SCIEPublish

Article

A Lightweight Visual Navigation and Control Approach to the 2022 RoboMaster Intelligent UAV Championship

Sijie Yang, Wenqi Song, Runxiao Liu and Quan Quan *

Reliable Flight Control Group, Beihang University, Beijing 100191, China

* Corresponding author. E-mail: qq_buaa@buaa.edu.cn (Q.Q.)

Received: 28 September 2023; Accepted: 11 January 2024; Available online: 26 January 2024

ABSTRACT: In this paper, an autonomous system is developed for drone racing. On account of their vast consumption of computing resources, the methods for visual navigation commonly employed are discarded, such as visual-inertial odometry (VIO) or simultaneous localization and mapping (SLAM). A series of navigation algorithms for autonomous drone racing, which can operate without the aid of the information on the external position, are proposed: one for lightweight gate detection, achieving gates detection with a frequency of 60 Hz; one for direct collision detection, seeking the maximum passability in-depth images. Besides, a velocity planner is adopted to generate velocity commands according to the results from visual navigation, which are enabled to perform a guidance role when the drone is approaching and passing through gates, assisting it in avoiding obstacles and searching for temporarily invisible gates. The approach proposed above has been demonstrated to successfully help our drone passing-through complex environments with a maximum speed of 2.5 m/s and ranked first at the 2022 RoboMaster Intelligent UAV Championship.

Keywords: MAV; Drone racing; Autonomous drone

1. Introduction

Recently, drones become a ubiquitous modern tool, which are widely used in search, rescue, surveillance, aerial photography or drone racing [1]. Benefitting from their mechanical simplicity, drones are one of the most agile aerial robots due to agile dynamics that professional human pilots can complete the competition for reaching multiple waypoints or inspection tasks at an astonishing speed with only image feedback from FPV glasses. Besides, drones can complete tasks independently of external systems, namely only with the aid of onboard sensors and computing resources. Whether or not drones can complete the task as ordinary people do under the same conditions and even exceed human pilots remains to be discussed. Consequently, the integration of navigation in complex environments and autonomous drone racing raises a great number of challenges in state estimation, perception, planning, and control. Although the individual problem could be easily solved in a desktop platform with rich computational resources and sensors, autonomous drone racing features limited resources, restricted by battery capacity, computational power, sensor weight, etc.

There are many competitions about autonomous drones around the world with various focuses. Autonomous drone racing was inaugurated in IROS 2016 [2] and continued in IROS 2017 [3], focusing on navigation in complex environments. The AlphaPilot Challenge [4] focuses on corner cases of visual navigation and pushes the limits in terms of speed. A competition is sponsored by DJI in 2022, named the RoboMaster Intelligent UAV Championship, which aims to spur technological innovation in the navigation and control of autonomous drones. In detail, the arena for autonomous drone racing is shown in Figure 1. A drone is expected to pass through all gates sequentially at the fastest speed and avoid obstacles at the same time and only be permitted to use onboard resources, which means all the calculations are supposed to be accomplished on the airborne platform. Gate No. 14 and No. 15 are square holes in the wall. Gate No. 8 is moving at a uniform speed reciprocated motion. More than 200 teams participate in the online simulation drone racing, and the top five teams are invited to compete in the 2022 RoboMaster Intelligent UAV Championship (https://www.robomaster.com/zh-CN/robo/drone?djifrom=nav_drone accessed on 15 September 2023), with each of them granted three opportunities during the competition.

https://doi.org/10.35534/dav.2024.10002

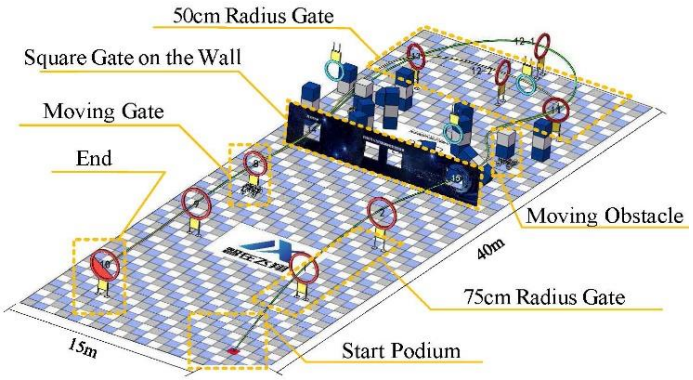


Figure 1. A schematic representation of the arena for autonomous drone racing.

This competition differs from others in three aspects: (1) the competitive environment is completely unknown and dense, with objects placed randomly; (2) moving gates and obstacles simultaneously exist in the racing course, which puts forward a significantly higher demand for autonomous drones; (3) the gates are in different sizes and shapes, with ring-shaped ones for radii of 50 cm, 60 cm and 75 cm and the square ones with a side length of 90 cm in the wall.

There are many solutions for flying a drone via multiple waypoints in a minimum amount of time. State-of-the-art approaches [5,6] for time-optimal, multi-waypoint flight split the problem into a planning task and a control task. The planning task generates a global, time-optimal trajectory, which requires definite waypoints. The control task tracks the generated trajectory, which is only available in indoor environments with the precision position provided by the motion caption system (MCS). In general, the most common methods to provide position information in GPS-denied environments include visual-inertial odometry (VIO) [7,8], lidar-inertial odometry (LIO) [9,10], simultaneous localization and mapping (SLAM) [11], etc. Despite excellent performance in general navigation, the lidar is not an ideal sensor for drones owing to its excessive weight, size, power-consuming, which enormously restricts the flight range and flight time.

