2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France

Integrated Motion Planner for Real-time Aerial Videography with a Drone in a Dense Environment

Boseong Jeon, Yunwoo Lee and H. Jin Kim

Abstract— **This work suggests an integrated approach for a drone (or multirotor) to perform an autonomous videography task in a 3-D obstacle environment by following a moving object. The proposed system includes 1) a target motion prediction module which can be applied to dense environments and 2) a hierarchical chasing planner. Leveraging covariant optimization, the prediction module estimates the future motion of the target assuming it efforts to avoid the obstacles. The other module, chasing planner, is in a bi-level structure composed of *preplanner* and *smooth planner*. In the first phase, we exploit a graph-search method to plan a *chasing corridor* which incorporates safety and visibility of target. In the subsequent phase, we generate a smooth and dynamically feasible trajectory within the corridor using quadratic programming (QP). We validate our approach with multiple complex scenarios and actual experiments.**

The source code and the experiment video can be found in `https://github.com/icsl-Jeon/traj_gen_vis` and `https://www.youtube.com/watch?v=_JSwXBwYR18`.

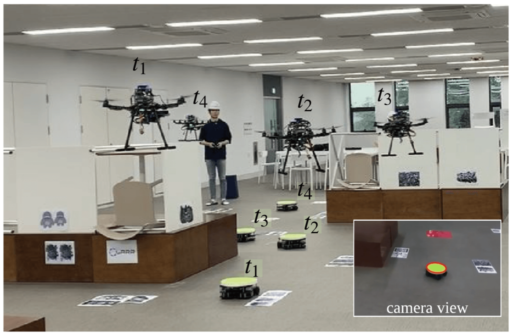


Fig. 1. Autonomous aerial video shooting using a drone for a moving target in plane classroom with cluttering objects. The drone plans a chasing trajectory on-the-fly to incorporate safety and visibility of the target against obstacles. The target (UGV with a green marker) is driven manually by a human operator.

I. Introduction

Video filming has been one of the most popular applications of unmanned aerial vehicles equipped with vision sensors, utilizing their maneuverability and technologies such visual odometry [1] and mapping [2], [3]. Still, the automation of the videographic tasks using drones remains as an open challenge especially in general dense environments.

We propose an online motion strategy developed for more realistic situations where multiple obstacles have arbitrary shapes and the future trajectory of target is not exactly known a priori except the location of sparse via-points which are predetermined for a filming purpose. Here, we assume that the arrival time at each point is also unknown. This setting supposes a scenario where a drone is deployed for shooting a ski game or racing where players are supposed to pass defined spots in a track.

A. Technical challenges

Regarding the autonomous target chasing mission, the followings can be pointed out as main issues, which should be handled jointly.

1) Smooth transition : first of all, the smoothness of flight path of a drone is essential for flight efficiency by avoiding jerky motion, which could cause increased actuation inputs and undesirable shooting quality.

2) Flight safety: the recording agent should be able to maintain its safety against arbitrary shape of obstacles not only simple obstacles (e.g. ellipse or sphere) for the broad applicability.

3) Occlusion against obstacles of general shape: occlusion should be carefully handled in obstacle environments. It could degrade the aesthetic quality of video. Also, a duration of occlusion of a dynamic object might interrupt the autonomous mission due to the failure of visual tracking.

4) Trade-off between global optimality and fast computation: as described in 1)-3), a motion strategy for videography in the considered cases aims to achieve multiple objectives simultaneously. Such a multi-objective problem is subject to local minima and might yield a poor solution if it relies entirely on numerical optimization. On the contrary, if we rely only on sampling or discrete search algorithm such as RRT* [4] and A* [5] to focus on the global optimality at the cost of online computation, a drone might not be able to respond fast enough for the uncertain motion of target on-the-fly. Therefore, trade-off between optimality and fast computation should be taken into account in a balanced manner.

5) Target prediction considering obstacles: for the operation in obstacle environments without exact information of the future movement of a target, reflecting the existence of obstacles is crucial for more reliable prediction.

