

Perception-aware trajectory generation for aggressive quadrotor flight using differential flatness

Varun Murali¹, Igor Spasojevic¹, Winter Guerra¹, Sertac Karaman¹

Abstract—Recent advances in visual-inertial state estimation have allowed quadrotor aircraft to autonomously navigate in unknown environments at operational speeds. In most cases, substantially higher speeds can be achieved by actively designing motion that reduces state estimation error. We are interested in autonomous vehicles running feature-based visual-inertial state estimation algorithms. In particular, we consider a trajectory optimization problem in which the goal is to maximize co-visibility of features, *i.e.* features are kept visible in the camera view from one keyframe to the next, increasing state estimation accuracy. Our algorithm is developed for autonomous quadrotor aircraft, for which position and yaw trajectories can be tracked separately. We assume that the desired positions of the vehicle are determined *a priori*, for instance, by a path planner that uses obstacles in the environment to generate a trajectory of positions with free yaw. This paper presents a novel algorithm that determines the yaw trajectory that jointly optimizes aggressiveness and feature co-visibility. The benefit of this algorithm was experimentally verified using a custom built quadrotor which uses visual inertial odometry for state estimation. The generated trajectories lead to better state estimation which contributes to improved trajectory tracking by a state-of-the-art controller under autonomous high-speed flight. Our results show that the root-mean-square error of the trajectory tracking is improved by almost 70%.

I. INTRODUCTION

Visual-inertial state estimation for high-speed autonomous quadrotor flight has received a large amount of interest in recent years. This has been fueled by the availability of low-cost cameras, inertial measurement units (IMUs), and powerful computational platforms available in ever smaller form factors [1], [2]. Typical visual-inertial navigation systems such as ORB-SLAM2 [3] employ a keyframe-based optimization technique to estimate the state of the robot from multiple visual features tracked across keyframes, as illustrated in Figure 1.

A robust state estimator is often critical in performing various agile maneuvers and loss of tracked features can result in collisions. Longer feature tracks often result in more robust visual inertial state estimation performance. Motion planning algorithms agnostic to the location of the visual features in the environment potentially result in trajectories for which the vehicle may face featureless areas of the environment (*e.g.*, empty walls). This leads to diminished state estimation accuracy, which degrades the performance of trajectory tracking, potentially resulting in catastrophic failure, such as

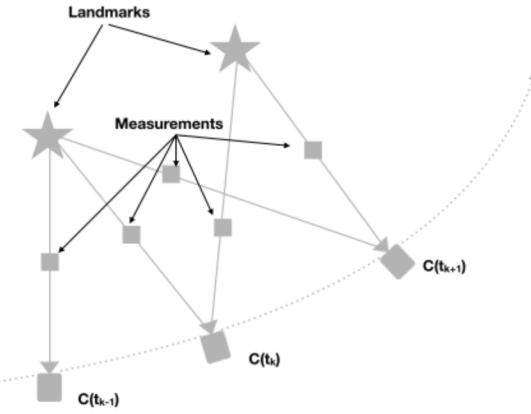


Fig. 1: Aspects of visual-inertial navigation. Observations of landmarks are acquired at discrete time intervals over a continuous trajectory.

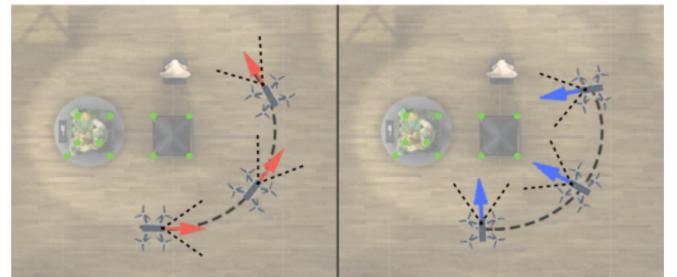


Fig. 2: An environment where most of the visual features are situated in the middle. Traditional planning approaches could consider facing forward as the vehicle navigates through a tight turn. Perception-aware planning algorithms, on the other hand, consider facing towards the visual features in the middle of the room while executing the same maneuver. This results in improved state-estimation performance, especially at high speeds.

a collision. *Perception-aware motion planning algorithms* consider trajectories that ensure better observation of visual features, improving state estimation accuracy and enabling faster navigation in complex environments (as illustrated in Figure 2).

We focus on planning perception-aware trajectories for quadrotor aircraft and restrict the class of admissible trajectories to those which pass through a finite sequence of predetermined waypoints. The perception objective consists

*This work was partly supported by the Office of Naval Research (ONR) and the Army Research Lab DCIST project.

¹Laboratory for Information and Decision Systems (LIDS), Massachusetts Institute of Technology (MIT), Cambridge, MA 02139, USA.
Emails: {mvarun, igorspas, winterg, sertac}@mit.edu

of observing a sparse set of triangulated landmarks from multiple keyframes in order to improve both the estimate of the position of the landmarks and the state of the aircraft itself. The perceptual sensor yields information about the relative transformation between consecutive keyframes, which motivates the *co-visibility* constraint we add to the optimal control problem. Nevertheless, this allows for significant freedom in specifying both the path and the orientation of the aircraft between the waypoints in order to optimize an objective of interest. In this paper, we propose a novel algorithm that maximizes both the aggressiveness of the flight and the co-visibility of the visual landmarks.

