Aggressive Quadrotor Flight through Narrow Gaps with Onboard Sensing and Computing using Active Vision

Davide Falanga, Elias Mueggler, Matthias Faessler and Davide Scaramuzza

***Abstract*— We address one of the main challenges towards autonomous quadrotor flight in complex environments, which is flight through narrow gaps. While previous works relied on off-board localization systems or on accurate prior knowledge of the gap position and orientation in the world reference frame, we rely solely on onboard sensing and computing and estimate the full state by fusing gap detection from a single onboard camera with an IMU. This problem is challenging for two reasons: (i) the quadrotor pose uncertainty with respect to the gap increases quadratically with the distance from the gap; (ii) the quadrotor has to actively control its orientation towards the gap to enable state estimation (i.e., active vision). We solve this problem by generating a trajectory that considers geometric, dynamic, and perception constraints: during the approach maneuver, the quadrotor always faces the gap to allow state estimation, while respecting the vehicle dynamics; during the traverse through the gap, the distance of the quadrotor to the edges of the gap is maximized. Furthermore, we replan the trajectory during its execution to cope with the varying uncertainty of the state estimate. We successfully evaluate and demonstrate the proposed approach in many real experiments, achieving a success rate of 80% and gap orientations up to $45°$. To the best of our knowledge, this is the first work that addresses and achieves autonomous, aggressive flight through narrow gaps using only onboard sensing and computing and without prior knowledge of the pose of the gap.**

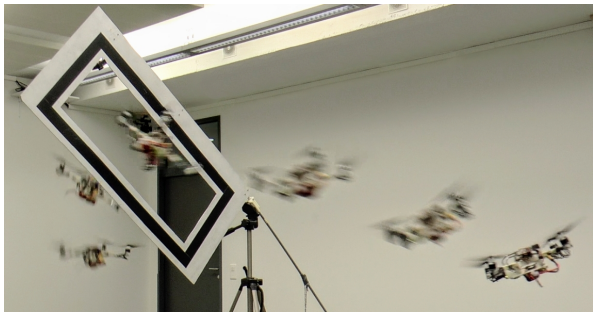
Supplementary Material

The accompanying video is available at:
`http://rpg.ifi.uzh.ch/aggressive_flight.html`

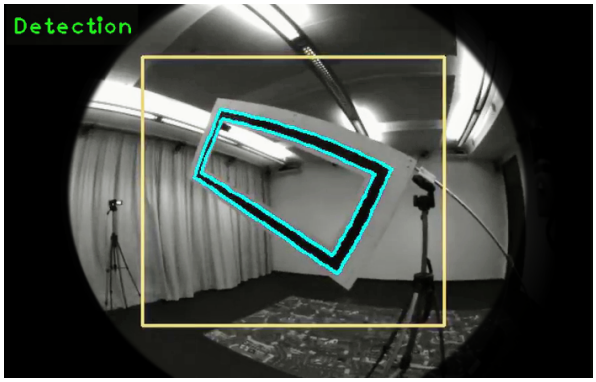
I. Introduction

Recent works have demonstrated that micro quadrotors are extremely agile and versatile vehicles, able to execute very complex maneuvers [1], [2], [3]. These demonstrations highlight that one day quadrotors could be used in search and rescue applications, such as in the aftermath of an earthquake, to navigate through buildings, by entering and exiting through narrow gaps, and to quickly localize victims.

In this paper, we address one of the main challenges towards autonomous quadrotor flight in complex environments, which is flight through narrow gaps. What makes this problem challenging is that the gap is very small, such that precise trajectory-following is required, and can be oriented arbitrarily, such that the quadrotor cannot fly through it in near-hover conditions. This makes it necessary to execute an aggressive trajectory (i.e., with high velocity and angular accelerations) in order to align the vehicle to the gap orientation (cf. Fig. 1).



(a) The quadrotor passing through the gap.



(b) View from the onboard camera

Fig. 1: Sequence of our quadrotor passing through a narrow, $45°$-inclined gap. Our state estimation fuses gap detection from a single onboard forward-facing camera with an IMU. All planning, sensing, control run fully onboard on a smartphone computer.

Previous works on aggressive flight through narrow gaps have focused solely on the control and planning problem and therefore relied on accurate state estimation from external motion-capture systems and/or accurate knowledge of the gap position and orientation in the world reference frame. Since these systems were not gap-aware, the trajectory was generated before execution and never replanned. Therefore, errors in the measure of the pose of the gap in the world frame were not taken into account, which may lead to a collision with gap. Conversely, we are interested in using only *onboard sensing and computing*, *without any prior knowledge* of the gap pose in the world frame. More specifically, we address the case where state estimation is done

The authors are with the Robotics and Perception Group, University of Zurich, Switzerland—`http://rpg.ifi.uzh.ch`. This research was funded by the DARPA FLA Program, the National Center of Competence in Research (NCCR) Robotics through the Swiss National Science Foundation and the SNSF-ERC Starting Grant.

by fusing gap detection through a single, forward-facing camera with an IMU. We show that this raises an interesting *active-vision* problem (i.e, coupled perception and control). Indeed, for the robot to localize with respect to the gap, a trajectory that guarantees that the quadrotor always faces the gap must be selected (perception constraint). Additionally, it must be replanned multiple times during its execution to cope with the varying uncertainty of the state estimate, which is quadratic with the distance from the gap. Furthermore, during the traverse, the quadrotor must maximize the distance from the edges of the gap (geometric constraint) to avoid collisions. At the same time, it must do so without relying on any visual feedback (when the robot is very close to the gap, it exits from the field of view of the camera). Finally, the trajectory must be feasible with respect to the dynamic constraints of the vehicle.

