CPA-Planner: Motion Planner with Complete Perception Awareness for Sensing-Limited Quadrotors

Qiuyu Yu[1], Chao Qin[2], Lingkun Luo[1], Hugh H.-T. Liu[2] and Shiqiang Hu[1]

***Abstract*—Motion planning in unknown environments suffers from the limited sensing range of cameras and occlusions. Therefore, perception awareness (PA) should be fully considered to ensure flight safety. Specifically, the unknown part of the planned trajectory should be sensed in advance (visibility constraint) at a safe distance (safety constraints). However, most of the optimization-based methods considered PA merely in the optimization stage in terms of the occlusions and ignored the limitation of the sensing model. As a result, the PA constraints remain violated due to poor initialization. Meanwhile, existing sampling methods can only ensure the visibility constraint rather than the safety constraint. In this paper, an optimization-based planning framework is proposed to solve the raised problems in real-time. Specifically, in the path-finding stage, a fast PA checking algorithm based on incremental convex sensing hulls is proposed and combined with the motion primitive propagation to find the initial path that satisfies the PA constraints. Subsequently, in the trajectory optimization stage, perception awareness with respect to occlusion and sensing model are jointly considered and formulated into the nonlinear optimization problem based on a grid map. Benchmarks with the state-of-the-art and physical experiments show that the proposed planner can generate safe trajectories with the highest success rate in different occluded environments and achieve comparable efficiency and aggressiveness simultaneously.**

***Index Terms*—Aerial Systems: Perception and Autonomy; Motion and path planning; Perception awareness.**

I. INTRODUCTION

AUTONOMOUS flight in obstacle-dense environments is essential for search, rescue, and other applications. Simply adopting the optimistic assumption that unknown areas are collision-free cannot guarantee flight safety [1], [2] due to the limited sensing range of the camera and occlusions caused by obstacles. Therefore, safe quadrotor flight in unknown environments requires the planner to have perception awareness (**PA**), which means the future trajectories it generates should be sensed in advance (visibility constraint) at a safe distance (safety constraint).

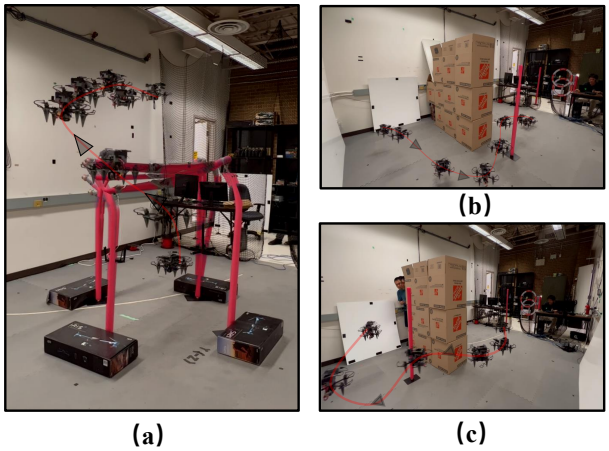

Fig. 1. composite image of the indoor experiment. **(a):** the quadrotor is required to fly up with only one forward camera. With the PA consideration w.r.t. sensing model, the proposed planner planned a safe spiral trajectory; **(b,c):** the quadrotor is required to fly to a specific target where existed occluded obstacles. With the PA consideration both in the path-finding stage and the optimization stage, the proposed planner generated safe trajectories.

However, perception awareness is not fully considered in the existing works. [3]–[11] Two mainstream research with PA consideration is categorized here. The first one is the sampling method, which aims to find a dynamically feasible path using motion primitive propagation. The second one is the optimization-based method, which finds an initial collision-free path in the path-finding stage and then generates an optimized trajectory based on the initial path in the optimization stage. Optimization-based motion planners considered the PA constraints w.r.t. occlusion mostly in the optimization stage based on Euclidean Signed Distance Field (**ESDF**) [3]–[5], which is redundant and computationally expensive. The perception range of the sensors is often ignored [6] or treated as omnidirectional [7]. As a result, if the initial path does not satisfy the PA constraints, the optimized trajectory may violate the constraints and lead to collisions. (Imagine that the target point is directly above the start point, and the quadrotor is equipped with only one forward depth camera). On the other hand, the sampling-based approach can only enforce the visibility of the constraints by constructing the sensing area conservatively [8]–[10] or ensuring the visibility of the future trajectory in a short horizon [11], preventing it from being

Manuscript received: August 05, 2022; Accepted November 28, 2022.

This paper was recommended for publication by Editor Tamim Asfour upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the National Natural Science Foundation of China (61773262, 62006152), and the China Aviation Science Foundation (20142057006).

[1]Qiuyu Yu, Lingkun Luo, Shiqiang Hu are with the school of Astronautics and Aeronautics, Shanghai Jiao Tong University, Shanghai, China. (email: joeyyu@sjtu.edu.cn; lolinkun1988@sjtu.edu.cn; sqhu@sjtu.edu.cn)

[2]Chao Qin and Hugh H.-T. Liu are with the University of Toronto Institute for Aerospace Studies, Toronto, ON, M3H5T6 Canada. (email: chao.qin@mail.utoronto.ca; hugh.liu@utoronto.ca)

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors.

Digital Object Identifier (DOI): see top of this page.

directly used for agile autonomous flight with limited sensing.

In this paper, we propose a motion planner with **c**omplete **P**erception **A**wareness called **CPA-P**lanner to address the research problems mentioned above. The PA constraints are considered in both the path-finding and trajectory optimization stages, thereby enhancing the motion planner with complete perception awareness and improved flight safety. At the same time, the efficiency of this motion planner is guaranteed by the following specially designed algorithms.

Specifically, we first propose a fast PA checking (**FPA**) algorithm based on a dynamic convex sensing hull. After that, a path-guided motion primitive propagation algorithm is combined with the FPA algorithm to generate a PA-satisfied initial path. In the optimization stage, a visibility metric that does not depend on ESDF is constructed to deal with the occlusion. In addition, the sensing model of the quadrotor is embedded into the optimization. Finally, a PA-satisfied trajectory is generated from the initial path via constrained nonlinear optimization.

The proposed planner is evaluated through comprehensive benchmark comparisons against state-of-the-art methods. The results demonstrate that the proposed method offers the highest level of safety, as well as comparable efficiency and aggressiveness.