Traditional methods for generating quadrotor trajectories exploit the differential flatness properties [4] of the dynamics of the quadrotor. The flat outputs, the trajectories of the x , y , z , and ψ components of the state of the quadrotor, can be specified independently, and furthermore, they uniquely determine the trajectories of the remaining components of the state as well as the required control inputs. Mellinger *et al.* [4] first proposed the idea of parametrizing flat outputs as polynomials in time, selecting coefficients that minimize the snap of the quadrotor. Richter *et al.* [5] proposed an extension to this work that bounds the actuator effort and adds a time penalty to the cost function to encourage fast flight. Other approaches, *e.g.*, proposed by Mueller *et al.* [6], make the assumption that the yaw of the quadrotor is constant during the execution of the trajectory and formulate the polynomial optimization problem in terms of minimum jerk.

In addition to planning dynamically feasible trajectories, autonomy entails accurate on-board state estimation. The difficulty of the latter task may vary significantly based on the choice of the trajectory. Prior work has considered planning to minimize various measures of state-estimation uncertainty with the goal of satisfying a chance constraint. The problem under consideration is to plan a path in the belief space that ensures the probability of collision with the environment is lower than a set threshold. Bry *et al.* [7] propose a sampling-based motion planning algorithm that generates a rapidly exploring random tree where the associated cost function accounts for both the cost of executing the path and the uncertainty associated with the action. The algorithm proposed by van den Berg *et al.* [8] accounts for uncertainty during the planning phase through forward propagation of the covariance matrix over paths planned using a RRT based planner. They use this to enumerate the paths and select the one with the lowest cost.

Bry *et al.* [7] and van den Berg *et al.* [8] solve problems involving chance constraints and require full knowledge of obstacle locations in the environment in addition to the locations of informative landmarks: beacons, visual landmarks, etc. The chance constraint can impose conditions on the full state of the robot under consideration, and these methods plan trajectories in the full state space of the robot. For instance as mentioned previously, quadrotor trajectory planning often takes place in the differentially flat space. However, to satisfy the chance constraint, these approaches would require either mapping the obstacles and landmarks

into this space or a projection of the differentially flat space back into the full state space to check the chance constraint during planning.

Approaches that specifically target visual-inertial navigation have been proposed. In [9], the authors propose a trajectory optimization method that first solves for a goal location given a target image and then plans the trajectory for which the reprojection error of the desired features is minimized during execution of the trajectory. In [10], the authors use a B-spline polynomial parameterization for the trajectory, and solve a minimum-time trajectory optimization problem with the objective of always keeping a certain set of features in the field of view. These methods are motivated by keeping a specified set of features in view. As such, they do not provide a flexible framework for allowing the algorithm to keep a smaller set of features in view to balance the perceptual constraints and the desired average speed while ensuring the actuator constraints are not violated. The set of possible trajectories that can be achieved using this constraint is often restrictive and not suitable to applications where aggression of the quadrotor is preferred over target tracking.

Falanga *et al.* [11] present an approach that adds a perception objective into their model predictive control framework. They model the cost as minimizing the velocity of a point of interest in the camera frame. They consider keeping a cluster of features within the field of view. While this holds when the visible features in the environment are clustered together at every position along the trajectory, there is room for improvement when the features are more dispersed. We address two main remaining challenges, (1) Enforcing the constraint that a significant number of features are visible in consecutive keyframes instead of enforcing that a certain set of features stay in view, (2) A computationally-efficient formulation to solve this problem while generating the reference trajectory to follow.

Passive methods for keeping features in view have also been studied. Attention and anticipation [12] has been studied for visual feature selection where the goal is to select features that will remain in the field of view without adding any actuator effort to the quadrotor. The authors of [12] use an information gain metric for a greedy algorithm to select features that will add the most information to the state estimate by forward simulating the dynamics of the quadrotor. However, this method separates perception from the planning and does not actuate the quadrotor to account for observing features.

The main contribution of this paper is an algorithm for the optimization of yaw trajectory to ensure the co-visibility of the landmarks from one frame to the next while ensuring agile navigation. Specifically, we first motivate the notion of landmark co-visibility between keyframes in the context of the trajectory generation problem for quadrotor aircraft. We further present an algorithm to generate feasible by construction trajectories that take into account the actuator limits and the perception constraint. We then present a differentiable relaxed constraint function that enables the optimizer to find a solution to the problem at keyframe rate.

II. PROBLEM STATEMENT

This section formally defines a new optimal control problem that embodies co-visibility constraints. The parameters of the camera include the following: π_x and π_y are the pixel width and height in meters; w and h are the width and height in pixels; and f is the focal length in meters. We reference coordinate frames fixed to the body of the vehicle and the world with subscripts b and W , respectively. The translation and orientation of the camera with respect to the world frame are represented by p_C^W and R_C^W , respectively. The coordinates of the i^{th} landmark with respect to the world frame are represented by l_i .

The dynamics of the vehicle are modeled by an ordinary differential equation of the following form:

$$\dot{\mathbf{X}}(t) = \mathbf{f}(\mathbf{X}(t), \mathbf{u}(t)),$$

where $\mathbf{X}(t)$ is the state, and $\mathbf{u}(t)$ is the input. Components of \mathbf{X} necessarily include the pose of the vehicle. The latter is defined by $x, y, z, \phi, \theta, \psi$, where x, y, z denote the position of the vehicle, and ϕ, θ, ψ denote the roll, pitch and yaw angles, respectively. The pose of the camera, rigidly attached to the body of the vehicle, is represented by $\mathbf{C}(t)$. The latter is a function of the pose of the vehicle:

$$\mathbf{C}(t) = \mathbf{g}(\mathbf{X}(t)),$$

for some function $\mathbf{g}(\cdot)$.