Our proposed trajectory generation approach is independent of the gap-detection algorithm being used; thus, to simplify the perception task, we use a gap with a black-and-white rectangular pattern (cf. Fig. 1) for evaluation and demonstration.

A. Related Work

A solution for trajectory planning and control for aggressive quadrotor flight was presented in [3]. The authors demonstrated their results with aggressive flight through a narrow gap, and by perching on inclined surfaces. The quadrotor state was obtained using a motion-capture system. To fly through a narrow gap, the vehicle started by hovering in a pre-computed position, flew a straight line towards a launch point, and then controlled its orientation to align with the gap. The method was not *plug-and-play* since it needed training through *iterative learning* in order to refine the launch position and velocity. This was due to the instantaneous changes in velocity caused by the choice of a straight line for the approach trajectory. Unlike their method, we use a technique that computes polynomial trajectories which are guaranteed to be feasible with respect to the control inputs. The result is a smooth trajectory, compatible with the quadrotor dynamic constraints, which makes learning unnecessary. Indeed, in realistic scenarios, such as search-and-rescue missions, we cannot afford training but must pass on the first attempt.

In [4], the same authors introduced a method to compute trajectories for a quadrotor solving a Quadratic Program, which minimizes the snap (i.e., the fourth derivative of position). In their experiments, agile maneuvers, such as passing through a hula-hoop thrown by hand in the air, were demonstrated using state estimation from a motion-capture system.

In [5], a technique that lets a quadrotor pass through a narrow gap while carrying a cable-suspended payload was presented and was experimentally validated using a motion-capture system for state estimation.

In [6], the authors proposed an unconstrained nonlinear model predictive control algorithm in which trajectory generation and tracking are treated as a single, unified problem. The proposed method was validated in a number of experiments, including a rotorcraft passing through an inclined gap. Like the previous systems, they used a motion-capture system for state estimation.

In [7], the authors proposed a vision-based method for autonomous flight through narrow gaps by fusing data from a downward and a forward-looking camera, and an IMU. Trajectory planning was executed on an external computer. However, the authors only considered the case of an horizontal gap, therefore no agile maneuver was necessary.

In [8], the authors proposed methods for onboard vision-based state estimation, planning, and control for small quadrotors, and validated the approach in a number of agile maneuvers, among which flying through an inclined gap. Since state estimation was performed by fusing input from a downward-looking camera and an IMU, rather than from gap detection, the gap position and orientation in the world reference frame had to be measured very accurately prior to the execution of the maneuver. The trajectory was generated before execution and never replanned. Therefore, errors in the measure of the pose of the gap in the world frame were not taken into account, which may lead to a collision with gap. To deal with this issue, the authors used a gap considerably larger than the vehicle size.

All the related works previously mentioned relied on the accurate state estimates from a motion-capture system or accurate prior knowledge of the gap position and orientation in the world reference frame. Additionally, in all these works but [6] and [8] trajectory generation was performed on an external computer. The advantages of a motion-capture system over onboard vision are that the state estimate is always available, at high frequency, accurate to the millimeter, and with almost *constant noise covariance* within the tracking volume. Conversely, a state estimate from onboard vision can be intermittent (e.g., due to misdetections); furthermore, its covariance increases *quadratically* with the distance from the scene and is strongly affected by the type of structure and texture of the scene. Therefore, to execute a complex aggressive maneuver, like the one tackled in this paper, while using only onboard sensing and *gap-aware* state estimation, it becomes necessary *to couple perception with the trajectory generation process* (i.e., active vision). Specifically, the desired trajectory has to render the gap always visible by the onboard camera in order to estimate its relative pose.

B. Contributions

Our method differs from previous works in the following aspects: (i) we rely solely on onboard, visual-inertial sensors and computing, (ii) we generate a trajectory that facilitates the perception task, while satisfying geometric and dynamic constraints, and (iii) we do not require iterative learning, neither do we need to know a priori the gap position and orientation in the world frame. To the best of our knowledge, this is the first work that addresses and achieves aggressive flight through narrow gaps with state estimation via gap detection from an onboard camera and IMU.

The remainder of this paper is organized as follows. Section II presents the proposed trajectory-generation algorithm. Section III describes the state-estimation pipeline. Section IV presents the experimental results. Section V discusses the results and provides additional insights about the approach. Finally, Section VI draws the conclusions.

II. Trajectory Planning

We split the trajectory planning into two consecutive stages. First, we compute a traverse trajectory to pass through the gap. This trajectory maximizes the distance from the vehicle to the edges of the gap in order to minimize the risk of collision. In a second stage, we compute an approach trajectory in order to fly the quadrotor from its current hovering position to the desired state that is required to initiate the traverse trajectory. While both trajectories need to satisfy *dynamic constraints*, the approach trajectory also satisfies *perception constraints*, i.e., it lets the vehicle-mounted camera always face the gap. This is necessary to enable state estimation with respect to the gap.

A. Traverse Trajectory

During the gap traversal, the quadrotor has to minimize the risk of collision. We achieve this by forcing the traverse trajectory to intersect the center of the gap while simultaneously lying in a plane orthogonal to the gap (see Fig. 2). In the following, we derive the traverse trajectory in this orthogonal plane and then transform it to the 3D space.