B. Related works

The previous works [6], [7] and [8] addressed the similar target following problem with consideration of flight

*This material is based upon work supported by the Ministry of Trade, Industry & Energy(MOTIE, Korea) under Industrial Technology Innovation Program. No.10067206, 'Development of Disaster Response Robot System for Lifesaving and Supporting Fire Fighters at Complex Disaster Environment'

Department of mechanical and aerospace engineering, Seoul national university of South Korea `{junbs95,yunwoo333}@gmail.com and hjinkim@snu.ac.kr`

978-1-7281-7395-5/20/$31.00 ©2020 IEEE

efficiency, safety and visibility (A1-A3) under continuous optimization formulation. [6] and [7] developed a receding horizon motion planner to yield a dynamically feasible path in real-time for dynamic situations. Nevertheless, they assumed an ellipsoidal shape for obstacles, which is not applicable to more general cases, having difficulty in fully satisfying A2 and A3. Also, the objective function contains multiple non-convex terms such as trigonometry and product of vectors. This formulation might not be able to produce a satisfactory solution due to local-minima as discussed in A4.

In the work [8], occlusion and collision were handled in a general environment which is represented with octomap. Nevertheless, they relied on numerical optimization for the entire objectives containing complex terms such as integration of signed distance field over a manifold. Such approach might not be able to guarantee the satisfactory optimality, similar to the case of [6], [9] (A4). In [10], the authors addressed the target following with consideration of A1, A2 and A4. The authors designed a hierarchical planner to consider the trade-off between optimality and online computation where a corridor is pre-planned to ensure safety and then a smooth path is generated to minimize the high-order derivatives in the following phase. Especially, [10] performed prediction with a polynomial regression. Still, it did not consider obstacles (A5) and the occlusion of the target was not included in their multi-layer planner, having difficulty in handling A3.

Regarding the target prediction in chasing tasks, [11] included the obstacles for the formulation of prediction to tackle A5. The authors performed Monte-Carlo sampling to estimate the distribution of future target position. In the paper, however, the dynamics and set of possible inputs of target were assumed to be known as a prior, which is difficult to be applied directly to more general cinematic scenarios. Also, the authors restricted the homotopy of the solution path of the robot from the discrete selection of actuation inputs.

To the best of our knowledge, there is few research which effectively handled A1 to A5 *jointly* for drones to be employed in the considered cinematic or chasing scenarios. In this work, we make the following contributions as extension of our previous work [12]:

- An integrated framework for motion strategy is proposed from prediction module to chasing planner, which could achieve the desired performance mentioned A1-A5 in our cinematic problem setting.
- We validate our method by multiple challenging scenarios and a real world experiment. Especially, the tested real platform is implemented to operate fully onboard handling target detection, localization, motion planning and control.

II. Overview

This section manifests the inputs and the objectives of the proposed method. We assume that the filming scene is available in the form of octomap before the mission. It is also assumed that an actor will pass through a set of via-points in sequence for a filming purpose. The viapoints are known a priori while the arrival time for each point is not available. As an additional specification on the behavior of target, it is assumed to move along a trajectory by minimizing high-order derivatives such as acceleration as assumed in [10].

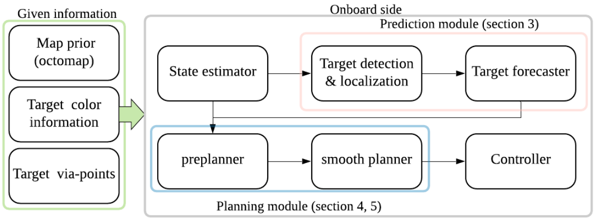


Fig. 2. A diagram for system architecture: we suppose the camera-drone has prior knowledge of the environment and target color for detection. Based on them, we implement a fully onboard system for automatically following a moving target with drone.

Based on these settings, we build a fully-onboard system integrating a target prediction module and a chasing planner for an autonomous target-chasing mission. Regarding the prediction, we aim to reflect on the presence of obstacles while regressing the past observation, which is explained in section III. For the chasing planner, the objective is to achieve the points A1-A4 simultaneously by designing a bi-level which is discussed in section IV and V. The overall pipeline is depicted in Fig. 2.

For the ease of discussion, several notations are defined as follows:

– $x_c \in \mathbb{R}^3$: Position of a chaser (drone).
– $x_p \in \mathbb{R}^3$: Position of a target.
– $L(x_1, x_2) = \{x|\ sx_1 + (1-s)x_2,\ 0 \leq t \leq 1\}$: The line segment connecting $x_1, x_2 \in \mathbb{R}^3$.
– $\chi \subset \mathbb{R}^3$: Configuration space.
– $\chi_{free} = \{x|\ P(x) < \epsilon\}$: Free space in χ, i.e. the set of points where the probability of occupancy $P(x)$ obtained from octomap is small enough.
– $\chi_{obs} = \chi \setminus \chi_{free}$: Space occupied by obstacles.
– $\chi_{vis}(x_p) = \{x|\ L(x, x_p) \cap \chi_{obs} = \emptyset\}$: A set of visible vantage points for a target position v_p.
– $\chi_{occ}(x_p) = \chi \setminus \chi_{vis}$: A set of occluded vantage points for a target position x_p.

