

# APACE: Agile and Perception-aware Trajectory Generation for Quadrotor Flights

Xinyi Chen, Yichen Zhang, Boyu Zhou and Shaojie Shen

**Abstract**—Various perception-aware planning approaches have attempted to enhance the state estimation accuracy during maneuvers, while the feature matchability among frames, a crucial factor influencing estimation accuracy, has often been overlooked. In this paper, we present APACE, an Agile and Perception-Aware trajeCtory gEneration framework for quadrotors aggressive flight, that takes into account feature matchability during trajectory planning. We seek to generate a perception-aware trajectory that reduces the error of visual-based estimator while satisfying the constraints on smoothness, safety, agility and the quadrotor dynamics. The perception objective is achieved by maximizing the number of covisible features while ensuring small enough parallax angles. Additionally, we propose a differentiable and accurate visibility model that allows decomposition of the trajectory planning problem for efficient optimization resolution. Through validations conducted in both a photorealistic simulator and real-world experiments, we demonstrate that the trajectories generated by our method significantly improve state estimation accuracy, with root mean square error (RMSE) reduced by up to an order of magnitude. The source code will be released to benefit the community<sup>1</sup>.

## I. INTRODUCTION

Visual onboard sensing has been popular on micro aerial vehicle platforms among perceptual modalities, thanks to recent advances in visual-inertial state estimation algorithms and camera’s advantages of lightweight, small size, affordable price, and low power consumption. For visual-based state estimation algorithms, such as VINS [1, 2] and ORB-SLAM2 [3], the accuracy of the estimator is significantly influenced by the captured image sequence. For instance, a perception-agnostic trajectory risks facing texture-less regions (e.g. a white wall), where diminished availability of features deteriorates state estimation and potentially results in catastrophic collisions. This motivates the development of *perception-aware planning*, which takes into consideration the impact of perception when planning trajectories, for safe and reliable maneuvers.

Although numerous works on perception-aware planning have been proposed, there are still some limitations under aggressive flight scenarios, which may lead to disastrous state estimation failure. Firstly, the aspect of feature matchability is often overlooked. Some approaches [4, 5] only focus on maximizing feature information gain, which implicitly assumes that a feature contributes to perception as long as

B. Zhou is with the School of Artificial Intelligence, Sun Yat-Sen University, Zhuhai, China. X. Chen, Y. Zhang and S. Shen are with the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong, China.

{xchenqc, yzhangec, eeshaojie}@connect.ust.hk  
zhouby23@mail.sysu.edu.cn

<sup>1</sup><https://github.com/HKUST-Aerial-Robotics/APACE>

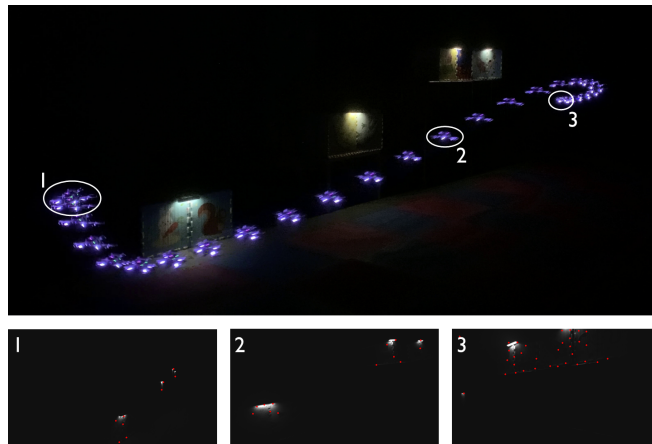


Fig. 1. A real-world aggressive flight experiment conducted in a challenging darkness environment. The LEDs on the quadrotor are only for photography purpose and disabled during the experiments. The bottom row shows snapshots of quadrotor’s first-person view at the corresponding circled positions, where the red dots represent the matched features.

it is visible. However, only those matched and triangulated features can yield relative transformation information and improve state estimation. Thus this assumption may not always hold, especially for high-speed maneuvers where the visual sensor captures rapidly changing images and the feature matchability among frames cannot be guaranteed. Secondly, there is a lack of systematic, flexible and computationally efficient perception-aware planning frameworks for quadrotor aggressive flights that simultaneously address perception objectives to improve state estimation accuracy while ensuring constraints on the smoothness, safety, agility and the quadrotor dynamics. One essential reason behind this gap is the absence of an accurate and differentiable visibility model allowing optimization decomposition with respect to position and yaw, resulting in challenging and inefficient trajectory optimization. Moreover, existing approaches considering feature matchability are mostly tailored towards tracking clustered features and may be inflexible to handle environments with features distributed in a more general pattern.

To address the above limitations, we present **APACE**, an Agile and Perception-Aware trajeCtory gEneration framework for quadrotors in this paper. Our motivation stems from the recognition that achieving accurate visual-based state estimation has two essential requirements: (1) the parallax angles of covisible features observed by the quadrotor should be sufficiently small to ensure reliable feature association; (2) the quadrotor should be able to observe adequate covisible features between consecutive frames for

feature triangulation. Hence we seek to generate trajectories that reduce the estimation error by ensuring small enough parallax angles while maximizing the number of covisible features. We observe that the parallax angle is independent of the yaw direction, which allows decomposing the optimization problem into two subproblems utilizing the differential flatness property of quadrotors. We also propose a novel accurate and differentiable visibility model, which enables the objective of covisibility to be optimized in two steps. This approach allows efficient resolution of the optimization problem, resulting in high-quality perception-aware trajectories for aggressive quadrotor flight.

