

Perception-aware receding horizon trajectory planning for multicopters with visual-inertial odometry

Xiangyu Wu¹, Shuxiao Chen², Koushil Sreenath², and Mark W. Mueller¹

Abstract—Visual inertial odometry (VIO) is widely used for the state estimation of multicopters, but it may function poorly in environments with few visual features or in overly aggressive flights. In this work, we propose a perception-aware collision avoidance trajectory planner for multicopters, that may be used with any feature-based VIO algorithm. Our approach is able to fly the vehicle to a goal position at high speed, avoiding obstacles in an unknown stationary environment while achieving good VIO state estimation accuracy. The proposed planner samples a group of minimum jerk trajectories and finds collision-free trajectories among them, which are then evaluated based on their speed to the goal and perception quality. Both the motion blur of features and their locations are considered for the perception quality. Our novel consideration of the motion blur of features enables automatic adaptation of the trajectory’s aggressiveness under environments with different light levels. The best trajectory from the evaluation is tracked by the vehicle and is updated in a receding horizon manner when new images are received from the camera. Only generic assumptions about the VIO are made, so that the planner may be used with various existing systems. The proposed method can run in real time on a small embedded computer on board. We validated the effectiveness of our proposed approach through experiments in both indoor and outdoor environments. Compared to a perception-agnostic planner, the proposed planner kept more features in the camera’s view and made the flight less aggressive, making the VIO more accurate. It also reduced VIO failures, which occurred for the perception-agnostic planner but not for the proposed planner. The ability of the proposed planner to fly through dense obstacles was also validated. The experiment video can be found at <https://youtu.be/qO3LZlrpwtQ>.

I. INTRODUCTION

Multicopters are useful for a wide range of applications such as aerial photography [1] inspection [2], search and rescue [3], and transportation [4] thanks to their simple design and high maneuverability. State estimation is necessary for these applications, which often use onboard sensors such as the GPS [5], camera [6], Lidar [7], and inertial measurement unit [8]. For indoor applications where special localization infrastructures can be deployed, a motion capture system [9] or an ultra-wideband system [10] could also be used. Among state estimation methods, visual inertial odometry (VIO) is a popular solution: it only requires light-weight, low-power, and low-cost onboard sensors – cameras and inertial measurement units (IMUs), suitable even for small aerial robots [11]. Additionally, VIO does not require other infrastructure in the operating environment. These advantages make it especially useful in applications where the GPS signal is unreliable, such as indoors, in the forest, or near tall buildings.

¹Authors are with the High Performance Robotics Laboratory (HiPeRLab), UC Berkeley. {wuxiangyu, mwm}@berkeley.edu

²Authors are with the Hybrid Robotics Group, UC Berkeley. {shuxiao.chen, koushils}@berkeley.edu



Figure 1. The perception-aware planner guides the quadcopter to fly close to areas with more visual features for better VIO accuracy – the paved road has much fewer visual features than the trees. The goal is 20 meters forward from the starting position of the vehicle, which is in the direction away from the camera taking this photo.

However, VIO may struggle when the vehicle flies in areas with few visual features or when the motion of the vehicle becomes too aggressive. As a result, the trajectory planning of a vehicle should include perception-awareness: it should consider not only the goal of the mission, but also the trajectory’s impact on the VIO. This topic has drawn increased research interest over the past few years [12].

One major category of perception-aware planners in the literature plan multicopter trajectories by solving an optimization problem, encoding the perception-awareness as a cost term or constraint. In [13], the authors add the visibility of features as a constraint in the optimization of B-spline trajectories. The differential flatness property of multicopters is used to speed up the optimization. The method is validated in simulation. In [14], [15], the vehicle’s trajectories are planned while maintaining a given set of landmarks within the field of view of its on-board camera. The first step is geometric path planning, followed by a time parameterization of the planned path to satisfy the kinodynamic constraints of the quadcopter. In addition to the visibility of the features, maximizing the covisibility of features is also helpful in reducing the state estimation error [16]. The goal is to keep features visible in the camera field of view from one keyframe to the next, instead of to maximize the visibility of features in each image. The authors first plan a minimum snap trajectory with only position, and then the yaw angle is planned to maximize the features’ covisibility. Indoor experiments are shown, with the locations of the landmarks in the environment are known a priori. Authors of [17] additionally take the

feature’s movement speed into account by adding a perception cost term to reduce the movement speed of the feature points’ centroid in the image and keep it close to the image center. Model predictive control (MPC) is used for trajectory planning and the optimization is accelerated by a sequential quadratic program (SQP) approximation. In [18], semantic information is used to plan trajectories in areas with more texture and to avoid places with unreliable visual features, such as lakes. Simulation results are shown with the semantic information from the ground truth of the simulator.

Another major category of perception-aware planners use sampling-based methods, where the sampled trajectories are evaluated based on both the mission goal and their impact on state estimation. In [19], the authors use the rapidly exploring random belief tree (RRBT) approach to evaluate multiple candidate paths in a known map and select paths with minimum pose uncertainty. The evaluation of the paths is done offline before the flight because of high computational cost. The pose covariance of the vehicle is estimated via bundle adjustment, by minimizing the reprojection errors of the 3D map points across all images (augmented by noise with constant covariance). The authors of [20] propose a planner that prefers feature-rich regions, by introducing a view-point score based on the visibility of the visual features and incorporating this score in RRT*. Perception-awareness is considered in [21], [22] to improve the mapping and state estimation accuracy during quadcopter exploration. The outer layer planner generates paths that explore the space using the rapidly-exploring random tree (RRT), and the inner layer planner aims to improve the mapping and state estimation accuracy. The authors propagate the state estimation for different paths found by the inner layer planner, and choose the one which minimizes the state estimation uncertainty. In [23], the task of reaching a given goal with the highest accuracy while avoiding obstacles in the environment is investigated. The planner generates candidate trajectories and evaluates them in terms of perception quality, collision probability, and distance to the goal. Given each sampled trajectory, the authors simulate the observed features if the trajectory is followed and construct a least squares problem to estimate the vehicle’s pose estimation error.

Unlike most of the work in the literature using feature-based methods for localization, the authors in [24] propose a perception-aware planner for direct visual inertial odometry. The direct VIO method estimates the vehicle’s pose based on the image pixels’ intensities and an introduction of it can be found at e.g. [25]. They adapt the RRT* framework to select trajectories that minimize the camera pose uncertainty according to photometric information (computed with dense image-to-model alignment). In the aforementioned papers, most use indoor experiments or simulations to validate their proposed methods, while both indoor and outdoor experiments are conducted in [19], [22].

In this work, we focus on the problem of flying a multi-copter to a desired position at a high speed, avoiding obstacles in the environment, while achieving good state estimation accuracy from the VIO. A stereo depth camera with an IMU is used for collision avoidance and VIO. To quickly

check if a trajectory is collision-free, we use a sampling-based planner named RAPPIDS [26], [27]. In the next step, given a sampled trajectory that is collision-free, we predict the pose of the vehicle (assuming perfect trajectory tracking) and then the position and velocity of the VIO features in the camera frame. The vehicle pose estimation is constructed as a least-squares problem, and each feature’s variance is estimated from its velocity in the image. We then evaluate the perception cost of the trajectory based on the vehicle’s predicted position estimation uncertainty if that trajectory is followed. In addition, the speed cost of the trajectory is the negative value of its average speed towards the goal. The trajectory that minimizes the perception cost plus the speed cost is selected as the trajectory to follow. Our planner runs in a receding horizon manner, and it replans every time a new image arrives.