III. Target future trajectory prediction

This section describes future motion estimation of the target, utilizing the observation history and prior map information. Here, the terms *path* and *trajectory* are differentiated for clarity as described in [13]. A path refers to a geometric path while a trajectory is a time-parameterized path. The proposed prediction module generates a geometric prediction path first, which will be followed by time estimation for the each point on the path.

A. Target path prediction

As mentioned in section II, we assume that the sequence of via-points of the target is available as $G = \{g_1, g_2, ..., g_M\}$ $(g_i \in \mathbb{R}^3)$ which are supposed to be passed in order and the arrival time for each point is not preset. Also,

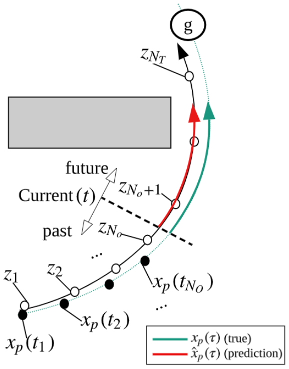


Fig. 3. Prediction of target motion over a horizon $(t, t+H]$. The black curve with white dots denotes the geometric prediction obtained from (1). By assigning the time knot for each point on the path, we predict the trajectory of the target as visualized in the red curve.

let us denote the value of the Euclidean signed distance field (ESDF) at a position $x \in \mathbb{R}^3$ of the prior map as $\phi(x)$.

Now, let us assume that the drone has collected target observation at discrete time steps $t_1, t_2, .., t_{N_o}$, representing $x_p(t_n)$ $(n = 1, .., N_o)$ as $x_{p,n}$. Then, we suppose that the target is heading to $g \in G$ after passing the previous waypoint. For a prediction request time $t > t_{N_o}$ and a future horizon H, we want to forecast the future target trajectory $\hat{x}_p(\tau)$. To obtain $\hat{x}_p(\tau)$, a positional path $\xi = \{z_1, z_2, ..., z_{N_T}\}$ $(z_i \in \mathbb{R}^3$ and $N_T > N_o)$ is generated first to provide a geometric prediction until the target reaches g, assuming the target efforts to avoid obstacles around it. We compute $\hat{x}_p(\tau)$ from the following optimization.

$$\underset{\xi}{min} \quad \underbrace{\frac{1}{2}\sum_{n=1}^{N_o} \exp(\gamma n)\|z_n - x_{p,n}\|^2}_{observation} + \underbrace{\frac{1}{2}\sum_{n=1}^{N_T-2} \|z_n - 2z_{n+1} + z_{n+2}\|^2}_{2nd\ order\ derivative} + \frac{1}{\rho}\underbrace{\sum_{n=1}^{N_T} f_{obs}(z_n)}_{obstacle}, \tag{1}$$

where γ in the first term is a positive constant to give weight to more recent observations. The second term implies that the target will minimize its derivatives the efficiency of its ego motion. The functional f_{obs} is a non-convex cost function inherited from $\phi(x)$ to account for the safe behavior of the target. For the functional, we adopt a similar form of functional introduced in [14].

The above optimization (1) can be arranged into the below, which is the standard form for covariant optimization [14].

$$\underset{\xi}{min} \quad \underbrace{\frac{1}{2}\rho\|A\xi - b\|^2}_{prior\ term} + \underbrace{f_{obs}(\xi)}_{obstacle\ term} \tag{2}$$

(2) is solved with the following covariant update rule where α is a step size.

$$\Delta\xi = -\alpha(A^T A)^{-1}(\rho(A^T A\xi - A^T b) + \nabla f_{obs}(\xi)) \tag{3}$$

From (1) – (3), a geometric path ξ is obtained where $z_{N_o+1}, ..., z_{N_T}$ represents the geometric prediction of the target until g (see fig. 3). We now proceed to complete a trajectory prediction from the discrete path.