We are interested in maximizing *co-visibility* of landmarks between consecutive frames. Let $\{t_0, t_1, \dots, t_k, \dots, t_K\}$ denote the time instances corresponding to each *keyframe* when the camera captures an image. For example, in combination with the previous definition, $\mathbf{C}(t_k)$ denotes the pose of the camera at time t_k .

To determine the projection of a point in the camera frame, we use a perspective camera model. The homogeneous coordinates, p'_i , of the projection of landmark i are given by

$$p'_i = (R_C^W)^T [I | -p_C^W] \begin{bmatrix} l_i \\ 1 \end{bmatrix}. \quad (1)$$

The conditions for visibility on the sensor array read

$$-\frac{w}{2}\pi_x \leq f \frac{p'_i[0]}{p'_i[2]} \leq \frac{w}{2}\pi_x, \quad (2)$$

$$-\frac{h}{2}\pi_y \leq f \frac{p'_i[1]}{p'_i[2]} \leq \frac{h}{2}\pi_y, \quad (3)$$

$$p'_i[2] > 0. \quad (4)$$

We define the *visibility function* via

$$v(\mathbf{C}(t_k), l_i) = \begin{cases} 1, & \text{if landmark } i \text{ is visible} \\ & \text{to camera at pose } \mathbf{C}(t_k); \\ 0, & \text{otherwise,} \end{cases} \quad (5)$$

and the *co-visibility measure* between two consecutive keyframes, say t_{k-1} and t_k , as

$$\mu = \sum_{i=1}^n W_i v(\mathbf{C}(t_k), l_i) v(\mathbf{C}(t_{k-1}), l_i), \quad (6)$$

where n denotes the number of landmarks in the environment. To vary the relative importance of individual landmarks, we assign each one a positive scalar weight W_i . The co-visibility metric evaluates to a large value when a large number of features are visible from one keyframe to the next.

Finally, we are ready to state the problem formulation. We seek to maximize the co-visibility metric, while minimizing the time of arrival at a goal state, *i.e.*,

$$\max \quad \sum_{k=1}^K \sum_{i=1}^n W_i v(\mathbf{C}(t_k), l_i) v(\mathbf{C}(t_{k-1}), l_i) - \gamma_T T \quad (7a)$$

$$\text{s.t.} \quad \dot{\mathbf{X}}(t) = \mathbf{f}(\mathbf{X}(t), \mathbf{u}(t)) \forall t \in [0 \dots T], \quad (7b)$$

$$\mathbf{C}(t) = \mathbf{g}(\mathbf{X}(t)) \forall t \in [0 \dots T], \quad (7c)$$

$$u_{min} \leq \mathbf{u}(t) \leq u_{max} \forall t \in [0 \dots T]. \quad (7d)$$

T is the *mission time* and γ_T is the *time penalty* parameter that quantifies the tradeoff between maximizing the co-visibility metric and minimizing the mission time.

We stress that this problem augments a time-optimal control problem with the co-visibility objective. In this paper, we are concerned with solving an instance of this problem, focusing on a quadrotor aerial vehicle. In the next section, we recall the differential flatness property of quadrotor dynamics which allows tracking position and yaw references independently. Specifically, we consider the special case when the translatory components of the path of the quadrotor are given by a collision-free trajectory around the obstacles, leaving the aircraft's average speed and yaw angle trajectory free.

III. ALGORITHM

This section is devoted to the presentation of a novel algorithm for the problem presented in the previous section. We describe the dynamics model under consideration and the bounds on the actuator limits in Section III-A. Next, we discuss the differentiable relaxation of the objective function that allows us to compute the trajectories in real time in Section III-B. The geometric interpretation of the relaxed function is defined in Section III-C. We then present the proposed algorithm and implementation details in Section III-D.

A. Quadrotor dynamics and differential flatness

In particular, we consider the dynamics of a quadrotor for the dynamics constraint (Equation (7b)), which can be written as:

$$\dot{p} = v \quad (8)$$

$$\dot{v} = -gz_w + c\vec{z}_b \quad (9)$$

$$\dot{R} = R\hat{w} \quad (10)$$

$$\dot{w} = J^{-1}(\tau - w \times Jw). \quad (11)$$

Mellinger *et al.* [4] show that the quadrotor model written in this form is differentially flat. The flat outputs are defined as:

$$\sigma = [x, y, z, \psi]. \quad (12)$$

This property is often exploited to plan the trajectories involving the states x, y, z, ψ separately. We utilize a time

scaling parameter α that intuitively speeds up or slows down time to ensure the given trajectories remain dynamically feasible while minimizing the total execution time. Indeed, the new trajectories, denoted by $\tilde{x}, \tilde{y}, \tilde{z}, \tilde{\psi}$, are defined by $\tilde{x}(t) = x(\alpha t)$, and similarly for $\tilde{y}, \tilde{z}, \tilde{\psi}$. Values of α greater than one induce more aggressive behavior; the execution time decreases, and the total thrust and torque increases. The level to which we can change α is constrained by the following inequalities obtained from the constraints defined by Equations (13) and (16).