Our proposed planner can reduce the state estimation error of the VIO by planning trajectories that guide the vehicle towards feature-rich areas and by preventing the vehicle from executing overly aggressive trajectories (causing motion blur). Meanwhile, the trajectories it plans are collision-free and dynamically feasible. The planner is also computationally efficient enough to run on an onboard embedded computer in real-time. Compared with the existing works in the literature, the contributions of this work are:

- 1) We propose a trajectory planner generating collision-free trajectories that guides the vehicle towards the target in unknown environments, while avoiding regions with few visual features, preventing overly aggressive flights, and avoiding obstacles.
- 2) We propose a perception cost function considering both the motion blur of the features and their locations.
- 3) Automatic adaptation of the trajectory’s aggressiveness under environments with different light levels.
- 4) Experimental validation in both indoor and outdoor environments with the algorithm running onboard in real-time, validating the effectiveness of the proposed method.

II. SYSTEM OVERVIEW

In this section, we give a brief overview of the proposed perception-aware planning system, a block diagram of which is shown in Fig. 2. The goal is to fly the drone to a goal position at a fast speed, avoid obstacles in the way, and achieve a good state estimation accuracy from the VIO. The goal position input to the perception-aware planner could be given directly by the user, or be given by a high-level global planner when used in an autonomous navigation framework. The environment is assumed to be stationary.

The perception-aware planner uses the depth images from the stereo depth camera to detect obstacles in the environment. We use the RAPPIDS planner [26], [27], a memory-less sampling-based planner, to generate a group of collision-free candidate trajectories at low computational cost. We use OpenVINS [28] for VIO, which uses monocular images from the depth camera and IMU measurements to estimate the state of the vehicle. It uses the monocular images from the left and right cameras on the stereo depth camera directly, instead

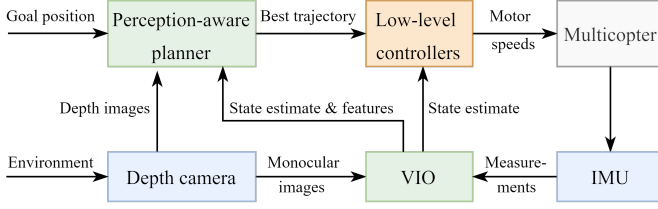


Figure 2. Block diagram of the proposed perception-aware planning system for multicopters. The perception-aware planner and VIO (with green background) run on the onboard computer. The goal position for the perception-aware planner could be given by the user, or by a high-level global planner.

of using the depth image for state estimation. Our proposed planner does not depend on a specific VIO method, and other feature-based VIO algorithms can also be used with it. The VIO also sends the 3D positions of the tracked features in the world frame to the perception-aware planner to evaluate the perception cost c_{perc} of each candidate trajectory.

One of the major contributions of this paper is the proposal of a novel perception-aware cost c_{perc} , whose derivation is detailed in Section III. The derived cost term not only considers the number and position of the VIO feature points, but also their uncertainty from motion blur. With this cost term, our proposed planner plans trajectories leading to reduced VIO state estimation uncertainty. Such trajectories guide the vehicle to fly close to areas with visual features to keep more feature points in the camera’s FOV, and they are not overly aggressive to cause excessive motion blur of the visual features. The camera’s exposure time is also considered in the perception cost: the cost term will be larger in darker environments where a longer exposure time makes images more prone to motion blur. This makes the planner prefer less-aggressive candidate trajectories in darker environments to prevent poor VIO accuracy.

In addition, we add a flight speed related cost c_{speed} (defined in Section IV-B) to encourage fast flight towards the goal. A parameter k_{perc} is introduced to determine the weight of the perception quality, and the collision-free trajectory with the minimum total cost $k_{\text{perc}}c_{\text{perc}} + c_{\text{speed}}$ is chosen as the best trajectory. The best trajectory is then tracked by the low-level position and attitude controllers of the multicopter.

The perception-aware planner runs in a receding horizon manner, and it replans every time a new depth image is received. The tracked trajectory is updated if a lower-cost trajectory is found. Recursive feasibility is guaranteed: all sampled trajectories have zero speed and acceleration at the end, and the planner assumes no other visual features except those already found by the VIO. As a result, in the worst case, the vehicle will follow the current trajectory to the end and stop safely if no new trajectories are found, avoiding collision or flying into areas without features.

III. PERCEPTION-AWARE COST DERIVATION

In this section, we introduce the derivation of the perception cost that helps to reduce the state estimation uncertainty. We define the world frame W , body frame B , and camera frame C , as shown in Fig. 3. Vector and matrix variables are written

in boldface. The notations used in this section are summarized in Table I.

Table I
NOTATIONS USED IN SECTION III.

| Symbol | Meaning |
|--|--|
| $c_{\text{perc}}, c_{\text{speed}}, c_{\text{tot}}$ | perception, speed, and total cost |
| k_{perc} | weight coefficient of perception cost |
| W, B, C | world frame, vehicle body frame, camera frame |
| $\mathbf{T}^{WB} = [\mathbf{R}^{WB} \ \mathbf{t}^{WB}]$ | pose of the vehicle body frame B in the world frame W |
| $\mathbf{T}^{BC} = [\mathbf{R}^{BC} \ \mathbf{t}^{BC}]$ | pose of the camera frame C in vehicle body frame B (rigid transformation) |
| $\mathbf{T}^{CW} = [\mathbf{R}^{CW} \ \mathbf{t}^{CW}]$ | camera extrinsics: pose of the world frame W in the camera frame C |
| subscript j | the j th sampled pose of the trajectory |
| subscript k | the k th VIO feature |
| I_j | set of visible features at sampled pose j |
| $\Gamma(t)$ $= [x(t) \ y(t) \ z(t) \ \psi(t)]^T$ | a trajectory consisted of 3D positions $x(t), y(t), z(t)$, and yaw angle $\psi(t)$ |
| $\mathbf{P}_k = [X_k \ Y_k \ Z_k]^T$, $\mathbf{P}'_{kj} = [X'_{kj} \ Y'_{kj} \ Z'_{kj}]^T$ | 3D coordinates of the k th feature in W , 3D coordinates of the k th feature in C when vehicle is at sampled pose j |
| $\hat{\mathbf{b}}_{kj} = [\hat{u}_{kj} \ \hat{v}_{kj}]^T$, $\mathbf{b}_{kj} = [u_{kj} \ v_{kj}]^T$ | k th feature’s observation in image, estimated projection in the image (when vehicle is at sampled pose j) |
| $S(\cdot)$ | function converting a \mathbb{R}^3 vector to its corresponding skew symmetric matrix |
| ξ | error of estimated camera extrinsics |
| $\mathbf{J}_{\xi, kj}$ | $\partial \mathbf{b}_{kj} / \partial \xi$, Jacobian of the k th feature with respect to ξ at sampled pose j |
| $\Sigma_{\xi j}$ | covariance of estimated ξ at sampled pose j |
| $\mathbf{v}_j, \boldsymbol{\omega}_j$ | vehicle’s velocity, angular velocity at sampled pose j |
| $\sigma_{kj, \parallel}^2, \sigma_n^2$ | feature’s variance because of speed, feature’s variance because of vibration |
| $\bar{\mathbf{b}}_{kj} = [\bar{u}_{kj} \ \bar{v}_{kj}]^T$ | the normalized speed of the k th feature in image at sampled pose j |
| t_{exp} | camera’s exposure time |
| $\Sigma_{\mathbf{b}, kj}$ | covariance of the k th feature in the image at sampled pose j |
| $\Sigma_{\mathbf{b} j}$ | covariance of visible features in the image at sampled pose j |

The pose of the vehicle’s body frame B with respect to the world frame W is given by $\mathbf{T}^{WB} = [\mathbf{R}^{WB} \ \mathbf{t}^{WB}]$, where \mathbf{R}^{WB} and \mathbf{t}^{WB} are the orientation and position of B with respect to W . We assume that the vehicle is equipped with a depth camera, whose pose in the vehicle’s body frame is given by a rigid body transformation $\mathbf{T}^{BC} = [\mathbf{R}^{BC} \ \mathbf{t}^{BC}]$. The extrinsics of the camera is given by $\mathbf{T}^{CW} = [\mathbf{R}^{CW} \ \mathbf{t}^{CW}]$.