Compared to previous works, our method is able to generate agile and perception-aware trajectory efficiently that achieves higher visual-based state estimation accuracy in diverse scenarios. We benchmark and validate our method in a photorealistic simulator and conduct real-world experiments. The result demonstrates that the trajectories generated by our method achieve higher state estimation accuracy and reliability during more aggressive maneuvers in challenging low-texture environments, reducing the root mean square error (RMSE) by up to an order of magnitude. In summary, the contributions of this work are:

- 1) A systematic, flexible and efficient perception-aware trajectory generation framework for quadrotor agile flight, which takes feature matchability into account.
- 2) An accurate and differentiable visibility model to evaluate the number of covisible features, which allows decomposition of the trajectory planning problem.
- 3) Method validation in a photorealistic simulator and real-world experiments. The source code will be released to benefit the community.

## II. RELATED WORK

Among recent perception-aware planning studies, information gain evaluated using various metrics has been used to bridge perception and planning. Fisher information is utilized in [4]–[6] as the metric for perception quality, which quantifies the information that the observations carry about the desired state. Bartolomei *et al.* [7] utilize semantic segmentation to avoid texture-less areas which may cause estimation failures. Sabdat *et al.* [8] suggest a perception quality metric consisting of mesh triangle density and visible surfaces viewing angle and integrates this metric into Rapidly-Exploring Random Tree [9]. These works commonly overlook feature matchability, assuming any visible features contribute to perception. However, this assumption may not always hold, especially in agile flight when the feature matchability between frames cannot be guaranteed.

Alternatively, feature matchability has been taken into consideration. Shekells *et al.* [10] optimize an execution trajectory minimizing the image feature reprojection error along the trajectory, given a goal image to align with. Falanga *et al.* [11] introduce a perception-aware model predictive control (MPC) framework, which maximizes the visibility of points of interest and minimizes their velocities in the image frame. Spasojevic *et al.* [12] propose a trajectory

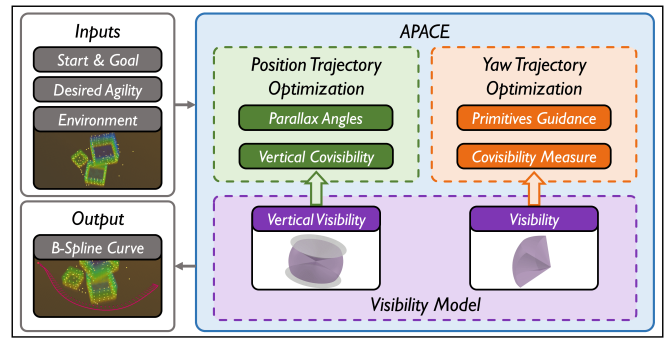


Fig. 2. An overview of the proposed trajectory generation method.

parameterization algorithm that minimizes the execution time while preserving a given set of features within the field of view. Penin *et al.* [13] solves a minimum-time optimization problem with a hard-constraint on keeping a certain set of features visible. In [14], obstacle avoidance in the environment and vision occlusion in the image space are further considered. However, the aforementioned works are commonly tailored for tracking clustered points of interest during the trajectory execution. As a result, their algorithms are not flexible enough to deploy in scenarios with features distributed in more general patterns when one needs to maintain a reduced set of features within the field of view to hit desired agility. Therefore, they may be unsuitable for our task where agent agility is preferred over target tracking.

Approaches that include feature matchability as a soft-constraint in the planning problem are also proposed. Greff *et al.* [15] propose a probabilistic model of feature matchability to be incorporated into an MPC chance constraint, ensuring the number of visible features is over a lower bound. However, their method is designed for a teach-and-repeat framework, which differs from our task. Murali *et al.* [16] propose an optimization algorithm for yaw trajectory with a predetermined position trajectory that ensures both feature covisibility and motion agility. However, perception quality can also be greatly influenced by the position trajectory, which should be considered in the optimization problem.

Various visibility functions have been proposed to model whether a feature lies in the camera fields of view. An axially symmetric conical model is deployed in [12, 13, 17], which oversimplifies the view frustum and may not be capable of handling situations when horizontal and vertical fields of view are different. In [16], the fields of view is modeled as the intersection of five half spaces, each of which corresponds to a facet of the view frustum. However, its high complexity leads to slow optimization convergence.

## III. SYSTEM OVERVIEW

Given start and goal positions and desired agility in terms of top velocity and environmental information including feature points and obstacles, we aim to generate a perception-aware trajectory that reduces the error of visual-based estimator while satisfying the constraints on smoothness, safety, agility and the quadrotor dynamics. The perception objective is achieved by maximizing the number of covisible features

and ensuring small enough parallax angles. The position and yaw trajectories are represented in the form of uniform B-spline curves (Sec.IV-A), where the control points of these curves serve as the optimization variables. Leveraging the independency of parallax angle from yaw direction (Sec.IV-B), the quadrotor's property of differential flatness (Sec.IV-C) and the decomposable visibility model proposed in Sec.V-A, we adopt a two-stage trajectory generation strategy, which includes position and yaw trajectory generation detailed in Sec.V-B and Sec.V-C respectively. The system overview is depicted in Fig.2.