The actuator constraints (Equation (7d)) for the quadrotor model are as follows. The total mass normalized thrust with respect to the world frame is given by $c(t) = [\ddot{x}(t), \ddot{y}(t), \ddot{z}(t) + g]^T$, where $[x, y, z]^T$ represents the position of the center of mass of the quadrotor as a function of time. This can be simply modeled by

$$\left\| \alpha^2 \begin{bmatrix} \ddot{x}(t) \\ \ddot{y}(t) \\ \ddot{z}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \right\| \leq c_{max} \quad (13)$$

For the purpose of modeling torque, we assume the body of the quadrotor is invariant under rotations by angle $\frac{\pi}{2}$ about its body z axis. This implies its moment of inertia with respect to the body frame takes on the form

$$\begin{bmatrix} I_x & 0 & 0 \\ 0 & I_x & 0 \\ 0 & 0 & I_z \end{bmatrix} \quad (14)$$

The coordinates of the torque with respect to the world frame, $\vec{\tau}^W$ can be obtained from

$$\vec{\tau}^W = \omega_3 I_z \vec{z}_b^W + \omega_3 (2I_x - I_z) \dot{\vec{z}}_b^W + I_x \vec{z}_b^W \times \ddot{\vec{z}}_b^W. \quad (15)$$

where \vec{z}_b^W denotes coordinates of the body z axis with respect to the world frame.

So, the bound on the torque can be applied using

$$\|\tau(t)\| \leq \tau_{max} \quad \forall t \in [0...T] \quad (16)$$

In practice, the maximum mass-normalized thrust c_{max} and the maximum torque τ_{max} are set empirically.

B. Modeling visibility using a relaxed differentiable function

The landmarks are projected into the camera using a perspective camera model. A point lies within the field of view if and only if it belongs to the viewing frustum. A convenient representation of the viewing frustum is an intersection of five half spaces. One half space corresponds to the region in front of the sensor of the camera. The remaining four half spaces are constrained by planes defined by the optical center of the camera and edges of the sensor. To leverage efficient nonlinear optimization techniques, we express membership of the field of view with a differentiable, relaxed indicator function. The indicator function is defined as the product of five differentiable functions, one assigned to each of the five aforementioned spaces. Each maps \mathbb{R}^3 to the interval $[0, 1]$; points further inside a given half space receive progressively higher values.

Finally, the scaling variable s determines the sharpness of the relaxed indicator function. We comment on its significance in Section III-C.

Let us derive the form of the differentiable indicator function, which we also refer to as the relaxed visibility function. A feature with position l_i with respect to the world frame has coordinates

$$z_i = R_W^C (l_i - p_C^W) \quad (17)$$

with respect to the camera frame. The displacements from the optical center to the vertices of the sensor are given by the following set of vectors, specified with respect to the camera frame:

$$v_{tr} = \begin{bmatrix} (w - c_x)\pi_x \\ (-c_y)\pi_y \\ f \end{bmatrix}, \quad (18)$$

$$v_{tl} = \begin{bmatrix} (-c_x)\pi_x \\ (-c_y)\pi_y \\ f \end{bmatrix}, \quad (19)$$

$$v_{lr} = \begin{bmatrix} (w - c_x)\pi_x \\ (h - c_y)\pi_y \\ f \end{bmatrix}, \quad (20)$$

$$v_{ll} = \begin{bmatrix} (-c_x)\pi_x \\ (h - c_y)\pi_y \\ f \end{bmatrix}. \quad (21)$$

These determine normals to four of the five half planes of the frustum: $v_{tr} \times v_{lr}$, $v_{tl} \times v_{tr}$, $v_{ll} \times v_{tl}$ and $v_{lr} \times v_{ll}$. The fifth normal is simply $[0, 0, 1]^T$. Using the vector

$$o = \begin{bmatrix} \frac{1}{2} \left(1 + \tanh \left(\frac{(v_{tr} \times v_{lr}) \cdot z_i}{s} \right) \right) \\ \frac{1}{2} \left(1 + \tanh \left(\frac{(v_{tl} \times v_{tr}) \cdot z_i}{s} \right) \right) \\ \frac{1}{2} \left(1 + \tanh \left(\frac{(v_{ll} \times v_{tl}) \cdot z_i}{s} \right) \right) \\ \frac{1}{2} \left(1 + \tanh \left(\frac{(v_{lr} \times v_{ll}) \cdot z_i}{s} \right) \right) \\ \frac{1}{2} \left(1 + \tanh \left(\frac{z_i \cdot e_3 - f}{s} \right) \right) \end{bmatrix}, \quad (22)$$

the relaxed visibility function can be defined as:

$$v_s(C(t_k), l_i) = \prod_{i=1}^5 o_i \quad (23)$$

C. Geometric interpretation of the relaxed function

In the limit as s tends to 0, the relaxed visibility function converges pointwise to the indicator of the interior of the view frustum. The lower the value of s , the closer the relaxed function represents whether or not a given feature lies within field of view. However, too small a value for s leads to vanishing gradients at a significant subset of features. To avoid this numerical problem, we set s to be the average distance of features from the body. A graphical illustration of the effect of s on the value of the relaxed indicator function is shown in Figure 3. In the figure, the camera is pointed upwards along the \vec{z} direction.

