Target Tracking for Quadrotors based on Deep Reinforcement Learning

Yan Gao[1*], Feiqiang Lin[1*], Changyun Wei[2], Raphael Grech[3] and Ze Ji[1]

***Abstract*— In this paper, we propose a deep reinforcement learning-based method for quadrotors to learn depth-based tracking policies autonomously. To this end, we present a novel reward function that guides the quadrotor to follow the target, avoid collisions, and keep the target close to the centre of the onboard camera's field of view without occlusions. In addition, to improve learning efficiency, we suggest using a teacher-student learning strategy. Specifically, we first train a state-based teacher policy encoding low-dimensional obstacle information, which then guides the vision-based student policy during training. Moreover, we introduce a variant of the Proximal Policy Optimisation algorithm based on the importance sampling algorithm. It facilitates the teacher-student learning process and enables the vision-based agent to escape local minima. The experimental results have demonstrated the satisfactory performance of our proposed method.**

I. Introduction

Autonomous aerial tracking is highly demanded for many applications, such as environmental surveillance, security, and aerial photography. However, it is challenging to allow a quadrotor, defined as a tracker, to autonomously track a moving target in unfamiliar and cluttered environments [1]. To ensure safety, the drone must accurately perceive targets and obstacles using onboard cameras and quickly respond to unforeseen obstacles within its limited field of view (FOV). This is challenging for drones constrained by size, weight, and power (SWaP), which limits their computing and sensing capabilities.

Most autonomous aerial tracking research focuses on non-learning, optimisation-based trajectory planning [2], [3]. Specifically, these methods decompose the tracking task into various sub-tasks, including sensing, mapping, planning, and trajectory optimisation [2]. However, this decomposition can increase processing latency due to computation and communication time between different components and may lead to a complex system [4].

In recent years, deep reinforcement learning (DRL) has shown promising performance in many research fields, including quadrotor control [5], [6]. With powerful representation learning capabilities, DRL methods bring about the possibility of learning control policies directly from raw sensory inputs, resulting in a simplified and straightforward system.

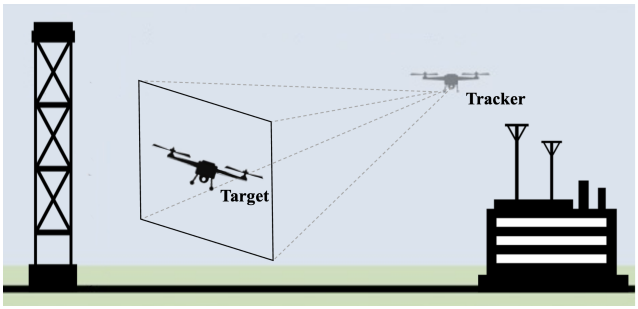


Fig. 1: A tracking task example. The tracker follows the target, keeps the target within the FOV and avoids obstacles simultaneously.

However, most works primarily focus on the problem of drone manoeuvrability [7] and navigation [8]. In particular, there is a lack of research on the use of DRLs for aerial tracking in cluttered environments.

In this paper, we propose a DRL-based and depth-based autonomous tracking method for quadrotors. In particular, first, we construct a novel reward function. It encourages the quadrotor to follow the target within a designated safe distance range, avoid collisions, and keep the target close to the FOV centre of the onboard camera without occlusions. In addition, perceiving environmental obstacles using depth images presents challenges in training deep neural networks from scratch due to potential instability and convergence issues. Although pre-training an autoencoder for encoding depth images is effective, it is time-consuming due to the image collection and training processes. Alternatively, since obstacle state information, including radius and relative positions, is available in simulation environments, a state-based teacher tracking policy can be trained. We suggest using obstacle state information instead of depth images, which can significantly accelerate teacher policy training. Once a satisfactory teacher policy is established, it guides the depth-based student policy during exploration.

Finally, to utilise such a teacher-student learning structure, we propose a variant Proximal Policy Optimisation (PPO) algorithm [9]. Under the teacher-student framework, some trajectory samples are collected from the teacher policy. However, training the vision-based student policy with these trajectory examples using the standard PPO algorithm [9] could lead to training collapse. This is due to the substantial initial differences between the teacher and the student policy, while the PPO algorithm imposes constraints on policy differences between two consecutive updates. To address this issue and train the policy effectively, we introduce a

*Yan and Feiqiang are joint first authors and contributed equally. Corresponding author: Ze Ji. Yan Gao and Feiqiang Lin thank the Chinese Scholarship Council (CSC) for providing the living stipend for their PhD programmes (Yan: No. 202008230171; Feiqiang: No. 201906020170).