B. Time prediction

In this sub-section, we will allocate time knots t_i for each point z_i in ξ with the constant velocity assumption for the simplicity. For the points z_n $(n \le N_o)$ which was used to regress on the past history of target, we simply assign the observation time stamps. For the predicted position z_n $(n > N_o)$, the following recursion is used for allocating times.

$$t_n = t_{n-1} + \frac{\|z_n - z_{n-1}\|}{v_{avg}} \tag{4}$$

where $v_{avg} = \dfrac{\sum_{n=1}^{N_o-1}\|x_{p,n} - x_{p,n+1}\|}{t_{N_o} - t_1}$ represents the average speed over the collected observation. That is, the passing times for the points obtained in (1) are estimated with the constant velocity assumption based on v_{avg}. With this allocation, the future trajectory of the target for a time window $(t, t+H]$ is predicted with the following interpolation:

$$\hat{x}_p(\tau) = \frac{(t_{n+1} - \tau)z_n + (\tau - t_n)z_{n+1}}{t_{n+1} - t_n} \quad (t_n < \tau < t_{n+1}) \tag{5}$$

In our videography setting, single prediction optimization routine runs at tens of Hz, showing its real-time performance. This helps us to re-trigger prediction when the estimation error exceeds a threshold. From the formulation including the cost f_{obs}, we were able to incorporate obstacles, which is the main objective described in A5.

IV. Preplanning for Chasing Corridor

This section introduces a method for generating a *chasing corridor* in order to provide the boundary regions for the chasing drone's trajectory. We first explain a metric to encode safety and visibility of the chaser's position, which are utilized as objectives in computing the corridor.

A. Metric for safety and visibility

For the safety metric, we reuse ESDF $\phi(x)$ as it can measure the risk of collision with nearby obstacles (see Fig. 4-(a)). Now, the visibility metric is introduced so that we can encode how robustly the drone can maintain its sight against 1) occluding obstacles and 2) unexpected motion of the target. For a target position x_p when seen from a chaser position x_c with line of sight (LOS) $L(x_p, x_c)$, we define the below as visibility:

$$\psi(x_c; x_p) = \min_{L(x_c, x_p)} \phi(x). \tag{6}$$

In the actual implementation, (6) is calculated over the grid field. That is, we can evaluate (6) with a simple min operation while traversing the voxels along $L(x, x_p)$ with linear time complexity. Because (6) means the minimum distance between obstacles and the LOS connecting the object and the drone (see Fig. 4 - (a)), a small value of which implies that the target could be lost more easily than the higher value as illustrated in fig. Fig. 4 - (b).

The proposed metric possesses multiple advantages. First, it can be directly computed from reusing ESDF which was utilized for the target prediction and safety metric for drone, without further calculation. Second, it can be defined without restriction of shape of obstacles in contrast to the research such as [6], [7] and [8].

B. Corridor generation

Based on the proposed metric for safety and visibility, computation of the sequence of corridors for chasing is explained given the prediction $\hat{x}_p(\tau)$ $(t \le \tau \le t+H)$ and the current position of the drone as $x_c(t)$. We first discretize the planning window $[t, t+H]$ into $(t_0, t_1, t_2, ..., t_N)$ writing $\hat{x}_{p,n} = \hat{x}_p(t_n)$ and $x_{c,0} = x_c(t_0)$. Now, we focus on the computation of a sequence of viewpoints $\sigma = \{v_0, v_1, v_2, ..., v_N\}$ where $v_n \in \mathbb{R}^3$, as a skeleton for the corridors. Rather than searching v_n in a continuous domain, we choose it from a finite set $V_n \subset \{x \mid d_l \le \|x - \hat{x}_{p,n}\| \le d_u,\ \ x \in \chi_{vis}(\hat{x}_{p,n})\}$ where x is a discrete point in the operating grid (e.g. grid from octomap) in case of $n > 0$. d_l and d_u are the minimum and maximum distance of tracking. The discrete path σ is obtained from the following optimization.

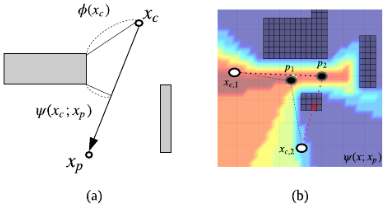
$$
\begin{aligned}
\underset{\sigma}{\text{argmin}} \quad & \sum_{n=1}^{N} c(v_{n-1}, v_n) \\
\text{subject to} \quad & v_0 = x_{c,0} \\
& \min_{x \in L(v_{n-1}, v_n)} \phi(x) \ge r_{safe} \\
& v_n \in V_n \\
& \|v_{n-1} - v_n\| \le d_{max} \\
\text{where} \quad & c(v_{n-1}, v_n) = \underbrace{\|v_{n-1} - v_n\|^2}_{\text{interval distance}} + \underbrace{w_v\ c_v(v_{n-1}, v_n)}_{\text{visibility}} \\
& + w_d \underbrace{(\|\hat{x}_{p,n} - v_n\| - d_{des})^2}_{\text{tracking distance}}
\end{aligned} \tag{7}
$$