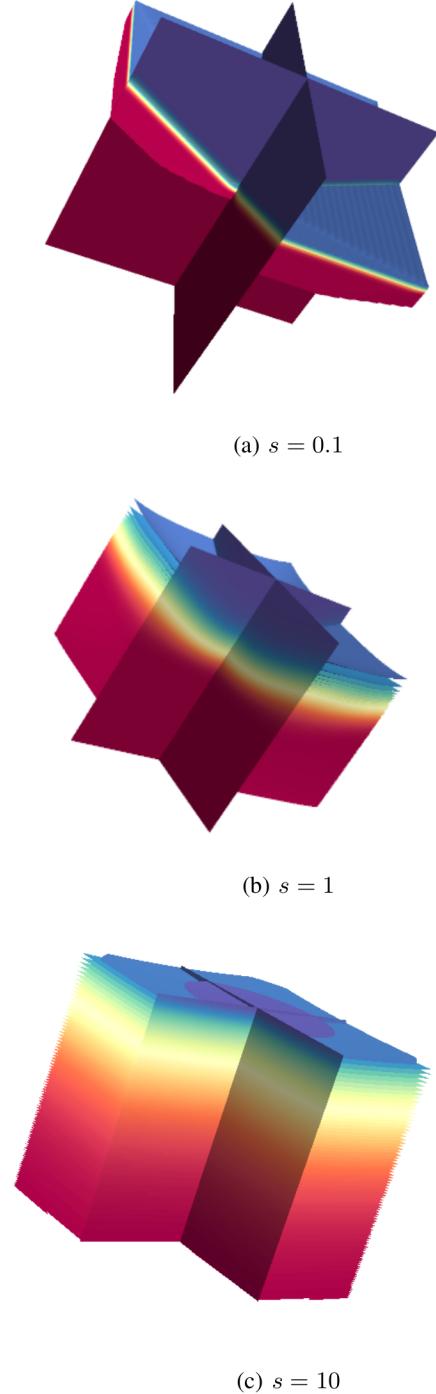


Fig. 3: Differentiable relaxation of the indicator function of the interior of the view frustum. Points at which the function takes values close to zero are colored red and values close to one are colored blue. We consider a family of such functions, indexed by a non-negative parameter s . The three figures above illustrate the effect of increasing s . Smaller values of s represent the true indicator function more accurately, whereas larger values of s lead to more numerically stable algorithms.

D. Proposed Algorithm

The numerical procedure is described in Algorithm 1. We let $\langle w_0, w_1, \dots, w_k, \dots, w_K \rangle$ denote the sequence of waypoints the quadrotor must follow and represent candidate trajectories of x , y , z , and ψ using polynomials $p_x(t)$, $p_y(t)$, $p_z(t)$, and $p_\psi(t)$, respectively.

Data: τ_{max}, c_{max}, w_k
Result: $\left[[p_x, p_y, p_z, p_\psi]_0^1, \dots, [p_x, p_y, p_z, p_\psi]_{k-1}^k \right]$

Step 1:

Find $p_x(t), p_y(t), p_z(t)$ that solve

$$\min \quad \alpha \gamma_T \sum_{i=1}^M T_i + \quad (24a)$$

$$\begin{bmatrix} p_1 \\ \vdots \\ p_K \end{bmatrix}^T Q(\alpha T_i) \begin{bmatrix} p_1 \\ \vdots \\ p_K \end{bmatrix}^T \quad (24b)$$

$$\text{s.t. } 0 \leq c(t) \leq c_{max} \forall t, \quad (24c)$$

$$\tau(t) \leq \tau_{max} \forall t, \quad (24d)$$

$$\mathcal{A}(\alpha T_i) \begin{bmatrix} p_1 \\ \vdots \\ p_K \end{bmatrix} = \begin{bmatrix} d_1 \\ \vdots \\ d_K \end{bmatrix} \quad (24e)$$

Step 2:

for $k = 1$ to K do

 find $p_\psi(t)$ for $t \in (t_{k-1}, t_k)$ that solves

$$\min_{p_\psi} \quad \int_{t_{k-1}}^{t_k} (\ddot{p}_\psi(t))^2 dt - \sum_{l \in L} \mu(l) \quad (25a)$$

$$\text{s.t. } \tau(t) \leq \tau_{max} \quad (25b)$$

$$\text{s.t. } \mathcal{A}(t_k) \begin{bmatrix} p_1 \\ \vdots \\ p_K \end{bmatrix} = \begin{bmatrix} d_1 \\ \vdots \\ d_K \end{bmatrix} \quad (25d)$$

end

Algorithm 1: Proposed two step algorithm to solve the optimal control problem defined in (7)

Step 1 generates trajectories of x , y , and z . Although numerous existing trajectory generation algorithms could be used, we choose the method given in [5]. It extends the widely used minimum snap polynomial optimization [4] approach using two key ideas. Firstly, it augments the minimum snap objective with minimizing the total completion time of all segments between consecutive waypoints. The block diagonal matrix $Q(T)$ concatenates matrices formed by taking (higher order) derivatives of polynomials p_x , p_y , and p_z with respect to their coefficients. In this particular case, we are interested in minimizing the snap, and so all the elements that do not correspond to the fourth derivative are set to 0. This matrix acts as the cost encouraging smoothness

of the fourth derivative (Equation (24b)). It has the form:

$$Q(T) = \begin{bmatrix} Q_1(T_1) & & \\ & \ddots & \\ & & Q_K(T_K) \end{bmatrix} \quad (26)$$

The matrix $\mathcal{A}(t)$ represents a mapping from the coefficients of the polynomials to their (higher order) derivatives, enforcing smoothness constraints at endpoints of adjacent segments. Second, the method of [5] incorporates actuator constraints during the optimization process by making sure they are satisfied at a set of collocation points.