Let W be our world frame. The vector $\mathbf{p}_G$ and the rotation matrix $\mathbf{R}_G$ denote the position of the geometric center of the gap and its orientation with respect to W, respectively. Let Π be a plane orthogonal to the gap, passing through its center and parallel to the longest side of the gap (cf. Fig. 2). Let $\mathbf{e}_1$ and $\mathbf{e}_2$ be the unit vectors spanning such a plane Π, whose normal unit vector is $\mathbf{e}_3$. The $\mathbf{e}_2$ axis is orthogonal to the gap and $\mathbf{e}_1 = \mathbf{e}_2 \times \mathbf{e}_3$.

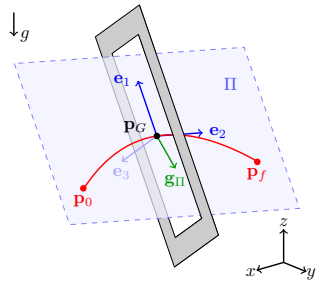


Fig. 2: An inclined gap and the corresponding plane Π.

Intuitively, a trajectory that lies in the plane Π and passes through the center of the gap, minimizes the risk of impact with the gap.

To constrain the motion of the vehicle to the plane Π, it is necessary to compensate the projection of the gravity vector $\mathbf{g}$ onto its normal vector $\mathbf{e}_3$. Therefore, a constant thrust of magnitude $\langle \mathbf{g}, \mathbf{e}_3 \rangle$ needs to be applied orthogonally to Π. By doing this, a 2D description of the quadrotor's motion in this plane is sufficient. The remaining components of $\mathbf{g}$ in the plane Π are computed as

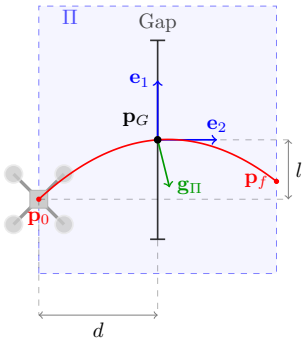


Fig. 3: The traverse trajectory in the plane Π.

$$\mathbf{g}_\Pi = \mathbf{g} - \langle \mathbf{g}, \mathbf{e}_3 \rangle \, \mathbf{e}_3. \tag{1}$$

Since this is a constant acceleration, the motion of the vehicle along Π is described by the following second order polynomial equation:

$$p_i(t) = p_i(t_0) + v_i(t_0)t + \frac{1}{2} g_{\Pi,i} t^2, \tag{2a}$$

$$v_i(t) = v_i(t_0) + g_{\Pi,i} t, \tag{2b}$$

where the subscript $i = \{1, 2\}$ indicates the component along the $\mathbf{e}_i$ axis. The quadrotor enters the traverse trajectory at time t_0, t is the current time, and p and v denote its position and velocity, respectively.

Equation (2) describes a ballistic trajectory. When $g_{\Pi,2} = 0$, it is the composition of a uniformly accelerated and a uniform-velocity motion. In other words, in these cases the quadrotor moves on a parabola in space.

Let l and d be the distance between $\mathbf{p}_G$ and the initial point of the trajectory, $\mathbf{p}_0$, along $\mathbf{e}_1$ and $\mathbf{e}_2$, respectively (cf. Fig. 3). These two parameters determine the initial position and velocity in the plane Π, as well as the time t_c necessary to reach $\mathbf{p}_G$. The values of d and l are determined through an optimization problem, as explained later in Sec. II-B.

For a generic orientation $\mathbf{R}_G$ of the gap, (2) is characterized by a uniformly accelerated motion along both the axes $\mathbf{e}_1$ and $\mathbf{e}_2$. Therefore, it is not possible to guarantee that the distance traveled along the $\mathbf{e}_2$ axis before and after the center of the gap are equal while also guaranteeing that the initial and final position have the same coordinate along the $\mathbf{e}_1$ axis. For safety reasons, we prefer to constrain the motion along the $\mathbf{e}_2$ axes, i.e., orthogonally to the gap, such that the distances traveled before and after the gap are equal.

Given the components of the unit vectors $\mathbf{e}_1$ and $\mathbf{e}_2$ in the world frame, it is now possible to compute the initial conditions $\mathbf{p}_0 = \mathbf{p}(t_0)$ and $\mathbf{v}_0 = \mathbf{v}(t_0)$ in 3D space as follows:

$$\mathbf{p}_0 = \mathbf{p}_G - l\mathbf{e}_1 - d\mathbf{e}_2, \tag{3a}$$

$$\mathbf{v}_0 = \left(\frac{l}{t_c} - \frac{1}{2} g_{\Pi,1} t_c \right) \mathbf{e}_1 + \left(\frac{d}{t_c} - \frac{1}{2} g_{\Pi,2} t_c \right) \mathbf{e}_2, \tag{3b}$$

where:

$$t_c = \sqrt{\frac{-2l}{g_{\Pi,1}}} \tag{4}$$

is the time necessary to reach the center of the gap once the traverse trajectory starts.

Note that this solution holds if $g_{\Pi,2} \geq 0$ which applies if $\mathbf{e}_2$ is horizontal or pointing downwards in world coordinates. The case $g_{\Pi,2} < 0$ leads to similar equations, which we omit for brevity. The final three-dimensional trajectory then has the following form:

$$\mathbf{p}(t) = \mathbf{p}_0 + \mathbf{v}_0 t + \frac{1}{2}\mathbf{g}_\Pi t^2, \tag{5a}$$

$$\mathbf{v}(t) = \mathbf{v}_0 + \mathbf{g}_\Pi t, \tag{5b}$$

$$\mathbf{a}(t) = \mathbf{g}_\Pi. \tag{5c}$$

This trajectory is inexpensive to compute since it is solved in closed form. Also, note that during the traverse the gap is no longer detectable. Nevertheless, since the traverse trajectory is short and only requires *constant control inputs* (a thrust of magnitude $\langle \mathbf{g}, \mathbf{e}_3 \rangle$ and zero angular velocities), it is possible to track it accurately enough to not collide with the gap, even without any visual feedback.