The perception cost of a trajectory c_{perc} represents the uncertainty of vehicle position estimation if the trajectory is followed. Similarly to [23], we estimate the variance of vehicle pose estimation by formulating a least-squares problem. However, we also estimate the variance of the feature points in the image according to their movement speed in the image, instead of assuming a constant variance for the features as in [23]. This helps to take motion blur into account and discourages

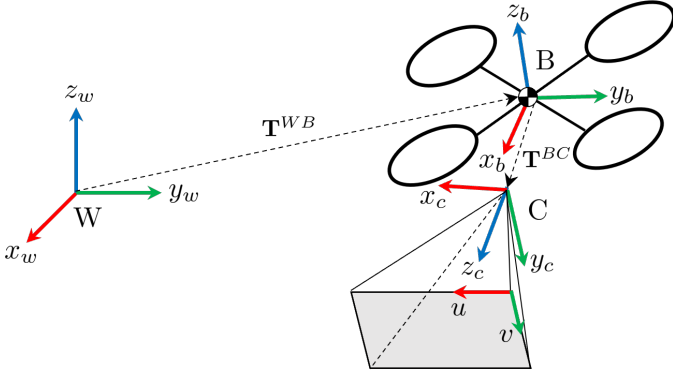


Figure 3. An illustration of the world frame W, body frame B and camera frame C. The position and attitude of the vehicle's body frame with respect to the world frame is given by T^{WB} . The position and attitude of the camera frame in the body frame is given by T^{BC} , which is a rigid body transformation.

the multicopter from executing an overly aggressive trajectory, which would adversely affect the VIO accuracy. The function mapping the pose estimation variance to the perception cost c_{perc} is also different, for faster computation and more intuitive tuning of the perception cost's weight coefficient k_{perc} .

A. Cost of a trajectory

Given a trajectory $\Gamma(t) = [x(t) \ y(t) \ z(t) \ \psi(t)]^T$ and the 3D positions of VIO features $F_{\text{feature}} := \{P_k\}_{k=1}^M$ in the world frame, we need to predict the pose estimation error if the given trajectory is followed. In $\Gamma(t)$, the 3D positions of the vehicle are represented as $[x(t) \ y(t) \ z(t)]^T$, and the yaw angle of the vehicle is represented as $\psi(t)$. We first sample the poses of the vehicle $\{T_j^{WB}\}_{j=1}^N$ along the trajectory at a fixed time interval. For a sampled pose T_j^{WB} , we can find the extrinsics of the camera T_j^{CW} and thus the visible features in F_{feature} , as shown in Fig. 4. We denote the indices of visible features at the camera pose T_j^{CW} as I_j .

Then if the vehicle moves to T_j^{WB} , the extrinsics of the camera T_j^{CW} can be estimated by the following least squares

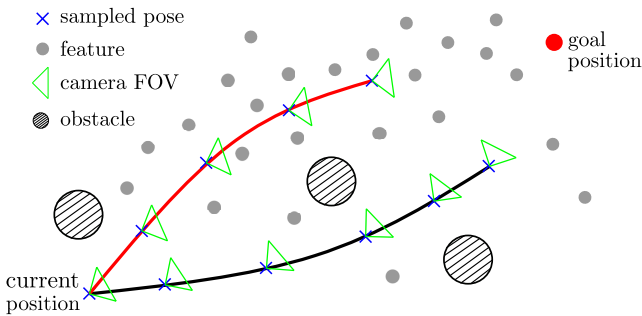


Figure 4. We discretize sampled trajectories at a fixed time interval to evaluate their perception cost. When the camera's exposure time is short and motion blur is not significant, the trajectory in red has lower perception cost than the trajectory in black. Because it keeps more features in the camera's view and is also closer to the features. On the trajectories, the camera always points towards the target point to better see the obstacles.

problem:

$$T_j^* = \arg \min_T \sum_{k \in I_j} \left\| \hat{b}_{kj} - \text{proj}(TP_k) \right\|_2^2, \quad (1)$$

where $\text{proj}(\cdot)$ represents the projection function of the camera, and $\hat{b}_{kj} = [\hat{u}_{kj} \ \hat{v}_{kj}]^T$ is the k th feature point's observation in the image, which has noise due to motion blur and camera lens imperfections.

The optimization problem (1) can usually be solved in an iterative way. It can be converted into the following form:

$$\xi^* = \arg \min_{\xi} \sum_{k \in I_j} \left\| \hat{b}_{kj} - \text{proj}(\exp(\xi) T_j^{CW} P_k) \right\|_2^2, \quad (2)$$

where ξ is in $\mathfrak{se}(3)$ and represents the error of the estimated camera extrinsics. The function $\exp(\cdot)$ maps $\mathfrak{se}(3)$ to $SE(3)$. The estimated camera extrinsics T_j^{CW} is updated iteratively by $T_j^{CW} = \exp(\xi) T_j^{CW}$.

Define $b_{kj} = \text{proj}(T_j^{CW} P_k)$, which is the projected coordinates of the k th feature in the image frame. The equation (2) can be solved through linearization at the current estimation of T_j^{CW} :

$$\begin{aligned} \xi^* &= \arg \min_{\xi} \sum_{k \in I_j} \left\| \hat{b}_{kj} - \text{proj}(T_j^{CW} P_k) - J_{\xi, kj} \xi \right\|_2^2 \\ &= \arg \min_{\xi} \sum_{k \in I_j} \left\| \hat{b}_{kj} - b_{kj} - J_{\xi, kj} \xi \right\|_2^2, \end{aligned} \quad (3)$$

where $J_{\xi, kj} = \frac{\partial b_{kj}}{\partial \xi}$, $k \in I_j$.

Denote $P'_{kj} = [X'_{kj} \ Y'_{kj} \ Z'_{kj}]^T = T_j^{CW} P_k$ as the coordinates of features in the camera frame, then we have:

$$J_{\xi, kj} = \frac{\partial b_{kj}}{\partial \xi} = \frac{\partial b_{kj}}{\partial P'_{kj}} \frac{\partial P'_{kj}}{\partial \xi}. \quad (4)$$

Denote the camera's focal length as f_x, f_y and its focal point's coordinates in the image as $[c_x \ c_y]^T$, from the pinhole camera model, we have:

$$b_{kj} = \begin{bmatrix} u_{kj} \\ v_{kj} \end{bmatrix} = \text{proj}(P'_{kj}) = \begin{bmatrix} f_x \frac{X'_{kj}}{Z'_{kj}} + c_x \\ f_y \frac{Y'_{kj}}{Z'_{kj}} + c_y \end{bmatrix}. \quad (5)$$

Besides, with some Lie algebra derivations, we can get

$$\frac{\partial P'_{kj}}{\partial \xi} = \begin{bmatrix} \mathbf{I}_{3 \times 3} & -S(P'_{kj}) \end{bmatrix}, \quad (6)$$

where the function $S(\cdot)$ converts a vector in \mathbb{R}^3 to its corresponding 3-by-3 skew-symmetric matrix. By combining (4), (5) and (6), we can get the expression of the Jacobian

$$J_{\xi, kj} = \begin{bmatrix} \frac{f_x}{Z'_{kj}} & 0 & -\frac{f_x X'_{kj}}{Z'^2_{kj}} & -\frac{f_x X'_{kj} Y'_{kj}}{Z'^2_{kj}} & f_x + \frac{f_x X'^2_{kj}}{Z'^2_{kj}} & -\frac{f_x Y'_{kj}}{Z'_{kj}} \\ 0 & \frac{f_y}{Z'_{kj}} & -\frac{f_y Y'_{kj}}{Z'^2_{kj}} & -f_y - \frac{f_y Y'^2_{kj}}{Z'^2_{kj}} & f_y \frac{X'_{kj} Y'_{kj}}{Z'^2_{kj}} & \frac{f_y X'_{kj}}{Z'_{kj}} \end{bmatrix} \quad (7)$$

As a result, the covariance of the estimated parameter ξ in (3) is given by

$$\Sigma_{\xi_j} = (\mathbf{J}_j^T \mathbf{J}_j)^{-1} \mathbf{J}_j^T \Sigma_{\mathbf{b}_{kj}} \mathbf{J}_j (\mathbf{J}_j^T \mathbf{J}_j)^{-1} = (\mathbf{J}_j^T \Sigma_{\mathbf{b}_{kj}}^{-1} \mathbf{J}_j)^{-1}, \quad (8)$$

where the matrix \mathbf{J}_j is stacked up of $\mathbf{J}_{\xi,k,j}$, and $\Sigma_{\mathbf{b}_{kj}}$ is the covariance of visible features, which is related to the feature's speed in the image plane and is derived in the following Section III-B.