With the trajectory corresponding to variables x , y , and z determined, step 2 computes trajectories of θ and ϕ by formulating the optimization problem (Equation (25)) as:

$$\min_{p_\psi} \int_{t_k}^{t_{k+1}} (\ddot{p}_\psi)^2 dt - \gamma_c \sum_{i,k} [v(C_k, l_i) v(C_{k-1}, l_i)], \quad (27)$$

where γ_c is a penalization term on the perception objective. Note we assume that all landmarks are weighted equally.

This is approximated in our problem using the relaxed indicator function, so Equation (27) reduces to

$$\min_{p_\psi} \int_{t_k}^{t_{k+1}} \gamma_\psi (\ddot{p}_\psi)^2 dt - \gamma_c \sum_{i,k} [v_s(C(t_k), l_i) v_s(C(t_{k-1}), l_i)]. \quad (28)$$

In Equation (28), the summation can be parallelized since all the elements of the sum are independent of each other, which leads to efficient implementation. We implement this algorithm using C++. This can be solved efficiently using an off the shelf nonlinear solver such as the SLQSP solver [13] in the NLOPT package [14]. Note, the problem as defined is non-convex in general and will only converge to a local optimum.

IV. EXPERIMENTS

In this section, we describe the experiments performed on a real quadrotor to validate our algorithm and present the results of these experiments. We fly generated reference trajectories using a custom built quadrotor platform in an 11m x 11m x 5.5m motion capture room, with 24 OptiTrack Prime 17W cameras providing the ground truth position of the quadrotor. For the purpose of state estimation, there is an Xsens MTi-3 IMU onboard the quadrotor. Visual data is generated in real-time using the FlightGoggles [2] photo-realistic image generation system. The visual inertial odometry system fuses the information from the IMU and camera to generate an estimate of the state of the quadrotor.

We design a trajectory to fly in the environment shown in Figure 4. The room has been emptied to demonstrate the effectiveness of the approach. The walls are textureless and thus devoid of visual landmarks but the center of the room has pillars and statues that are landmark-rich. The reference trajectory generated by our proposed method is shown in Figure 5.

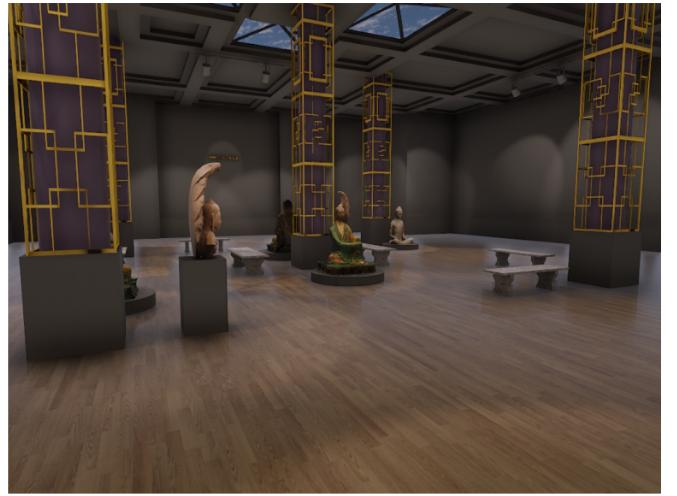


Fig. 4: The figure shows the environment that is used for the experiments. This environment has no landmarks on the walls and all the objects are in the center of the room.

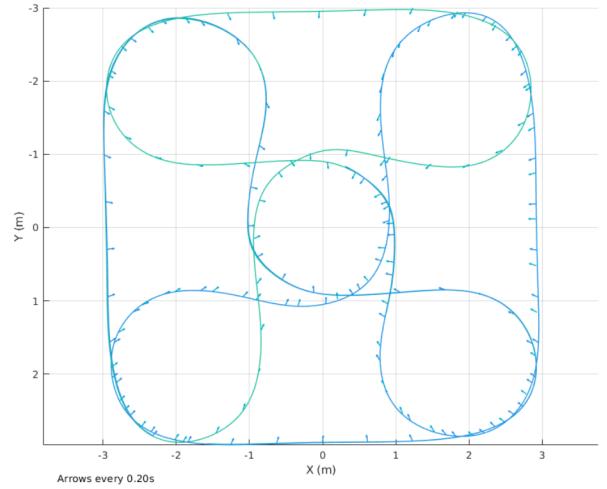


Fig. 5: The figure shows the optimized trajectory generated by the proposed algorithm after taking the perceptual constraints into account. As can be seen in the figure, the quadrotor prefers facing towards the objects in the environment and avoids facing the empty walls.

In this section, we assume locations of the landmarks in the environment are known a priori. This is typical in some autonomous quadrotor aircraft scenarios such as drone racing, industrial warehouses, and more general scenarios where the environment has been mapped beforehand and the topological visual map of the environment is known. In practice, the landmark locations used could be replaced by accumulating triangulated landmarks from visual inertial odometry over a short window. For the purpose of experimentation, we relax this constraint.

