–[5] simply ignore the shape and orientation of the drone, by planning a position trajectory ($\mathbb{R}^3$ planning). Although $\mathbb{R}^3$ planning is computationally efficient, they are over-conservative since they prohibits some trajectories which are really feasible if the drone is at a certain attitude.

Some existing works consider the drone's attitude (SE(3) planning). However, two main challenges remain to be addressed: **(1) Planning in unknown and unstructured environments**: some existing works either make a strong assumption about the environment (*e.g.*, [6]–[10] assume the shape of small gaps is known or have specific visual feature) or have a low success rate in unknown, cluttered environment [11], which are not suitable for real-world applications. **(2) Computational efficiency**: Although some methods like [11], [12] can be modified to accommodate online planning, they take hundreds of milliseconds to a few seconds to compute, which limit their application in real-world online quadrotor navigation.

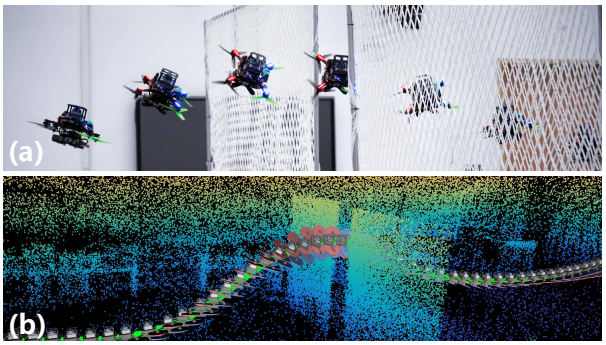

Fig. 1. Performing an aggressive maneuver to fly through a narrow gap. (a) The composite image of one real-world test. (b) The point cloud view of the same flight. The trajectory is planned online without any prior information of the environment, including the narrow gap. All sensing, localization, planning, and control are running on the onboard computer in real-time.

To address above challenges, in this paper, we propose a multi-resolution search method, which actively discovers narrow areas requiring certain drone attitude and large areas that impose no such requirement. Distinguishing narrow areas from large areas allows to decouple SE(3) planning, which is very time-consuming, from normal $\mathbb{R}^3$ planning, which is much more efficient, thus reducing the overall computation time significantly. Moreover, the awareness of narrow areas allows more dedicated planning around that area, which also improves the success rate. Requiring no prior information (*e.g.*, position, pose, shape) of the narrow area, our method is applicable to general unknown and unstructured environments. To sum up, the contributions of this paper are listed below:

1) We propose a novel parallel multi-resolution search (MR-search) method which actively discovers narrow areas requiring certain drone attitude and large areas that impose no such requirement.
2) Based on the multi-resolution search, we propose a

*These two authors contributed equally to this work.

Y. Ren, F. Zhu, G. Lu and F. Zhang are with the Department of Mechanical Engineering, University of Hong Kong {renyf, zhufc}@connect.hku.hk, {gzlu, fuzhang}@hku.hk, S. Liang is with School of Mechanical Engineering and Automation, Harbin Institute of Technology sqliang@stu.hit.edu.cn.

2023 IEEE International Conference on Robotics and Automation (ICRA) | 979-8-3503-2365-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICRA48891.2023.10160767

hierarchical planning framework that separates the SE(3) planning from the $\mathbb{R}^3$ planning.

3) We present a compact LiDAR-based quadrotor that is fully autonomous and integrated onboard sensing, localization, planning, and control modules. To the best of our knowledge, this is the first time a fully autonomous quadrotor can perform online whole-body motion planning and navigation in unknown and unstructured environments.
4) Various simulation benchmarks comparisons show the advantages of the proposed method in terms of computation efficiency and success rate over the state-of-the-art baselines. Furthermore, extensive real-world tests show the proposed method's outstanding real-world performance.

II. Related Works

Collision avoidance is a key component of robot motion planning. The mainstream planning approaches modeled the robot as a sphere [1]–[3], [13]–[16]. In this way, the configuration space (C-space) can be easily obtained by inflating the obstacles by the largest axis length of the robot and then simplifying the drone into a mass-point and planning in the C-space (i.e., $\mathbb{R}^3$ planning). However, this leads to conservatism by prohibiting some trajectories that are feasible if the attitude is right. Planning both the robot's attitude and position is commonly referred to as whole-body planning, or SE(3) planning, which is necessary to navigate a robot in narrow, cluttered environments [12].

One problem of whole-body motion planning is distinguishing narrow areas requiring full pose planning. [8], [10] focus more on the planning and control problem while the pose and shape of the narrow gap in the environment are priorly known. Falanga *et al*. [17] use an onboard fish-eye camera to detect the narrow gap. However, it requires the gap to have specific visual features (*e.g*., black and white lines) and pre-known shapes; hence cannot apply to general unstructured environments. Our proposed method uses a novel multi-resolution search strategy to distinguish narrow areas requiring SE(3) planning from normal areas. The proposed method does not require any instrumentation of the environment nor the specific shape of the narrow area, thus applicable to general unstructured environments.