For computational efficiency and more intuitive tuning of k_{perc} , we only consider the estimation variance of the first three elements of ξ , which corresponds to the estimated position of the vehicle. The perception cost of a trajectory is defined as:

$$c_{\text{perc}} = \sum_{j=1}^N \frac{\sqrt{\Sigma_j^{(1,1)}} + \sqrt{\Sigma_j^{(2,2)}} + \sqrt{\Sigma_j^{(3,3)}}}{3N}, \quad (9)$$

which corresponds to the mean sum of per-axis standard deviation of the position estimates over the sampled times.

B. Feature variance estimation

When the vehicle moves to \mathbf{T}_j^{WB} , the 3D position of a visible VIO feature in the camera's frame $\mathbf{P}'_{kj} = [X'_{kj} \ Y'_{kj} \ Z'_{kj}]^T$ is given by:

$$\mathbf{P}'_{kj} = \mathbf{T}_j^{CW} \mathbf{P}_k = \mathbf{R}_j^{CW} (\mathbf{P}_k - \mathbf{R}_j^{WB} \mathbf{t}^{BC} - \mathbf{t}_j^{WB}). \quad (10)$$

To obtain the feature's velocity in the camera frame, we differentiate (10) with respect to time:

$$\begin{aligned} \dot{\mathbf{P}}'_{kj} &= [\dot{X}'_{kj} \ \dot{Y}'_{kj} \ \dot{Z}'_{kj}]^T \\ &= -\mathbf{R}^{CB} S(\omega_j) \mathbf{R}^{CB} \mathbf{P}'_k \\ &\quad - \mathbf{R}^{CB} S(\omega_j) \mathbf{t}^{BC} - \mathbf{R}_j^{CW} \mathbf{v}_j^{WB} \end{aligned} \quad (11)$$

where \mathbf{v}_j^{WB} is the vehicle's velocity in the world frame and ω_j is the vehicle's angular velocity in the body frame. They can be predicted given a trajectory $\Gamma(t)$ [29]. Also, $\mathbf{R}^{CB} = (\mathbf{R}^{BC})^{-1}$.

We then differentiate (5) with respect to time, to get the feature's speed in the image plane:

$$\dot{u}_{kj} = f_x \frac{\dot{X}'_{kj} Z'_{kj} - X'_{kj} \dot{Z}'_{kj}}{Z'^2_{kj}}, \quad \dot{v}_{kj} = f_y \frac{\dot{Y}'_{kj} Z'_{kj} - Y'_{kj} \dot{Z}'_{kj}}{Z'^2_{kj}}. \quad (12)$$

Due to the movement of the camera, a feature point will have some motion blur in the image. Denote the camera's exposure time as t_{exp} , and the feature could be approximated by a straight line of length $t_{\text{exp}} \sqrt{\dot{u}_{kj}^2 + \dot{v}_{kj}^2}$ when the exposure time is relatively short. We assume that the feature point is uniformly distributed on this straight line, whose direction is the feature's speed ($\dot{\mathbf{b}}_{kj}$) direction in the image plane. The feature's variance in this direction is approximated by:

$$\sigma_{k,j,\parallel}^2 = \frac{t_{\text{exp}}^2 (\dot{u}_{kj}^2 + \dot{v}_{kj}^2)}{12}. \quad (13)$$

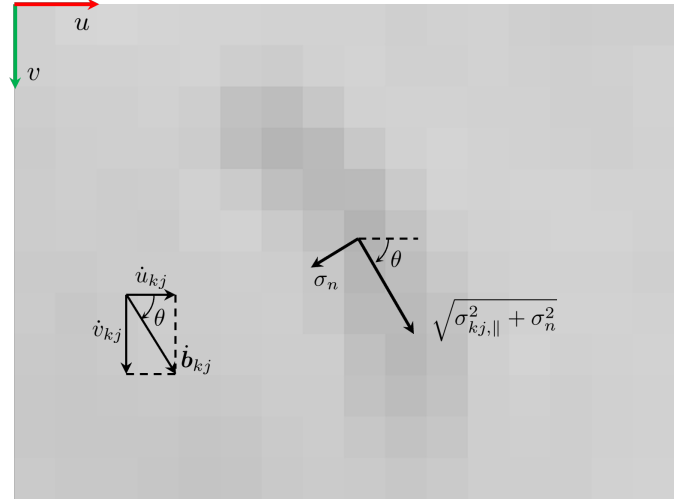


Figure 5. A feature point becomes a blurry line in the image because of motion blur. The angle of the line is θ , which is the same as the feature's speed in the image plane $\dot{\mathbf{b}}_{kj}$. The standard deviation of the feature's position on the speed direction is $\sqrt{\sigma_{k,j,\parallel}^2 + \sigma_n^2}$, and is σ_n on the perpendicular direction.

In addition, due to vehicle vibration and the imperfect lens, the feature has an additional variance σ_n^2 , which we assume to be omnidirectional and can be measured experimentally.

Define the normalized feature speed in the image to be $\bar{\mathbf{b}}_{kj} = [\bar{u}_{kj} \ \bar{v}_{kj}]^T$. Then, the covariance of the feature in the image plane is given by:

$$\begin{aligned} \Sigma_{\mathbf{b},kj} &= \begin{bmatrix} \bar{u}_{kj} & -\bar{v}_{kj} \\ \bar{v}_{kj} & \bar{u}_{kj} \end{bmatrix} \begin{bmatrix} \sigma_{k,j,\parallel}^2 + \sigma_n^2 & 0 \\ 0 & \sigma_n^2 \end{bmatrix} \begin{bmatrix} \bar{u}_{kj} & -\bar{v}_{kj} \\ \bar{v}_{kj} & \bar{u}_{kj} \end{bmatrix}^T = \\ &= \begin{bmatrix} \bar{u}_{kj}^2 \sigma_{k,j,\parallel}^2 + (\bar{u}_{kj}^2 + \bar{v}_{kj}^2) \sigma_n^2 & \bar{u}_{kj} \bar{v}_{kj} \sigma_{k,j,\parallel}^2 \\ \bar{u}_{kj} \bar{v}_{kj} \sigma_{k,j,\parallel}^2 & \bar{v}_{kj}^2 \sigma_{k,j,\parallel}^2 + (\bar{u}_{kj}^2 + \bar{v}_{kj}^2) \sigma_n^2 \end{bmatrix} \end{aligned} \quad (14)$$

We can then get the covariance matrix of the visible features in (8):

$$\Sigma_{\mathbf{b}_j} = \begin{bmatrix} \Sigma_{\mathbf{b},k_1j} & 0 & \dots & 0 \\ 0 & \Sigma_{\mathbf{b},k_2j} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \Sigma_{\mathbf{b},k_mj} \end{bmatrix}, \quad (15)$$

where $k_1, k_2, \dots, k_m \in \mathbf{I}_j$. By substituting (15) into (8) and (9), we can get the perception cost c_{perc} of a sampled trajectory.

IV. PERCEPTION-AWARE TRAJECTORY PLANNING

In this section, we briefly introduce the RAPPIDS planner that we use to generate collision-free trajectories, which is first introduced in [26] and is improved in [27]. In addition, this section introduces how the perception-aware cost is integrated with the RAPPIDS planner to reduce the multicopter's state estimation uncertainty of VIO.