B. Optimization of the Traverse Trajectory

To safely pass through the gap, the quadrotor must reach the initial position and velocity of the traverse trajectory described by (3a)-(3b) with an acceleration equal to $\mathbf{g}_\Pi$ at time t_0. An error in these initial conditions is propagated through time according to (5a)-(5c), and therefore may lead to a collision. The only viable way to reduce the risk of impact is to reduce the time duration of the traverse. More specifically, (4) shows that one can optimize the value of l to reduce the time of flight of the traverse trajectory. On the other hand, (3b) and (4) show that reducing l leads to an increase in the norm of the initial velocity $\mathbf{v}_0$. Intuitively speaking, this is due to the fact that, for a given value of d, if the time of flight decreases, the velocity along the $\mathbf{e}_2$ axis has to increase to let the vehicle cover the same distance in a shorter time. The initial velocity also depends on d, which can be tuned to reduce the velocity at the start of the traverse. The value of d cannot be chosen arbitrarily small for two reasons: (i) it is necessary to guarantee a safety margin between the quadrotor and the gap at the beginning of the traverse; (ii) the gap might not be visible during the final part of the approach trajectory. For this reason, we compute the values of the traverse trajectory parameters solving the following optimization problem:

$$\min_{d,l} t_c \quad \text{s.t.} \quad \|\mathbf{v}_0\| \leq \mathrm{v}_{0,\max}, \quad d \geq d_{\min}, \tag{6}$$

where $\mathrm{v}_{0,\max}$ and $d_{\min}$ are the maximum velocity allowed at the start of the traverse and the minimum value of d, respectively. We solve the nonlinear optimization problem described by (6) with Sequential Quadratic Programming (SQP [9], using to the NLopt library [10]. Thanks to the small dimensionality of the problem, it can be solved on-board in few tens of milliseconds.

C. Approach Trajectory

Once the traverse trajectory has been computed, its initial conditions (namely, position, velocity, and acceleration) are known. Now we can compute an approach trajectory from a suitable start position to these initial conditions. Note that this start position is not the current hover position but also results from the proposed trajectory generation method. Our goal in this step is to find a trajectory that not only matches the initial conditions of the traverse trajectory, but also enables robust perception and state estimation with respect to the gap.

Robust state estimation with respect to the gap can only be achieved by always keeping the gap in the field of view of a forward-facing camera onboard the quadrotor. Since it is difficult to incorporate these constraints into the trajectory generation directly, we first compute trajectory candidates and then evaluate their suitability for the given perception task. To do so, we use the approach proposed in [11], where a fast method to generate feasible trajectories for flying robots is presented. In that paper, the authors provide both a closed-form solution for motion primitives that minimize the jerk and a feasibility check on the collective thrust and angular velocities. The benefit of using such a method is twofold. First, it allows us to obtain a wide variety of candidate trajectories within a very short amount of time by uniformly sampling the start position and the execution time within suitable ranges. This way we can quickly evaluate a large set of candidate trajectories and select the best one according to the optimality criterion described in Sec. II-E. Each of these candidate trajectories consists of the quadrotor's 3D position and its derivatives. Second, and most importantly, since the computation and the verification of each trajectory takes on average a two tenths of millisecond, it is possible to replan the approach trajectory at each control step, counteracting the effects of the uncertainty in the pose estimation of the quadrotor when it is far away from the gap. Each new approach trajectory is computed using the last state estimate available. In the following, we describe how we plan a yaw-angle trajectory for each candidate and how we select the best candidate to be executed.

D. Yaw-Angle Planning

In [4], the authors proved that the dynamic model of a quadrotor is *differentially flat*. Among other things, this means that the yaw angle of the quadrotor can be controlled independently of the position and its derivatives. In this section, we present how to compute the yaw angle such that a camera mounted on the quadrotor always faces the gap. Ideally, the camera should be oriented such that the center of the gap is projected as close as possible to the center of the image, which yields the maximum robustness for visual state estimation with respect to the gap against disturbances on the quadrotor.

To compute the desired yaw angle, we first need to compute the ideal orientation of the camera. Let $\mathbf{p}_G$ be the coordinates of the center of the gap with respect to the world frame W. Furthermore, let $\mathbf{R}_{WC}$ and $\mathbf{p}_C$ be the extrinsic

parameters of the camera: $\mathbf{p}_C$ is the camera's position and the rotation matrix $\mathbf{R}_{WC} = (\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3)$ defines the camera orientation with respect to the world frame, where $\mathbf{r}_3$ is the camera's optical axis.

For a given trajectory point, we can compute the vector from the camera to the center of the gap $\mathbf{d} = \mathbf{p}_G - \mathbf{p}_C$. Ideally, we can now align the camera's optical axis $\mathbf{r}_3$ with $\mathbf{d}$ but since the trajectory constrains the quadrotor's vertical axis $\mathbf{z}_b$, we can generally not do this. Therefore, we minimize the angle between $\mathbf{d}$ and $\mathbf{r}_3$ by solving the following constrained optimization problem:

$$\mathbf{r}_3^* = \arg\max_{\mathbf{x}} \langle \mathbf{x}, \mathbf{d} \rangle \quad \text{s.t. } \|\mathbf{x}\| = 1, \quad \langle \mathbf{x}, \mathbf{z}_b \rangle = k, \tag{7}$$

where the last constraint says that the angle between the quadrotor's vertical body axis $\mathbf{z}_b$ and the camera's optical axis is constant and depends on how the camera is mounted on the vehicle. For example, $k = 0$ if the camera is orthogonal to the $\mathbf{z}_b$ axis as it is the case in our setup with a forward-facing camera.