Since we would like to optimize the polynomials over segments defined by keyframes, we want the optimization to be performed at keyframe rate at the very least. In

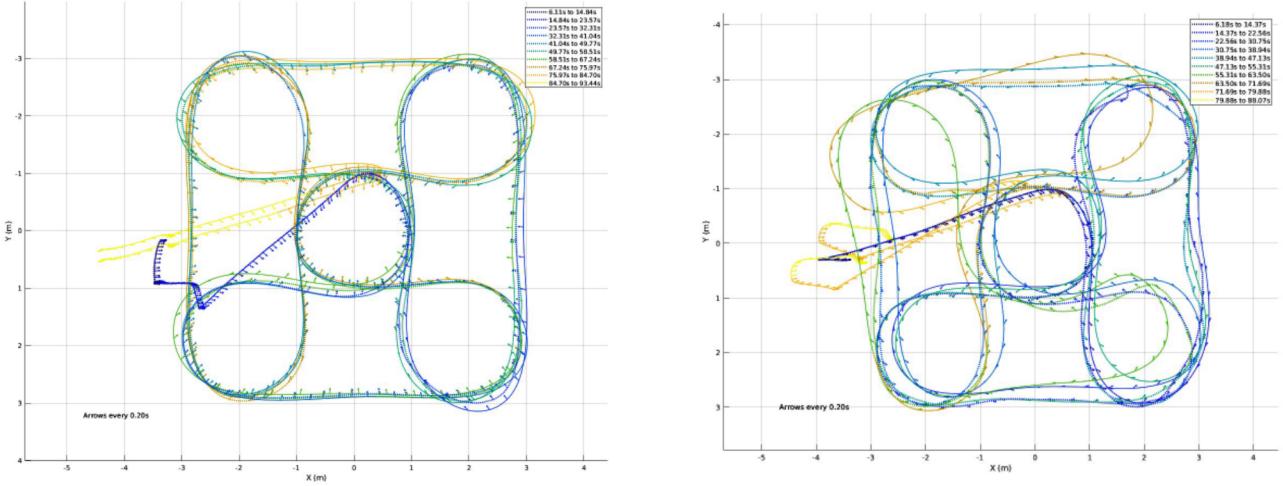


Fig. 6: The Figure above shows the qualitative comparison between the quadrotor following the reference trajectory at $2.7ms^{-1}$ closed by visual inertial odometry. On the left, the trajectory flown with optimized yaw is shown. On the right, the trajectory flown with a forward facing yaw is shown.

practice, we use a keyframe rate of $10Hz$ and indeed our implementation computes the polynomials in less than $0.1s$. Our timing experiments were run on a 10 core Intel® i9-7900X CPU with 32GB of RAM. As mentioned previously, it is possible to write an efficient implementation that uses a GPU which would fit on the Tegra TX2 which is currently onboard our quadrotor aircraft platform.

In the rest of this section, we present two experiments. In the first experiment, the state estimate for closing the control loop is provided by the motion capture system, whereas in the second experiment, the state estimation is provided by visual inertial odometry. We compare the effect on the performance of the visual front-end and the tracking error of the controller with respect to the reference trajectory and our optimized trajectory. The reference trajectory follows a forward facing objective for the yaw. The parameters for the system are kept constant across both of the experiments.

A. Experiment 1

For the first experiment the quadrotor is commanded to perform the trajectory at various top speeds ($1.7ms^{-1}$, $2.7ms^{-1}$, $3.4ms^{-1}$) with the state estimation from the motion capture system being used to close the control loop. The reference trajectory that is generated by applying our approach is shown in Figure 5. This trajectory is controlled using a nonlinear dynamics inversion based controller as described in [15]. In the figure, the arrows represent the heading angle of the quadrotor. As can be seen in the figure, the quadrotor tries to keep the yaw angle pointed towards the feature rich part of the environment which is the expected behavior.

In this experiment, the average number of tracked landmarks between consecutive keyframes is measured and compared between a forward facing trajectory and the optimized yaw trajectory. The parameters for both the state estimation

	Forward Facing	Optimized Yaw
$1.7ms^{-1}$	134.1836	180.0496
$2.7ms^{-1}$	126.7671	179.1885
$3.4ms^{-1}$	122.4460	179.1367

TABLE I: The average number of tracked features between keyframes in Experiment 1 is shown above.

and the controller are kept the same across all speeds. The results of this experiment is shown in Table I. As can be seen in the table, a significantly larger number of landmarks is tracked across different speeds, which shows that the perception objectives are maximized by our proposed method. Since the number of landmarks tracked across keyframes is significantly larger, this directly validates that our approach is able to generate trajectories that can maintain co-visibility of a large number of landmarks between keyframes.

B. Experiment 2

In this experiment, the quadrotor is commanded to fly the same trajectory three times with top speeds of $1.7ms^{-1}$ and $2.7ms^{-1}$ with the state estimate to close the control loop provided by visual inertial odometry for both the forward facing yaw and the optimized yaw. The parameters are again kept the same for every trial. The qualitative comparison is shown in Figure 6. As can be seen from the figure, the quadrotor fails to follow the trajectory while attempting to face forward, but is successful when the optimized reference trajectory is used. It is important to note here that the state estimation error drifts because the walls are featureless and the poor state estimation results in unsuccessful tracking.