[1]School of Engineering, Cardiff University, Cardiff, UK {gaoy84, linf6, jiz1}@cardiff.ac.uk

[2]College of Mechanical and Electrical Engineering, Hohai University, Changzhou, China c.wei@hhu.edu.cn

[3]Spirent Communications, Paignton, UK raphael.grech@spirent.com

variant of the PPO policy based on the importance sampling algorithm [10].

Our main contributions are as follows:

- We propose a DRL-based method that enables the quadrotors to track moving targets in cluttered environments.
- We introduce a novel reward function that encourages the quadrotor to follow the target, keep the target close to the centre of the FOV without occlusions, and avoid obstacles simultaneously.
- We suggest using a teacher-student learning strategy to improve learning efficiency. Also, we reconstruct the standard PPO algorithm based on the importance sampling algorithm to ensure the convergence of the teacher-student framework.
- Experimental results have demonstrated the remarkable performance of the proposed DRL-based autonomous tracking method.

The rest of this paper is organised as follows. Section II discusses the related work. Section III provides important preliminaries, and Section IV describes the proposed method in detail. Section V introduces our experiment setup and results. Section VI summarises our work.

II. Related Work

Conventional non-learning aerial tracking methods can be roughly divided into two categories: control-based methods [11], [12] and trajectory planning-based methods [13], [14]. Control-based methods tackle the tracking task by directly computing optimal drone control signals, such as drone velocities or drone attitude. Most works utilise PID control methods to reduce tracking errors defined in the image plane, ensuring that the target appears in the horizontal centre of the image and occupies enough pixels [11], [12]. However, these works present experiments in simple environments and do not require the avoidance of obstacle collision or occlusion. Also, the target to be tracked is static or moves at a relatively low speed.

Trajectory planning-based methods, instead of optimising control commands, optimise trajectories that are collision-free, occlusion-free, and dynamically feasible, ensuring targets remain centred in the observation. Most trajectory planning-based methods [13], [14] have achieved satisfactory results in simple environments. The target's trajectory is also easy to predict as it moves straightforwardly. In cluttered environments with random target movements, a coarse path is typically pre-planned using graph-based search [15], [16], A^* [17], or kinodynamic search [1], [2]. Subsequently, a smooth trajectory is optimised to fit the pre-planned coarse path. With the coarse path, collision-free flight corridors will be produced serving as constraints during the trajectory smoothing phase. By optimising trajectories, the algorithm can plan further into the future, enabling it to deal with complex environments. However, this benefit comes at the expense of requiring more computational resources and being more time-consuming, such as constructing the Euclidean Signed Distance Field (ESDF).

Learning-based algorithms can simplify the system in an end-to-end way, bypassing all the intermediate steps of the conventional methods. Imitation learning [18], DQN [19], DDPG [20] and PPO [21] have been applied to tackling the tracking task with simplified settings. These works are based on many assumptions, such as non-real quadrotor physics, minimal or no environmental obstacles, and predictable target movements. The obstacle information is also presumed to be available for decision-making. These assumptions may limit the capability of the methods for real-world tracking tasks.

III. Preliminaries

In this section, we will briefly review the important preliminaries on which our variant PPO algorithm is built, including Markov Decision Processes (MDPs) and standard PPO.

MDP: RL solves problems under the MDP framework. Physical processes are described by a state transition model $p(s_{t+1}|s_t, a_t)$ where states s are in the state space S and actions a belong to action space A. The reward function $R_t(s_t, a_t)$: $S \times A \rightarrow R$ is designed to evaluate action decisions based on task requirements. Strategies can be optimised by pursuing maximum overall discounted rewards $R(\tau) = \sum_{k=0}^{\infty} \gamma^k r_t$ received during the whole process. State-action value functions $Q^\pi(s_t, a_t) = E_{\tau\sim\pi}[R(\tau)|s_t, a_t]$ and/or state value function $V^\pi(s_t) = E_{\tau\sim\pi}[R(\tau)|s_t]$ are usually constructed to reconfigure the objective function, where π: $a \sim \pi(\cdot|s, \theta)$ represented by policy parameters θ representing. The advantage function is defined as $A^\pi(s, a) = Q^\pi(s, a) - V^\pi(s)$.