#### IV. PRELIMINARIES

##### A. Uniform B-spline Curves

A B-spline curve  $\mathcal{C}(t) \in \mathbb{R}^3$  is a piecewise polynomial curve which can be uniquely defined by its **degree**  $p$ , a set of  $n+1$  **control points**  $\{\mathbf{Q}_0, \mathbf{Q}_1, \dots, \mathbf{Q}_n\}$  and a **knot vector**  $T = (t_0, t_1, \dots, t_m)$ , where  $\mathbf{Q}_i \in \mathbb{R}^3, t_j < t_{j+1}$  and  $m = n + p + 1$ . A B-spline curve  $\mathcal{C}(t)$  can be parameterized using time  $t$ , the domain of which is  $[t_p, t_{m-p}]$ . For a knot  $t_j \in T, j = p, p+1, \dots, m-p$ , the point  $\mathcal{C}(t_j)$  on the curve corresponds to  $t_j$  is referred as a **knot point**, giving the set of knot points  $\{\mathbf{P}_p, \mathbf{P}_{p+1}, \dots, \mathbf{P}_{m-p}\}$ . For a general time instance  $t \in [t_k, t_{k+1}) \subset [t_p, t_{m-p}]$ , the value  $\mathcal{C}(t)$  can be evaluated using a matrix representation [18]

$$\begin{aligned} \mathcal{C}(t) &= \mathbf{U}(t)\mathbf{M}_{p+1}(k)\mathbf{V}(k)^T, \\ \mathbf{U}(t) &= [1 \quad u(t) \quad u^2(t) \quad \dots \quad u^p(t)], \\ \mathbf{V}(k) &= [\mathbf{Q}_{k-p} \quad \mathbf{Q}_{k-p+1} \quad \dots \quad \mathbf{Q}_k], \end{aligned} \quad (1)$$

where  $u(t) = (t - t_k)/(t_{k+1} - t_k)$  and  $\mathbf{M}_{p+1}(k)$  is the  $k$ -th B-spline basis matrix of degree  $p$ . For a **uniform B-spline curve**, the **knot spans** between consecutive knots are identical, i.e.  $t_{j+1} - t_j = \Delta t$  and the basis matrix  $\mathbf{M}_{p+1}(k)$  is a constant matrix determined by the degree  $p$ .

##### B. Parallax Angle

Parallax angle is defined as the angle between the rays from a feature  $\mathbf{f}$  to two camera positions  $\mathbf{c}$  and  $\mathbf{c}'$  at different time instances, describing the apparent displacement or difference of the feature as a result of the sensor positional offset. The parallax angle is only related to the position of the sensor, and independent of its yaw direction. A large parallax angle may lead to feature matching failure [19] and deteriorate state estimation accuracy, which should be avoided during the maneuvers.

##### C. Differential Flatness

Mellinger *et al.* [20] show the differential flatness of quadrotor dynamics, which means the states and the inputs of the quadrotor can be expressed as algebraic functions of the flat outputs and their derivatives. The flat outputs are selected as  $\sigma = [x, y, z, \psi]$ , where  $\mathbf{p}_i = (x, y, z)$  is the position coordinate of the quadrotor and  $\psi$  is its yaw rotation. Differential flatness is often exploited to independently plan trajectories for the flat outputs, enabling the decomposition of the trajectory planning problem into position and yaw optimization.

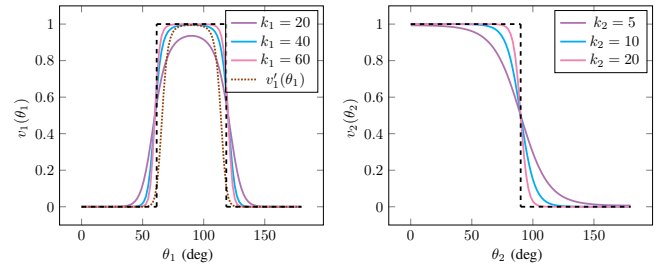


Fig. 3. Visualization of the sigmoid functions (4) and (6) with  $\alpha_v = 60^\circ$  and different  $k$  values where the black dashed lines indicate the corresponding step functions to be approximated. The brown dotted curve in the left plot illustrates the constant weight  $v'_1(\theta_1)$  used in (14).

#### V. METHODOLOGY

##### A. Visibility Model

The field of view (FoV) of a camera is commonly expressed as an angle of view pair  $(\alpha_h, \alpha_v)$ , where  $\alpha_h, \alpha_v$  are the horizontal and vertical FoV angles. For a quadrotor state  $\mathbf{s}_i = (\mathbf{p}_i, \psi_i), \mathbf{p}_i = (x_i, y_i, z_i)$  and a feature  $\mathbf{f}_j$  in world frame, the exact visibility function takes value 1 if feature  $\mathbf{f}_j$  is within camera FoV at state  $\mathbf{s}_i$  and 0 otherwise. We present a differentiable visibility model that approximates this exact visibility function more accurately and allows optimization decomposition of the covisibility objective for position and yaw trajectories.

Define  $\mathbf{n}_1$  and  $\mathbf{n}_y$  as the unit thrust direction of the quadrotor and the unit yaw direction in world frame respectively. Then

$$\mathbf{n}_1 = \frac{\mathbf{a} - \mathbf{g}}{|\mathbf{a} - \mathbf{g}|}, \quad \mathbf{n}_y = (\cos \psi, \sin \psi, 0) \quad (2)$$

where  $\mathbf{a}, \mathbf{g}$  and  $\psi$  are the acceleration, gravity and the yaw angle. For simplicity, we assume the quadrotor is equipped with a forward-looking camera and the optical axis of the camera  $\mathbf{n}_2$  is orthogonal to the thrust direction  $\mathbf{n}_1$ . The model can be generalized to different camera configuration with minor adjustments. Then  $\mathbf{n}_2$  can be calculated as

$$\mathbf{n}_3 = \frac{\mathbf{n}_1 \times \mathbf{n}_y}{|\mathbf{n}_1 \times \mathbf{n}_y|}, \quad \mathbf{n}_2 = \mathbf{n}_3 \times \mathbf{n}_1.$$