The contributions of this paper are summarized as follows:

- A fast PA checking algorithm based on incremental convex sensing hull is proposed to speed up the checking procedure of the PA constraints.
- A visibility-guaranteed motion primitive propagation procedure is proposed to ensure that PA is satisfied in the path-finding stage.
- A two-phase trajectory optimization algorithm that considered the PA constraints w.r.t. occlusion and sensing model is proposed to ensure flight safety.

II. Related Work

A. Perception Awareness Path Planning

In general, sampling-based methods deal with the unknown area by shortening the planning horizon within the visible area. It assumed the unknown part would become visible as the quadrotor moves forward, [8], [10], leading to conservative flight strategies. Lopez et al. further mitigated the conservation in their work by allowing the selection of the past motion primitives [9]. However, checking the visibility only at the nodes decreases the number of motion primitives that can be kept, thus still introducing conservation. Florence et al. propose a specific data structure to quickly access the former sensing data at each frame [11], while the short planning horizon is still a remaining problem. Additionally, the aforementioned sampling methods can only ensure the new node is sensed in advance but not sensed in advance at a safe distance. In this paper, the sampling method is combined with a specially designed sensing convex hull algorithm to quickly find a long-distance collision-free path that satisfies the PA constraints.

B. Perception Awareness Trajectory Optimization

Optimization-based methods are widely investigated because of their flexibility in embedding constraints that can be adapted to different tasks. Tordesillas et al. [5] propose a method for always maintaining alternative trajectories to ensure safety in unknown environments. However, this method still does not consider active sensing of unknown areas, leading to incoherent flights. Recently, Wang et al. [4], [12] defined a visibility metric based on ESDF and adjusted the trajectory of the quadrotor based on this visibility metric to eliminate the occlusion effect along the trajectory. Although this approach achieves active awareness of the unknown areas, it does not guarantee that all the unknown trajectory points can be sensed at a safe distance. Zhou et al. [3] use an iterative optimization approach based on the visibility metric to ensure that the quadrotor can detect potential obstacles at a safe distance. Despite the excellent balance between safety and aggressiveness it achieved, the construction of the ESDF map imposes a non-negligible computational burden, and the sensing model is not considered. In addition, none of the aforementioned optimization-based methods considers the visibility constraint during the pathfinding stage. In other words, there is no guarantee that the PA constraints can be satisfied if there is already a PA violation at the initial path.

In this paper, PA constraints w.r.t. occlusion and sensing model are jointly considered and ensured by iterative optimization. Due to ill-conditioning problems, the L-BFGS utilized in [13] can not be directly applied. Therefore, we formulated the trajectory optimization problem into a general nonlinear constraint problem and solved it with the Augmented Lagrangian method [14].

Algorithm 1 FPA: Fast PA checking through incremental Sensing Convex hull

Input: Node n_c, n_c.parent
Output: True or False

```
n_tmp = n_c;
C_i ∈ S_c ← Samplebetween(n_c, n_c.parent);
n_c.H_C = QuickConvexHull(n_c.parent.H_C, H_c);
while n_tmp do
    if (evaluate (4) with n_tmp.H_C)==Fasle then
        if n_tmp == n_c then
            return False;
        else
            n_f = n_tmp break;
        end
    else
        n_tmp = n_tmp.parent;
    end
end
for C_i ∈ S_f do
    if (evaluate (5) with C_i, n_c)==True then
        return True;
    end
end
return False;
```

III. Path-finding with visibility checking

A. Incremental Convex Hull Algorithm for Quick Visibility Checking

Assume that the camera has fixed sensing distance h, equivalent vertical and horizontal sensing angle α, and the

sensing axis is aligned with the x-axis $\mathbf{x}^b(t_j)$ in quadrotor body frame. For an initial polynomial trajectory $\mathbf{P}(t) \in \mathbb{R}^3$, The sensing range of the camera at timestamp t can be represented by a convex cone $C \subset \mathbb{R}^3$:

$$C(t) = \{\mathbf{x} = R(t)\mathbf{E}\mu\hat{h} + \mathbf{P}(t) + \bar{h}R(t)\mathbf{e_1} \mid \|\mu\|_\infty \leq 1, \bar{h} \leq h\} \quad (1)$$

where $R(t)$ is the rotation matrix of the quadrotor at time step t, $\mathbf{E} = (\mathbf{e_2}, \mathbf{e_3}) \in \mathbb{R}_{3\times 2}$, $\mu \in \mathbb{R}_2$, $\hat{h} = \bar{h} \cdot tan(\alpha)$.

Since the contour of the sensing cone along the path is not convex, it is critical to design an efficient algorithm to check the visibility of a path node. In this paper, motion primitive in jerk space is adopted to propagate an initial path. The original problem is relaxed to ensure the PA constraints at the expanding nodes.

We first reorganized the sensing cone at the $i-th$ nodes as the intersection of five half-planes, as illustrated in Fig. 2.

$$C_i = \{(x, y, z) | a_k^x x + a_k^y y + a_k^z z \leq b_k \ k = 1, 2...5\} \quad (2)$$

The accumulated sensing area $\mathcal{S}_j \subset \mathbb{R}^3$ at time t_j can be calculated as:

$$\mathcal{S}_j = \cup C_i \ i = 1, 2...j \quad (3)$$

In the pathfinding stage, the PA constraints are added as: for any expanding nodes $\mathbf{P}(t_i)$, it should be first seen at some t_j before t_i and satisfies the safety criterion:

$$\mathbf{P}(t_i) \in \mathcal{S}_j \quad (4)$$

$$\|\mathbf{v}(t_i)\|^2 - 2\|\mathbf{a}_{max}\|\|\mathbf{P}(t_i) - \mathbf{P}(t_j)\| < 0 \quad (5)$$

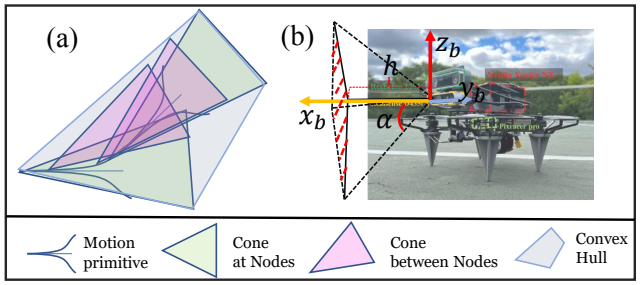

Fig. 2. Definition of the sensing cone and the sensing hull. **a**The incremental convex hull is constructed on the node of the motion primitive, which takes the convex hull of the last node and all the sensing cones between the last node and the current node as input. **(b)**: quadrotor used in the experiment and the relation between the sensing cone and the body frame