Another problem is the trajectory generation. Liu *et al*. [12] proposed a search-based method for quadrotor SE(3) planning. They generate a set of motion primitives and perform a graph search. To select a feasible primitive, they model the drone to an ellipsoid, then perform a collision check by verifying if any ellipsoid on the trajectory intersects with obstacles. While requiring no instrumentation of the environment or specific shape of the narrow area similar to our method, this approach cannot guarantee finding a feasible solution due to the limited discretization resolution of the motion primitive library. Also, it takes several seconds to find a trajectory, preventing this approach from an online (re-)planning framework. [9]–[11] follow a two-step pipeline: they first perform convex decomposition to represent the free space by a series of convex polyhedrons. Then take the polyhedrons as the geometry constraints of the optimization problem to ensure whole-body collision avoidance. Those methods are one to several orders of magnitude faster than [12] in computation time. However, they still need hundreds of milliseconds (considering the computation time of both steps) to generate a trajectory due to the large number of constraints brought by considering the robot's attitude. Moreover, without distinguishing the narrow areas, these methods' success rates are not satisfactory, as they often generate infeasible corridors in narrow areas. In contrast, our proposed method actively discovers narrow areas and thus can perform a dedicatedly designed corridor generation strategy in narrow areas, significantly increasing the planning success rate. Furthermore, the active discovery of narrow areas makes it possible to decompose of the time-consuming SE(3) planning from normal $\mathbb{R}^3$ planning without planning the entire trajectory in SE(3) [9]–[11]. This hierarchical planning strategy significantly reduces the computation time. The high success rate and high computational efficiency enable the proposed method to perform online quadrotor whole-body planning in unknown and unstructured environments.

III. Preliminaries

A. System Modeling and Polynomial Trajectory

A quadrotor system is proved to be differential flat [18] with the flat output $\sigma = [x, y, z, \psi]^T$ where $\mathbf{p} = [x, y, z]^T$ is the position of the quadrotor in the world frame and ψ the yaw angle. Since the quadrotor yaw is decoupled, we specify the yaw angle trajectory $\Phi(t)$ as the tangent direction of $\mathbf{p}(t)$ such that the quadrotor is always facing forward during a flight. In this way, we only need to plan the position trajectory $\mathbf{p}(t)$. Define the quadrotor's state as: $\mathbf{s} = [\mathbf{p}, \mathbf{v}, \mathbf{a}, \mathbf{j}] \in \mathbb{R}^{12}$, where $\mathbf{v}, \mathbf{a}, \mathbf{j}$ are the associated velocity, acceleration and jerk, respectively. And the kinodynamic constraints includes: $\|\mathbf{v}(t)\| \leq v_{\max}, \|\mathbf{a}(t)\| \leq a_{\max}, \|\mathbf{j}(t)\| \leq j_{\max}$.

Following [10], we adopt piece-wise polynomials to represent the trajectory:

$$\mathbf{p}(t) = \begin{cases} \mathbf{p}_1(t) = \mathbf{c}_1^T \beta(t-0) & 0 \leq t < T_1 \\ \vdots & \vdots \\ \mathbf{p}_M(t) = \mathbf{c}_M^T \beta(t - T_{M-1}) & T_{M-1} \leq t < T_M \end{cases} \tag{1}$$

where $\mathbf{c}_1, \ldots, \mathbf{c}_M \in \mathbb{R}^{2s \times 3}$ is the coefficient matrix of the M pieces, and $\beta(t) = [1, t, t^2, \ldots, t^{2s-1}]^T$ is the time basis vector. The state of the quadrotor can be easily obtained by taking derivative of the polynomial trajectory. In this paper, we use $s = 4$ to ensure the continuity up to jerk in adjacent pieces.

B. Safety Constraints

To avoid collisions, we use safe flight corridors (SFC) (which is a series of convex polyhedrons representing the free space) as the explicit spatial constraints. Each polyhedron is described by its $\mathcal{H}$-representation [19]:

$$\mathcal{P} = \{\mathbf{x} \in \mathbb{R}^3 | \mathbf{A}\mathbf{x} \preceq \mathbf{b}\} \tag{2}$$

Each piece of a polynomial is constrained in one polyhedron. For different planning missions, we introduce two types of corridor constraints.

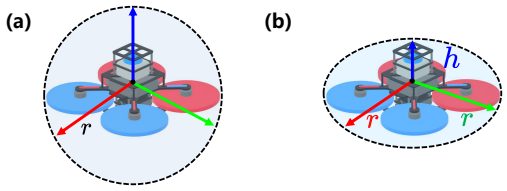

Fig. 2. (a) The quadrotor is modeled as a sphere with radius equals to the largest axis length r in $\mathbb{R}^3$ planning. (b) The quadrotor is modeled as an ellipsoid in SE(3) planning.