Letting $\mathbf{d}_{\perp z_b} = \mathbf{d} - \langle \mathbf{d}, \mathbf{z}_b \rangle \mathbf{z}_b$ be the component of $\mathbf{d}$ perpendicular to $\mathbf{z}_b$, the solution of (7) is

$$\mathbf{r}_3^* = \sqrt{1 - k^2} \frac{\mathbf{d}_{\perp z_b}}{\|\mathbf{d}_{\perp z_b}\|} + k\mathbf{z}_b, \tag{8}$$

which is a vector lying in the plane spanned by $\mathbf{d}$ and $\mathbf{z}_b$, and the minimum angle between the ideal and the desired optical axis is $\arccos(\langle \mathbf{r}_3^*, \mathbf{d} \rangle / \|\mathbf{d}\|)$, i.e.,

$$\theta_{min} = \arccos\big((\sqrt{1 - k^2} \|\mathbf{d}_{\perp z_b}\| + k \langle \mathbf{d}, \mathbf{z}_b \rangle) / \|\mathbf{d}\| \big). \tag{9}$$

Once $\mathbf{r}_3^*$ is known, we can compute the yaw angle such that the actual camera optical axis $\mathbf{r}_3$ is aligned with $\mathbf{r}_3^*$.

Observe that in the particular case of a trajectory point that allows to align $\mathbf{r}_3$ with $\mathbf{d}$, we have $\langle \mathbf{d}, \mathbf{z}_b \rangle = k\|\mathbf{d}\|$ and the solution of (7) reduces to $\mathbf{r}_3^* = \frac{\mathbf{d}}{\|\mathbf{d}\|}$, with a minimum angle $\theta_{\min} = \arccos(\langle \mathbf{r}_3, \mathbf{d} \rangle / \|\mathbf{d}\|) = \arccos(1) = 0$.

E. Selection of the Approach Trajectory to Execute

In the previous sections, we described how we compute a set of candidate trajectories in 3D space and yaw for approaching the gap. All the candidate trajectories differ in their start position and their execution time. From all the computed candidates, we select the one that provides the most reliable state estimate with respect to the gap. As a quality criterion for this, we define a cost function J composed of two terms:

- the Root Mean Square (RMS) θ_{rms} of (9) over every sample along a candidate trajectory;
- the straight-line distance d_0 to the gap at the start of the approach.

More specifically:

$$J = \frac{\theta_{\text{rms}}}{\bar{\theta}} + \frac{d_0}{\bar{d}}, \tag{10}$$

where $\bar{\theta}$ and $\bar{d}$ are normalization constants that make it possible to sum up quantities with different units, and render the cost function dimensionless. This way, the quadrotor executes the candidate approach trajectory that keeps the center of the gap as close as possible to the center of the image for the entire trajectory, and at the same time prevents the vehicle from starting too far away from the gap.

F. Recovery after the Gap

Since we localize the quadrotor with respect to the gap in order to traverse it, the quadrotor is left with no state estimate after the traversal. Therefore, at this point it has to recover a vision-based state estimate and then hover in a fixed position without colliding with the environment. We solve this problem using the automatic recovery system detailed in [12], where the authors provide a method to let a quadrotor stabilize automatically after an aggressive maneuver, e.g. after a manual throw in the air.

III. State Estimation

A. State Estimation from Gap Detection

Our proposed trajectory generation approach is independent of the gap-detection algorithm being used; thus, to simplify the perception task, we use a black-and-white rectangular pattern to detect the gap (cf. Fig. 1). A valid alternative to cope with real-world gaps would be to use monocular dense-reconstruction methods, such as REMODE [13]; however, they require more computing power (GPUs).

We detect the gap in each image from the forward-facing camera by applying a sequence of steps: first, we run the Canny edge detector, undistort all edges, and group close edges [14]; then, we search for quadrangular shapes and run geometrical consistency checks. Namely, we search for a quadrangle that contains another one and check the area ratio of these two quadrangles. Finally, we refine the locations of the eight corners to sub-pixel accuracy using line intersection.

Since the metric size of the gap is known, we estimate the 6-DOF pose by solving a Perspective-n-Points (PnP) problem (where $n = 8$ in our case). As a verification step, we require that the reprojection error is small. We then refine the pose by minimizing also the reprojection error of all edge pixels. To speed up the computation, we only search the gap in a region of interest around the last detection. Only when no detection is found, the entire image is searched. The detector runs with a frequency of more than 30 Hz onboard the quadrotor.

Finally, we fuse the obtained pose with IMU measurements to provide a full state estimate using the multi-sensor fusion framework of [15].

IV. Experiments

A. Experimental Setup

We tested the proposed framework on a custom-made quadrotor, assembled from off-the-shelf hardware, 3D printed parts, and self-designed electronic components (see Fig. 4). The frame of the vehicle is composed of a 3D printed center cross and four carbon fiber profiles as arms. Actuation is guaranteed by four RCTimer MT2830 motors, controlled by Afro Slim ESC speed controllers. The motors are tilted by 15° to provide three times more yaw-control action, while only losing 3 % of the collective thrust.