As demonstrated in Algorithm 1, given a piece of motion primitive and a new node n_c to be added to the open set. A series of points n_i along with it will be sampled, and the corresponding sensing cones are constructed C_i at each point. After that, construct the convex hull $\mathcal{H}_c$ of these sensing cone, and update $n_c.\mathcal{H}_C$ as the convex hull of $n_c.parent.\mathcal{H}_C$ and $\mathcal{H}_c$. We then find the first node n_f at which n_c is not inside the $n_f.\mathcal{H}_C$. If n_f is exactly n_c, then it means n_c can't be seen by any former trajectory point. Otherwise, we further check the sample point n_j between n_f and $n_f.children$ for (5), if any n_j satisfies (5), n_c can be regarded as a visible and safe node to be expanded.

A convex hull can be represented by its vertex points. To find if the checking node is inside the convex hull or not, we formulated it as a linear separability problem and solved it with Seidel's algorithm [15].

$$\begin{aligned} \min_{\mathbf{a}\in\mathbb{R}_3, b\in R} \quad & \mathbf{a}^\top \mathbf{x}_n \\ & \mathbf{a}^\top \mathbf{x}_i + b \leq 0 \quad \forall \mathbf{x}_i \in V_H \\ & \mathbf{a}^\top \mathbf{x}_n + b > 0. \end{aligned} \quad (6)$$

where V_H is the set of the vertex of the current convex hull, $\mathbf{x}_n$ is the expanding node. Given a node with M parent nodes, the expected time complexity to reject an invalid node by checking every former sensing cone is $O(MN)$. Meanwhile, given the vertex number H of the current convex hull, the expected time complexity of the quick-hull algorithm and the linear separability problem is $O((H+N)\log(H+N))$ and $O(H)$ respectively. Through careful engineering, H can be ensured with the same order as N, and the time complexity is $O(N \log N)$. With $M \gg N$, which is common in long distance path-finding, the proposed PA constraint-checking algorithm is more efficient.

B. Path Guided Visibility Guaranteed Motion Primitive Propagation

The KinodynamicAStar algorithm in [16] suffered from the balance between efficiency and the resolution of the input discretization. In this paper, a resolution-increasing mechanism is designed to activate when a direct collision-free path is found analytically by the Pontryagin minimum principle [17]. More specifically, instead of directly accepting the analytical path, the discretization degree of the input will be increased, and the motion primitive propagation will keep working until the new node is near the target. We name this high-resolution discretization stage the analytically approaching stage.

Algorithm 2 VGPF: Visibility Guaranteed Path Finding

Input: Goal $\mathbf{g}_c$, GridMap $\mathcal{M}_c$, Initial path $\mathcal{P}_c$, Current Position $\mathbf{P}_c$, input resolution $\mathbf{r}_u$;
Output: PA satisfied path $\mathcal{P}$
while $\neg$***nearEnd*** $\wedge$ $\neg$**AnaliticalApproaching do**
  $\mathcal{P} \leftarrow$ **KinoPA**$(\mathcal{P}, \mathbf{r_u})$
end
if $\neg$**nearEnd then**
  $\mathbf{r}_u \leftarrow \alpha \mathbf{r}_u$;
  $\mathcal{P} \leftarrow$ **KinoPAfit**$(\mathcal{P}, \mathbf{r_u})$;
end
return $\mathcal{P}$;

In the analytically approaching stage, to guide the motion primitive quickly converge to the target goal, the original heuristic function in KinodynamicAStar is replaced by:

$$h_c = \omega_l d_l + \omega_s d_d - \omega_v \|\mathbf{v}\| \quad (7)$$

where d_l is the longitude distance to the point of the analytical path at the same time stamp. d_s is the lateral distance to the path. $\mathbf{v}$ represents the velocity point to the target. The procedure of the **V**isibility **G**uaranteed **P**ath **F**inding (**VGPF**) is detailed in Algorithm2. The **KinoPA** is the combination of the

KinodynamicAStar algorithm and the proposed FPA algorithm. FPA functions in the new node checking phase to reject new nodes that violate PA constraints. **KinoPAfit** is the same as **KinoPA** except that it uses the heuristic function in (7).

IV. Perception Awareness trajectory optimization

A. Path Guided Warm-up Optimization

In this paper, trajectories are represented by uniform cubic B-spline curves for efficient cost evaluations. As shown in Fig. 3, whenever replanning is triggered, a collision-free, PA-satisfied path is found by VGPF as the guiding path. Unlike the Astar algorithm which is adopted by [13], the proposed VGPF may generate an initial path that is not close to the surface of the original obstacles. Therefore, to generate enough pairs of intersection points P and out-obstacle vectors V to avoid potential new obstacles, an initial trajectory $P_i(t)$ that intersects with obstacles (may not) is obtained by arch length sampling and minimum snap fitting.

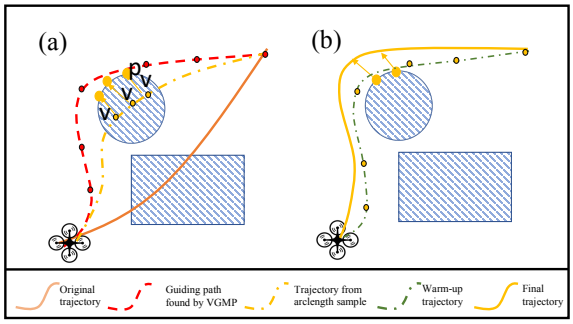

Fig. 3. Path guided trajectory optimization. (a) obtain the guiding path and the initial control points; (b) warm-up optimization;

In the first stage, we generated a warm-up trajectory $P_w(t)$ from $P_i(t)$ via an unconstrained quadratic problem so that it has the shape of the guiding path and is as smooth as possible. The cost function J_{p1} is:

$$\begin{aligned} J_{p1} =& \lambda_s f_s + \lambda_g f_g \\ =& \lambda_s \sum_{i=p_b-1}^{N-p_b+1} \|\mathbf{Q}_{i+1} - 2\mathbf{Q}_i + \mathbf{Q}_{i-1}\|^2 \\ &+ \lambda_g \sum_{i=p_b}^{N-p_b} \|\mathbf{Q}_i - \mathbf{g}_i\|^2 \end{aligned} \tag{8}$$

where $\mathbf{Q}_i \in \mathbb{R}^3$ is the control point of the trajectory and also the decision variable of the trajectory optimization. N is the number of the control points, $p_b = 3$ is the order of the B-spine, and $\mathbf{g_i}$ is the corresponding guiding point.