PPO: PPO is a policy gradient-based algorithm, which directly optimises the strategy π_θ. Reliability is achieved by limiting consecutive network updates during training to avoid large deviations that can cause training collapse. This constraint can be implemented by introducing a penalty to significant update differences evaluated using KL-divergence (PPO-Penalty) or by clipping update values (PPO-Clip) [9]. We adopt PPO-clip in this work, which has the following update process:

$$\theta_{k+1} = \arg\max_\theta \mathop{\mathrm{E}}_{s,a\sim\pi_{\theta_k}} [L(s, a, \theta_k, \theta)] \tag{1}$$

where L is given by

$$L(s, a, \theta_k, \theta) = \min\left(\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)} A^{\pi_{\theta_k}}(s, a), g(\epsilon, A^{\pi_{\theta_k}}(s, a))\right) \tag{2}$$

in which ϵ is a small hyperparameter and $g(\epsilon, A) = \begin{cases} (1+\epsilon)A & A \geq 0 \\ (1-\epsilon)A & A < 0. \end{cases}$

IV. Method

We propose a DRL-based method for addressing the dynamic target-tracking tasks in cluttered environments. The overall requirements encompass visibility, safe distance, occlusion avoidance, and collision avoidance. To simplify the

problem, we assume the obstacles to be spherical in shape. Then, the problem can be formulated as

$$\min_{\pi(x)} \mathcal{J}_o = \int_0^T \|^C p_{target}(x,y,t)\|^2 \mathrm{d}t, \tag{3a}$$

$$s.t. \quad x_{t+1} = f(x_t, \pi(x_t)), \tag{3b}$$

$$x_0 \in \mathcal{X}, \tag{3c}$$

$$p(t) \in \mathcal{P}, \ \forall t \in [0,T], \tag{3d}$$

$$p(t) \in \mathcal{V}_t, \ \forall t \in [0,T], \tag{3e}$$

$$d_l \leq \|p(t) - p_{target}(t)\| \leq d_u, \ \forall t \in [0,T], \tag{3f}$$

where x_t represents the quadrotor states such as position p_t and attitude q_t at timestep t. ${}^C p_{target}(x,y,t)$ is the target coordinate with respect to the image coordinate frame at timestep t. Otherwise, representations are under the world frame without specifications. The visibility requirement can be re-configured as Eq. 3a that puts the target in the centre of the $x-y$ plane. Eq. 3b shows the dynamics of the drone. $\mathcal{P}$ denotes obstacle collision-free area and $\mathcal{V}_t$ represents the occlusion-free area at timestep t. d_l and d_u are the distance boundaries describing the tracking distance requirement.

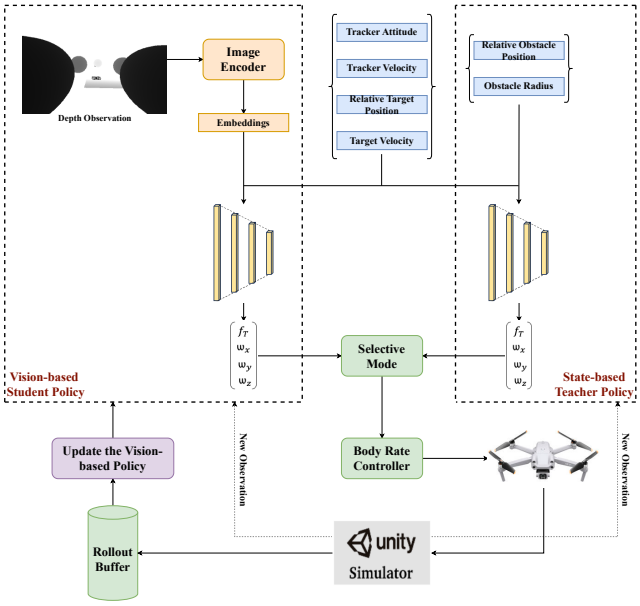


Fig. 2: Teacher-student learning pipeline.