With the mutually orthogonal unit vectors  $\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3$  and the bearing vector  $\mathbf{b} = \mathbf{f}_j - \mathbf{p}_i$ , we can now define our differentiable visibility model as

$$v(\mathbf{s}_i, \mathbf{f}_j) = v_1(\theta_1) \cdot v_3(\theta_3) \cdot v_2(\theta_2), \quad (3)$$

$$v_1(\theta_1(i, j)) = (1 + e^{-k_1(\sin \theta_1 - \sin \alpha_1)})^{-1}, \quad (4)$$

$$v_3(\theta_3(i, j)) = (1 + e^{-k_3(\sin \theta_3 - \sin \alpha_3)})^{-1}, \quad (5)$$

$$v_2(\theta_2(i, j)) = (1 + e^{-k_2(\cos \theta_2 - \cos \alpha_2)})^{-1}, \quad (6)$$

where  $\alpha_1 = (\pi - \alpha_v)/2, \alpha_2 = \pi/2, \alpha_3 = (\pi - \alpha_h)/2$ , and  $\theta_1, \theta_2, \theta_3$  are the angles between the bearing vector  $\mathbf{b}$  and the corresponding unit vectors  $\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3$  respectively as shown in Fig.4(a-c). Then

$$\sin \theta_1 = \frac{|\mathbf{n}_1 \times \mathbf{b}|}{|\mathbf{b}|}, \quad \cos \theta_2 = \frac{\mathbf{n}_2 \cdot \mathbf{b}}{|\mathbf{b}|}, \quad \sin \theta_3 = \frac{|\mathbf{n}_3 \times \mathbf{b}|}{|\mathbf{b}|}. \quad (7)$$

Intuitively, the idea behind this visibility model construction is a deductive scheme as illustrated in Fig. 4. In the

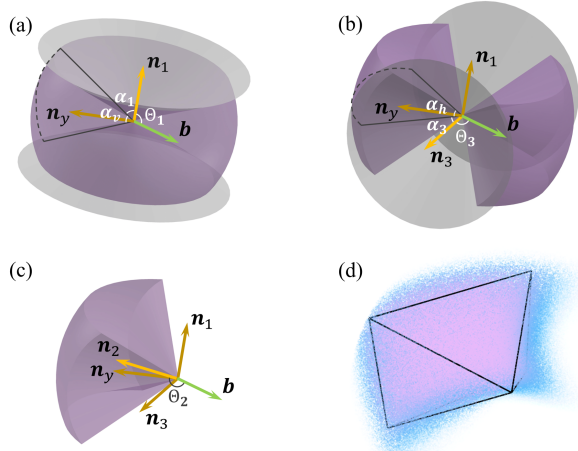


Fig. 4. Illustration of the proposed visibility model under FoV setting  $(\alpha_h, \alpha_v) = (90^\circ, 60^\circ)$ . The violet shape in (a-c) represents visible regions. (d) visualizes the proposed visibility model by dense sampling. Points with visibility model take values close to zero are transparent and colored blue, and values close to one are opaque and colored pink.

beginning, a sphere is used to represent omni-directional sensing within FoV depth. Then the visibility components  $v_1(\theta_1)$  and  $v_3(\theta_3)$  carve out two pairs of axially symmetric cones in grey as illustrated in Fig. 4(a) and Fig. 4(b), accounting for the vertical and horizontal FoV respectively. Consequently, we name  $v_1(\theta_1)$  and  $v_3(\theta_3)$  as *vertical visibility* and *horizontal visibility*. Lastly,  $v_2(\theta_2)$  is used to choose the correct pyramidal shape among the remaining two according to  $\theta_2$  resulting in Fig. 4(c). In (4)-(6), we use sigmoid functions to approximate the step functions. We illustrate (4) and (6) in Fig. 3 for different  $k$  and we opt for  $(k_1, k_2, k_3) = (40, 10, 20)$  in our implementation.

Moreover, we can further induce the covisibility measure for states  $\mathbf{s}_1$  and  $\mathbf{s}_2$  as

$$\mu(\mathbf{s}_1, \mathbf{s}_2) = \sum_{j \in J} v(\mathbf{s}_1, \mathbf{f}_j) v(\mathbf{s}_2, \mathbf{f}_j), \quad (8)$$

where  $J$  is the set of features within FoV depth of both states and  $v(\mathbf{s}_i, \mathbf{f}_j)$  is the visibility model specified in (3).

We decompose the optimization of the covisibility measure into two phases based on the observation that  $\theta_1$  is independent of the yaw direction. The reason is that  $\theta_1$  depends only on  $\mathbf{n}_1$  and  $\mathbf{b}$  according to (7), which can be completely determined by the position trajectory and its derivatives. In the first phase where we optimize the position trajectory, we aim to maximize the *vertical covisibility* considering  $\theta_1$  to include as many covisible features as possible within the ring-shape FoV (Fig.4(a)) along the trajectory. In the second phase, we optimize the yaw trajectory given the position trajectory, which yields the best sector (Fig.4(c)) to further maximize the covisibility measure  $\mu$ . The two-phase strategy reduces the computational complexity of the covisibility optimization problem, facilitating efficient convergence towards high-quality trajectories. Although this two-step optimization may not guarantee maximum covisibility, it is capable of generating a sufficiently informative trajectory to support accurate state estimation during maneuvers.

## B. Position Trajectory Optimization

The position trajectory is represented using a degree  $p = 3$  uniform B-spline curve defined by  $n + 1$  control points  $\{\mathbf{Q}_0, \mathbf{Q}_1, \dots, \mathbf{Q}_n\}$  and a uniform knot span  $\Delta t$ , both of which serve as the variables in the position trajectory optimization. The initial value of the trajectory is set as an initial B-spline curve that fits the collision-free path provided by standard method like A\* and RRT\*. The overall cost function objective is defined as

$$f_{\text{total}} = \lambda_{\text{vc}} f_{\text{vc}} + \lambda_{\text{para}} f_{\text{para}} + \lambda_o f_o, \quad (9)$$

where  $f_{\text{vc}}$  and  $f_{\text{para}}$  are the vertical covisibility and the parallax cost.  $f_o$  is the cost function for other factors including smoothness, safety, agility and dynamic feasibility. These factors are traded off by  $\lambda_{\text{para}}, \lambda_{\text{vc}}, \lambda_o$ .