As a rich sensor, vision can provide an assortment of information. The capabilities of VIO are based on feature detection and tracking. While the drone is at high translational and rotational velocity, detecting and tracking features will be difficult, and thus cause substantial drift in state estimate [12]. Several approaches have been developed to overcome this difficulty, for instance, utilizing some structures in drone racing as landmarks. In [13,14], gate information is extracted by the "snake gate detection" method from images [15], which fuses with an attitude estimator and guides the drone toward the gate. The approach is lightweight but struggles with scenarios in which multiple gates are visible. In [16], the convolutional neural network (CNN) is used to retrieve a bounding box of the gate. Although CNN methods stand out when the gates are visible, they fail to apply to invisible gates.

Owing to the method of location for drones, which exploits obstacles and gates as landmarks, is inapplicable to the situation where external objects are randomly placed. In this case, this paper puts forward an effective one for autonomous flight that can be implemented without the aid of the information on self-positioning.

Specifically speaking, controls on velocity and yaw rate are put in place to ensure the smooth and reliable flight of drones. The velocity and yaw rate response are assumed fast enough so the delay time of actuation can be ignored. A unified velocity planner is designed to generate velocity commands for various tasks, which can be categorized into three phases: exploring, approaching, and passing through. In the exploring phase, the drone moves forward until the gates can be observed, which is followed by the beginning of the approaching phase. During the approaching phase, the information about those gates is extracted via the conventional method of image processing with lightweight computation. The velocity commands are calculated by a visual servo method. In particular, the detection approach functions efficiently for the situation with multiple gates, even just a small fraction of which is visible.

Owing to unreliable VIO estimated position and huge computing resource consumption, the position feedback method and position control are discarded. IMU and visible landmarks are used for velocity estimation with only using RGB-D images to complete the whole task.

The contributions of this paper herein are: (1) extending the method in [15] to detect ring-shaped gates; (2) completing the task according to images to eliminate the dependence on the information of the drone's position; (3) employing a direct method for collision avoidance which generates a collision-free direction rather than a collision-free trajectory; (4) designing a velocity planner inspired by [17] to generate velocity commands, which guides the drone to approach and pass through the gate and avoid obstacles to ensure collision-free flight.

2. System Overview

To meet the requirements of the RoboMaster Intelligent UAV Championship, an autonomous drone is designed as one with a diameter of approximately 0.35 m, a weight of 1.3 kg, and a thrust-to-weight ratio of around 2.7, as shown in Figure 2. There are three coordinate systems used in this paper [18]: $o_b - x_b y_b z_b$ is the body frame {b} in red, $o_c - x_c y_c z_c$ is the camera frame {c} in green, $o_e - x_e y_e z_e$ is the earth frame {e} in blue. The drone is equipped with Jetson NX as the computing unit to handle all computation onboard. A forward-looking D435i camera is mounted on the drone to get RGBD images for perception, with a CUAV V5 nano Autopilot for compact size as the flight controller.

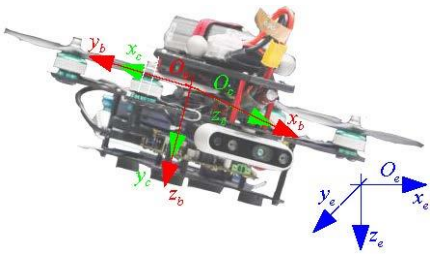


Figure 2. The autonomous platform used for real-world flight. The red arrows depictured body frame {b}; the green arrows depictured the camera frame {c} and the blur arrows depictured the earth frame {e}.

Additionally, three coordinate frames are introduced. The earth frame {e} is a world-fixed frame, with its x-axis, y-axis and z-axis pointing to the rectangle arena's long side, short side and downwards, respectively. The body frame {b} and a forward-looking camera frame {c} are attached to the drone's center of mass. The relationship among the frames {b}, {e} and {c} are shown in Figure 2. The conversion from frame {b} to {e} is $\mathbf{R}_b^e$. The flight controller controls the total thrust f along the z-axis of body coordinate frame {b} and the angular velocity ${}^b\omega$ also in {b}.

The software architecture is shown in Figure 3. The sensor and pre-knowledge part provide the current information and vague information about the environment already known. Then, information is extracted via the navigation algorithms. The velocity planner generates velocity commands to pass through gates to avoid collision and explore invisible gates. The IMU on PX4 is used to estimate velocity via an EKF estimator. The velocity-controlling scheme is implemented with proportional attitude control and throttle control.

The details of how to solve middleware from vision are shown in Section 3, which focuses on image processing, corresponding to the perception part in Figure 3. And the velocity planner is designed by converting the middleware to velocity command in Section 4.