A. Variant PPO Algorithm

In this work, we propose a variant PPO algorithm with a teacher-student learning structure to train the tracking policy.

State-based Teacher PPO Policy: We suggest training a state-based teacher using ground truth obstacle information. Then, this teacher will serve as an expert to guide the training of the vision-based student policy, thereby significantly improving learning efficiency. We utilise the standard PPO algorithm to train the teacher policy, and the network parameter is denoted as ξ.

Vision-based Student PPO Policy: Fig. 2 outlines the training process for the vision-based student policy. Both the teacher and student policies receive current observations during sample collection, making action decisions based on shared and individual components. A selective module picks actions from the teacher policy with probability ε, otherwise, from the current student policy. These actions, body rate $[w_x, w_y, w_z]$ and collective thrust f_T commands, are fed to a body rate controller. Transitions $[s_t, a_t, s_{t+1}, r_t]$ are produced by the environment and stored in the rollout buffer. Upon collecting a set number of transitions, the student policy is updated by randomly sampling data from the buffer. This is summarised in Algorithm 1.

We treat the training samples as if they all come from the teacher's policy and use importance sampling to get a non-biased cost function estimation.

$$\mathop{\mathrm{E}}_{s,a\sim\pi_{\theta_k}}[L] = \int \pi_{\theta_k} L dx = \int \frac{\pi_{\theta_k}\pi_\xi}{\pi_\xi} L dx = \mathop{\mathrm{E}}_{s,a\sim\pi_\xi}\left[\frac{\pi_{\theta_k}}{\pi_\xi} L\right] \tag{4}$$

Hence Eq. 2 can be approximated by:

$$\theta_{k+1} = \arg\max_\theta \mathop{\mathrm{E}}_{s,a\sim\pi_{\theta_k}}[L] = \arg\max_\theta \mathop{\mathrm{E}}_{s,a\sim\pi_\xi}\left[\frac{\pi_{\theta_k}}{\pi_\xi} L\right] \tag{5}$$

Algorithm 1 Variant Proximal Policy Optimisation

Input: teacher policy π_ξ, initial student policy parameters θ_0, initial value function parameters ϕ_0;
for $k = 0, 1, 2, \ldots$ **do**
  Initialise rollout buffer $\mathcal{D}_k$;
  for step $= 1, \ldots, m$ **do**
    With probability ε sample an action from $\pi(s, \xi)$; otherwise sample from $\pi(s, \theta_k)$;
    Execute action a_t in the environment; receive reward r_t and new state s_{t+1};
    Store transition (s_t, a_t, r_t, s_{t+1}) ;
  end for
  Compute rewards-to-go $R(t)$;
  Compute advantage estimates $\hat{A}_t$ based on the teacher value function V_ξ;
  Update the policy by maximising the importance sampling weighted objective:

$$\theta_{k+1} = \arg\max_\theta \frac{1}{|\mathcal{D}_k|T} \sum_{\forall\tau\in\mathcal{D}} \sum_{k=0}^{T} \frac{\pi_{\theta_k}}{\pi_\xi} L \tag{6}$$

  via stochastic gradient ascent with Adam;
  Fit value function via gradient descent algorithm:

$$\phi_{k+1} = \arg\min_\phi \frac{1}{|\mathcal{D}_k|T} \sum_{\forall\tau\in\mathcal{D}} \sum_{t=0}^{T} (V_\phi - R(t))^2 \tag{7}$$

end for

B. Tracking Policy

In this section, we will explain the implementation details of our method.

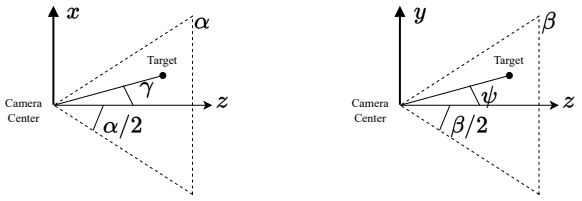
Target Dynamics: We model the target as a 2D vehicle using a velocity model that can move freely along x and y axes by controlling the acceleration speed of the corresponding axis. The altitude of the vehicle is fixed. The target moves freely without colliding with obstacles. Decoupling movements along the x and y axes of the target challenges the tracking task and enhances the quadrotor's general tracking capabilities.