We define the vertical covisibility cost utilizing the angle  $\theta_1$  in the visibility model introduced in Sec. V-A. For a pair of adjacent knot points  $\mathbf{P}_i; \mathbf{P}_{i+1}$  (defined in Sec.IV-A) and a feature  $\mathbf{f}_j$ , we first calculate their vertical visibility angles  $\theta_1(i, j)$  and  $\theta_1(i + 1, j)$  respectively. Here,  $\theta_1(k, j)$  is the angle between the thrust direction  $\mathbf{n}_1(k)$  at knot points  $\mathbf{P}_k$  and the bearing vector  $\mathbf{b}(k, j) = \mathbf{f}_j - \mathbf{P}_k$ , where  $k = i, i + 1$ , i.e.

$$\theta_1(k, j) = \arccos \frac{\mathbf{n}_1(k) \cdot \mathbf{b}(k, j)}{\|\mathbf{n}_1(k)\| \|\mathbf{b}(k, j)\|}. \quad (10)$$

Then the vertical covisibility cost  $f_{\text{vc}}$  is calculated as

$$f_{\text{vc}} = \sum_{i \in I} \sum_{j \in J_i} \underbrace{(1 + g(\theta_1(i, j)))(1 + g(\theta_1(i + 1, j)))}_{h(i, j)} - 1, \quad (11)$$

where  $I$  is the set of knot points,  $J_i$  is the set of features within FoV depth of both  $\mathbf{P}_i$  and  $\mathbf{P}_{i+1}$ , and  $g(\theta_1(i, j))$  is a differentiable potential function on  $\theta_1(i, j)$  defined as

$$g(\theta_1(k, j)) = \begin{cases} a_1(\theta_1(k, j) - \theta_{1, \max})^2, & \text{if } \theta_1(k, j) > \theta_{1, \max} \\ a_1(\theta_1(k, j) - \theta_{1, \min})^2, & \text{if } \theta_1(k, j) < \theta_{1, \min} \\ 0, & \text{otherwise} \end{cases} \quad (12)$$

where  $\theta_{1, \max} = (\pi - \alpha_v)/2, \theta_{1, \min} = (\pi + \alpha_v)/2$ , and  $a_1$  is a scalar constant. When the feature  $\mathbf{f}_j$  is vertically covisible from both  $\mathbf{P}_i$  and  $\mathbf{P}_{i+1}$ , the cost function  $h(i, j)$  takes value zero. Otherwise, it takes a positive value, penalizing the loss of vertical covisibility.

For the parallax cost  $f_{\text{para}}$ , we first calculate the parallax angle  $\rho(i, j)$  for adjacent knot points  $\mathbf{P}_i, \mathbf{P}_{i+1}$  and a feature  $\mathbf{f}_j$  as

$$\rho(i, j) = \arccos \frac{(\mathbf{P}_i - \mathbf{f}_j) \cdot (\mathbf{P}_{i+1} - \mathbf{f}_j)}{\|\mathbf{P}_i - \mathbf{f}_j\| \|\mathbf{P}_{i+1} - \mathbf{f}_j\|}. \quad (13)$$

Then the parallax cost  $f_{\text{para}}$  is defined as

$$f_{\text{para}} = \sum_{i \in I} \sum_{j \in J_i} w_{i, j} h(\rho(i, j)), \quad (14)$$

where the constant weight  $w_{i, j} = v'_1(\theta_1(i, j))$  with parameter  $k'_1 = 60, \alpha'_1 = (\pi - 0.8\alpha_v)/2$  (shown in Fig. 3) is used, giving lower weights to features near vertical FoV boundary and negligible weights to vertically invisible ones.  $h(\rho(i, j))$

is a differentiable potential function penalizing large parallax angle which exceeds threshold  $\rho_{\text{thr}}$ ,

$$h(\rho(i, j)) = \begin{cases} a_2(\rho(i, j) - \rho_{\text{thr}})^2, & \text{if } \rho(i, j) > \rho_{\text{thr}} \\ 0, & \text{otherwise} \end{cases} \quad (15)$$

where  $a_2$  is a scalar constant and  $\rho_{\text{thr}} = \nu \rho_{\text{max}} \Delta t$  with image frame frequency  $\nu$ , the acceptable maximum parallax angle between consecutive frames  $\rho_{\text{max}}$ , and the knot span  $\Delta t$ .

The formulation of the cost function  $f_o$  is similar to [21] and [22]. The smoothness, safety, agility and dynamic feasibility are achieved by minimizing jerk, collision risk, execution time and infeasible velocity and acceleration penalty respectively.

### C. Yaw Trajectory Generation

We first search a yaw primitives path (Sec.V-C.1), which serves as a guidance for the subsequent yaw trajectory optimization (Sec.V-C.2), avoiding local minima trap.