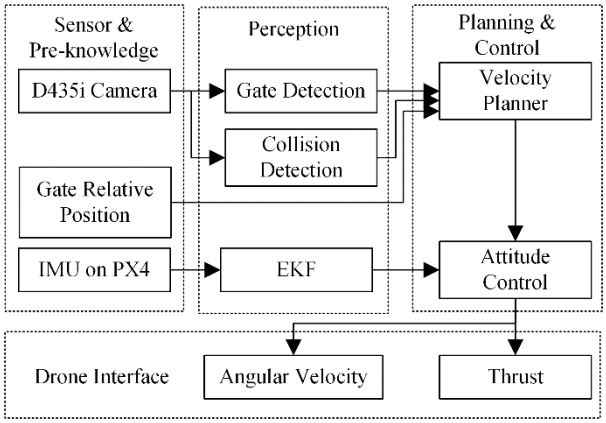


Figure 3. Overview of the system architecture.

3. Perception

3.1. Gate Detection

The method for the detection of multiple gates comes up with the inspiration in [15]. Different from the Hoff-circle detection method and the machine learning method, it even performs excellently in the circumstance where the gate is too large to be presented entirely in the image.

Gates are made of wood with red paint on the surface. The proposed ring-shaped gate target detection process is shown in Figure 4. In the first place, the image I is converted from the RGB color space to the HSV color space. Next, a binarization image is extracted through color filtering. Subsequently, morphological processing is employed to eliminate some noises, and divides image I into several sub-images I_i, which probably contains ring-shaped targets. With connected components $C_i = \{(u,v) \mid (u,v) \in I_i\}$, where c_{xi} and c_{yi} are the center of sub-image I_i, 20% data of I_i is used to check whether the circle is complete. The center of I_i is considered to be the center of the circle if the circle is complete. Otherwise, the circle fitting algorithm is performed on the 20% data of I_i by least squares. Then the center c_x, c_y and the radius r are figured out.

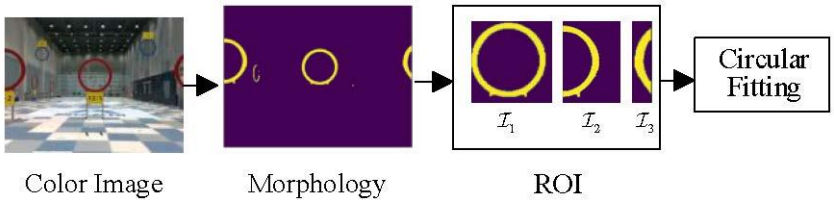


Figure 4. Flow chart of ring target detection.

The radius r can be used to calculate the depth from the drone to the circle plane when the depth image is unavailable. As a matter of fact, according to the pinhole imaging model, the relationship between pixel radius r and depth s is

$$\frac{R}{s+f} = \frac{r}{f} \tag{1}$$

where R is the real radius in meter, f is the focus of the camera. Provided that the minimum s_i from I_i is selected as the target gate, the relative position of the drone to the target gate ${}^{c}\mathbf{P}_t = [{}^{c}X_t \; {}^{c}Y_t \; {}^{c}Z_t]^{\mathrm{T}}$ is obtained through the camera-calibrated parameter matrix $\mathbf{K}$ with

$$s\mathbf{p} = \mathbf{K} \cdot {}^{c}\mathbf{P}_t \tag{2}$$

where $\mathbf{p} = [c_x \; c_y]^{\mathrm{T}}$ is the pixel center of target. Moreover, if the depth image is available, depth s is directly employed to figure out the result of Equation (2). In Section 4, more details are elaborated on how to design velocity commands to approach and passing-through static and moving gates using ${}^{c}\mathbf{P}_t$.

3.2. Collision Detection

The risks of collision in the autonomous racing process predominantly lie in obstacles. When the drone passes through a gate, the visual servo method can guarantee the flight is collision-free. Under the inspiration from the velocity obstacle method in [19], a new one for direct obstacle detection from the FPV view is put forward. When some obstacles occupy the center of the camera, a collision occurs inevitably.

Accordingly, the predominant task is to find a safe direction for drones. According to the FPV view, the field of view is divided into nine parts, with the center part shown in Figure 5 labeled the color green. The safety radius safety is r_a, and the safety depth s_a could be calculated via:

$$s_a = \frac{r_a}{2\tan\left(\theta_f / 6\right)} \tag{3}$$

where θ_f is the field of view. If all depth values in area D_5 surpasses s_a, the drone is collision-free for moving forward. Otherwise, collision might occur with θ_{cp} as

$$\theta_{cp} = \begin{cases} 1 & \text{if } \forall s_i \in D_5 : s_i > s_a \\ 0 & \text{if } \exists s_i \in D_5 : s_i \le s_a \end{cases} \tag{4}$$

where D_5 is the center area shown in Figure 5b.

Inputting a depth image D with size $w \times h$, the foreground and background are segmented by OTSU method [20] in area D_4, D_5, D_6. Then the direction for collision avoidance $\mathbf{p}_{ca}$ is calculated from background by finding the maximum area passable, as shown in Figure 6. The foreground and background are annotated with gray and pink respectively. The value of $\mathbf{p}_{ca}$ can be figured out via

$$\mathbf{p}_{ca} = \left[\frac{w}{2} - c_{xd} \quad \frac{h}{2}\right]^{\mathrm{T}} \tag{5}$$

where c_{xd} is the center of the largest area passable, a red point in Figure 6(e).