A. RAPPIDS planner overview

1) *Trajectory sampling*: The RAPPIDS planner samples fifth-order minimum jerk polynomial trajectories $s(t)$ with different duration T and end position s_T using a computationally efficient planner proposed in [30]. The sampled trajectories then go through checks to find if they are input feasible, below the flight speed limit (for safety), and collision-free. The planner replans when a new depth image arrives, to take into account the latest obstacle information.

The sampled trajectories are described as:

$$s(t) = \frac{\alpha}{120}t^5 + \frac{\beta}{24}t^4 + \frac{\gamma}{6}t^3 + \frac{\ddot{s}(0)}{2}t^2 + \dot{s}(0)t + s(0), \quad t \in [0, T] \quad (16)$$

where $s(0)$, $\dot{s}(0)$, and $\ddot{s}(0)$ are the position, velocity, and acceleration of the vehicle at the start time of the trajectory. The terminal condition is selected to be at rest to ensure safety, which requires $s(T) = s_T$ and $\dot{s}(T) = \ddot{s}(T) = 0$. Even if the planner cannot find a new collision-free trajectory (when new depth images arrive), the vehicle can continue following the current trajectory and stop safely. The coefficients α , β and γ can be solved in closed form [30].

The yaw angle $\psi(t)$ of the trajectory is selected such that the multicopter always faces to the target point, to better detect the obstacles and free space on the way and thus increase the chance of finding trajectories with lower cost:

$$\psi(t) = \arctan2\left(\begin{bmatrix} 0 & 1 & 0 \end{bmatrix} (s_G - s(t)), \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} (s_G - s(t))\right), \quad (17)$$

where s_G and $s(t)$ are the positions of the goal and the vehicle's position in the world frame, respectively. Note that we do not require the vehicle to be placed facing towards the goal point at the beginning of the flight, since the planner will adjust the vehicle's yaw towards the goal automatically.

2) *Collision check*: The RAPPIDS planner checks whether a sampled trajectory collides with obstacles by partitioning the free space into rectangular pyramids and checks if the trajectory remains in the union of these pyramids. The idea of free space partitioning helps to reduce the computational cost of finding collision-free trajectories [31], [32], [33], [34]. In collision check, the vehicle is simplified as the smallest sphere S_c containing the vehicle. The space partitioning process is illustrated in Fig. 6. Firstly, we can get the free space \mathcal{F} and the occupied space \mathcal{O} based on the depth image. To avoid collisions with potential unseen obstacles, we treat all spaces \mathcal{U} outside the depth camera's field of view that are l distance away from the vehicle as occupied. The next step is to search for the depth pixel closest to the end position of the trajectory $s(T)$, marked as \bar{s}_1 in the figure. Then, starting with the nearest depth pixel and reading the surrounding depth pixels in a spiral sequence, we find the largest possible rectangular space \mathcal{P}_{exp1} that does not intrude into the occupied space \mathcal{O} . Finally, a pyramid \mathcal{P}_1 is created by shrinking the expanded pyramid \mathcal{P}_{exp1} with the vehicle's radius r .

Whether the sampled trajectory is within the union of generated collision-free pyramids can be determined efficiently using the method proposed in [35]. If a sampled trajectory intersects with the union of existing pyramids, the algorithm tries to generate a new pyramid \mathcal{P}_2 , starting the search from

the intersection point, marked as \bar{s}_2 in Fig. 6. The pyramid generation process continues until the trajectory is within the union of the pyramids – the trajectory is collision-free, or when no new pyramid could be generated – the trajectory can collide with obstacles. A detailed description of this algorithm can be found at [26].

B. Selection of the best trajectory

To encourage fast flight towards the target point (following [27]), we define the speed cost c_{speed} as:

$$c_{speed} = -\frac{\|s_G - s(0)\|_2 - \|s_G - s(T)\|_2}{T}, \quad (18)$$

which is the negative of the average flight speed towards the goal for a sampled trajectory.

There is a trade-off between fast flight and state estimation quality, especially in indoor environments where the camera's exposure time is long. Flying at a fast speed will increase the motion blur, which makes the VIO less accurate and causes the perception cost c_{perc} to increase. As a result, we introduce a coefficient k_{perc} to determine the importance of the VIO quality. The larger we set k_{perc} , the more weight we put on the state estimation quality over fast flight and vice versa.

We choose the reference trajectory from the sampled trajectories that pass the RAPPIDS collision check (Section IV-A2) to avoid obstacles. Among them, the trajectory with the minimum total cost $c_{tot} = k_{perc}c_{perc} + c_{speed}$ is chosen as the trajectory to follow. The proposed perception-aware planner runs in a receding horizon manner, to take into account the latest information of the obstacles in the environment and the new feature points found by the VIO. It tries to find a trajectory with lower total cost c_{tot} every time a new depth image arrives and the current trajectory will continue to be followed if no better trajectory is found. Since each generated trajectory has zero speed and acceleration at the end, and the planner assumes the only features in the environment are those found by the VIO, the planned trajectory is always safe. The

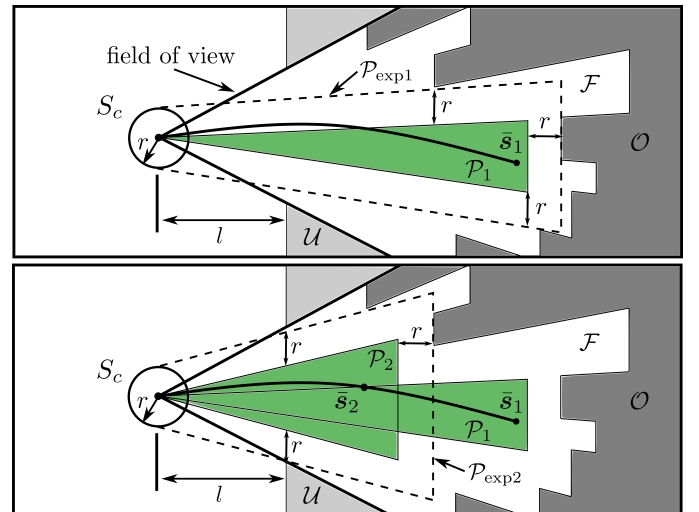


Figure 6. The RAPPIDS planner partitions the free space with rectangular pyramids (shown in green) for fast collision check of a sampled trajectory.

vehicle would stop safely if no new collision-free trajectory that allows the camera to see tracked VIO features could be found, avoiding the vehicle flying to areas with no VIO features.

V. EXPERIMENTAL EVALUATION

We validate the effectiveness of the proposed perception-aware planning method in both stationary indoor and outdoor environments, by comparing it with the original perception-agnostic RAPPIDS planner. The perception-aware planner is shown to improve the VIO’s state estimation accuracy and the number of tracked features in the camera’s field of view. The experiment video can be found at <https://youtu.be/qO3LZIrptwQ>.

A. Hardware setup

A custom-built quadcopter was used during the experiments, as shown in Fig. 7. It has a mass of 1.5 kg, and the distance between two diagonal motors is 500 mm. The diameter of each propeller is 254 mm (10 inches). The vehicle is equipped with a forward-looking Intel Realsense D455 depth camera for collision avoidance and VIO (with the structure light turned off). The VIO (openVINS [28] in the experiments) uses monocular images from the left and right cameras on the D455 stereo depth camera at 15Hz and IMU data at 400Hz to give the estimated states of the vehicle. It extracts the visual features from the monocular images instead of from the depth images. The RAPPIDS planner uses depth images (15Hz) to generate collision-free trajectories.

The proposed perception-aware planner and the VIO run on a small onboard computer (Qualcomm RB5). The trajectories generated by the perception-aware planner are sent to the Pixracer flight controller running the standard PX4 firmware [36], and are tracked by the low-level position and attitude controllers run on the flight controller. The vehicle detects if it reaches the goal point according to the state estimation from VIO – the goal is reached when the distance between the estimated position of the vehicle and the goal point is below a user-defined threshold.



Figure 7. The quadcopter used in the experiments. The distance between two diagonal motors is 500 mm.