1) Corridor constraints for $\mathbb{R}^3$ planning: For $\mathbb{R}^3$ planning, we model it to a sphere with a radius equal to its largest axis length r (see Fig. 2(a)). Then, we can get the configuration space by inflating all obstacle points with r. Finally, the corridors are generated in the configuration space. In this way, the corridor constraints for the i-th piece can be written as:

$$\mathbf{A}_i\mathbf{p}(t) - \mathbf{b}_i \preceq \mathbf{0}, \forall t \in [T_{i-1}, T_i] \tag{3}$$

2) Corridor constraints for SE(3) planning: For SE(3) planning, We model the geometrical shape of the quadrotor as its outer John ellipsoid (see Fig. 2(b)) following [10]:

$$\mathcal{E}(t) = \{\mathbf{R}(t)\mathbf{Q}\mathbf{x} + \mathbf{p}(t) \mid \|\mathbf{x}\|_2 \leq 1\}, \tag{4}$$

where $(\mathbf{R}(t), \mathbf{p}(t))$ is the drone pose computed from the state $\mathbf{s}(t)$ based on the quadrotor differential flatness [18] and $\mathbf{Q} = \mathbf{diag}(r, r, h)$ is the shape matrix where r is the radius of the drone and h is half the height of the UAV.

As described in [10], the ellipsoid along the i-th piece polynomial trajectory $\mathbf{p}_i$ inside the i-th polyhedron $\mathcal{P}_i$ can be expressed as

$$\begin{aligned} &\left[\left[\mathbf{A}_i\mathbf{R}(t)\mathbf{Q}\right]^2 \mathbf{1}\right]^{\frac{1}{2}} + \mathbf{A}_i\mathbf{p}_i(t) - \mathbf{b}_i \preceq \mathbf{0}, \\ &\forall t \in [T_{i-1}, T_i], \end{aligned} \tag{5}$$

where $\mathbf{1}$ is an all-ones vector with an appropriate length, $[\cdot]^2$ and $[\cdot]^{\frac{1}{2}}$ are entry-wise square and square root, respectively.

C. Trajectory Generation

In this paper, the $\mathbb{R}^3$ trajectory generation problem is described as: for a given start state $\mathbf{s}_s = [\mathbf{p}_s, \mathbf{v}_s, \mathbf{a}_s, \mathbf{j}_s]$ and goal state $\mathbf{s}_g = [\mathbf{p}_g, \mathbf{v}_g, \mathbf{a}_g, \mathbf{j}_g]$, generates a trajectory that is collision-free and satisfies all kinodynamic constraints. The collision-free condition is described in (3). Since quadrotor's $\mathbb{R}^3$ planning is well studied, our $\mathbb{R}^3$ trajectory generation method is a combination of [20] and [10]. We first generate a series of SFC $\mathcal{S} = \{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_N\}$ using RILS [20], then perform MINCO trajectory optimization [10] with the SFC constraints. We called this process **GenerateR3Trajectory($\mathbf{s}_s$, $\mathbf{s}_g$)**, which will be used in the sequel.

For the SE(3) trajectory generation, we modeled the quadrotor to an ellipsoid as mentioned in Sec. III-B.2. Our SE(3) trajectory optimization is based on [10], which generates a trajectory satisfying all kinodynamic constraints and safety constraints (5) within a given SFC $\mathcal{S}$. The SE(3) trajectory generation is encapsulated as **GenerateSE3Trajectory($\mathcal{S}$)**.

IV. Planner

The overview of the proposed method is shown in Fig. 3, including: **1)** Colliding segments extraction for sub-problem generation (Sec. IV-A); **2)** Parallel multi-resolution search for planning problem decomposition (Sec. IV-B); **3)** SFC generation and trajectory planning for SE(3) sub-problems. (Sec. IV-C); **4)** $\mathbb{R}^3$ planning and trajectory stitching (Sec. IV-D).

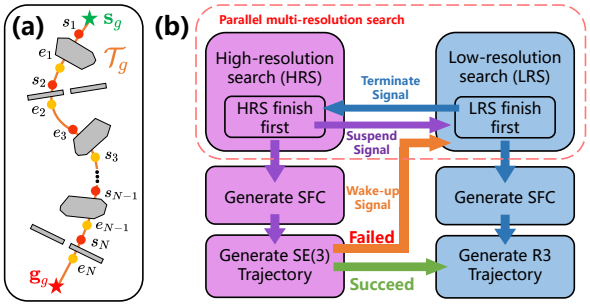

Fig. 3. The overview of our proposed method.

A. Colliding Segments Extraction

In the first step, we generate a global optimal trajectory $\mathcal{T}_g$ to connect the global start state $\mathbf{s}_g$ and the global goal state $\mathbf{g}_g$ (see orange curve in Fig. 3(a)) without consider any obstacles by solving the *Linear Quadratic Minimum Time* (LQMT) problem following [21]. Then a collision check process is performed to extract N pairs of colliding segments $\mathbf{S} = \{\mathbf{S}_i = (s_i, e_i) | i = 1, 2, \ldots, N\}$, where s_i is the start point and e_i is the end point (see red and yellow points in Fig. 3(a)).