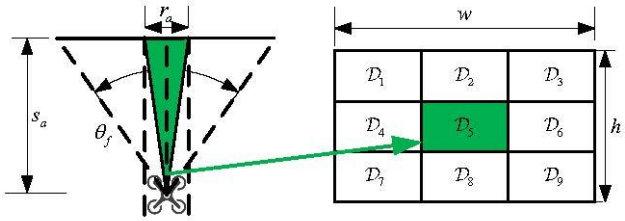


Figure 5. Camera field of view model and depth image partition.

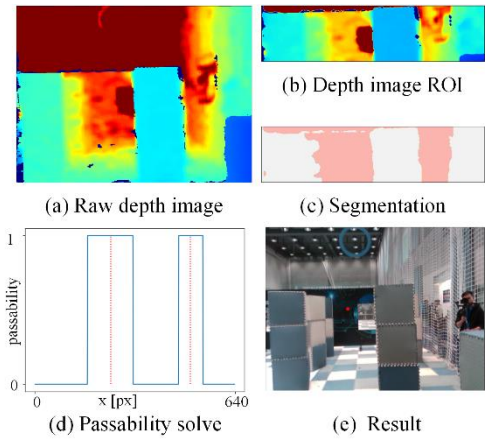


Figure 6. An example of solving c_{xd} from the depth image D.

Then the middleware ${}^{c}\mathbf{p}_{ca}$ can be calculated by using Equation (2) with minimum depth value in the non-passability area as depth s_{ca}. This subsection elaborates on the details for how to obtain ${}^{c}\mathbf{p}_{ca}$ from depth image D.

4. Velocity Planner

With some methods for acquiring middleware from images presented in Section III, the whole arena for autonomous drone racing is divided into different parts as displayed in Figure 7a, accompanied by a corresponding phase in Figure 7b. The principal task of the velocity planner is to generate velocity commands for guiding the drone to accomplish the task within the areas.

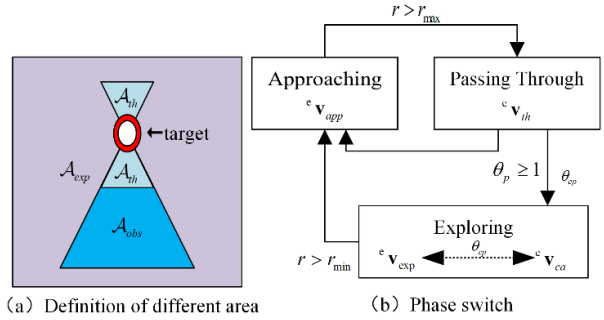


Figure 7. The overview of velocity planner.

As shown in Figure 7a, A_{obs} is the area where the drone can observe the gate; A_{th} is the area where the motion of passing through the gate occurs; A_{exp} is the area where the gate is too small to be observed or out of camera view. Besides, the pixel threshold of radius r_{max} and r_{min} are set by experiments. Provided that the pixel radius of a gate satisfies the inequality $r_{max} < r < r_{min}$, the drone will be regarded in A_{obs}.

The velocity planner generates velocity commands that follow certain criteria as follows: (1) ${}^{e}\mathbf{v}_{app}$ for commanding the drone to move from A_{obs} to A_{th}; (2) ${}^{e}\mathbf{v}_{th}$ for commanding the drone to pass through the gate and enter A_{exp} of the next gate; (3) ${}^{e}\mathbf{v}_{exp}$ for commanding the drone to move from A_{exp} to A_{obs}; (4) ${}^{e}\mathbf{v}_{ca}$ for collision avoidance.

4.1. Approaching Phase

In this section, a method to approach the static and moving gate is introduced. After obtaining ${}^{c}\mathbf{P}_{t}$ by the method proposed in Section 3, the command of approaching velocity ${}^{e}\mathbf{v}_{app}$ is initiated to make the drone approach the target i.e., $\|{}^{c}\mathbf{P}_{t}\| \rightarrow 0$.

This solution aims to reduce lateral error ${}^{e}Y_t$ and longitudinal error ${}^{e}Z_t$ in the earth frame {e} as follows:

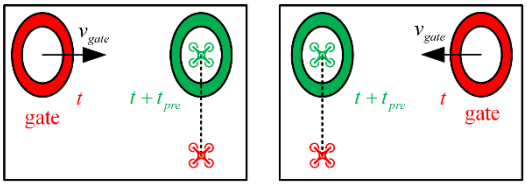
$$ {}^{e}\mathbf{P}_t = \mathbf{R}_{b}^{e} \cdot \mathbf{R}_{c}^{b} \cdot {}^{c}\mathbf{P}_t \tag{6} $$

$$ t_{pre} = \frac{{}^{e}X_t C_{scale}}{v_{tgt}} \tag{7} $$

$$ {}^{e}\mathbf{v}_{app} = \begin{bmatrix} v_{tgt} & \dfrac{{}^{e}Y_t}{t_{pre}} & \dfrac{{}^{e}Z_t}{t_{pre}} \end{bmatrix}^{\mathrm{T}} \tag{8} $$

where v_{tgt} is the specified speed and C_{scale} is the time scale coefficient. Controlled by the command velocity ${}^{e}\mathbf{v}_{app}$, the drone will approach the ring-shaped target. The distance between the drone and the ring-shaped gate $\|{}^{c}\mathbf{P}_t\|$, will be decreased through the method mentioned in this section, with the prerequisite that ${}^{c}\mathbf{P}_t$ could be solved from image I. However, the gate cannot always be detected in all areas. When the drone approaches a gate target to an extent where the gate is too large to be presented in image I, it is regarded to arrive at the area A_{th}, where the motion of passing-through occurs.