1) *Primitives Search*: We model the problem as a graph search problem which seeks to maximize the cumulative covisibility measure along the trajectory while ensuring yaw dynamics feasibility. At each knot point  $\mathbf{P}_i \in \{\mathbf{P}_p, \mathbf{P}_{p+1}, \dots, \mathbf{P}_{m-p}\}$  of the optimized position trajectory obtained from Sec.V-B, a set of yaw angles  $L_i = \{\psi_{i,0}, \psi_{i,1}, \dots, \psi_{i,l}\}$  are sampled uniformly, each of which serves as a graph node. An edge is built between two nodes  $\psi_{i_1, j_1}$  and  $\psi_{i_2, j_2}$  if and only if they belong to consecutive sets and fulfill the yaw dynamics, i.e.  $|i_1 - i_2| = 1$  and  $|\psi_{i_1, j_1} - \psi_{i_2, j_2}| < \dot{\psi}_{\text{max}} \Delta t$ . For each edge, the gain associated with it is calculated as the covisibility measure according to (8). Since there is no constraint on the start and end yaw values, we add two extra graph nodes  $\psi_{p-1,0}, \psi_{m-p+1,0}$  and connect them with all nodes in set  $L_p$  and  $L_{m-p}$  respectively with zero-gain edges. Then the problem is formulated as a shortest path problem from node  $\psi_{p-1,0}$  to  $\psi_{m-p+1,0}$ , where the cost of each edge is the negative of the associated gain, and it can be efficiently solved by the Dijkstra algorithm.

2) *Trajectory Optimization*: The yaw trajectory  $\psi(t)$  is also represented as a degree  $p = 3$  uniform B-spline curve defined by a same number of control points as the position trajectory  $\Xi = \{\xi_0, \xi_1, \dots, \xi_n\}$  and a same knot span  $\Delta t$ . Then the yaw knot points  $\{\psi_p, \psi_{p+1}, \dots, \psi_{m-p}\}$  correspond to the same time instance to the yaw primitives  $\Psi^* = \{\psi_p^*, \psi_{p+1}^*, \dots, \psi_{m-p}^*\}$  obtained from Sec.V-C.1. Then the optimization problem is formulated as

$$\arg \min_{\Xi} \int_{t_p}^{t_{m-p}} \ddot{\psi}(t)^2 dt + \sum_{i \in I} (\lambda_1 (\psi_i - \psi_i^*)^2 - \lambda_2 \mu(\mathbf{s}_i, \mathbf{s}_{i+1})), \quad (16)$$

Here the first term minimizes the jerk and ensures smoothness. The second term accounts for yaw primitives guidance. For the covisibility term, the definitions (3)(8) gives

$$\begin{aligned} \mu(\mathbf{s}_i, \mathbf{s}_{i+1}) &= \sum_{j \in J_i} v(\mathbf{s}_i, \mathbf{f}_j) v(\mathbf{s}_{i+1}, \mathbf{f}_j) \\ &= \sum_{j \in J_i} \underbrace{v_1(i) v_1(i+1) v_2(i) v_2(i+1) v_3(i) v_3(i+1)}_{\text{constant}}, \end{aligned} \quad (17)$$

TABLE I  
ESTIMATION ACCURACY BENCHMARK

Top Vel	Method	Path (m)	Time (s) <sup>◇</sup>	Fail Rate <sup>‡</sup>	GE(m) <sup>†</sup>		RMSE(m)	
					Avg	Std	Avg	Std
2.0 m/s	PAG [21]	18.3	<b>0.05</b>	10/10	-	-	-	-
	PAW [16]	19.9	4.17	1/10	3.23	2.36	1.69	1.16
	Ours	19.9	0.24	<b>0/10</b>	<b>0.12</b>	<b>0.04</b>	<b>0.13</b>	<b>0.03</b>
4.0 m/s	PAG [21]	18.6	<b>0.11</b>	10/10	-	-	-	-
	PAW [16]	19.9	4.17	8/10	7.98	1.75	4.15	1.06
	Ours	20.9	0.25	<b>0/10</b>	<b>0.14</b>	<b>0.09</b>	<b>0.16</b>	<b>0.04</b>

<sup>◇</sup> Average trajectory generation time.

<sup>†</sup> GE abbreviates for goal error.

<sup>‡</sup> If GE exceeds 10m, the task is deemed fail and the result will not be counted towards GE and RMSE statistics.

where we write  $v_k(\theta_k(i, j))$  as  $v_k(i)$  for simplicity without ambiguity. Thanks to the decomposable visibility model, we can treat the first four terms of the covisibility as a constant and only optimize the last two terms with respect to yaw.

## VI. EXPERIMENTS

### A. Experimental Setup

For simulation experiments, we utilize a photorealistic simulator AirSim [23, 24]. All simulation experiments have been run on a computer equipped with an Intel Core i7-13700F, GeForce RTX 3060 12G and 32GB memory with identical input feature cloud. For real-world experiments, we use a custom built light-weight and high-agility quadrotor platform, which has a take-off weight of 861g and thrust-to-weight ratio  $\sim 5.3$ . The quadrotor is equipped with an NVIDIA Jetson Orin NX 16 GB and an NxtPx4 autopilot<sup>2</sup>. The visual data is provided by a forward-looking global-shutter camera RealSense<sup>3</sup> with 86° horizontal FoV and 57° vertical FoV. The odometry ground truth is provided by OptiTrack motion capture system, which is also utilized as controller feedback. In both simulation and real-world experiments, we adopt VINS-Fusion [2] as our visual state estimator, where inertial measurement unit (IMU) information is excluded to eliminate its contribution to the visual odometry (VO). For implementation details, we use  $a_1 = 10, a_2 = 20, \nu = 20\text{Hz}, \rho_{\text{max}} = 10^\circ$  and solve the optimization with a general nonlinear solver<sup>4</sup>.