B. Perception Awareness for Occlusion

Given a warm-up trajectory $P_w(t)$ that is already collision-free, the second step is getting this trajectory ready for the unknown region, which starts with obtaining the boundaries between the unknown area and the known-free area. In this subsection, the PA constraints w.r.t. occlusion are first discussed and formulated into penalty functions, and the problem of sensing range is detailed in the following subsection.

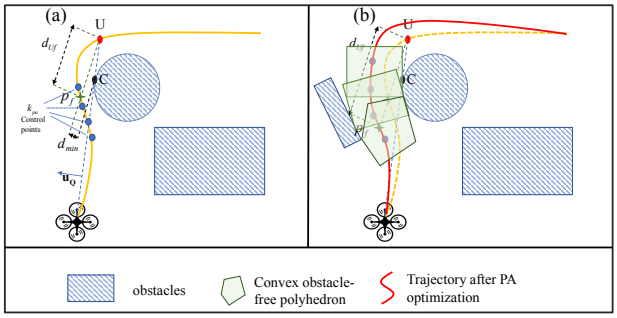

Fig. 4. Perception awareness trajectory optimization; (a) the definition of the visibility metrics; (b) Optimization according to the PA cost and the corridor constraints

As shown in Fig. 4 The first unknown point $\mathbf{U}$ in $P_i(t)$ could be easily obtained from the grid map within $O(n)$. Accordingly, the corner point $\mathbf{C}$ is the intersection point of the line $L_{\mathbf{P_c U_0}}$ and the obstacle.

1) Visibility metric: With all the notions mentioned above, the visibility metric between a trajectory point $\mathbf{Q}$ and unknown point $\mathbf{U}$ is defined as:

$$d_Q = \left(\frac{(\mathbf{C}-\mathbf{Q})\cdot(\mathbf{U}-\mathbf{Q})}{\|\mathbf{Q}-\mathbf{U}\|^2} (\mathbf{Q}-\mathbf{U}) \right) \cdot \mu_{\mathbf{Q}} - (\mathbf{C}-\mathbf{Q}) \cdot \mu_{\mathbf{Q}} \tag{9}$$

where $\mu_{\mathbf{Q}}$ is perpendicular to $\mathbf{L_{CU}}$ and points to the outside of the obstacle. In general, d_Q is positive if $\mathbf{U}$ is visible to $\mathbf{Q}$, otherwise d_Q is negative.

With the visibility metric, we can find the first trajectory point $\mathbf{p}_f$ that satisfied $d_{\mathbf{p}_f} \geq d_{min}$. Then we check if the following criterion is satisfied:

$$\|\mathbf{v}_f\|^2 / 2\|\mathbf{a}_{max}\| \leq d_{\mathbf{U}f} \tag{10}$$

where $\mathbf{v}_f$ is the velocity at $\mathbf{p}_f$, $d_{\mathbf{U}f}$ represents the distance between $\mathbf{p}_f$ and unknown point $\mathbf{U}$. If the inequality above is satisfied, return $\mathbf{P}_i(t)$ directly. If it is not satisfied, then the unknown point needs to be seen in advance. Instead of adding cost on the future $\mathbf{p}_f$ and starting with a guess, we add the constraint directly to the k_{pa} control points before $\mathbf{p}_f$. k_{pa} is related to the distance of the initial control point $dist_{cp}$, and is initialized as:

$$k_{pa} = Ceil(\alpha_0 \frac{\|\mathbf{v}_f\|^2/2\|\mathbf{a}_{max}\| - d_{\mathbf{U}f}}{dist_{cp}}) \tag{11}$$

2) PA penalty w.r.t. occlusion: Two criterion items: distance criterion c_d^i and velocity criterion c_v^i are constructed for any control point $\mathbf{Q}_i$ that belongs to the k_{pa} control points. $\mathbf{Q}_i$ will be penalized according to the criterion.

$$c_d^i = \alpha_1 \frac{s\|\mathbf{Q}_i - \mathbf{U}\|^2}{\|\mathbf{p}_f - \mathbf{U}\|^2} - d_{\mathbf{Q}_i}, c_v^i = \|\mathbf{V}_i\|^2 - 2a_{max}\|\mathbf{Q}_i - \mathbf{U}\| \tag{12}$$

The gradients of c_d^i and c_v^i w.r.t $\mathbf{Q_i}$ are:

$$\frac{\partial c_d^i}{\partial \mathbf{Q}_i} = 2\alpha_1 \frac{\mathbf{Q}_i - \mathbf{U}}{\|\mathbf{p}_f - \mathbf{U}\|^2} - \frac{\partial d_{\mathbf{Q}i}}{\partial \mathbf{Q}_i} \tag{13}$$

$$\frac{\partial c_v^i}{\partial \mathbf{Q}_i} = -2\mathbf{V}_i - 2a_{max}\frac{\mathbf{Q}_i - \mathbf{U}}{\|\mathbf{Q}_i - \mathbf{U}\|} \tag{14}$$

To get a smooth penalty function and controllable constraints accuracy, the penalty function and corresponding gradient are defined respectively:

$$f_{pa}(c) = \begin{cases} 0 & (c \le 0) \\ c^3 & (0 < c \le \mu_{pa}) \\ 3\mu_{pa}c^2 - 3\mu_{pa}^2 c + \mu_{pa}^3 & (c_i > \mu_{pa}) \end{cases} \tag{15}$$

$$\frac{\partial f_{pa}(c)}{\partial c} = \begin{cases} 0 & (c \le 0) \\ 2c^2 & (0 < c \le \mu_{pa}) \\ 6\mu_{pa}c - 3\mu_{pa}^2 & (c_i > \mu_{pa}) \end{cases} \tag{16}$$

where μ_{pa} is the parameter to control the accuracy of the constraints. The total perception cost is obtained from combining costs on each $\mathbf{Q}_i$.

$$J_{po} = \sum_{i=F-k_{pa}}^{F} f_{pa}\left(c_d^i\right) + \sum_{i=F-k_{pa}}^{F} f_{pa}\left(c_v^i\right) \tag{17}$$

where F is the index of the last control point of the trajectory segment in which $\mathbf{p}_f$ is located. Given (13-17), the gradients of the total perception cost can be easily calculated using the chain rule.