In particular, the drone is permitted to approach the moving gate only by flying forward. For the sake of simplicity, it is assumed that the gate is located to the left of the drone, as shown in Figure 8. The gate and the drone are marked with red and green corresponding at the timestamp t and timestamp $t + t_{pre}$. The proposed strategy works as follows: the drone hovers in the area in which it can observe the moving gate and moves forward at timestamp t with specified speed v_{tgt}. After a duration of v_{pre}, the drone approaches and passes through the moving gate. Due to the predetermined speed of the moving gate v_{gate}, it is easy to estimate the timestamp t for the drone to start the approaching phase.



(a) The moving gate is at the left of the drone (b) The moving gate is at the right of the drone

Figure 8. The strategy for approaching the moving gate.

4.2. Passing-through Phase

An estimator is put in place to address the potential problems during the process of passing through the gate. Provided that the process of passing-through commences with the situation where $r \geq r_{max}$ and ends with the circumstance in which $\theta_p \geq 1$. The process descriptor θ_p will satisfy:

$$ \theta_p(t) = \frac{t - t_{start}}{2t_{th}} \tag{9} $$

where t_{start} refers to the timestamp when the drone moves from A_{obs} to A_{th}, with t as the current timestamp and t_{th} as the last frame image I in A_{obs} computed by Equation (7). And ${}^{e}\mathbf{v}_{th}$ keeps the same value as the last ${}^{e}\mathbf{v}_{app}$ until the end of the passing-through phase.

4.3. Exploring Phase

As mentioned above, after the drone accomplishes its phase of passing through the gate, it enters a new phase in A_{obs} or A_{exp}. If the drone is in A_{obs}, it begins with a passing-through phase using the method in Section 4. If the drone is in A_{exp}, where the target may not appear in the camera view, the exploring phase will trigger and the drone will move from A_{exp} to A_{obs}. Thus, with the position of the gate target given roughly in the range of 1 m, the exploring velocity, namely ${}^{e}\mathbf{v}_{exp}$ is calculated via:

$$ {}^{e}\mathbf{v}_{exp} = v_{tgt}\, {}^{c}\mathbf{e}_{exp} \tag{10} $$

$$ {}^{c}\mathbf{e}_{exp} = \frac{{}^{e}\mathbf{P}_r(i+1) - {}^{e}\mathbf{P}_r(i)}{\left\| {}^{e}\mathbf{P}_r(i+1) - {}^{e}\mathbf{P}_r(i) \right\|} \tag{11} $$

where v_{tgt} is the specified speed and ${}^{c}\mathbf{e}_{exp}$ is the direction generated ahead of time from pre-knowledges. The drone can move from A_{exp} to A_{obs} controlled by velocity commands ${}^{e}\mathbf{v}_{exp}$.

It is supposed that the collision only occurs during the exploring phase. If $\theta_{cp} = 1$, which implies collision happening as some depth values in area D_5 are smaller than safety depth value s_a. Then the collision avoidance will be operated. Similar to the previous approaching phase, the velocity command ${}^{e}\mathbf{v}_{ca}$ for collision avoidance is designed based on the Equation (6)–(8), using ${}^{c}\mathbf{p}_{ca}$ in Equation (6) and supposing $C_{scale} = 1$ in Equation (7). The drone, controlled by velocity commands ${}^{e}\mathbf{v}_{ca}$, is enabled to move from non-collision-free area where $\theta_{cp} = 1$ to collision-free area where $\theta_{cp} = 0$.

5. Results

The proposed system is employed in the 2022 RoboMaster Intelligent UAV Championship, where the course, with a total length of around 74 m, consisted of 10 gates including 1 moving gate and many other obstacles. A schematic representation of the arena is depicted in Figure 9. Only three teams completed the full race course in the real-flight competition coming from two hundred teams that participated in the competition in total. Compared to other teams in Table 1, we win the first prize with the lightest drone and the shortest time of 58.98 s. The drone flies at an average speed of 1.25 m/s, with a peak speed of 2.5 m/s.

Table 1. Timing and Drone Weight Statistics.

Rank	Team	Institution	Weight [g]	Timing [s]
1	RFLY	BUAA	1306	58.98
2	Critical HIT	HITSZ	1500	84.23
3	Skyrocket	CETC-32	1661	159.74

The results of the system's main components are demonstrated in the video (https://rfly.buaa.edu.cn/videos/agile_crossing_video_v5_20230306_v3.mp4, accessed on 15 September 2023) attached to the paper. Due to missing the ground truth, the position of the drone is recorded by T265 while the position of gates is measured roughly by manual as shown in Figure 9. Since the drone can accomplish autonomous racing only by images, it can be observed that with controls on velocity without feedback of the absolute position.

According to the results illustrated in Figure 9, compared with the Hough-circle detection algorithm, the method proposed in this paper is insensitive to parameters and is more robust. It also shows that an excellent detection performance for circular objects including incomplete circular objects by comparing with the right side of Figure 10a and 10c. The proposed method detects a new circular object while the calculation frame rate satisfies the real-time requirements.