B. Parallel Multi-resolution Search

The key of our multi-resolution search is a dual-resolution map respectively called *low-resolution map* (LRM) and *high-resolution map* (HRM). The LRM's

resolution is equal to the length of the quadrotor's largest axis length (i.e., r), while the HRM's is equal to the smallest axis length (i.e., h), both maps are inflated by one grid to represent the configuration space. As a result, a path in the free space of the LRM is guaranteed to be collision-free regardless of the quadrotor attitude, and a path in the free space of the HRM requires a certain attitude of the quadrotor.

As shown in Fig. 3(b), for the segment $\mathbf{S}_i$, to determine whether it requires SE(3) planning (*e.g.*, $\mathbf{S}_2$) or normal $\mathbb{R}^3$ planning (*e.g.*, $\mathbf{S}_1$), we perform two parallel A* search, one on the high-resolution map (i.e., the high-resolution search (HRS)), and the other on the low-resolution map (i.e., low-resolution search (LRS)), both from s_i to e_i. If the LRS completes first, this segment is marked as a $\mathbb{R}^3$ sub-problem, and the HRS is terminated. Otherwise, if the HRS completes first, this segment is marked as a candidate SE(3) sub-problem, and the LRS is suspended and may be waken up later if the candidate SE(3) sub-problem does not has a feasible trajectory.

C. SE(3) Trajectory Generation

If a segment $\mathbf{S}_i$ has been marked as a candidate SE(3) sub-problem with searched A* path $\mathcal{Q}$ connecting s_i to e_i, we proceed to plan a SE(3) trajectory around the A* path. To do so, we adopt the framework in [10], which is encapsulated as **GenerateSE3Trajectory($\mathcal{S}$)** as explained in Sec. III-C. A prerequisite for the algorithm is SFC $\mathcal{S}$, which consists of a series of convex polyhedrons, connecting the starting s_i and end point e_i of $\mathbf{S}_i$. However, generating convex polyhedrons around narrow areas like a small gap is not easy, existing methods like [9], [11], [20] that generates SFC directly from $\mathcal{Q}$ often lead to small overlaps between two adjacent corridors. As shown in Fig. 4(a,b), the overlapped part is too narrow to accommodate a quadrotor, making the trajectory optimization infeasible.

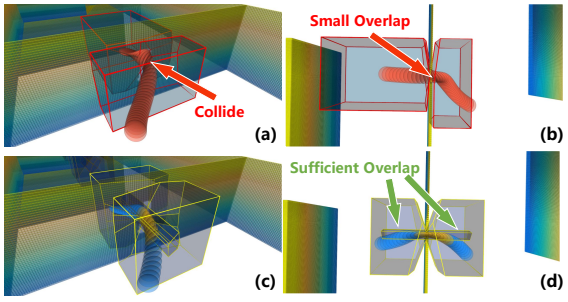

Fig. 4. The generated corridor comparison. (a,b) The corridor generated by [11] has two polyhedrons whose overlapped part cannot accommodate a quadrotor, making the optimization problem infeasible (the shown trajectory collides with the gap). (c,d) The corridor generated by the proposed method has three overlapped polyhedrons, and the shape is close to the actual free space.

To address this problem, we present a simple seed generation method shown in Fig. 5. The goal is to

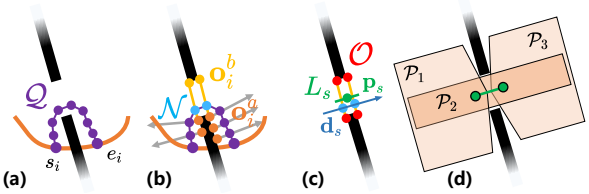

Fig. 5. The proposed line seed generation processes.

generate a line seed denoted by $L_s(\mathbf{d}_s, \mathbf{p}_s, l_s)$, where $\mathbf{d}_s$ is the direction, $\mathbf{p}_s$ is the midpoint and l_s is the length of the line segment. To do so, we leverage the path $\mathcal{Q}$ generated by the high-resolution search (see purple points in Fig. 5(a)) and the high resolution map (HRM) without any inflation. For each waypoint $\mathbf{q}_i \in \mathcal{Q}$, we find its nearest occupied grid on the HRM $\mathbf{o}_i^a$ (see orange points in Fig. 5(b)). Then, we search the first occupied grid on the HRM $\mathbf{o}_i^b$ (see yellow points in Fig. 5(b)) along the direction of $\mathbf{q}_i - \mathbf{o}_i^a$ (see gray arrows in Fig. 5(b)) and within distance $d_c = 2r$ from $\mathbf{q}_i$. If such point $\mathbf{o}_i^b$ is found, $\mathbf{o}_i^b$ and $\mathbf{o}_i^a$ are added to $\mathcal{O}$, the set of points on the narrow area obstacle (see red points in Fig. 5(c)), and $\mathbf{q}_i$ is added to $\mathcal{N}$, the set of path waypoint passing through the narrow area (see blue points in Fig. 5(b)). If no such point $\mathbf{o}_i^b$ is found, we proceed to the next point on $\mathcal{Q}$. After this process, the line direction $\mathbf{d}_s$ is the average direction of $\mathcal{N}$ (see blue arrow in Fig. 5(c)) and the midpoint $\mathbf{p}_s$ is center of $\mathcal{O}$ (see the green point in Fig. 5(c)). The length of the line segment l_s is set to the length of $\mathcal{N}$ when projected to $\mathbf{d}_s$. The found line seed L_s is depicted by the green line in Fig. 5(c).