Note that the initial heading direction of the vehicle did not need to face the target position, since the planned trajectory would automatically adjust the yaw angle to always face to the goal during the flight. As a result, the ability of the planner to reach the goal is not affected by the initial yaw angle. We chose the yaw angle to always face the goal to better detect obstacles and free space, as we mentioned in Section IV-A1.

B. Indoor experiments

In indoor experiments, we used a motion capture system to provide the ground truth for the vehicle’s position. As shown in Fig. 8, the vehicle first took off to 1.2 meters in height, flew to the target point 4 meters forward avoiding the obstacles, and then landed. The experiments were repeated for 12 times each

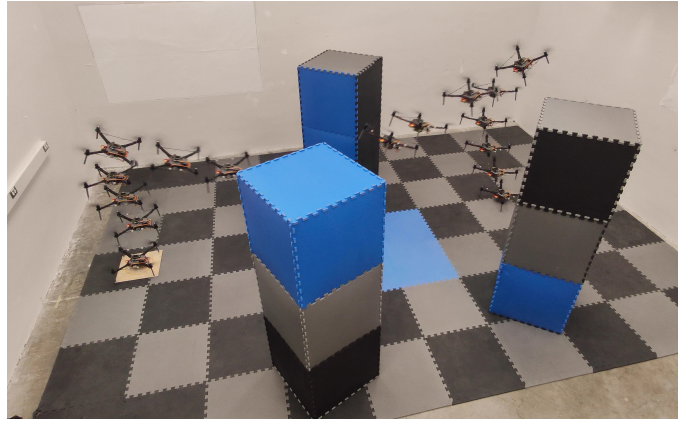


Figure 8. The indoor experiment setup. The vehicle’s goal is 4 meters ahead of the starting point and the vehicle flies from left to right in the image. The proposed perception aware planner guides the vehicle to reach the goal avoiding the obstacles on its way, while ensuring good state estimation quality from the VIO.

for the proposed perception-aware planner and the original RAPPIDS planner, respectively. The weight of the perception cost k_{perc} was set to 100. The camera’s exposure time t_{exp} was set to 8 milliseconds. Since the planner was only used in the collision avoidance flight, not in the taking-off and landing stages, we exclude the landing stage when comparing the two planners to minimize the uncertainty they introduce.

In the 12 tests, the VIO diverged twice for the original perception-agnostic RAPPIDS planner, while the VIO divergence did not occur for the perception-aware planner. Excluding the VIO diverged cases, the performance of the perception-agnostic and perception-aware planners is compared in Table II. We can see that, on average, the perception-aware planner reduced the final position estimation error (root mean square error) by 19%. Furthermore, the perception-aware planner significantly reduced the aggressiveness of the flights, which can be seen in the reduction of the angular velocity by 24%. This helped reduce the motion blur of the VIO feature points and improved the accuracy of the state estimation. The flight speed was only slightly slower than the perception-agnostic planner by approximately 6%. In addition, the perception-aware planner kept slightly more feature points within the camera’s field of view (6%). The difference in the

average feature number was small due to the space-constrained experiment setup, making the geometric paths the vehicle could take similar.

Table II
COMPARISON OF INDOOR EXPERIMENTAL RESULTS BETWEEN THE PROPOSED PERCEPTION-AWARE PLANNER AND THE ORIGINAL PERCEPTION-AGNOSTIC PLANNER.

| | original | proposed | difference |
|--------------------------------|----------|----------|------------|
| mean final pos. est. error [m] | 0.141 | 0.114 | -19.1% |
| mean angular velocity [rad/s] | 1.390 | 1.052 | -24.3% |
| mean feature number in FOV | 97.892 | 103.806 | +6.0% |
| mean speed [m/s] | 1.376 | 1.298 | -5.7% |

C. Outdoor experiments

1) Environment with distinct visual feature distribution:

The first group of outdoor experiments were conducted along a road near a small forest, where there were much more visual features on the trees than on the road, so that only seeing the road will result in poor VIO accuracy. The vehicle first took off to 2 meters in height, flew to the specified target point 20 meters forward, and then landed. The perception-aware planner guided the vehicle to fly closer to the forest to see more features and at a closer distance to increase the VIO's accuracy, as shown in Fig. 1. On the contrary, the original perception-agnostic planner flew the vehicle directly to the goal from the starting position.

Table III

COMPARISON BETWEEN PLANNERS IN AN ENVIRONMENT WITH DISTINCT VISUAL FEATURE DISTRIBUTION

| | original | proposed | difference |
|-------------------------------|----------|----------|------------|
| mean pos. est. std [m] | 0.0259 | 0.0213 | -17.8% |
| mean angular velocity [rad/s] | 0.821 | 0.789 | -3.9% |
| mean feature number in FOV | 45.379 | 50.997 | +12.4% |
| mean speed [m/s] | 2.419 | 2.225 | -8.0% |

The experiment was repeated three times for each planner (the original perception agnostic planner and the proposed perception-aware planner), and the experimental results are summarized in Table III. Since both planners were only used in the forward flight, we exclude the taking-off and landing stages. We can see that the proposed perception-aware planner reduced the standard deviation of position estimation by 17.8% and increased the number of features in the camera's field of view (FOV) by 12.4% compared to the original perception-agnostic planner. The detailed result for each test is shown in Fig. 9. In addition, the perception-aware planner prevented the vehicle from seeing very few features in the camera's FOV, which happened at around 1 second in test 3 of the perception-agnostic planner (marked in red in Fig. 9) and would cause the triangulation to fail. In test 2 of the perception-agnostic planner, the camera saw a small number of features at around 6 seconds (marked by ①), causing an increase in position estimation uncertainty. The standard deviation of the position estimation (i.e. (9) at the current pose of the vehicle, $N = 1$) is used to evaluate the performance of the VIO instead of the error of position estimation in indoor tests, due to the

lack of ground truth from the motion capture system. Because there was some randomness in the VIO's feature selection, the distribution of features was different at the beginning of the flights, causing the position estimation's standard deviation to be different (marked by ②).

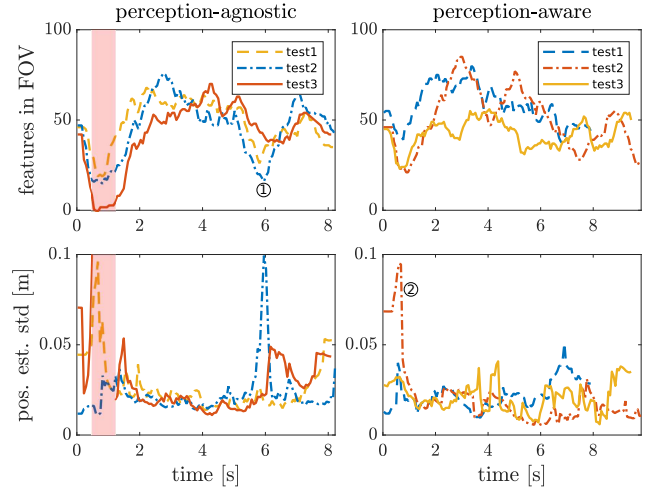


Figure 9. Comparison between the perception-agnostic planner (column 1) with the perception-aware planner (column 2) in the first set of outdoor experiments. For each planner, the tests are repeated three times, as shown in lines with different color and line type. The red shaded area marks the time when triangulation fails in test 3 of the perception-agnostic planner. The proposed perception-aware planner was able to keep more VIO features in the camera's FOV and also reduce position estimation's standard deviation of the vehicle.