The objective function in (7) penalizes the interval distance between each point while rewarding the high score of visibility. The second term is defined by

$$
c_v(v_{n-1}, v_n) = \left(\sqrt{\int_{L(v_{n-1}, v_n)} \psi(x; \hat{x}_{p,n-1}) dx \int_{L(v_{n-1}, v_n)} \psi(x; \hat{x}_{p,n}) dx} \right)^{-1} . \tag{8}
$$

where the integral is computed as an accumulative sum in the grid field.

The last term in $c(v_{n-1}, v_n)$ of (7) aims to keep the relative distance as d_{des}. w_v and w_d are the weights for visibility and relative distance respectively. Among the constraints, the second one enforces a safe clearance r_{safe} of each line $L(v_{n-1}, v_n)$ and the third constraint means that v_n should be a visible viewpoint for the predicted target at $\hat{x}_{p,k}$. The last constraint bounds the maximally connectable distance between two points v_n, v_{n+1} to prevent the excessive velocity of the drone. Due to the page limit, we refer the readers to our previous work [12] where the optimal sequence of v_n is computed from the graph search algorithm.



(a) (b)

Fig. 4. (a): Safety metric $\phi(x_c)$ and visibility metric $\psi(x_c; x_p)$ for a target position x_p and drone x_c. (b): Visibility field for x_p in colormap. Red-coloured field denotes higher visibility and the occluded region χ_{occ} is illustrated with the same uniform color (dark blue). As an illustrative example, we consider the case where the object moves $p_1 \to p_2$ for a short time. Both positions $x_{c,1}$ and $x_{c,2}$ were able to see p_1. While the camera-drone at $x_{c,1}$ can still observe the target which moved to p_2, the vantage at $x_{c,2}$ fails to maintain the visibility.

From optimal viewpoint σ computed from (7), we now generate a set of corridors connecting two consecutive viewpoints in σ as visualized in Fig 5-(a). Once the width of corridor $r_c < r_{safe}$ is chosen, we can write the box region connecting v_{n-1} and v_n as a linear inequality $A_n x \le b_n$. Due to the formulation of (7), every point in $A_n x \le b_n$ maintains a safe margin $r_c < r_{safe}$ guaranteeing the strict safety for the trajectories bounded in the region. Also, the predicted point $\hat{x}_{p,n}$ can be observed from $v_n \in V_n$ without occlusion by obstacles as all the elements of V_n belongs to $\chi_{vis}(\hat{x}_{p,n})$. For a large value of w_v and small enough d_{max}, we empirically found that every point in the corridor $A_n x \le b_n$ can maintain visibility for the prediction $\hat{x}_p(\tau)$ for $\tau \in [t_{n-1}, t_n]$.

These two properties of the chasing corridor provides a base to achieve A2 and A3 to a continuous trajectory within it, which is discovered in the following section. We also highlight that the optimal solution of the multi-objectives non-convex optimization of (7) was computed using graph-search rather than a numerical optimization. The approach can ensure the quality of solutions, which was mentioned in A4. In the remaining process, we will rapidly generate a continuous trajectory receiving the results of preplanning: viewpoints and the chasing corridors.

V. Dynamically feasible trajectory generation

In this section, we generate a dynamically feasible trajectory for position and yaw $[x_c(\tau)^T,\ r(\tau)]^T \in \mathbb{R}^4$ using v_k and $A_k x \le b_k$. The position trajectory $x_c(\tau)$ is represented

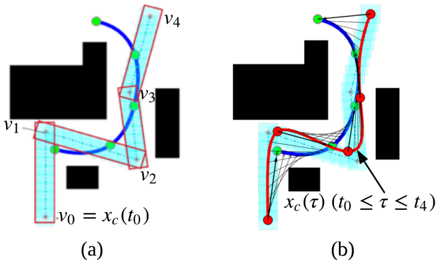


Fig. 5. (a): Illustrative example of the chasing corridor with the skeleton viewpoints v_n $(1 \le n \le 4)$. The blue line denotes target prediction. The red edged boxes denote corridors between v_n and v_{n+1}. (b): Smooth path (red) is generated within the corridors from (a). LOS stamps are also drawn with black arrows.