State Space: For teacher policy, the state $s = [p_{\text{obstacle}} * n, r_{\text{obstacle}} * n, v_{\text{tracker}}, q_{\text{tracker}}, p_{\text{target}}, v_{\text{target}}]$ consists of relative positions p_{obstacle} and radii r_{obstacle} of the n closest obstacles, quadrotor states: v_{tracker} (linear velocity) and q_{tracker} (attitude), and target states: p_{target} (relative position) and v_{target} (linear velocity). In real-world applications, the low-dimensional state-based obstacle information is not available. Instead, we provide depth image $O_{\text{depth}}$to perceive obstacle information. Therefore, the state for the vision-based student policy is $s = [O_{\text{depth}}, v_{\text{tracker}}, q_{\text{tracker}}, p_{\text{target}}, v_{\text{target}}]$.

Action Space: The teacher policy and the student policy share the same action space. We are utilising body-rate control $a = [f_T, w_x, w_y, w_z]$ which is capable of agile flying as suggested in [22]. This action command will be carried out by a body-rate controller, which computes individual propeller thrust to control the quadrotor.

Reward Function: We construct the reward function with several components to satisfy the task requirements, including visibility, safe distance, occlusion avoidance, and collision avoidance, as described below:

$$R = R_{\text{visibility}} + R_{\text{safe\_distance}} + R_{\text{occlusion}} + R_{\text{collision}} \tag{8}$$



Horizontal View Vertical View

Fig. 3: Relative horizontal and vertical angles between the target and the camera. z is the camera-looking direction.

The visibility reward component $R_{\text{visibility}}$ is to punish the angle of the target to the camera centre vector in the camera frame, deviating away from the camera z axis. Fig. 3 describes the relative angles where α and β are the horizontal and vertical FOV respectively.

$$R_{\text{visibility}} = w(R_{\text{horizontal}} + R_{\text{vertical}}) \tag{9}$$

where w is a weight factor. The horizontal view is controlled by the quadrotor yaw, which can be decoupled from the control of the target pose. Thus, it can be constructed as a hard constraint :

$$R_{\text{horizontal}} = -0.2(e^{|\gamma|-\alpha/2} - e^{-\alpha/2}) \tag{10}$$

The vertical angle deviation can not always be guaranteed to be zero as it is controlled by the pitch of the quadrotor. Hence, we only punish the action when the relative vertical angle is larger than half of the vertical field of view, which makes the target disappear from the image.

$$R_{\text{vertical}} = \begin{cases} -0.2e^{|\psi|-\beta/2} & \psi \geq \beta/2 \\ 0 & otherwise \end{cases} \tag{11}$$

$R_{\text{safe\_distance}}$ ensures a safe distance between the quadrotor and the target:

$$R_{\text{safe\_distance}} = \begin{cases} 0 & d_l \leq \|p_{tracker} - p_{target}\| \leq d_u \\ -10 & otherwise \end{cases} \tag{12}$$

$R_{\text{collision}}$ is activated when the distance between the quadrotor and an obstacle is smaller than the sum of the obstacle's radius and the quadrotor's radius.

$$R_{\text{collision}} = \begin{cases} 0 & d^i_{obstacle} > r^i + r_{quadrotor} | i = 0, ..., N \\ -10 & otherwise \end{cases} \tag{13}$$

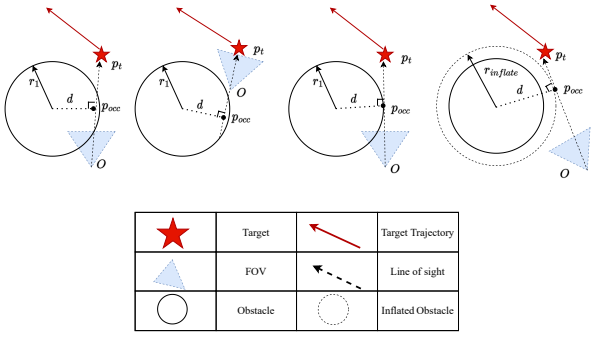


Fig. 4: An illustration of occlusion.