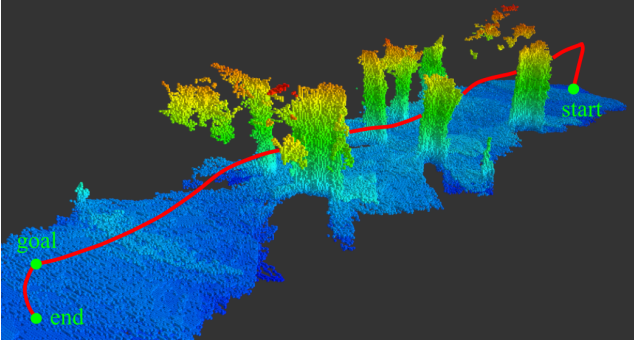
The tests were conducted during a sunny day and there was abundant light. As a result, the exposure time t_{exp} for the camera was set to 0.05 millisecond, much faster than during the indoor experiments. This short exposure time meant that the motion blur was much smaller compared to indoors, and the features' motion speed played a very small role in a feature's covariance (14). As a result, the angular velocity of the perception-aware planner was only 3.9% lower than that of the perception-agnostic planner, showing similar aggressiveness in flight. Its average flight speed was slightly slower than the perception-agnostic planner, similar to the indoor experiments.

2) Environment with dense obstacles: The second set of outdoor experiments were conducted near a small dense forest, where the vehicle was commanded to take off to 2 meters in height, fly 30 meters forward to pass through the forest, and then land, as shown in Fig. 10. The aim of these experiments is to show the ability of our proposed perception-aware planner to avoid dense obstacles, and simultaneously improve the VIO estimation accuracy. The tests were conducted during a cloudy day, and the camera's exposure time was set to 0.5 millisecond.

The experiment was repeated three times for each planner (the original perception agnostic planner and the proposed perception-aware planner), and the experimental results are summarized in Table IV, like in previous experiments, we exclude the taking-off and landing stages because both planners were only used in the forward flight. We can see that the perception-aware planner was able to fly the vehicle through the dense forest like the perception agnostic planner. In addi-



(a) Experiment environment with dense obstacles.



(b) Visualization of the flight path.

Figure 10. The perception-aware planner guides the vehicle through a small forest, avoiding obstacles while improving the VIO accuracy. (a) The small forest with the vehicle at the beginning of the forward flight. (b) In this sub-figure we visualize the tree obstacles using the RTAB-Map [37] offline. Our proposed planner didn't use the map during operation. The path of the vehicle is marked in red line, while the start point, goal point and end point are marked with green dots. The distance between the start point and the end point is 30 meters.

tion, it improved the number of visual features in the camera's FOV by 20.6%, and also reduced the standard deviation of position estimation by 25.6%. The detailed number of VIO features in the camera's FOV and the standard deviation of position estimation for each of the tests are visualized in Fig. 11. We saw in the tests that the perception-aware planner flew the vehicle closer to the forest ground, as there were more features on the ground than on the trees. This helped the camera to keep more features in the camera's FOV and keep a closer distance to them – both were helpful for increasing the VIO's accuracy. The dramatic drop of the number of visual features in the camera's FOV (the first row) at around 8 seconds was because the vehicle flew out of the forest, and a landscape with less visual features (road and grass land) entered the camera's view.

Because the experiments were conducted during a cloudy day, the camera exposure time of 0.5 milliseconds was between the 8 milliseconds exposure time in the indoor experiments (Section V-B) and the 0.05 milliseconds in the first set of outdoor experiments during a sunny day (Section V-C1). This led to a degree of motion blur between these two previous experiments for a feature point moving at the same speed in the image plane. As a result, the reduction in the flight aggressiveness of 14.1%, was also in between them (24.3%

for the indoor tests and 3.9% in the first set of outdoor experiments). Like in the previous experiments, the introduction of the perception cost slightly decreased the flight speed, since the aim was not solely fast flight towards the goal as for the perception-agnostic planner. This led to a 5.56% reduction in flight speed compared with the perception-agnostic planner.

Table IV
COMPARISON BETWEEN THE PLANNERS IN AN ENVIRONMENT WITH DENSE OBSTACLES

| | original | proposed | difference |
|-------------------------------|----------|----------|------------|
| mean pos. est. std [m] | 0.0172 | 0.0128 | -25.6% |
| mean angular velocity [rad/s] | 0.810 | 0.710 | -14.1% |
| mean feature number in FOV | 45.469 | 54.846 | +20.6% |
| mean speed [m/s] | 2.392 | 2.259 | -5.56% |

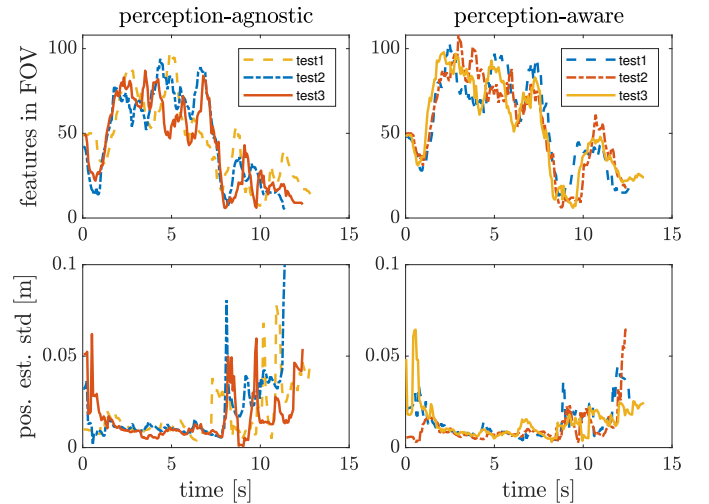


Figure 11. Comparison between the perception-agnostic planner (column 1) with the perception-aware planner (column 2) in the second set of outdoor experiments. For each planner, the tests are repeated three times, as shown in lines with different color and line type. Both planner were able to fly through the dense tree obstacles. The proposed perception-aware planner was able to keep more VIO features in the camera's FOV and also reduce position estimation's standard deviation of the vehicle.

VI. CONCLUSION AND FUTURE WORK

In this work, we proposed a receding horizon perception-aware local planner for multicopters, which is able to guide the vehicle to areas with rich visual features and reduce the features' motion blur by reducing the planned trajectory's aggressiveness. By taking the motion blur of the features into consideration, our proposed method is able to adjust the trajectory's aggressiveness based on the light conditions. We conducted both indoor and outdoor experiments to show the effectiveness of the proposed method in improving the VIO's position estimation accuracy and reducing the VIO's failure rate. The ability of the proposed planner to fly through dense obstacles was also demonstrated through the experiments. Thanks to the computationally efficient collision checking and cost function design, our method is capable of running in real time on a small embedded computer onboard the vehicle.

A potential future extension of this work is to combine the proposed perception-aware local planner with a global planner for large-scale perception-aware navigation of multicopters in complex environments. The global planner can supplement the proposed local planner by giving high-level waypoints and taking into account the semantic information to avoid areas with potential unreliable visual features (like water surface). Another potential extension is to take dynamic obstacles into account: a module of detecting and predicting the motion of moving obstacles needs to be added. This module can also be used for preventing the VIO from using the visual features on the moving objects and improve its accuracy.

ACKNOWLEDGEMENTS

This work was partially supported by the AFRI Competitive Grant no. 2020-67021-32855/project accession no. 1024262 from the USDA National Institute of Food and Agriculture. The AFRI Competitive Grant is being administered through AIFS: the AI Institute for Next Generation Food Systems, <https://aifs.ucdavis.edu>. Research was also partially sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-20-2-0105. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for government purposes notwithstanding any copyright notation herein.

The authors would thank Patrick Geneva of the Robot Perception and Navigation Group, University of Delaware, for his helpful support and suggestions over the usage of the OpenVINS. The experimental testbed at the HiPeRLab is the result of contributions of many people, a full list of which can be found at hiperlab.berkeley.edu/members/.