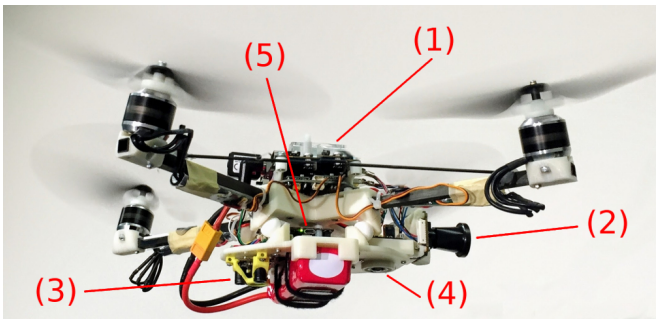


Fig. 4: The quadrotor platform used in the experiments. (1) Onboard computer. (2) Forward-facing fisheye camera. (3) TeraRanger One distance sensor and (4) downward-facing camera, both used *solely during the recovery phase*. (5) PX4 autopilot. The motors are tilted by 15° to provide three times more yaw-control action, while only losing $3\,\%$ of the collective thrust.

Our quadrotor is equipped with a PX4FMU autopilot that contains an IMU and a micro controller on which our custom low-level controller runs. Trajectory planning, state estimation and high-level control run on an Odroid-XU4 single-board computer. Our algorithms have been implemented in ROS, running on Ubuntu 14.04. Communication between the Odroid and the PX4 runs over *UART*.

Gap-detection is done through a forward-facing fisheye camera (MatrixVision mvBlueFOX-MLC200w 752×480-pixel monochrome camera with a 180° lens), which ensures that the gap can be tracked until very close. To allow the robot to execute the recovery maneuver after traversing the gap, we mounted the same hardware detailed in [12], which consists of a TeraRanger One distance sensor and a downward-facing camera. Notice, however, that these are *not used* for state estimation before passing the gap but *only* to recover and switch into stable hovering after the traverse.

The overall weight of the vehicle is $830\,\text{g}$, while its dimension are $55 \times 12\,\text{cm}$ (largest length measured between propeller tips). The dimensions of the rectangular gap are $80 \times 28\,\text{cm}$. When the vehicle is at the center of the gap, the tolerances along the long side and short sides are only $12.5\,\text{cm}$, and $8\,\text{cm}$, respectively (cf. Fig. 5). This highlights that the traverse trajectory must be followed with centimeter accuracy to avoid a collision.

The parameters of the traverse trajectory (Sec. II-B) have been set as $\text{v}_{0,\text{max}} = 3\,\text{m/s}$, $\text{d}_{\text{min}} = 0.25\,\text{cm}$. The normalization constants $\bar{\theta}$ and $\bar{d}$, introduced in Sec. II-E, have been manually tuned to let the quadrotor start the maneuver close enough to render vision-based pose estimation reliable and, at the same time, keep the gap as close as possible to the center of the image.

The dynamic model and the control algorithm used in this work are the same presented in [12]. We refer the reader to that for further details.

Fig. 5: Our quadrotor during a traverse.

B. Results

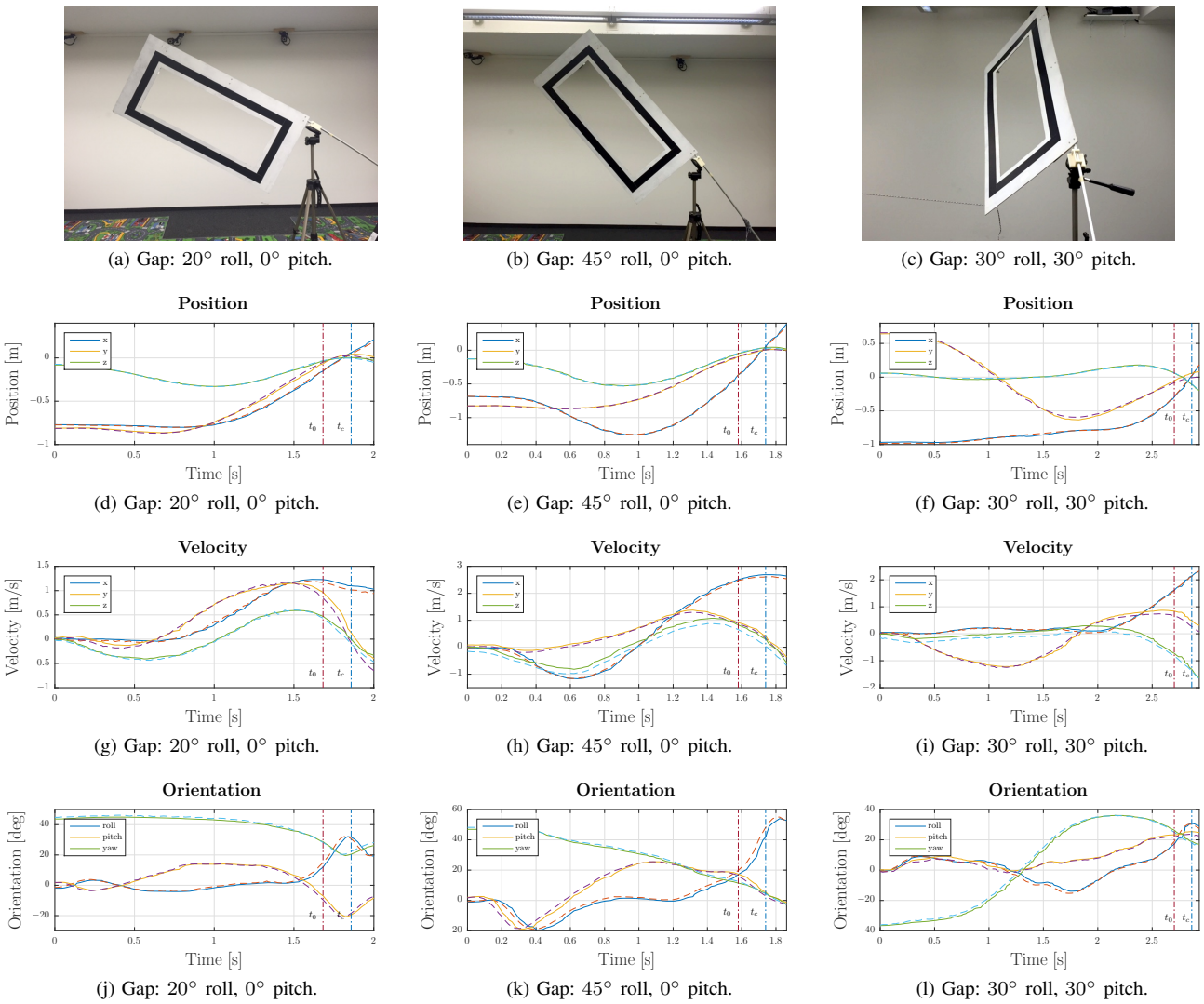
To demonstrate the effectiveness of the proposed method, we flew our quadrotor through a gap inclined at different orientations. We consider both rotations around the world x and y axes, and denote them as *roll* and *pitch*, respectively. Overall, we ran 35 experiments with the roll angle ranging between 0° and 45° and the pitch angle between 0° and 30°. We discuss the choice of these values in Sec. V-C. With the gap inclined at 45°, the quadrotor reaches speeds of $3\,\text{m/s}$ and angular velocities of $400\,^\circ\text{/s}$.