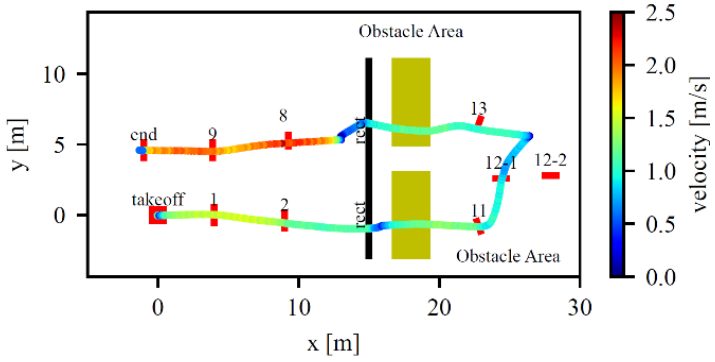
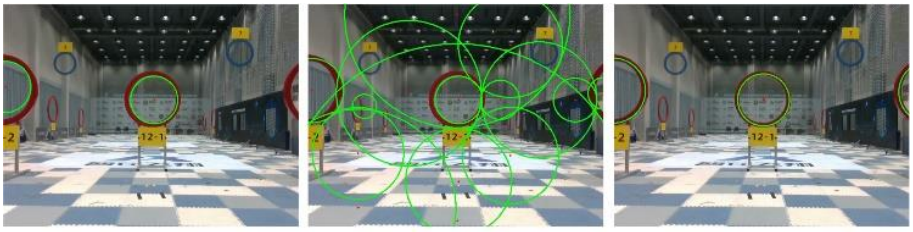


Figure 9. Trajectory with speed profile color.



(a) Hough-circle detect at 50fps while parameter2=20

(b) Hough-circle detect at 80fps while parameter2=25

(c) Proposed method at 60fps

Figure 10. The results of gate detection are calculated on Jetson NX. The parameter in Hough-circle detection is the threshold of the center accumulator. The smaller parameter is, the more fake circles there will be.

We achieve a success rate of over 90% in more than 30 trials (successfully passing through 214 out of 236 gates) of passing-through ring-shaped or square targets sequentially. A trial refers to a process in which a drone departs at a random position near the takeoff point, and arrives at the targets placed in predefined areas randomly, after autonomously passing through given gates in sequence. Success is achieved when the drone passes through all the gates without collision, yet the trial ends in failure if a collision occurs or the tasks are completed in a wrong order.

Figure 11 displays an example of how a drone explores, approaches and passes through gates. When timestamp t = 30 s, the drone is in A_{exp} without a visible gate. Controlled by ${}^{e}\mathbf{v}_{exp}$, it moves from A_{exp} to A_{obs} and enters into the approaching phase when timestamp t = 31 s. From timestamp t = 31 s to t = 33 s, the drone approaches the gate by ${}^{e}\mathbf{v}_{app}$. While the gate is nearly out of view at timestamp t = 34 s, it starts to pass through the gate with velocity commands ${}^{e}\mathbf{v}_{th}$. The drone finishes the passing-through process due to $\theta_p \geq 1$ at timestamp t = 35 s. After the passing-through phase, the drone explores the gate target if there is no gate in the image. Otherwise, the approaching phase or passing-through phase continues at the next visible gate.

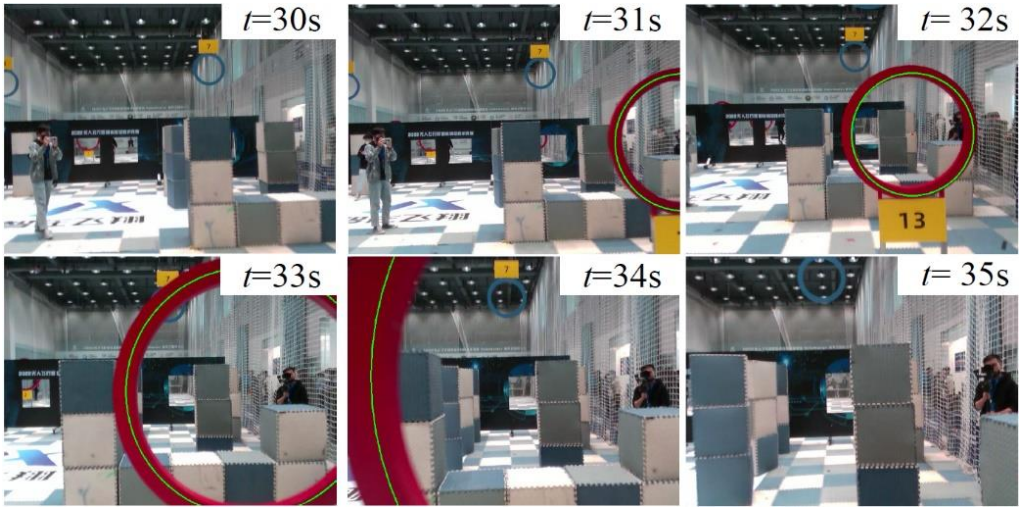


Figure 11. An example of the drone in the exploring phase, approaching phase and passing-through phase.