with piece-wise polynomials as below:

$$x_c(\tau) = \begin{cases} \sum_{k=0}^{K} p_{1,k}\tau^k & (t_0 \le \tau < t_1) \\ \sum_{k=0}^{K} p_{2,k}\tau^k & (t_1 \le \tau < t_2) \\ \dots & \\ \sum_{k=0}^{K} p_{N,k}\tau^k & (t_{N-1} \le \tau < t_N) \end{cases} \tag{9}$$

where $p_{n,k} \in \mathbb{R}^3$ $(n = 1, ..., N, \;\; k = 1, ..., K)$ is the coefficients of the polynomials and K denotes the order of it. Polynomial coefficients of the chaser's trajectory are computed from the optimization below (10). The yaw $y(\tau)$ is chosen so that $x_c(\tau)$ heads toward $x_p(\tau)$ at each time step if observation of the target at τ is acquired.

$$\begin{aligned} \min \quad & \int_{t_0}^{t_N} \left\| {x_c}^{(3)}(\tau) \right\|^2 d\tau \;+\; \lambda \sum_{n=1}^{N} \| x_c(t_n) - v_n \|^2 \\ \text{subject to} \quad & x_c(t_0) = x_0 \\ & \dot{x}_c(t_0) = \dot{x}_0 \\ & \ddot{x}_c(t_0) = \ddot{x}_0 \\ & A_n x_c(\tau) \le b_n \; (t_{n-1} < \tau < t_n, \; n = 1, ..., N). \end{aligned} \tag{10}$$

Our optimization setup (10) tries to minimize the magnitude of jerk along the trajectory and the deviation of $x_c(t_k)$ from viewpoints v_k where λ is an importance weight. In the constraints, x_0, $\dot{x}_0$ and $\ddot{x}_0$ are the states of the drone when the planning was triggered and used as the initial conditions. Additionally, we enforce continuity conditions on the knots t_n. The last constraint restricts smooth path to be generated within the chasing corridors to ensure safety and visibility.

The formulation in (10) can be converted into the quadratic programming (QP) with respect to the coefficients of the polynomials in a similar manner with [10], [15], which can be solved efficiently with the algorithm such interior point [16]. As investigated in [15], $x_c(\tau)$ can be executed by the virtue of differential flatness of quadrotors. From the formulations (10), we captured the objective A1 by generating a trajectory with reduced high-order derivatives. Also, the QP framework yields the optimal solution in the order of milliseconds in our implementation, which also satisfies one of objectives of A4.

Combining the smooth trajectory generation with the preplanning process where chasing corridors and viewpoints were obtained, a receding horizon planner for target following is constructed as follows: during mission, we first predict target future trajectory for a time window based on recent observations and the current cinematographic via-point g by solving (1). Based on the prediction, we compute the viewpoints σ and the corresponding chasing corridor, which is followed by the continuous trajectory generation proposed in this section. If observation becomes unreliable where the accumulated estimation error exceeds a defined threshold, prediction is re-triggered. This loop continues until the end of the videographic mission.

VI. Validation results

A. Simulations

We validate the proposed algorithm in a dense environment with four target trajectories. For the simulation, we used *complex city* (see our previous work [12] for the 3D models) where five target via-points (from left to right) are defined as blue circles in Fig. 6. *Complex city* includes multiple non-convex obstacles, and the target was operated to hide behind the obstacles at the moment denoted as green boxes (see Fig. 6). In the simulation, we used *rotors simulator* [17] for the chaser drone, and the target (turtlebot) was manually operated with a keyboard. A vision sensor is fixed on the drone (13°pitching down). The simulations were performed using gazebo in a laptop having Intel i7 CPU and 16GB RAM. Given target trajectories, chasing strategies with two different levels of visibility weights w_v were tested, totalling 8 simulations.

Other than the visibility weight w_v, the other parameters were set to the same values for all tests. We used the receding horizon period with $H = 4s$ and discretization $N = 4$. The results are summarized in Fig. 6 (A)-(D) and Table I. For each target scenario, the history of $\phi(x_p)$ is plotted in the bottom row in Fig. 6 where a small value of $\phi(x_p)$ implies difficulty for securing visibility due to the proximity to obstacles. As one can observe, the planner with high visibility $w_v = 5.0$ tries to attain more visibility score $\psi(x_c; x_p)$ resulting in the less occlusion duration compared to planner with $w_v = 1.0$, at the cost of longer detours. In the all simulations, the safety of drone chaser was strictly satisfied during entire mission. The average computation times were 0.02s, 0.15s and 0.03s for prediction, preplanning and smooth trajectory generation respectively. We observed that the entire pipeline of the receding horizon planner ran at 5-6Hz, showing the capability pointed in A4 by re-planing fast enough.