The quantitative results for this evaluation are shown in Table II. The metric we use to measure the tracking performance of the controller with the reference trajectories is the absolute trajectory error statistics. These are the

	Forward Facing	Optimized Yaw
$1.7ms^{-1}$	Mean	0.3675m
	Median	0.3234m
	RMSE	0.4838m
$2.7ms^{-1}$	Mean	0.5152m
	Median	0.5185m
	RMSE	0.5958m

TABLE II: The average absolute trajectory error statistics over 3 trials of the trajectory are shown in the above table.

average root-mean-square error (RMSE), mean, and median of the trajectory errors over the three separate trials for each instance of the forward facing and the optimized yaw trajectory. As can be seen in the table, there is a significant improvement in the absolute trajectory error statistics for the optimized yaw trajectory over the forward facing behavior. As noted in the qualitative comparison, this can be attributed to having a better state estimate due to a larger number of constraints between keyframes.

V. CONCLUSIONS

We present an addition to the optimal control problem for a quadrotor aircraft in the form of *co-visibility* constraints and an algorithm that generates reference trajectories aware of the perception constraints of the aircraft. From our results, Table I shows that the reference trajectory generated by our approach leads to a higher number of tracked landmarks, which in turn results in a better state estimate. This also serves as a good indicator that there are a large number of co-visible landmarks between keyframes generated by our method.

Our experiments demonstrate that with this approach, the performance of a state-of-the-art control system (using state estimates provided by visual-inertial odometry) is improved significantly. For example, the RMSE error decreases by 53.57% in $1.7ms^{-1}$ trajectories and 69.75% for $2.7ms^{-1}$ trajectories. The parameter α sets the aggressiveness of the quadrotor, which in turn partially constrains the orientations of the camera along the trajectory being followed. For applications where a known subset of landmarks must always be kept in view, optimizing only the yaw trajectory of the quadrotor might not be sufficient for the required objective. In such cases, the aggressiveness variable can be adjusted to accommodate the orientation targets of the drone since the viewpoint is defined geometrically. This can be achieved by specifying the roll and pitch as endpoint constraints (Equation (25d)) in the optimization problem using the described framework.

As mentioned in Section III, the problem definition is non-convex in general and the proposed solution only converges to a local minimum. As such, it is purely exploitative in nature and tries to maintain the *visibility* of already observed features in the environment. In future work, we propose to use this to our advantage to generate trajectories that make small variations on a pre-defined behavior for applications where the forward facing or exploratory behavior of the

quadrotor aircraft is important. This can be used to balance the exploitative behavior in cases where we also want to discover new landmarks in the environment.

This method can also be easily extended to accumulate landmarks over short time windows from onboard visual inertial odometry for online trajectory planning. This could be achieved by first initializing landmarks with a fixed yaw angle and then adjusting the yaw trajectory as a sufficient number of landmarks have been accumulated. Occlusion detection and handling could also be included into the optimization process. This could allow the optimizer to weight features that could be in occlusion in the future lower. The indicator function is only designed to handle static landmarks and can be extended to include dynamic objects in the environment.

REFERENCES

- [1] G. Loianno, C. Brunner, G. McGrath, and V. Kumar, “Estimation, control, and planning for aggressive flight with a small quadrotor with a single camera and imu,” *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 404–411, 2017.
- [2] T. Sayre-McCord, W. Guerra, A. Antonini, J. Arneberg, A. Brown, G. Cavalheiro, Y. Fang, A. Gorodetsky, D. McCoy, S. Quilter, F. Richter, E. Tal, Y. Terzioglu, L. Carlone, and S. Karaman, “Visual-inertial navigation algorithm development using photorealistic camera simulation in the loop,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018.
- [3] R. Mur-Artal and J. D. Tardós, “Orb-slam2: An open-source slam system for monocular, stereo, and rgbd cameras,” *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.
- [4] D. Mellinger and V. Kumar, “Minimum snap trajectory generation and control for quadrotors,” in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2520–2525.
- [5] C. Richter, A. Bry, and N. Roy, “Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments,” in *Robotics Research*. Springer, 2016, pp. 649–666.
- [6] M. W. Mueller, M. Hehn, and R. D’Andrea, “A computationally efficient motion primitive for quadrocopter trajectory generation,” *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1294–1310, 2015.
- [7] A. Bry and N. Roy, “Rapidly-exploring random belief trees for motion planning under uncertainty,” in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 723–730.
- [8] J. Van Den Berg, P. Abbeel, and K. Goldberg, “Lgg-mp: Optimized path planning for robots with motion uncertainty and imperfect state information,” *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 895–913, 2011.
- [9] M. Sheckells, G. Garimella, and M. Kobilarov, “Optimal visual servoing for differentially flat underactuated systems,” in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 2016, pp. 5541–5548.
- [10] B. Penin, R. Spica, P. R. Giordano, and F. Chaumette, “Vision-based minimum-time trajectory generation for a quadrotor uav,” in *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*. IEEE, 2017, pp. 6199–6206.
- [11] D. Falanga, P. Foehn, P. Lu, and D. Scaramuzza, “PAMPC: Perception-aware model predictive control for quadrotors,” in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2018.
- [12] L. Carlone and S. Karaman, “Attention and anticipation in fast visual-inertial navigation,” in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 3886–3893.
- [13] D. Kraft, “A software package for sequential quadratic programming,” *Forschungsbericht- Deutsche Forschungs- und Versuchsanstalt fur Luft- und Raumfahrt*, 1988.
- [14] S. G. Johnson, “The nlopt nonlinear-optimization package,” 2014.
- [15] E. Tal and S. Karaman, “Precision tracking of aggressive quadrotor trajectories using incremental nonlinear dynamic inversion and differential flatness,” in *Decision and Control (CDC), 2018 IEEE 57th Conference on*. IEEE, 2018.