We define an experiment as successful if the quadrotor passes through the gap without collision and recovers and locks to a hover position. We achieved a remarkable success rate of 80%. When failure occurred, we found this to be caused by a persistent absence of a pose estimate from the gap detector during the approach trajectory. This led to a large error in matching the initial conditions of the traverse trajectory, which resulted in a collision with the frame of the gap.

Figure 6 shows the estimated position, velocity, and orientation against ground truth for some of the most significant experiments and for different orientations of the gap (namely: 20° roll, 0° pitch; 45° roll, 0° pitch; and 30° roll, 30° pitch). Ground truth is recorded from an OptiTrack motion-capture system. It can be observed that the desired trajectories were tracked remarkably well. Table I reports the statistics of the errors when the quadrotor passes through the plane in which the gap lies (i.e., at $t = t_c$), measured as the distance between actual and desired state. These statistics include both the successful and the unsuccessful experiments. The average of the norm of the position error at the center of the gap was $0.06\,\text{m}$, with a standard deviation of $0.05\,\text{m}$. The average of the norm of the velocity error was below $0.19\,\text{m/s}$, with a standard deviation of $0.20\,\text{m/s}$. We refer the reader to the attached video for further experiments with different orientations of the gap. Figure 7 shows a picture of one of the experiments with the executed approach and traverse trajectories marked in color.

V. Discussion

In this section, we discuss our approach and provide more insights into our experiments.



(a) Gap: 20° roll, 0° pitch. (b) Gap: 45° roll, 0° pitch. (c) Gap: 30° roll, 30° pitch.

(d) Gap: 20° roll, 0° pitch. (e) Gap: 45° roll, 0° pitch. (f) Gap: 30° roll, 30° pitch.

(g) Gap: 20° roll, 0° pitch. (h) Gap: 45° roll, 0° pitch. (i) Gap: 30° roll, 30° pitch.

(j) Gap: 20° roll, 0° pitch. (k) Gap: 45° roll, 0° pitch. (l) Gap: 30° roll, 30° pitch.

Fig. 6: Comparison between ground truth and estimated position (top), velocity (center), and orientation (bottom). Each column depicts the result of an experiment conducted with a different configuration of the gap: (d), (g) and (j) 20° of roll and 0° of pitch; (e), (h) and (k) 45° of roll and 0° of pitch; (f), (i) and (l) 30° of roll and 30° of pitch. The approach trajectory starts at $t = 0$ and ends at $t = t_0$, when the traverse trajectory is executed. The quadrotor reaches the center of the gap at $t = t_c$ and starts the recovery maneuver at the final time of each plot. We refer the reader to the accompanying video for further experiments with different orientations of the gap.

	Position [m]			**Velocity** [m/s]			**Orientation** [°]	
	x	**y**	**z**	**x**	**y**	**z**	**roll**	**pitch**
μ	0.04	0.04	0.03	0.09	0.15	0.08	6.04	8.89
σ	0.03	0.02	0.03	0.08	0.10	0.06	3.70	5.85

TABLE I: Position, velocity and orientation error statistics at time $t = t_c$. The mean error μ and the standard deviation σ are computed using ground truth data gathered from 35 experiments conducted with the gap at different orientations.

A. Replanning

The method we use to compute the approach meneuver [11] can fail to verify whether a trajectory is feasible or not, as also highlighted by the authors. This usually happens when the time duration of the trajectory is short. In such a case, we skip the replanning and provide the last available approach trajectory to our controller.