B. Real world experiment

We tested the proposed method in an actual experiment in an indoor classroom without GPS. In the test, the drone was equipped with ZED (stereo vision sensor) and pixhawk2 auto-pilot for flight control. The hardware of the drone chaser can be found in Fig. 7. The vision-related algorithm ran on Jetson TX2 while planning and control was processed in the

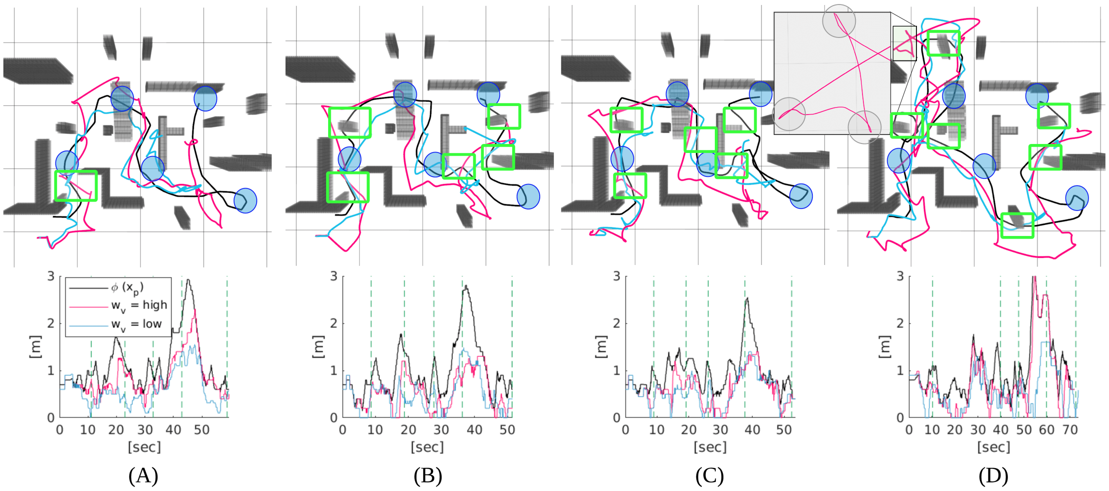


Fig. 6. Flight results of the drone from the receding horizon planner given four different trajectories of the target. **Top:** the target's history is denoted as a black line and the chaser's history is depicted with skyblue for low visibility, and magenta for high visibility respectively. The size of the grid in the figure is 4m. From (A) to (D), the target moves faster and it is more unpredictable due to its hiding behind obstacles, which increases difficulty for chasing. For all the simulations, the target passes through five via-points one-by-one (blue circles). The green boxes denote the locations where the target was intentionally operated to hide behind obstacles with an abrupt maneuver. To confirm the smoothness of the flight trajectory, a zoom-in image is provided in (D). **Bottom:** History of distance field value of the target (black) and visibility score for a small w_v $(= 1.0)$ and a large w_v $(= 5.0)$ are plotted in magenta and blue respectively. The dotted vertical line (green) denotes the the target's arrival time to each via-point.

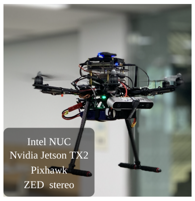


Fig. 7. The camera-drone for the fully-onboard implementation of the proposed algorithm.

	A		B		C		D	
target speed [m/s]	0.36		0.57		0.67		0.75	
w_v	1.0	5.0	1.0	5.0	1.0	5.0	1.0	5.0
avg. $\psi(x_c; x_p)$[m]	0.6084	0.8433	0.4817	0.5330	0.5543	0.6051	0.5566	0.7791
occ. duration [sec]	0.099	0	7.59	4.323	4.29	2.145	9.768	6.435
flight dist.[m]	40.7761	49.9106	34.8218	47.5424	36.3040	50.6000	55.7393	77.2377

TABLE I

SIMULATION RESULT

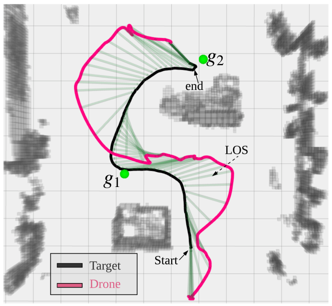


Fig. 8. The experimental result where a fully onboard camera drone is chasing a ground robot in an indoor environment depicted in Fig. 1. The two viapoints g_1 and g_2 were predetermined. The drone takes two detours to maintain the visibility toward the target when the obstacles are expected.

onboard computer (NUC7i7BNH). The target is a ground robot with green color (see Fig. 1). The target was operated manually by a human operator with linear velocity of 0.2-0.3 m/s. The experimental environment and the paths taken by the target and chaser are shown in Fig. 8. We can observe that the safety and visibility were satisfied for the entire duration. The while pipeline ran at 10Hz for the future horizon of 4s.