With the found line seed $L_s(\mathbf{d}_s, \mathbf{p}_s, l_s)$, three high-quality corridors $\mathcal{S} = \{\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3\}$ can be generated by RILS [20] on the HRM without any inflation as shown in Fig. 5(d). The $\mathcal{P}_1$ and $\mathcal{P}_3$ are respectively generated by taking the two endpoints of the L_s as point seeds, and $\mathcal{P}_2$ passing through the small gap is generated by taking L_s as a line seed. Then we perform trajectory optimization with **GenerateSE3Trajectory($\mathcal{S}$)** as described in Sec. III-C. If the optimization is successful (which means all safety and dynamic constraints are satisfied), this segment will be marked as an SE(3) segment; If not, this segment is marked as a $\mathbb{R}^3$ sub-problem, and the LRS is waken up, which will continue to finding a detour path in LRM.

D. $\mathbb{R}^3$ Trajectory Generation

With the steps mentioned above, we have a set of K pieces of SE(3) trajectories with start states $\mathbf{s}_1^{\text{wb}}, \mathbf{s}_2^{\text{wb}}, \ldots, \mathbf{s}_K^{\text{wb}}$ and goal states $\mathbf{g}_1^{\text{wb}}, \mathbf{g}_2^{\text{wb}}, \ldots, \mathbf{g}_K^{\text{wb}}$, respectively. Between the global start state $\mathbf{s}_g$ and the first SE(3) trajectory, two consecutive SE(3) trajectories, or the last SE(3) trajectory and the global goal state $\mathbf{g}_g$, there are $\mathbb{R}^3$ path searched from the LRS (yellow dotted line in Fig. 6) and collision-free trajectory segments from $\mathcal{T}_g$ (orange curve in Fig. 6), we use them as the seed

to produce flight corridors on the inflated low-resolution map (LRM) and plan a smooth trajectory within the corridor. The trajectory generation process is encapsulated by **GenerateR3Trajectory**($\mathbf{s}, \mathbf{g}$) as explained in Sec. III-C, where $\mathbf{s}$ is the global start state or the end state of the preceding SE(3) trajectory and $\mathbf{g}$ is the global goal state or the start state of the succeeding SE(3) trajectory.

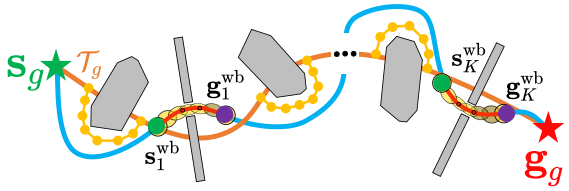

Fig. 6. The blue curves represent the $\mathbb{R}^3$ trajectories and the red curves represent the SE(3) trajectory. The start and end states of the $\mathbb{R}^3$ trajectory are restricted to be the same as the adjacent SE(3) trajectories, ensuring the continuity up to jerk of the final trajectory

V. Experiments

A. Benchmark Comparison

In this section, we compare the proposed method with a search-based SE(3) planning method [12] (*Liu et al.*), and a corridor optimization-based method [11] (*Han et al.*). Note that for *Liu et al.*'s [12] method, it generates a set of motion primitives and performs a graph search on it. However, it heavily suffers from the curse of dimensionality. Only a search on the x-y plane with a fixed height (i.e., 2D search) can find a feasible solution in all of the test environments below. So we manually adjust the height of the searching plane to fit the start position, goal position, and narrow gaps.

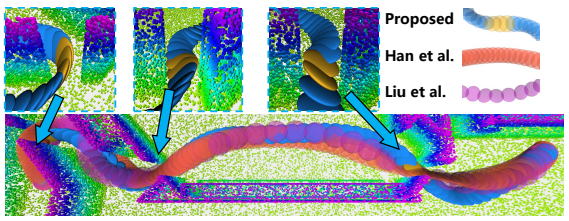

Fig. 7. The generated trajectory of the three methods in **Office**.

We make the comparison in three different simulation environments, including: **1) Office**, which is open sourced[1] from Liu *et al.* [12]. **2) Zhangjiajie**, which is open sourced[2] from Han *et al.* [11]. **(3) Maze**: our custom environment. In each environment, the dynamic constraints are set to the same for all approaches.