C. Perception Awareness for Sensing Model

Considering the sensing model we constructed in Section III-A, the PA constraints (5) are highly sequentially related. To release the sequential effect, we first record the time stamp t_i^{pro} of the farthest trajectory point the quadrotor can see at each control point Q_i, and the violation of the PA constraints can be categorized into the following three situations:

1) Trajectory shape: Under this situation, the quadrotor can not see any future trajectory point since the angle between the desired velocity and the sensing axis is larger than the angle of the sensing cone. Therefore, the following inequality constraint should be satisfied at control point $\mathbf{Q}_i$:

$$\cos\alpha < \mathbf{V}_i^u \cdot \mathbf{h}_i,\ \mathbf{V}_i^u = \frac{\mathbf{V}_i}{\|\mathbf{V}_i\|},\ \mathbf{h}_i = \mathbf{q}_i \odot \mathbf{e}_x, \tag{18}$$

where $\mathbf{V}_i, \mathbf{q}_i$ is the velocity and the quaternion at control point $\mathbf{Q}_i$, and $\odot$ is multiplication of quaternions.

2) Velocity limit exceeded: Under this situation, the furthest visible trajectory point $\mathbf{P}(t_i^{pro})$ is inside the sensing cone of $\mathbf{Q}_i$ and at the longest sensing distance. Thus, the velocity needs to be constrained as:

$$\|\mathbf{V_i}\|^2 \le 2 \cdot a_{max} \cdot S,\ S = ||\mathbf{P}(t_i^{pro}) - \mathbf{Q}_i|| \tag{19}$$

3) Pitch violation: Under this situation, the first unknown point $\mathbf{P}(t_i^{pro})$ is not far enough to satisfy the PA constraints (5), but a farther trajectory point $P(t_s)$ can be found which satisfied the PA constraints, and the angle between $P(t_s)$ and $\mathbf{e}_x$ is no more than the angle of the sensing cone α. Thus, the pitch angle needs to be reduced (F-L-U body axis adopted) to see that point:

$$\cos\alpha < \mathbf{S}_i^u \cdot \mathbf{h}_i,\ \mathbf{S}_i^u = \frac{\mathbf{P}(t_s) - \mathbf{Q}_i}{\|\mathbf{P}(t_s) - \mathbf{Q}_i\|} \tag{20}$$

4) Triggered when Needed Polyhedron Constraints: Additionally, to ensure the safety of the adjusted trajectory, we perform convex decomposition [18] of free space based on these k_{pa} control points, obtaining $k_{pa} - 1$ convex polyhedron. For each control point $\mathbf{Q}_i$, it will have k_i halfplane constraints:

$$\mathbf{n}_{ij}^{\mathrm{T}}\mathbf{Q}_i + d_{ij} < 0 \quad j = 1, 2, \cdots k_i \tag{21}$$

D. PHR Augmented Lagrangian Optimization

Algorithm 3 PHR Augmented Lagrangian Perception awareness trajectory optimization

Input: Warm up trajectory $\mathbf{P}_i(t)$
Output: Perception awareness trajectory $\mathbf{P}_r(t)$
while ¬ *(25)* **do**
  $\mathbf{p}_f, t_i^{pro} \leftarrow$ **VisMetricConstruction**$(\mathbf{P}_i(t))$;
  $\mu \leftarrow \max[\mu + \rho g(x), 0]$, $\rho \leftarrow \min[(1+\gamma)\rho, \beta], \alpha_0 \uparrow, \alpha_1 \uparrow$;
  while ¬ *(24)* **do**
    $\mathbf{P}_r(t) \leftarrow$ **InterOptimization**$(\mathbf{P}_i(t), \mathbf{p}_f, t_i^{pro})$;
  end
  $\mathbf{P}_i(t) \leftarrow \mathbf{P}_r(t)$
end
return $\mathbf{P}_r(t)$;

The PA constraints for occlusion are determined by the environment, it's appropriate to model them as penalty functions to trade constraints accuracy for efficiency and robustness slightly. However, the PA constraints for the sensing model of the quadrotor are determined by the trajectory itself and should be accurate enough to get higher safety. Thus, the perception awareness trajectory optimization problem takes the form of the following:

$$\begin{aligned} &\min_{\mathbf{Q}_i} \sum_{i=1}^{N} J_{ego} + \lambda_{po} J_{po} \\ &\text{s.t. } \cos\alpha < \mathbf{V}_i^u \cdot \mathbf{h}_i;\ \|\mathbf{V}_o\|^2 \le 2 \cdot a_{max} \cdot S \\ &\cos\alpha < \mathbf{S}_i^u \cdot \mathbf{h}_i;\ \mathbf{n}_{ij}^{\mathrm{T}}\mathbf{Q}_i + d_{ij} < 0 \end{aligned} \tag{22}$$

where J_{ego} is the rebound cost function in [13] for obstacle avoidance, trajectory smoothing, and dynamical feasibility. To avoid the ill-conditioned problem and heavy computational burden, the PHR Augmented Lagrangian Method is adopted for iterative optimization. The PHR Augmented Lagrangian is defined as:

$$\mathcal{L}_\rho(x, \mu) := f(x) + \frac{\rho}{2}\left\|\max\left[g(x) + \frac{\mu}{\rho}, 0\right]\right\|^2 - \frac{1}{2\rho}\|\mu\|^2 \tag{23}$$

where x, $f(x)$, and $g(x)$ are the decision variable, objective function, and inequality constraints described in (22) respectively. $\rho \in \mathbb{R}$ is the regular parameter. $\mu \ge 0$ is the lagrangian multiplier for inequality constraints. As summarized in Algorithm 3, in the inner optimization loop, t_i^{pro}, ρ, μ are fixed and an unconstrained optimization is conducted using L-BFGS. In the outer optimization loop, the checking point $\mathbf{p}_f$ and t_i^{pro} are constructed according to the new variables, meanwhile, ρ and μ are updated for higher accuracy.