VII. CONCLUSION AND FUTURE WORKS

In this work, we proposed a chasing planner to handle safety and occlusion against obstacles. The preplanning phase provides a chasing corridor where the objectives such as visibility, safety and travel distance (A1-A3) are incorporated, yielding the high-quality sketch for the path in the grid space. Receiving the results of preplanner, a dynamically feasible path is rapidly computed leveraging the QP framework in smooth planning step (A1). The bi-level structure could handle the tradeoff commented in A4. We also proposed a prediction module which allows the camera-drone to forecast the future motion during a time horizon, which can be applied in obstacle cases (A5). The whole pipeline was validated in various simulation scenarios, satisfying all the objectives A1-A5. Additionally, we explored the effect of visibility weights to the two conflicting objectives: travel distance and visibility. We also implemented a real drone which operates fully onboard to perform the autonomous videography. In the future, we will extend the proposed algorithm for the multi-target chasing scenario. Also, we plan to enhance the algorithm for the case of unknown map where the drone has to explore to gather information to generate more efficient trajectory.

References

[1] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[2] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: An efficient probabilistic 3d mapping framework based on octrees," *Autonomous robots*, vol. 34, no. 3, pp. 189–206, 2013.

[3] H. Oleynikova, Z. Taylor, R. Siegwart, and J. Nieto, "Safe local exploration for replanning in cluttered unknown environments for microaerial vehicles," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1474–1481, 2018.

[4] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, 2011.

[5] F. Duchoň, A. Babinec, M. Kajan, P. Beňo, M. Florek, T. Fico, and L. Jurišica, "Path planning with modified a star algorithm for a mobile robot," *Procedia Engineering*, vol. 96, pp. 59–69, 2014.

[6] T. Nägeli, J. Alonso-Mora, A. Domahidi, D. Rus, and O. Hilliges, "Real-time motion planning for aerial videography with dynamic obstacle avoidance and viewpoint optimization," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1696–1703, 2017.

[7] B. Penin, P. R. Giordano, and F. Chaumette, "Vision-based reactive planning for aggressive target tracking while avoiding collisions and occlusions," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3725–3732, 2018.

[8] R. Bonatti, Y. Zhang, S. Choudhury, W. Wang, and S. Scherer, "Autonomous drone cinematographer: Using artistic principles to create smooth, safe, occlusion-free trajectories for aerial filming," *arXiv preprint arXiv:1808.09563*, 2018.

[9] 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)*, pp. 6199–6206, IEEE, 2017.

[10] J. Chen, T. Liu, and S. Shen, "Tracking a moving target in cluttered environments using a quadrotor," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 446–453, IEEE, 2016.

[11] P. Švec, A. Thakur, E. Raboin, B. C. Shah, and S. K. Gupta, "Target following with motion prediction for unmanned surface vehicle operating in cluttered environments," *Autonomous Robots*, vol. 36, no. 4, pp. 383–405, 2014.

[12] B. F. Jeon and H. J. Kim, "Online trajectory generation of a mav for chasing a moving target in 3d dense environments," *arXiv preprint arXiv:1904.03421*, 2019.

[13] A. Gasparetto, P. Boscariol, A. Lanzutti, and R. Vidoni, "Path planning and trajectory planning algorithms: A general overview," in *Motion and operation planning of robotic systems*, pp. 3–27, Springer, 2015.

[14] N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa, "Chomp: Gradient optimization techniques for efficient motion planning," 2009.

[15] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *2011 IEEE International Conference on Robotics and Automation*, pp. 2520–2525, IEEE, 2011.

[16] S. Mehrotra, "On the implementation of a primal-dual interior point method," *SIAM Journal on optimization*, vol. 2, no. 4, pp. 575–601, 1992.

[17] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart, "Rotors—a modular gazebo mav simulator framework," in *Robot Operating System (ROS)*, pp. 595–625, Springer, 2016.