$R_{\text{occlusion}}$ is set to avoid occlusion. Occlusion occurs when obstacles intersect with the line of sight connecting the tracker and the target. This can be detected by ensuring that the distance between the obstacle and the line of sight exceeds the obstacle's radius, or by verifying that the closest point from the line of sight to the obstacle centre is behind the camera. We increase the obstacle occlusion radius such that the occlusion avoidance is robust, as illustrated by Fig. 4.

Network Structure: The network structure of our vision-based policy is shown in Fig. 5. The depth image carries high-dimensional information and will first be fed through the CNN module to obtain a low-dimensional latent feature vector. This vector will be concatenated with the remaining information ($[v_{\text{tracker}}, q_{\text{tracker}}, p_{\text{target}}, v_{\text{target}}]$) to provide inputs for the fully connected layers of the value network and the policy network.

V. Evaluation

A. Experiments Setup

In our work, we use the Flightmare simulator [23], which is based on the Unity game engine. To train our policy, in each episode, the target is spawned at a fixed altitude and will navigate to a randomly assigned goal position. The

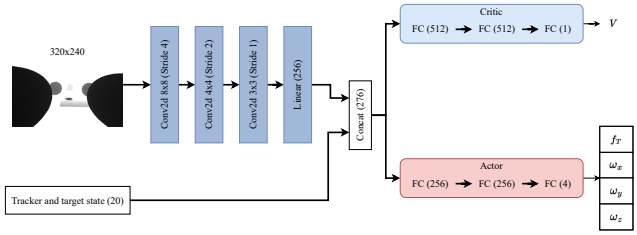


Fig. 5: Network structure of vision-based student policy.

obstacles are also randomly distributed. The quadrotor is spawned 2m away from the target at the beginning of each episode. A training environment example is shown in Fig. 6. The parameter w in Eq. 9 is set as 1.

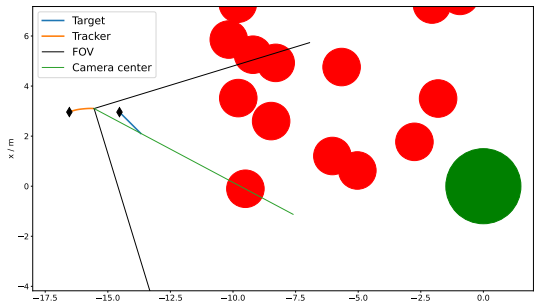


Fig. 6: A training environment example. Red circles are the obstacles and the green circle is the navigation destination of the target. Black thin diamonds are the starting points for the target and the quadrotor.

To study the performance of our proposed method, we carried out comprehensive experiments and analyses from two aspects, as follows:

- Effectiveness of the proposed reward function.
- Performance of the variant PPO algorithm.

B. Effectiveness of our proposed reward function

This section studies the effect of our proposed reward function. We train a state-based policy with the standard PPO algorithm using our proposed reward function. It will also serve as the teacher policy to guide the vision-based agent in this work. Fig. 7 demonstrates the training result of the state-based policy. From the figure, we can observe that the average reward of our state-based policy steadily increases and finally converges to -1.5. This indicates good tracking performance where nearly all constraints are met. It proves the effectiveness of our reward function. Fig. 8 shows an example of when the state-based policy is tested after training. The performance is considered satisfactory, where the quadrotor can closely follow the target and avoid obstacles.

C. Performance of the variant PPO algorithm

To validate the design of our variant PPO algorithm, we train two vision-based agents: 1) Vision-standard agent: the agent is trained with the standard PPO algorithm; 2) Vision-student agent: the agent is trained with our proposed variant PPO algorithm.

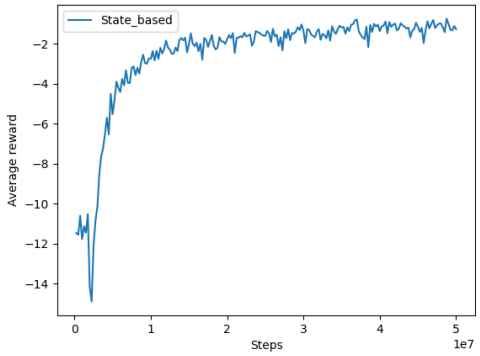


Fig. 7: Average rewards achieved by the state-based agent.