The stopping conditions of the inner optimization and outer optimization are:

$$\left\|\nabla_x \mathcal{L}_\rho(x,\mu)\right\|_\infty < \xi^k \min\left[1, \left\|\max\left[g(x), -\frac{\mu}{\rho}\right]\right\|_\infty\right] \tag{24}$$

$$\left\|\max\left[g(x), -\frac{\mu}{\rho}\right]\right\|_\infty < \epsilon_{\text{cons}}, \left\|\nabla_x \mathcal{L}_\rho(x,\mu)\right\|_\infty < \epsilon_{\text{prec}} \tag{25}$$

where ξ^k is positive and gradually converged to 0, $\epsilon_{\text{cons}} = 1e-6$ and $\epsilon_{\text{prec}} = 1e-8$ are accuracy parameters. $\beta = 1.0e+3, \gamma = 1.0$;

V. BENCHMARK

In this section, extensive benchmark comparisons of path finding, perception awareness planning, and complete planning performance in several different simulation environments are conducted on an AMD 5900HX CPU laptop.

A. Evaluation for Path Planning

We first conducted path finding comparison in obstacle-free environments between the proposed method, Nano-Map [11], and Search-Limited [10] in 200 trials. Fig. 5 shows the result of one specific experiment, and Table I shows the average results of 200 experiments. Attributed to the fast PA checking and the high resolution of discretization, the proposed VGPF produces a path with the highest average speed of 1.96 m/s and the shortest average path length of 12.41m. VGPF also achieved the highest safety percentage since it considered the safety constraint (5) in path-finding. VGPF achieves the same efficiency level as Nano-map but with a slightly longer finding time.

TABLE I
PATH PLANNING COMPARISON FOR SENSING CONSTRAINTS

Method	t_{find} (ms)	t_{traj} (s)	Path Length(m)	v_{avg} (m/s)	safe length(%)
SE3 limited	12.01	10.30	17.70	1.71	79.41
Nano-Map	**5.32**	8.41	15.94	1.89	91.72
VGPF(**proposed**)	6.67	**6.33**	**12.41**	**1.96**	**96.38**

B. Evaluation For Perception Awareness Trajectory Optimization

For perception awareness evaluation, we benchmark our algorithm in 10 Gazebo worlds (five of which contain vertical obstacles and narrow vertical aisles, and the other fives contain horizontal occlusions) with EGO-Planner [13], and RAPTOR [3]. For each world, we switch the start and goals, and a total of 20 trials of experiments are conducted for each method. The results of the experiments are demonstrated in Table II. Without the consideration of PA, Ego-Planner was successful only four times out of 20 experiments in environments containing horizontal occlusion. Raptor had PA consideration w.r.t occlusion in the optimization stage, but the sensing model was ignored, so it was successful only two times out of 10 vertical scenes. The proposed method achieved the highest success rate, and the only scenario that failed was due to the initial position being less than 0.2 m away from the vertical obstacle.

Screenshots of the experiments that highlight the effectiveness of the proposed PA mechanism are presented in Fig. 6.

TABLE II
PERCPETION AWARENESS COMPARISON

Method	$\mathbf{N}_{suc}$	$\mathbf{N}_{stop}$	$\mathbf{N}_{obsSensed}$
Ego-Planner	4	9	(0/4)
Raptor(velocity-tracking yaw)	12	/	(2/12)
CPA(**proposed**)	**19**	**2**	**(9/19)**

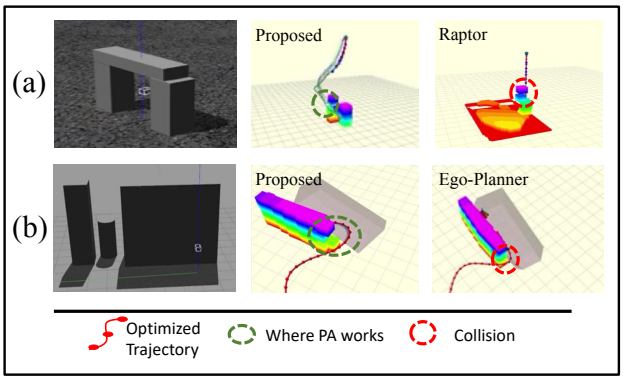

Fig. 6. Screenshots that highlight the effectiveness of the proposed perception awareness mechanism.

C. Planner Comparison

To fully test the performance of the proposed planner from the aspect of long-distance navigation, we benchmarked it against two state-of-the-art methods: Ego-Planner [13], Raptor [3]. To make the comparison fair enough, we added the proposed VGPF as the global planning module to Ego-planner since it does not have collision-free initialization originally. 200 experimental

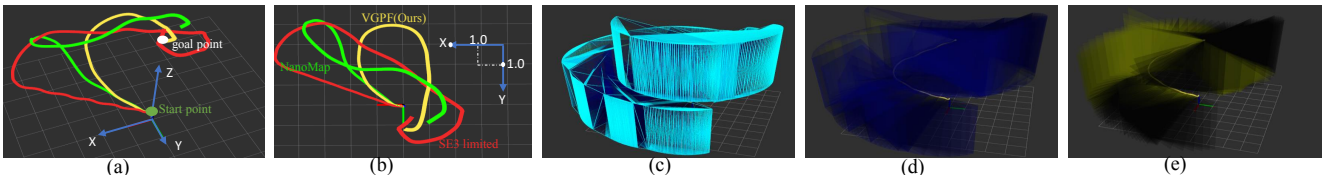

Fig. 5. One specific experiment of the sensing constrained path planning. The start point is at $(0,0,1)$ and the goal point is at $(0,0,7)$. **(a):** Planning comparison from a 3-dimensional view; **(b):** Planning comparison from top view; **(c-d):** The sensing convex hulls of the path found by VGPF (visualized with edges and meshes); **(e):** The sensing cones.

flights in several forest environments are conducted for each method, and the results are presented in Table III. the screenshots of one convincing experiment of the participated methods together with the velocity profile of the proposed method are shown in Fig. 7. We refer readers to the video[1] for more details about the experiments.