Since environment 1) and 2) are both static, the generated trajectory of the three methods are all the same in different trials. The generated trajectories are shown

[1]https://github.com/sikang/mpl_ros
[2]https://github.com/ZJU-FAST-Lab/Fast-Racing

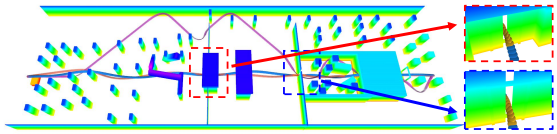

Fig. 8. The generated trajectory of the three methods in **Zhangjiajie**.

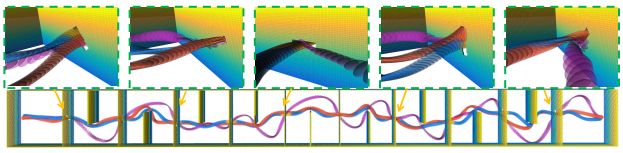

Fig. 9. The generated trajectory of the three methods in **Maze**.

in Fig. 7 and Fig. 8. The computation time t_c, trajectory tracking execution time t_e, and the total trajectory length l, are shown in Table. I with bold font indicating better physical meaning. The proposed method can generate trajectories with similar quality, but takes substantially less computation time.

Our customized environment randomly generates narrow gaps on the walls with different random seeds. We set different start and goal positions making the trajectory pass through one to ten narrow gaps. For each number of walls and gaps, we tested ten times with ten different random seeds and counted the success rate. A planning is successful only if it connects the start and goal points while satisfying all dynamic constraints and ensuring collision-free. The generated trajectories are shown in Fig. 9. And Fig. 10 shows the computation time and success rate. As can be seen, [10] has a low success rate due to the reason explained in Fig. 4. [12] achieves a high success rate due to exhaustive search in a fine-grade motion library, but requires tremendously high computation time. Our proposed method is several

TABLE I
Run Time Comparison

	Office(20.5 m)			Zhangjiajie(158 m)		
	t_c(s)	l (m)	t_e(s)	t_c(s)	l (m)	t_e(s)
Liu *et al.*	1.683	**24.80**	5.800	38.210	174.234	14.400
Han *et al.*	0.171	24.930	**5.364**	3.433	**159.065**	**13.845**
Proposed	**0.0874**	25.52	5.985	**0.179**	159.863	14.267

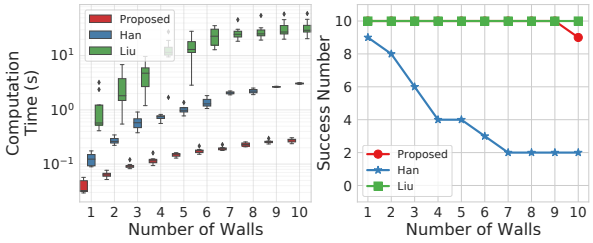

Fig. 10. The computation time and success number of the three methods in **Maze**.

orders of magnitude faster than the baselines while maintaining a high success rate.

B. Real-world Tests

To verify the real-world performance of the proposed method, we build a LiDAR-based quadrotor platform equipped with an Intel NUC onboard computer with CPU i7-10710U. The platform has a total weight of $1.5\,\mathrm{kg}$ and a thrust-to-weight ratio over 4.0.

High-accuracy and robust localization and mapping modules (*e.g.*, [22]–[24]) are necessary for the quadrotor to perform aggressive maneuvers. We use the Livox Mid360 LiDAR and Pixhawk's built-in IMU running FAST-LIO2 [24], which provides $100\,\mathrm{Hz}$ high-accuracy state estimation and $50\,\mathrm{Hz}$ point cloud. The extrinsic and time-offset between the LiDAR and IMU are pre-calibrated by [25]. We use an on-manifold model predictive controller [26] to perform high-accuracy trajectory tracking. To realize online planning and cope with newly sensed obstacles during the flight, we adopt a distance-triggered receding horizon planning scheme from our previous work [3]. The planning horizon is set to $D = 15\,\mathrm{m}$.

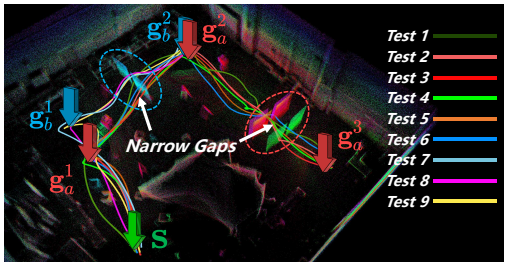

Fig. 11. The composite image of the 9 tests. The point cloud is superimposed by the nine independent tests, and in each test, there is only one narrow gap. All tests start from $\mathbf{s}$. Test 1-6 run the path **a**): $\mathbf{s} \to \mathbf{g}_a^1 \to \mathbf{g}_a^2 \to \mathbf{g}_a^3$, and pass through the narrow gap in red dashed ellipsoid; Test 7-9 run the path **b**): $\mathbf{s} \to \mathbf{g}_b^1 \to \mathbf{g}_b^2$ and pass through the narrow gap in blue dashed ellipoid. The location of the narrow gap in each test is changed to increase the diversity.