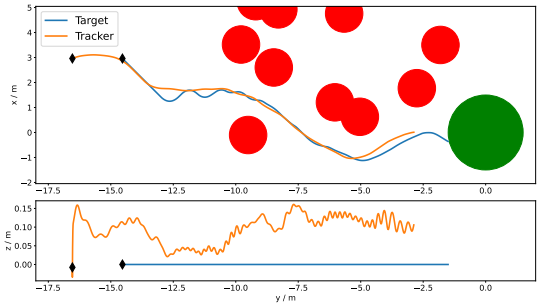


Fig. 8: A test example of the state-based agent after training.

Fig. 9 shows the average rewards achieved by different agents during training. From the figure, the average rewards achieved by the vision-standard agent do not show any indication of improvement with increased training steps. The average rewards fluctuate significantly at the early stage and remain at -10 after a period of training. This is because the agent has not yet learned the policy to avoid occlusion initially. Consequently, the target cannot stay in the centre of the camera as required. Such behaviour will be punished through Eq.10 and Eq. 11. Then, the agent will try to avoid such punishment by destroying itself as quickly as possible. It is achieved by flying away from the target at a high velocity. With this behaviour, the agent will only be punished with a reward of -10. From the perspective of optimisation, this is a local minima.

Mitigating this issue involves reducing the visibility punishment by using a smaller weight factor w as shown in Eq.9. We also train another agent using w of 0.1 (vision-standard2). The average rewards are also shown in Fig. 9. From the figure, we can see that this adjustment also results in encountering the same local minimum problem.

In contrast, the average rewards achieved by the agent with our proposed algorithm converge to around -1.5, which is similar to the state-based agent. It proves that the use of the variant PPO algorithm effectively resolves the local minimum issues. Fig. 10 illustrates a test example of our

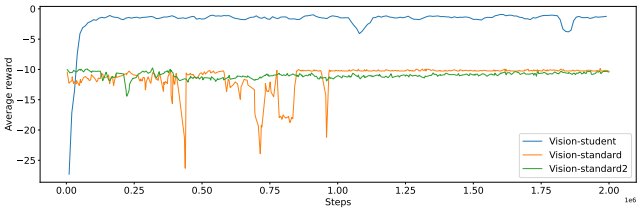


Fig. 9: Average training reward of vision-based agents.

method. As can be seen, the agent can simultaneously follow the target and avoid obstacles.

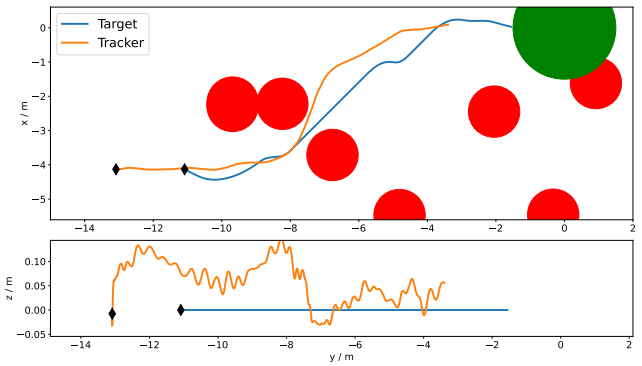


Fig. 10: An example of tests of our method after training.

VI. Conclusion

In this work, we proposed a DRL-based approach for aerial tracking tasks. A novel reward function and a variant PPO algorithm are introduced for training the policy. The reward function consists of several components designed to encourage the quadrotor to follow the target within a safe distance range, avoid collisions, and keep the target close to the centre of the onboard camera's FOV without occlusions. In addition, we suggest using a teacher-student learning pipeline to improve learning efficiency. A variant PPO method based on importance sampling is proposed to help the vision-based agent escape local minima. Experimental results and analyses highlight the effectiveness of our proposed reward function and the variant PPO algorithm in learning the tracking policy.

For future research, we will scale up our model to handle more difficult tracking tasks. In this work, all obstacles are assumed to be spherical in shape. We will deploy more realistic and general obstacles in the environment. Moreover, we aim to deploy this work on real robots and evaluate performance in real-world environments.

References

[1] Q. Wang, Y. Gao, J. Ji, C. Xu, and F. Gao, "Visibility-aware trajectory optimization with application to aerial tracking," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 5249–5256.

[2] Z. Han, R. Zhang, N. Pan, C. Xu, and F. Gao, "Fast-tracker: A robust aerial system for tracking agile target in cluttered environments," in *2021 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2021, pp. 328–334.