TABLE III
BENCHMARK OF THE PROPOSED PLANNER IN RANDOM ENVIRONMENT

method	Density (obs/m^2)	Suc (%)	DTC (m)	v_{max} (m/s)	v_{avg} (m/s)	Energy (m^2/s^5)	t_{plan} (ms)
Ego	0.2	87.0	2.13	3.34	2.57	260.44	**0.91**
	0.4	73.5	1.31	3.04	2.26	747.29	**1.14**
Raptor	0.2	95.5	3.31	3.10	2.14	**100.31**	8.9
	0.4	89.5	2.33	3.01	1.83	**413.28**	10.1
CPA(**Proposed**)	0.2	**97.5**	**3.42**	**3.97**	**2.89**	271.16	3.42
	0.4	**93.5**	**2.61**	**3.68**	**2.65**	670.41	3.91

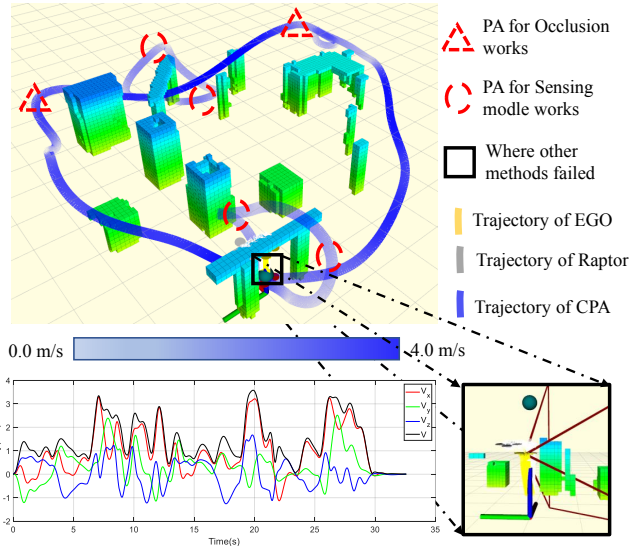

Fig. 7. Screenshots of one convincing simulation comparison. **Top**: executed trajectory and mapping results of the proposed CPA-planner; **Left bottom**: Velocity file of the CPA-planner; **Right bottom:** where Ego-planner and Raptor didn't sense the vertical obstacle and crashed.

As the results demonstrated, the success rate of the proposed **CPA**-Planner is the highest among all other methods. Attributed to the complete consideration of PA, the proposed method finds the obstacle on the trajectory with the longest distance. The planning time (excluding path finding time) of the proposed method is slightly longer than Ego-Planner but significantly faster than Raptor. Considering the functional complexity of the proposed method, we argue that the proposed algorithm is efficient. The trajectory energy of the proposed method is close to EGO-Planner but slightly higher than other methods, which may be caused by the longer trajectory length due to PA constraints.

VI. REAL-WORLD EXPERIMENTS

Two indoor experiments have been conducted utilizing a self-assembled quadrotor as shown in Fig. 2(b). The tracking controller is developed from [19], all the state estimation, mapping, planning, and control are conducted on the Nvidia Xavier NX onboard computer[2]. The software of the proposed planner will be released online[3].

The first experiment is aimed at demonstrating the effectiveness of the proposed sensing-constrained PA mechanism. The quadrotor is required to fly up and down with only a forward depth camera and without any pre-built map. As shown in Fig. 8, there are three different combinations of obstacles, which are door-like vertical barriers, cross-barriers, and cross-barriers with the board above. The narrowest gap within these environments is just 0.7 m and the drone is only 0.5 m from the nearest vertical obstacle at the initial state. Despite the limitation of the sensing range and the narrow indoor environment, the quadrotor can always foresee and avoid obstacles, while achieving a max velocity of 1.02 m/s.

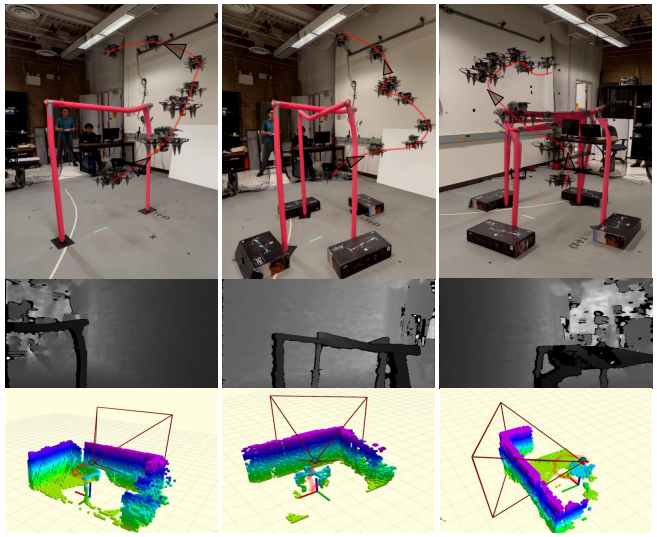

Fig. 8. Indoor experiments of vertical planning. **Top:** composite image of the experiments with different vertical obstacle settings; **Middle:** corresponding depth image when the quadrotor first reacted to the obstacle; **Bottom:** mapping and planning visualization in rviz.

The second experiment is to illustrate the effectiveness of the PA mechanism w.r.t. occlusion. The quadrotor is required to fly by a series of given way-points in a cluttered indoor environment. The obstacles on the way are always occluded by big boxes at the beginning. Every time the quadrotor arrived at a target point, the obstacle will be placed again and the grid map will be cleaned. As shown in Fig. 9, despite all the limited perception and occlusion, the quadrotor can always foresee the obstacle at safe distance, while achieving a max velocity at 1.31 m/s. With all the results presented above, we argue the proposed planner is capable of truly 3-dimensional perception awareness and achieves comparable efficiency and aggressiveness as some state-of-the-art gradient-based planners.