B. Trajectory Computation Times

The trajectory planning approach we adopt for the approach phase is fast enough to compute and test $40,000$ trajectories in less then one second, even with the additional computational load induced by our check on the gap perception. The computation of each trajectory on the on-board computer takes on average (0.240 ± 0.106) ms, including: (i) generation of the trajectory; (ii) feasibility check; (iii) trajectory sampling and computation of the yaw angle for

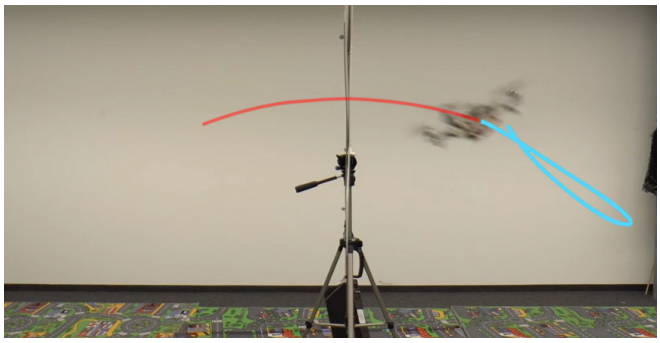


Fig. 7: Our quadrotor executing the whole trajectory split into approach (blue), traverse (red).

each sample; (iv) evaluation of the cost function described in (10); (v) comparison with the current best candidate. It is important to point out that these values do not apply to the *replanning* of the approach trajectory during its execution, since the initial state is constrained by the current state of the vehicle and there is no cost function to evaluate. In such a case, the computation is much faster and for each trajectory it only takes (0.018 ± 0.011) ms on average.

C. Gap configuration

Our trajectory generation formulation is able to provide feasible trajectories with any configuration of the gap, e.g., when the gap is perfectly vertical (90° roll angle) or perfectly horizontal (90° pitch angle). However, in our experiments we limit the roll angle of the gap between 0° and 45° and the pitch angle between 0° and 30°. We do this for two reasons. First, when the gap is heavily pitched, the quadrotor needs more space to reach the initial conditions of the traverse from hover. This renders the gap barely or not visible at the start of the approach, increasing the uncertainty in the pose estimation. Second, extreme configurations, such as roll angles of the gap up to 90°, require high angular velocities in order to let the quadrotor align its orientation with that of the gap. This makes gap detection difficult, if not impossible, due to motion blur. Also, our current experimental setup does not allow us to apply the torques necessary to reach high angular velocities because of the inertia of the platform and motor saturations.

D. Dealing with Missing Gap Detections

The algorithm proposed in Sec. III-A fuses the poses from gap detection with IMU readings to provide the full state estimate during the approach maneuver. In case of motion blur, due to high angular velocities, or when the vehicle is too close to the gap, the gap detection algorithm does not return any pose estimate. However, these situations do not represent an issue during short periods of time (a few tenths of a second). In these cases, the state estimate from the sensor fusion module is still available and reliable through the IMU.

VI. Conclusion

We developed a system that lets a quadrotor vehicle safely pass through a narrow inclined gap using only onboard sensing and computing. Full state estimation is provided by fusing gap detections from a forward-facing onboard camera and an IMU.

To tackle the problems arising from the varying uncertainty from the vision-based state estimation, we coupled perception and control by computing trajectories that facilitate state estimation by always keeping the gap in the image of the onboard camera.

We successfully evaluated and demonstrated the approach in many real-world experiments. To the best of our knowledge, this is the first work that addresses and achieves autonomous, aggressive flight through narrow gaps using only onboard sensing and computing, and without requiring prior knowledge of the pose of the gap. We believe that this is a major step forward autonomous quadrotor flight in complex environments with onboard sensing and computing.

References

[1] M. Müller, S. Lupashin, and R. D'Andrea, "Quadrocopter ball juggling," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2011, pp. 4972–4978.

[2] M. Cutler and J. How, "Analysis and control of a variable-pitch quadrotor for agile flight," *ASME Journal of Dynamic Systems, Measurement and Control*, vol. 137, no. 10, Oct 2015.

[3] D. Mellinger, N. Michael, and V. Kumar, "Trajectory generation and control for precise aggressive maneuvers with quadrotors," in *Int. Symp. Experimental Robotics (ISER)*, Dec 2010.

[4] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2011, pp. 2520–2525.

[5] S. Tang and V. Kumar, "Mixed integer quadratic program trajectory generation for a quadrotor with a cable-suspended payload," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2015, pp. 2216–2222.

[6] M. Neunert, C. de Crousaz, F. Furrer, M. Kamel, F. Farshidian, R. Siegwart, and J. Buchli, "Fast nonlinear model predictive control for unified trajectory optimization and tracking," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2016.

[7] E. Lyu, Y. Lin, W. Liu, and M. Q. H. Meng, "Vision based autonomous gap-flying-through using the micro unmanned aerial vehicle," in *Electrical and Computer Engineering, IEEE Canadian Conference on*, May 2015, pp. 744–749.

[8] 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, April 2017.

[9] D. Kraft, *A Software Package for Sequential Quadratic Programming*, ser. Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt Köln: Forschungsbericht. Wiss. Berichtswesen d. DFVLR, 1988.

[10] S. G. Johnson, "The NLopt nonlinear-optimization package." [Online]. Available: http://ab-initio.mit.edu/nlopt

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

[12] M. Faessler, F. Fontana, C. Forster, and D. Scaramuzza, "Automatic re-initialization and failure recovery for aggressive flight with a monocular vision-based quadrotor," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2015, pp. 1722–1729.

[13] M. Pizzoli, C. Forster, and D. Scaramuzza, "REMODE: Probabilistic, monocular dense reconstruction in real time," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2014, pp. 2609–2616.

[14] S. Suzuki and K. Abe, "Topological structural analysis of digitized binary images by border following," *Computer Vision, Graphics, and Image Processing*, vol. 30, no. 1, pp. 32–46, 1985.

[15] S. Lynen, M. Achtelik, S. Weiss, M. Chli, and R. Siegwart, "A robust and modular multi-sensor fusion approach applied to MAV navigation," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2013.