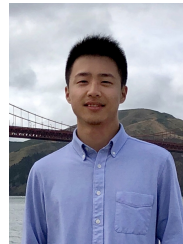
REFERENCES

- [1] H. Kang, H. Li, J. Zhang, X. Lu, and B. Benes, "Flycam: Multitouch gesture controlled drone gimbal photography," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3717–3724, 2018.
- [2] Y. Wu, Y. Qin, Z. Wang, and L. Jia, "A uav-based visual inspection method for rail surface defects," *Applied Sciences*, vol. 8, no. 7, 2018.
- [3] M. Tranzatto, F. Mascarich, L. Bernreiter, C. Godinho, M. Camurri, S. Khattak, T. Dang, V. Reijgwart, J. Loeje, D. Wisth *et al.*, "Cerberus: Autonomous legged and aerial robotic exploration in the tunnel and urban circuits of the darpa subterranean challenge," *arXiv preprint arXiv:2201.07067*, 2022.
- [4] X. Wu, J. Zeng, A. Tagliabue, and M. W. Mueller, "Model-free online motion adaptation for energy-efficient flight of multicopters," *IEEE Access*, vol. 10, pp. 65 507–65 519, 2022.
- [5] H. Huang, G. M. Hoffmann, S. L. Waslander, and C. J. Tomlin, "Aerodynamics and control of autonomous quadrotor helicopters in aggressive maneuvering," in *2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 3277–3282.
- [6] S. Chen, X. Wu, M. W. Mueller, and K. Sreenath, "Real-time geolocalization using satellite imagery and topography for unmanned aerial vehicles," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 2275–2281.
- [7] S. W. Chen, G. V. Nardari, E. S. Lee, C. Qu, X. Liu, R. A. F. Romero, and V. Kumar, "Sloam: Semantic lidar odometry and mapping for forest inventory," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 612–619, 2020.
- [8] X. Wu and M. W. Mueller, "Using multiple short hops for multicopter navigation with only inertial sensors," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 8559–8565.
- [9] S. Lupashin, M. Hehn, M. W. Mueller, A. P. Schoellig, M. Sherback, and R. D'Andrea, "A platform for aerial robotics research and demonstration: The flying machine arena," *Mechatronics*, vol. 24, no. 1, pp. 41–54, 2014.
- [10] M. W. Mueller, M. Hamer, and R. D'Andrea, "Fusing ultra-wideband range measurements with accelerometers and rate gyroscopes for quadcopter state estimation," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 1730–1736.
- [11] D. Scaramuzza and Z. Zhang, "Visual-inertial odometry of aerial robots," *arXiv preprint arXiv:1906.03289*, 2019.
- [12] M. W. Mueller, S. J. Lee, and R. D'Andrea, "Design and control of drones," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, no. 1, pp. 161–177, 2022.
- [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)*, 2017, pp. 6199–6206.
- [14] 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)*, 2020, pp. 3213–3219.
- [15] —, "Joint feature selection and time optimal path parameterization for high speed vision-aided navigation," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 5931–5938.
- [16] V. Murali, I. Spasojevic, W. Guerra, and S. Karaman, "Perception-aware trajectory generation for aggressive quadrotor flight using differential flatness," in *2019 American Control Conference (ACC)*, 2019, pp. 3936–3943.
- [17] 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)*, 2018, pp. 1–8.
- [18] L. Bartolomei, L. Teixeira, and M. Chli, "Perception-aware path planning for uavs using semantic segmentation," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 5808–5815.
- [19] M. W. Achtelik, S. Lynen, S. Weiss, M. Chli, and R. Siegwart, "Motion- and uncertainty-aware path planning for micro aerial vehicles," *Journal of Field Robotics*, vol. 31, no. 4, pp. 676–698, 2014.
- [20] 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)*, 2014, pp. 3870–3875.
- [21] C. Papachristos, S. Khattak, and K. Alexis, "Uncertainty-aware receding horizon exploration and mapping using aerial robots," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 4568–4575.
- [22] C. Papachristos, F. Mascarich, S. Khattak, T. Dang, and K. Alexis, "Localization uncertainty-aware autonomous exploration and mapping with aerial robots using receding horizon path-planning," *Autonomous Robots*, vol. 43, no. 8, pp. 2131–2161, 2019.
- [23] Z. Zhang and D. Scaramuzza, "Perception-aware receding horizon navigation for mavs," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 2534–2541.
- [24] G. Costante, C. Forster, J. Delmerico, P. Valigi, and D. Scaramuzza, "Perception-aware path planning," *arXiv preprint arXiv:1605.04151*, 2016.
- [25] M. Irani and P. Anandan, "About direct methods," in *Vision Algorithms: Theory and Practice*, B. Triggs, A. Zisserman, and R. Szeliski, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 267–277.
- [26] N. Bucki, J. Lee, and M. W. Mueller, "Rectangular pyramid partitioning using integrated depth sensors (RAPPIDS): A fast planner for multicopter navigation," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4626–4633, 2020.
- [27] J. Lee, X. Wu, S. J. Lee, and M. W. Mueller, "Autonomous flight through cluttered outdoor environments using a memoryless planner," in *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2021, pp. 1131–1138.
- [28] P. Geneva, K. Eickenhoff, W. Lee, Y. Yang, and G. Huang, "OpenVINS: A research platform for visual-inertial estimation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 4666–4672.

- [29] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 2520–2525.
- [30] M. W. Mueller, M. Hehn, and R. D'Andrea, "A computationally efficient motion primitive for quadcopter trajectory generation," *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1294–1310, 2015.
- [31] F. Gao, Y. Lin, and S. Shen, "Gradient-based online safe trajectory generation for quadrotor flight in complex environments," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 3681–3688.
- [32] J. Chen, T. Liu, and S. Shen, "Online generation of collision-free trajectories for quadrotor flight in unknown cluttered environments," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1476–1483.
- [33] Z. H. Azizul, S. K. Yap, and Y. K. Loh, "Waypoint planning for mono visual drone using the minimal bounded space method," in *RiTA 2020*. Springer, 2021, pp. 160–170.
- [34] J. Tordesillas, B. T. Lopez, M. Everett, and J. P. How, "Faster: Fast and safe trajectory planner for navigation in unknown environments," *IEEE Transactions on Robotics*, vol. 38, no. 2, pp. 922–938, 2021.
- [35] N. Bucki and M. W. Mueller, "Rapid collision detection for multicopter trajectories," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 7234–7239.
- [36] PX4. (2022) Px4 drone autopilot. [Online]. Available: <https://github.com/PX4/PX4-Autopilot/tree/v1.12.1>
- [37] M. Labbé and F. Michaud, "Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation," *Journal of Field Robotics*, vol. 36, no. 2, pp. 416–446, 2019.



XIANGYU WU received his bachelor of science degree from Beijing Institute of Technology, China in 2017 and master of science degree from University of California, Berkeley, USA in 2019. He is currently a Ph.D. candidate at the High Performance Robotics Lab at UC Berkeley. His current research interests are the state estimation and path planning of multicopters.



SHUXIAO CHEN received his bachelor of engineering degree from the University of Nottingham, UK in 2017 and master of engineering degree from the University of California, Berkeley, USA in 2019. He is currently a Ph.D. student at the Hybrid Robotics Group at UC Berkeley. His current research interests are the perception and state estimation for legged robots.



KOUSHIL SREENATH is an Associate Professor of Mechanical Engineering, at UC Berkeley. He received the Ph.D. degree in Electrical Engineering and Computer Science from the University of Michigan at Ann Arbor, MI, in 2011. He was a Postdoctoral Scholar at the GRASP Lab at University of Pennsylvania from 2011 to 2013 and an Assistant Professor at Carnegie Mellon University from 2013 to 2017. His research interests center on dynamic robotics, applied nonlinear control, and safety-critical control. He received the NSF CAREER, Hellman Fellow, Best Paper Award at the Robotics: Science and Systems (RSS), and the Google Faculty Research Award in Robotics.



MARK W. MUELLER is an assistant professor of Mechanical Engineering at the University of California, Berkeley, and runs the High Performance Robotics Laboratory (HiPeRLab). He received a Dr.Sc. and M.Sc. from the ETH Zurich in 2015 and 2011, respectively, and a BSc from the University of Pretoria in 2008. His research interests include aerial robotics, their design and control, and especially the interactions between physical design and algorithms.