### B. Photorealistic Simulation Benchmarks

We perform benchmark comparison with a perception-agnostic (PAG) planner [21] and a state-of-the-art perception-aware (PAW) planner [16]. As there is no open-source code available for [16], we use our implementation. The experiment setup consists of a pile of texture-rich boxes in the middle of a textureless room as shown in Fig.5. The quadrotor is commanded to execute each trajectory in different agilities in terms of top velocity for 10 runs and the result statistics are shown in Table I. We also illustrate the generated trajectories along with those estimated by VO in Fig.5. As the position trajectories from [21] [16] are not optimized for perceptual objectives, they tend to take short routes which pass too

<sup>2</sup><https://github.com/HKUST-Aerial-Robotics/NxtPX4>

<sup>3</sup><https://www.intelrealsense.com/depth-camera-d435>

<sup>4</sup><https://nlopt.readthedocs.io/en/latest>

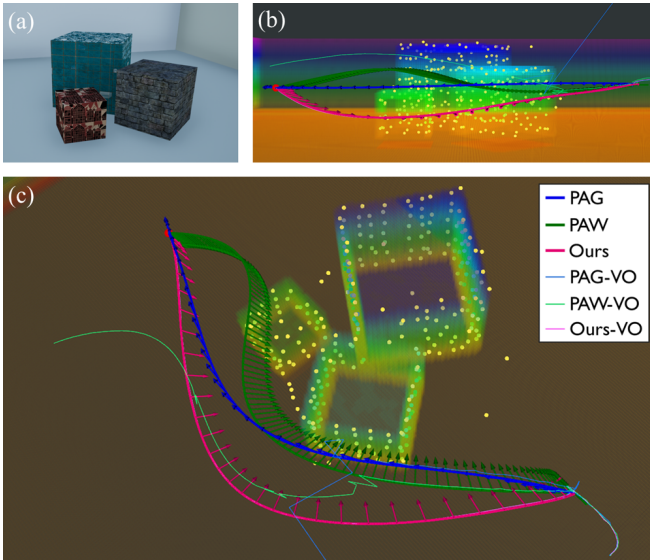


Fig. 5. The setup in the photorealistic simulator Airsim is shown in (a). The trajectories generated by PAG [21], PAW [16] with top velocity  $2.0m/s$  and our planner with top velocity  $4.0m/s$  along with the visual odometries (VO) are illustrated from side and top views in (b) and (c) respectively. The yaw directions are visualized as arrows at interval of  $0.2s$ .

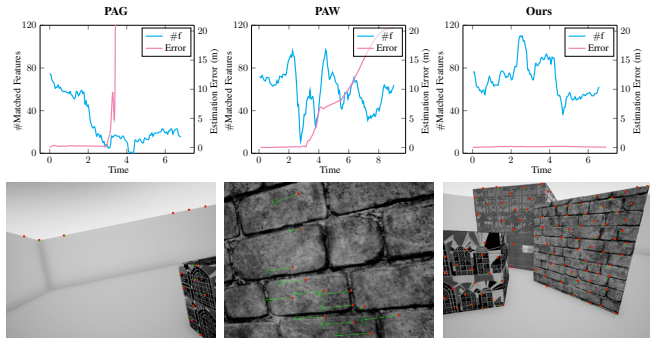


Fig. 6. The top row shows plots of matched features number and state estimation error over time for the three trajectories, i.e. PAG [21], PAW [16] and ours, at a top velocity of  $4.0m/s$  respectively from left to right. The bottom row showcases corresponding snapshots of quadrotor’s first-person view, where the red dots represent the matched features and the green arrows indicate the pixel of matched features from the previous frame.

close to the grid-texture grey box resulting in large parallax angles. The perception-aware yaw trajectory from [16] barely allowed task completion, yielding poor localization accuracy, while the head-forwarding yaw trajectory from [21] caused the quadrotor to face the white wall, leading to task failure. In contrast, the trajectory given by us is able to maintain a reasonable distance from the grey box to ensure small enough parallax angles and includes the small red box in the FoV to ensure sufficient covisible features for localization. This resulted in a sufficient and stable number of matched features as shown in Fig. 6 and consequently accurate state estimation with low goal error and RMSE. Moreover, our method is capable of generating trajectories efficiently in less than  $0.25s$ , thanks to the decomposable visibility model, while [16] suffers from computational complexity due to a large number of features.

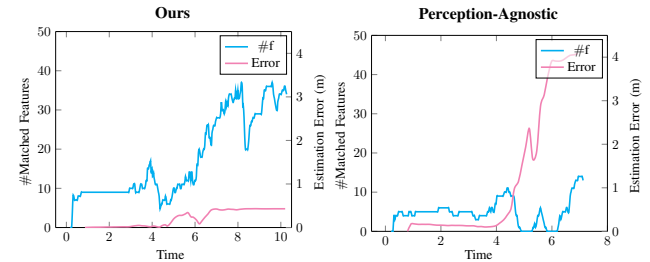
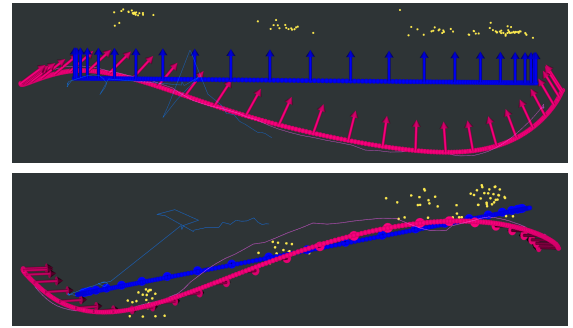


Fig. 7. The figures show the trajectory generated by our methods in pink and a perception-agnostic trajectory in blue at top velocity of  $3.7m/s$  along with their VO results from top and side views. The start and goal of the perception-agnostic trajectory have been adjusted to guarantee existence of visible features at these two points. The bottom row shows plots of matched features number and state estimation error over time for the two trajectories.