The results of collision avoidance are shown in Figure 12. At the timestamp t = 36 s, the depth value of the depth image center, around 3 m, is smaller than safety depth s_a, which signifies an inevitable collision coming soon if the drone maintains its movement. Hence, collision avoidance starts. In Figure 12, the yellow point depicted $\mathbf{p}_{ca}$ is solved by methods in Section 3. And then the drone is controlled by ${}^{e}\mathbf{v}_{ca}$, which makes the collision probability θ_{cp} return to 0. The collision avoidance ends at timestamp t = 38 s, due to θ_{cp} = 0 perceived from the depth image D.

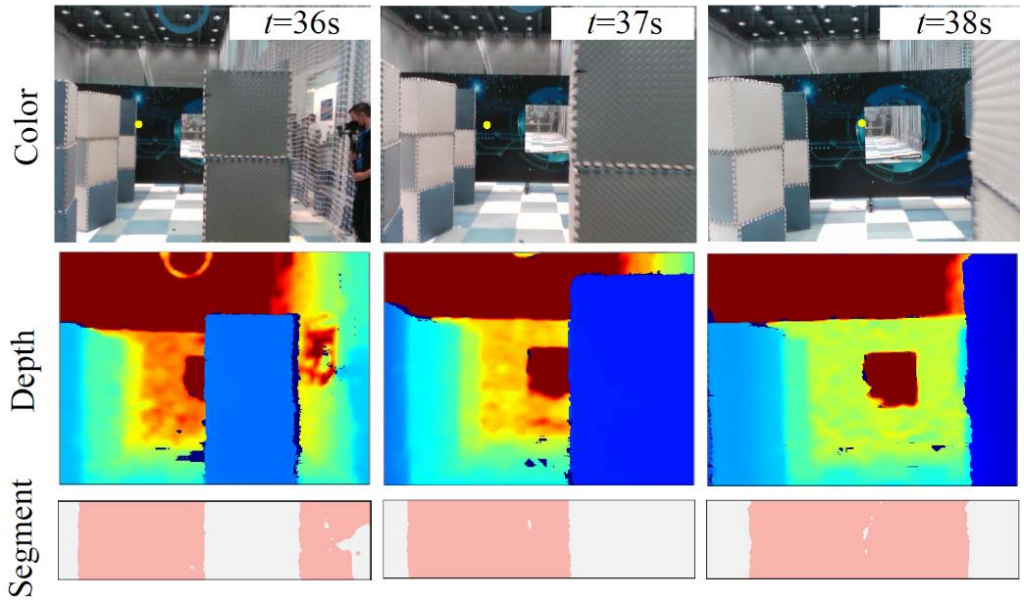


Figure 12. An example of the collision avoidance phase. The foreground and background are annotated with gray and pink respectively.

6. Discussion

Although the method proposed in this paper is still far from state-of-the-art, it brings some value to drone racing: (1) A software and hardware autonomous system is developed for drone racing, which could help researchers simulate and operate an autonomous drone. (2) We present a baseline of autonomous racing, including dynamic obstacle avoidance and velocity planner, which make the baseline applicable to more general scenarios. (3) Our system facilitates the understanding and development of perception, planning, and control. There is no learning-based method in our system, hence, the entire system is explainable.

The drone racing task is broken into three main components: perception, planning, and control. The main task for the perception is to estimate the vehicle state and to perceive the environment using onboard sensors. The most common solution for state estimation of flying vehicles is VIO. The next step is to plan. A feasible, time-optimal trajectory is planned subject to the limits of the physical platform and the environmental constraints. The final step is control. Control means making a real-time maneuver to lead the vehicle to track given trajectories, even with poor sensor information or mismatched models.

Learning-based approaches are the trend of the future. The first autonomous drone racing challenge at IROS 2016 reached a top speed of 0.6 m/s [2]. The team used a neural network to detect the gate. Two years later, the fastest with speeds up to 2.0 m/s in the third iteration of the ADR challenge [3]. The 2019 AlphaPilot Challenge [4] and the NeurIPS 2019's Game of Drones [21] provided further opportunities for researchers to compare them. The top speeds approaching 10 m/s at the challenging drone racing course with more learning-based approaches participated in the system. In 2022, the Swift system [22], designed and developed by the University of Zurich, successfully applied reinforcement learning to autonomous aircraft and became a milestone in being able to outperform human world champions in real competitions with a strategy similar to human pilots (speeds exceeded 20 m/s).

In conclusion, various presented learning-based approaches could improve the performance of drones. These approaches replace one, two, or all of the planner, controller, and perception stack with neural networks. In the perception stack, learning-based approaches inspire the end-to-end vision-based odometry algorithms, which could be specialized for drone racing tasks and potentially outperform classical VIO approaches. In the planning stack, the planned trajectory can be considered as an intermediate representation, generating something to help the platform perform the task under the constraint of the physics dynamics and the environment. The learning-based approaches could directly convert sensor observations to actuation commands. One of the biggest benefits of learning-based approaches is that different planners can be designed for various tasks, such as search and rescue, not only racing. In a control stack, a learning-based approach could make real-time decisions even in conditions with poor sensor information or mismatched models.

Overall, autonomous drone racing is a field full of passion and technical challenges. Its development not only promotes the advancement of drone technology but also opens up new possibilities for a wider range of drone applications in the future.