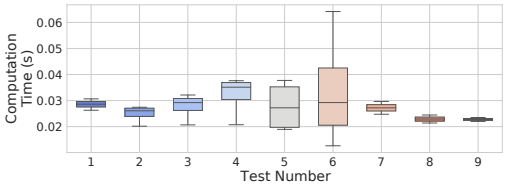

Fig. 12. The computation time of each (re-)planning process in the nine real-world tests.

We conduct nine tests and succeed in all of them (see Fig. 11). The computation time of (re-)planning in each test is shown in Fig. 12, which shows the proposed method can perform over $30\,\mathrm{Hz}$ online (re-)planning with an onboard computation unit.

Due to the space limit, we only show the detail of one typical experiment called *Test 1* and more details can be found in the attached video[3].

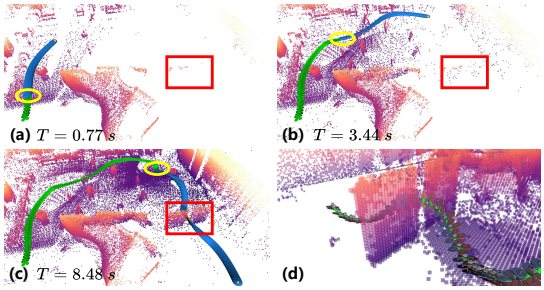

Fig. 13. The point cloud during the *Test 1* flight. For (a-c): the green curves indicate the executed trajectory, and the blue curves are the planned trajectory. The narrow gap (framed in the red box) is gradually sensed as the drone moves forward. The drone's position at $T = 0.77, 3.44, 8.48$ s are circled in yellow. (d) shows the trajectory of the quadrotor flying through the narrow gap.

In *Test 1*, as the drone travels forward, it automatically avoids obstacles and gradually senses the narrow gap (see Fig. 13(a-c)). Then it plans a piece of SE(3) trajectory as shown in Fig. 13(d). The maximum position tracking error in the middle of the gap is less than $5\,\mathrm{cm}$.

VI. Conclusion and Future Work

In this work, we proposed an online planning framework for quadrotor whole-body motion planning in unknown and unstructured environments. We first perform a parallel multi-resolution search to decompose the planning problem into several SE(3) and $\mathbb{R}^3$ sub-problems. Then SE(3) problem is solved with a dedicatedly designed seed generation approach which significantly increases the success rate. The overall hierarchical planning process dramatically reduces the computation time, making it possible to perform online aggressive flights in cluttered environments with a fully autonomous drone.

One limitation of the proposed method is that the sensor can only perceive one side of the narrow area, if the narrow area is long (not a thin wall), the narrow area is not actually passable since the quadrotor cannot maintain a nonzero roll angle without inducing lateral motion. One possible direction to address this problem is to adopt perception-aware planning, which actively detects the narrow areas and reduces the influence of partial perception. In the future, we will explore these designs and extend this method to more challenging environments and missions.

Acknowledgment

This work was supported by the Hong Kong General Research Fund (GRF) (17206920). The authors gratefully acknowledge DJI and Livox Technology for equipment support during the project. The authors would like to thank Wei Xu and Yixi Cai for the helpful discussions, and Prof. Ximin Lyu for the support of the experiment site.

[3]https://youtu.be/0q5mA9vijMY

REFERENCES

[1] Jesus Tordesillas, Brett T Lopez, Michael Everett, and Jonathan P How. Faster: Fast and safe trajectory planner for navigation in unknown environments. *IEEE Transactions on Robotics*, 38(2):922–938, 2021.

[2] Xin Zhou, Zhepei Wang, Hongkai Ye, Chao Xu, and Fei Gao. EGO-Planner: An ESDF-free gradient-based local planner for quadrotors. *IEEE Robotics and Automation Letters*, 6(2):478–485, 2021.

[3] Yunfan Ren, Fangcheng Zhu, Wenyi Liu, Zhepei Wang, Yi Lin, Fei Gao, and Fu Zhang. Bubble planner: Planning high-speed smooth quadrotor trajectories using receding corridors. *arXiv preprint arXiv:2202.12177, (Accepted by 2022 IROS)*, 2022.

[4] Boyu Zhou, Fei Gao, Jie Pan, and Shaojie Shen. Robust real-time UAV replanning using guided gradient-based optimization and topological paths. In *2020 IEEE International Conference on Robotics and Automation*, pages 1208–1214. IEEE, 2020.

[5] Sikang Liu, Nikolay Atanasov, Kartik Mohta, and Vijay Kumar. Search-based motion planning for quadrotors using linear quadratic minimum time control. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 2872–2879. IEEE, 2017.