VII. CONCLUSIONS

In this paper, aimed at a lightweight trajectory optimization framework with complete perception awareness, we first propose

[1][Online]. Available: https://youtu.be/v-TtTod7Pfk

[2][Online]. Available: https://www.nvidia.com

[3][Online]. To be released: https://github.com/Joeyyuenjoyslife/CPA-Planner

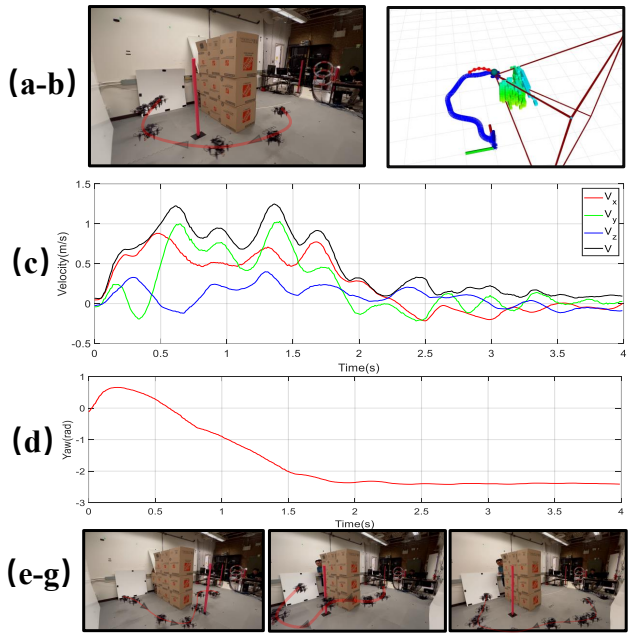

Fig. 9. Planning with occlusion. **(a):** composite image of one specific occlusion planning experiment; **(b):** corresponding executed trajectory and mapping visualization in rviz; **(c,d):** velocity and yaw file of the experiment; **(e-g):** composite image of several other trials.

the VGPF algorithm to find PA-satisfied paths with high efficiency. After that, perception awareness w.r.t occlusion and sensing model are considered and formulated into nonlinear constrained optimization separately to make sure the PA constraints are not violated. Finally, the PHR-Augmented Lagrangian method is introduced to solve the iterative nonlinear constrained problems. The benchmark with state-of-the-art demonstrates the superiority of **CPA**-Planner in flight safety and aggressiveness under unknown obstacle-rich environments. In the future, different settings and combinations of cameras, and more suitable time allocation will be considered. We also try to find the specific trajectory class with PA constraints built in to separate the PA cost and other customized objectives to get higher efficiency and flexibility.

References

[1] V. Usenko, L. von Stumberg, A. Pangercic, and D. Cremers, "Real-time trajectory replanning for mavs using uniform b-splines and a 3d circular buffer," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Vancouver, Canada, Sept. 2017, pp. 215–222.

[2] S. Liu, M. Watterson, K. Mohta, K. Sun, S. Bhattacharya, C. J. Taylor, and V. Kumar, "Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-d complex environments," *IEEE Robot. Automat. Lett.*, vol. 2, no. 3, pp. 1688–1695, July 2017.

[3] B. Zhou, J. Pan, F. Gao, and S. Shen, "Raptor: Robust and perception-aware trajectory replanning for quadrotor fast flight," *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 1992–2009, Dec. 2021.

[4] Q. Wang, B. He, Z. Xun, C. Xu, and F. Gao, "Gpa-teleoperation: Gaze enhanced perception-aware safe assistive aerial teleoperation," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 5631–5638, Apr. 2022.

[5] J. Tordesillas, B. T. Lopez, M. Everett, and J. P. How, "Faster: Fast and safe trajectory planner for navigation in unknown environments," *IEEE Trans. Robot.*, Apr. 2021.

[6] T. Liu, Q. Wang, X. Zhong, Z. Wang, C. Xu, F. Zhang, and F. Gao, "Star-convex constrained optimization for visibility planning with application to aerial inspection," in *Proc. IEEE Int. Conf. Robot. Autom.*, Philadelphia, USA, May 2022, pp. 7861–7867.

[7] W. Gao, K. Wang, W. Ding, F. Gao, T. Qin, and S. Shen, "Autonomous aerial robot using dual-fisheye cameras," *J. Field Robot.*, vol. 37, no. 4, pp. 497–514, Feb. 2020.

[8] B. T. Lopez and J. P. How, "Aggressive 3-d collision avoidance for high-speed navigation." in *Proc. IEEE Int. Conf. Robot. Autom.*, Singapore, May 2017, pp. 5759–5765.

[9] B. T. Lopez and J. P. How, "Aggressive collision avoidance with limited field-of-view sensing," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Vancouver, Canada, Sept. 2017, pp. 1358–1365.

[10] S. Liu, M. Watterson, S. Tang, and V. Kumar, "High speed navigation for quadrotors with limited onboard sensing," in *Proc. IEEE Int. Conf. Robot. Autom.*, Stockholm, Sweden, May 2016, pp. 1484–1491.

[11] P. R. Florence, J. Carter, J. Ware, and R. Tedrake, "Nanomap: Fast, uncertainty-aware proximity queries with lazy search over local 3d data," in *Proc. IEEE Int. Conf. Robot. Autom.*, Brisbane, Australia, May 2018, pp. 7631–7638.

[12] Q. Wang, Y. Gao, J. Ji, C. Xu, and F. Gao, "Visibility-aware trajectory optimization with application to aerial tracking," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Prague, Czech Republic, Sept. 2021, pp. 5249–5256.

[13] X. Zhou, Z. Wang, H. Ye, C. Xu, and F. Gao, "Ego-planner: An esdf-free gradient-based local planner for quadrotors," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 478–485, Apr. 2021.

[14] E. G. Birgin and J. M. Martinez, "Improving ultimate convergence of an augmented lagrangian method," *Optim. Methods Softw.*, vol. 23, no. 2, pp. 177–195, Apr. 2008.

[15] R. Seidel, "Small-dimensional linear programming and convex hulls made easy," *Discrete Comput. Geom.*, vol. 6, no. 3, pp. 423–434, June 1991.

[16] B. Zhou, F. Gao, L. Wang, C. Liu, and S. Shen, "Robust and efficient quadrotor trajectory generation for fast autonomous flight," *IEEE Robot. Automat. Lett.*, vol. 4, no. 4, pp. 3529–3536, Oct. 2019.

[17] M. W. Mueller, M. Hehn, and R. D'Andrea, "A computationally efficient motion primitive for quadrocopter trajectory generation," *IEEE Trans. Robot.*, vol. 31, no. 6, pp. 1294–1310, Dec. 2015.

[18] S. Liu, M. Watterson, K. Mohta, K. Sun, S. Bhattacharya, C. J. Taylor, and V. Kumar, "Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-d complex environments," *IEEE Robot. Automat. Lett.*, vol. 2, no. 3, pp. 1688–1695, July 2017.

[19] M. Faessler, A. Franchi, and D. Scaramuzza, "Differential flatness of quadrotor dynamics subject to rotor drag for accurate tracking of high-speed trajectories," *IEEE Robot. Automat. Lett.*, vol. 3, no. 2, pp. 620–626, Apr. 2017.