[3] M. Lyu, Y. Zhao, C. Huang, and H. Huang, "Unmanned aerial vehicles for search and rescue: A survey," *Remote Sensing*, vol. 15, no. 13, p. 3266, 2023.

[4] A. Loquercio, E. Kaufmann, R. Ranftl, M. Müller, V. Koltun, and D. Scaramuzza, "Learning high-speed flight in the wild," *Science Robotics*, vol. 6, no. 59, p. eabg5810, 2021.

[5] N. O. Lambert, D. S. Drew, J. Yaconelli, S. Levine, R. Calandra, and K. S. Pister, "Low-level control of a quadrotor with deep model-based reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4224–4230, 2019.

[6] H. Hua and Y. Fang, "A novel reinforcement learning-based robust control strategy for a quadrotor," *IEEE Transactions on Industrial Electronics*, vol. 70, no. 3, pp. 2812–2821, 2022.

[7] E. Kaufmann, L. Bauersfeld, A. Loquercio, M. Müller, V. Koltun, and D. Scaramuzza, "Champion-level drone racing using deep reinforcement learning," *Nature*, vol. 620, no. 7976, pp. 982–987, 2023.

[8] J. Fu, Y. Song, Y. Wu, F. Yu, and D. Scaramuzza, "Learning deep sensorimotor policies for vision-based autonomous drone racing," *arXiv preprint arXiv:2210.14985*, 2022.

[9] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[10] J. P. Hanna, S. Niekum, and P. Stone, "Importance sampling in reinforcement learning with an estimated behavior policy," *Machine Learning*, vol. 110, no. 6, pp. 1267–1317, 2021.

[11] J. Kim and D. H. Shim, "A vision-based target tracking control system of a quadrotor by using a tablet computer," in *2013 international conference on unmanned aircraft systems (icuas)*. IEEE, 2013, pp. 1165–1172.

[12] A. G. Kendall, N. N. Salvapantula, and K. A. Stol, "On-board object tracking control of a quadcopter with monocular vision," in *2014 international conference on unmanned aircraft systems (ICUAS)*. IEEE, 2014, pp. 404–411.

[13] J. Thomas, J. Welde, G. Loianno, K. Daniilidis, and V. Kumar, "Autonomous flight for detection, localization, and tracking of moving targets with a small quadrotor," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1762–1769, 2017.

[14] 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.

[15] B. F. Jeon and H. J. Kim, "Online trajectory generation of a mav for chasing a moving target in 3d dense environments," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 1115–1121.

[16] B. Jeon, Y. Lee, and H. J. Kim, "Integrated motion planner for real-time aerial videography with a drone in a dense environment," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 1243–1249.

[17] J. Chen, T. Liu, and S. Shen, "Tracking a moving target in cluttered environments using a quadrotor," in *IEEE International Conference on Intelligent Robots and Systems*, vol. 2016-November, 2016, pp. 446–453.

[18] F. Schilling, J. Lecoeur, F. Schiano, and D. Floreano, "Learning vision-based flight in drone swarms by imitation," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4523–4530, 2019.

[19] J. Moon, S. Papaioannou, C. Laoudias, P. Kolios, and S. Kim, "Deep reinforcement learning multi-uav trajectory control for target tracking," *IEEE Internet of Things Journal*, vol. 8, no. 20, pp. 15 441–15 455, 2021.

[20] B. Li and Y. Wu, "Path planning for uav ground target tracking via deep reinforcement learning," *IEEE access*, vol. 8, pp. 29 064–29 074, 2020.

[21] H. Duoxiu, D. Wenhan, X. Wujie, and H. Lei, "Proximal policy optimization for multi-rotor uav autonomous guidance, tracking and obstacle avoidance," *International Journal of Aeronautical and Space Sciences*, vol. 23, no. 2, pp. 339–353, 2022.

[22] E. Kaufmann, L. Bauersfeld, and D. Scaramuzza, "A benchmark comparison of learned control policies for agile quadrotor flight," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 10 504–10 510.

[23] Y. Song, S. Naji, E. Kaufmann, A. Loquercio, and D. Scaramuzza, "Flightmare: A flexible quadrotor simulator," in *Conference on Robot Learning*, 2020.