7. Conclusions

In this paper, a software and hardware system, which fulfills ring detection, collision avoidance as well as the passing of static or moving gates via RGBD image, and operates without the aid of information on the external position, has been developed for the 2022 RoboMaster Intelligent UAV Championship. The proposed system can help the drone to accomplish its flight through an array of gates with a 74 m course at a peak speed of 2.5 m/s and a success passing-through rate of over 90%.

In terms of the limitations, there is, however, still a long way to go for autonomous drones to outperform human pilots in terms of speed within a complex and dynamic environment. The drone may initially explore, discover and memorize the passable areas in unknown environments, and afterward speed up in the subsequent series of attempts, eventually reaching an astonishingly high speed. There are numerous techniques such as reinforcement learning, or iterative learning control that arouse our interests.

Author Contributions

Conceptualization, Methodology and Software, S.Y.; Software, S.Y., W.S. and R.L.; Validation and Hardware Resources, W.S. and R.L.; Writing, S.Y.; Review & Editing Q.Q.

Funding

This research received no external funding.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Loianno G, Scaramuzza D. Special issue on future challenges and opportunities in vision-based drone navigation. *J. Field Robot.* **2020**, *37*, 495–496.
2. Moon H, Sun Y. The IROS 2016 Competitions. *IEEE Robot. Autom. Mag.* **2017**, *24*, 20–29.
3. Moon H, Martinez-Carranza J, Cieslewski T, Faessler M, Falanga D, Simovic A, et al. Challenges and implemented technologies used in autonomous drone racing. *Intell. Serv. Robot.* **2019**, *12*, 137–148.
4. Foehn P, Brescianini D, Kaufmann E, Cieslewski T, Gehrig M, Muglikar M, et al. AlphaPilot: Autonomous Drone Racing. *Auton. Robots.* **2022**, *46*, 307–320.
5. Foehn P, Romero A, Scaramuzza D. Time-optimal planning for quadrotor waypoint flight. *Sci. Robot.* **2021**, *6*, eabh1221.
6. Penicka R, Scaramuzza D. Minimum-time quadrotor waypoint flight in cluttered environments. *IEEE Robot. Autom. Lett.* **2022**, *7*, 5719–5726.
7. Bloesch M, Burri M, Omari S, Hutter M, Siegwart R. Iterated extended Kalman filter based visual-inertial odometry using direct photometric feedback. *Int. J. Robot. Res.* **2017**, *36*, 1053–1072.
8. Qin T, Li P, Shen S. VINS-Mono: A robust and versa- tile monocular visual-inertial state estimator. *IEEE Trans. Robot.* **2018**, *34*, 1004–1020.
9. Xu W, Zhang F. FAST-LIO: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter. *IEEE Robot. Autom. Lett.* **2021**, *6*, 3317–3324.
10. Liu K, Zhao Y, Nie Q, Gao Z, Chen BM. Weakly Supervised 3D Scene Segmentation with Region-Level Boundary Awareness and Instance Discrimination. In Proceedings of the Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, 23–27 October 2022.
11. Campos C, Elvira R, Rodríguez JJG, Montiel JM, Tardós JD. ORB-SLAM3: An accurate open-source library for visual, visual-inertial and multi-map slam. *IEEE Trans. Robot.* **2021**, *37*, 1874–1890.
12. Delmerico J, Cieslewski T, Rebecq H, Faessler M, Scaramuzza D. Are we ready for autonomous drone racing? The UZH-FPV Drone Racing Dataset. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019.
13. Li S, van der Horst E, Duernay P, De Wagter C, de Croon GC. Visual model-predictive localization for computationally efficient autonomous racing of a 72-g drone. *J. Field Robot.* **2020**, *37*, 667–692.
14. Qi Y, Jiang J, Wu J, Wang J, Wang C, Shan J. Autonomous landing solution of low-cost quadrotor on a moving platform. *Robot. Autonom. Syst.* **2019**, *119*, 64–76.
15. Li S, Ozo MM, De Wagter C, de Croon GC. Autonomous drone race: A computation- ally efficient vision-based navigation and control strategy. *Robot. Autonom. Syst.* **2020**, *133*, 103621.
16. Kaufmann E, Gehrig M, Foehn P, Ranftl R, Dosovitskiy A, Koltun V, et al. Beauty and the beast: Optimal methods meet learning for drone racing. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019.
17. Yang K, Quan Q. An autonomous intercept drone with image-based visual servo. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Virtual Conference, 31 May 2020.
18. Quan Q. *Introduction to Multicopter Design and Control*; Springer: Singapore, 2017.
19. Fiorini P, Shiller Z. Motion planning in dynamic environments using velocity obstacles. *Int. J. Robot. Res.* **1998**, *17*, 760–772.
20. Otsu N. A threshold selection method from gray-level histograms. *IEEE Trans. Syst. Man Cybern.* **1979**, *9*, 62–66.
21. Game of Drones. Available online: https://github.com/microsoft/AirSim-NeurIPS2019-Drone-Racing (accessed on 15 September 2023).
22. Kaufmann E, Bauersfeld L, Loquercio A, Müller M, Koltun V, Scaramuzza D. Champion-level drone racing using deep reinforcement learning. *Nature* **2023**, *620*, 982–987.