[6] Giuseppe Loianno, Chris Brunner, Gary McGrath, and Vijay Kumar. Estimation, control, and planning for aggressive flight with a small quadrotor with a single camera and imu. *IEEE Robotics and Automation Letters*, 2(2):404–411, 2016.

[7] Toru Hirata and Makoto Kumon. Optimal path planning method with attitude constraints for quadrotor helicopters. In *Proceedings of the 2014 International Conference on Advanced Mechatronic Systems*, pages 377–381. IEEE, 2014.

[8] Jiarong Lin, Luqi Wang, Fei Gao, Shaojie Shen, and Fu Zhang. Flying through a narrow gap using neural network: an end-to-end planning and control approach. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3526–3533. IEEE, 2019.

[9] Shaohui Yang, Botao He, Zhepei Wang, Chao Xu, and Fei Gao. Whole-body real-time motion planning for multicopters. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9197–9203. IEEE, 2021.

[10] Zhepei Wang, Xin Zhou, Chao Xu, and Fei Gao. Geometrically constrained trajectory optimization for multicopters. *IEEE Transactions on Robotics*, 2022.

[11] Zhichao Han, Zhepei Wang, Neng Pan, Yi Lin, Chao Xu, and Fei Gao. Fast-racing: An open-source strong baseline for se(3) planning in autonomous drone racing. *IEEE Robotics and Automation Letters*, 6(4):8631–8638, 2021.

[12] Sikang Liu, Kartik Mohta, Nikolay Atanasov, and Vijay Kumar. Search-based motion planning for aggressive flight in se (3). *IEEE Robotics and Automation Letters*, 3(3):2439–2446, 2018.

[13] Boyu Zhou, Fei Gao, Luqi Wang, Chuhao Liu, and Shaojie Shen. Robust and successful quadrotor trajectory generation for fast autonomous flight. *IEEE Robotics and Automation Letters*, 4(4):3529–3536, 2019.

[14] Fanze Kong, Wei Xu, Yixi Cai, and Fu Zhang. Avoiding dynamic small obstacles with onboard sensing and computation on aerial robots. *IEEE Robotics and Automation Letters*, 6(4):7869–7876, 2021.

[15] Zhuozhu Jian, Zihong Yan, Xuanang Lei, Zihong Lu, Bin Lan, Xueqian Wang, and Bin Liang. Dynamic control barrier function-based model predictive control to safety-critical obstacle-avoidance of mobile robot. *arXiv preprint arXiv:2209.08539*, 2022.

[16] Zhuozhu Jian, Zihong Lu, Xiao Zhou, Bin Lan, Anxing Xiao, Xueqian Wang, and Bin Liang. Putn: A plane-fitting based uneven terrain navigation framework. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7160–7166. IEEE, 2022.

[17] Davide Falanga, Elias Mueggler, Matthias Faessler, and Davide Scaramuzza. Aggressive quadrotor flight through narrow gaps with onboard sensing and computing using active vision. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 5774–5781. IEEE, 2017.

[18] Daniel Mellinger and Vijay Kumar. Minimum snap trajectory generation and control for quadrotors. In *2011 IEEE international conference on robotics and automation*, pages 2520–2525. IEEE, 2011.

[19] Satyan L Devadoss and Joseph O'Rourke. *Discrete and computational geometry*. Princeton University Press, 2011.

[20] Sikang Liu, Michael Watterson, Kartik Mohta, Ke Sun, Subhrajit Bhattacharya, Camillo J Taylor, and Vijay Kumar. Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-d complex environments. *IEEE Robotics and Automation Letters*, 2(3):1688–1695, 2017.

[21] Mark W Mueller, Markus Hehn, and Raffaello D'Andrea. A computationally efficient motion primitive for quadrocopter trajectory generation. *IEEE transactions on robotics*, 31(6):1294–1310, 2015.

[22] Jiarong Lin and Fu Zhang. R^3live: A robust, real-time, rgb-colored, lidar-inertial-visual tightly-coupled state estimation and mapping package. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 10672–10678, 2022.

[23] Jiarong Lin and Fu Zhang. R^3live++: A robust, real-time, radiance reconstruction package with a tightly-coupled lidar-inertial-visual state estimator. *arXiv preprint arXiv:2209.03666*, 2022.

[24] Wei Xu, Yixi Cai, Dongjiao He, Jiarong Lin, and Fu Zhang. Fast-lio2: Fast direct lidar-inertial odometry. *IEEE Transactions on Robotics*, 2022.

[25] Fangcheng Zhu, Yunfan Ren, and Fu Zhang. Robust real-time lidar-inertial initialization. *arXiv preprint arXiv:2202.11006, (Accepted by 2022 IROS)*, 2022.

[26] Guozheng Lu, Wei Xu, and Fu Zhang. Model predictive control for trajectory tracking on differentiable manifolds. *arXiv preprint arXiv:2106.15233*, 2021.