### C. Real-world Experiments

We further conduct experiments with a real quadrotor in a challenging darkness scenario with four illuminated spots distributed at different height levels as depicted in Fig.1. Without perception-aware planning, a visual state estimator would not be able to sustain reliable performance in such a severe environment since the VO would start drifting drastically once the quadrotor encounters blindly complete darkness without any illuminated features. As exemplified in Fig. 7, we showcase a perception-agnostic trajectory and the plots reveal that it encounters zero features, resulting in task failure. In contrast, our perception-aware trajectory of  $11.3m$  length maintains a sufficient number of matched features and yields accurate VO with an average goal error of  $0.54m$  and RMSE of  $0.40m$  in 5 runs. The results validate the feasibility of our perception-aware planner under challenging low-texture environments, thereby expanding the application boundaries of visual state estimation techniques.

## VII. CONCLUSIONS

This paper presents a perception-aware trajectory generation method for quadrotors aggressive flight. Our approach takes the feature matchability into consideration by maximizing the number of covisible features and maintaining small-enough parallax angles. Leveraging the decomposable visibility model proposed, we adopt a two-stage strategy to generate perception-aware trajectories that enhances visual-based estimator accuracy while satisfying the constraints on smoothness, safety, agility and the quadrotor dynamics. Our method is validated in photorealistic simulations and real-world experiments, demonstrating significant improvement in state estimation accuracy with root mean square error (RMSE) reduced by up to an order of magnitude.

## REFERENCES

- [1] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot. (TRO)*, vol. 34, no. 4, pp. 1004–1020, 2018.
- [2] T. Qin, J. Pan, S. Cao, and S. Shen, "A general optimization-based framework for local odometry estimation with multiple sensors," *arXiv preprint arXiv:1901.03638*, 2019.
- [3] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.
- [4] Z. Zhang and D. Scaramuzza, "Perception-aware receding horizon navigation for mavs," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 2534–2541, 2018.
- [5] G. Costante, J. Delmerico, M. Werlberger, P. Valigi, and D. Scaramuzza, "Exploiting photometric information for planning under uncertainty," in *Robotics Research*, A. Bicchi and W. Burgard, Eds. Cham: Springer International Publishing, 2018, vol. 2, pp. 107–124.
- [6] D. Kim, G. C. Kim, Y. Jang, and H. J. Kim, "Topology-guided path planning for reliable visual navigation of mavs," *IEEE International Conference on Intelligent Robots and Systems*, pp. 3117–3124, 2021.
- [7] L. Bartolomei, L. Teixeira, and M. Chli, "Perception-aware path planning for uavs using semantic segmentation," *IEEE International Conference on Intelligent Robots and Systems*, pp. 5808–5815, 2020.
- [8] S. A. Sadat, K. Chutskoff, D. Jungic, J. Wawerla, and R. Vaughan, "Feature-rich path planning for robust navigation of mavs with monoslam," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. Hong Kong, China: IEEE, May 2014, pp. 3870–3875.
- [9] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research*, vol. 30, pp. 846–894, 2011.
- [10] M. Shekells, G. Garimella, and M. Kobilarov, "Optimal visual servoing for differentially flat underactuated systems," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 5541–5548.
- [11] D. Falanga, P. Foehn, P. Lu, and D. Scaramuzza, "Pampc: Perception-aware model predictive control for quadrotors," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1–8.
- [12] I. Spasojevic, V. Murali, and S. Karaman, "Perception-aware time optimal path parameterization for quadrotors," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 3213–3219.
- [13] B. Penin, R. Spica, P. R. Giordano, and F. Chaumette, "Vision-based minimum-time trajectory generation for a quadrotor uav," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 6199–6206.
- [14] B. Penin, P. R. Giordano, and F. Chaumette, "Vision-based reactive planning for aggressive target tracking while avoiding collisions and occlusions," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3725–3732, 2018.
- [15] M. Greeff, T. D. Barfoot, and A. P. Schoellig, "A perception-aware flatness-based model predictive controller for fast vision-based multirotor flight," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 9412–9419, 2020.
- [16] V. Murali, I. Spasojevic, W. Guerra, and S. Karaman, "Perception-aware trajectory generation for aggressive quadrotor flight using differential flatness," *Proceedings of the American Control Conference*, vol. 2019-July, pp. 3936–3943, 2019.
- [17] Z. Zhang and D. Scaramuzza, "Beyond point clouds: Fisher information field for active visual localization," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2019-May, pp. 5986–5992, 2019.
- [18] K. Qin, "General matrix representations for b-splines," *The Visual Computer*, vol. 16, no. 3, pp. 177–186, 2000.
- [19] S. Song, D. Kim, and S. Choi, "View path planning via online multiview stereo for 3-d modeling of large-scale structures," *IEEE Transactions on Robotics*, vol. 38, no. 1, pp. 372–390, 2021.
- [20] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Proc. of the IEEE Intl. Conf. on Robot. and Autom. (ICRA)*, Shanghai, China, May 2011, pp. 2520–2525.
- [21] B. Zhou, F. Gao, L. Wang, C. Liu, and S. Shen, "Robust and efficient quadrotor trajectory generation for fast autonomous flight," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3529–3536, 2019.
- [22] B. Zhou, J. Pan, F. Gao, and S. Shen, "Raptor: Robust and perception-aware trajectory replanning for quadrotor fast flight," *arXiv preprint arXiv:2007.03465*, 2020.
- [23] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity virtual and physical simulation for autonomous vehicles," in *Field and Service Robotics: Results of the 11th International Conference*. Springer, 2018, pp. 621–635.
- [24] Z. Han, Z. Wang, N. Pan, Y. Lin, C. Xu, and F. Gao, "Fast-racing: An open-source strong baseline for se(3) planning in autonomous drone racing," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 8631–8638, 2021.