Space Noncooperative Object Active Tracking With Deep Reinforcement Learning

DONG ZHOU, GUANGHUI SUN, WENXIAO LEI, LIGANG WU

Actively tracking an arbitrary space noncooperative object relied on visual sensor remains a challenging problem. In this article, we provide an open-source benchmark for space noncooperative object visual tracking including simulated environment, evaluation toolkit, and a position-based visual servoing (PBVS) baseline algorithm, which can facilitate the research in this topic especially for those methods based on deep reinforcement learning. We also present an end-to-end active visual tracker based on deep Q-learning, named as DRLA VT, which learns approximately optimal policy merely took color or RGBD images as input. To the best of authors knowledge, it is the first intelligent agent used for active visual tracking in aerospace domain. The experiment results show that our DRLA VT achieves an excellent robustness and real-time performance compared with the PBVS baseline, benefitted from the design of complex neural network and efficient reward function. In addition, the multiple targets training adopted in this article effectively guarantees the transferability of DRLA VT by forcing the agent to learn optimal control policy with respect to motion patterns of the target.

Manuscript received 20 December 2021; revised 21 May 2022 and 28 July 2022; accepted 27 September 2022. Date of publication 3 October 2022; date of current version 6 December 2022.

DOI. No. 10.1109/TAES.2022.3211246

Refereeing of this contribution was handled by B. B. Gupta.

This work was supported by the National Key R&D Program of China under Grant 2019YFB1312001.

Authors’ address: The authors are with the Department of Control Science and Engineering, Harbin Institute of Technology, Harbin 150001, China, E-mail: (dongzhou@hit.edu.cn; guanghuisun@hit.edu.cn; leiwenxiao@stu.hit.edu.cn; ligangwu@hit.edu.cn). (Corresponding author: Guanghui Sun.)

0018-9251 © 2022 IEEE

NOMENCLATURE

AEL Average episode length.
AER Average episode reward.
BN Batch normalization.
CoM Center of mass.
DoF Degrees-of-freedom.
IBVS Image-based visual servoing.
PBVS Position-based visual servoing.
RPN Region proposal network.
\( r^* \) Position of target in \( F_B \).
\( r^* \) Expected distance between chaser and target.
\( \Delta t \) Sampling time of simulation.
\( \theta_i \) Parameter of Q-network at \( i \)th update.
\( \theta_i^- \) Parameter of target network at \( i \)th update.
\( F_I \) Reference frame.
\( F_B \) Chaser body-fixed frame.
\( F_C \) Vision camera frame.
\( F_T \) Target body-fixed frame.
\( C_{\text{pos}} \) Constant translational speed of target.
\( C_{\text{ang}} \) Constant rotation speed of target.
\( M_{\text{intr}} \) Camera intrinsic matrix.
\( Q^*(s, a) \) Optimal action value function.

I. INTRODUCTION

Space noncooperative object visual tracking, as one of the key components of vision system on spacecraft, has wide applications in space debris removal, malfunctioning spacecraft maintenance, asteroid exploration, and autonomous rendezvous and docking [1], [2], [3], [4], [5], [6].

According to the presence of control commands, space noncooperative object visual tracking can be categorized into passive and active methods. Many efforts have been devoted to studying passive methods [7], [8], [9], [10]. However, this kind of methods can easily fail, because of wide-range complex 6-degrees-of-freedom (DoF) motion of space noncooperative objects and low-resolution vision sensors with small field of view (FOV) mounted on spacecraft. This problem dramatically shortens observation period and severely blocks subsequent missions to proceed.

To this end, space noncooperative object active visual tracking has garnered much more concerns, which keeps the target in the field of view or makes the relative position and attitude between the chaser and the target to an expected state with vision sensors. It can provide more relaxed condition for follow-up tasks. In general, active visual tracking is typically classified into the following two main categories [3]: position-based visual servoing (PBVS) and image-based visual servoing (IBVS).

The essence of PBVS methods is that visual perception and controller design are solved, respectively. It can directly and naturally navigate chaser to the target with expected relative pose. Dong and Zhu [3] proposed a PBVS scheme for space robotic manipulator to capture noncooperative object, in which photogrammetry and adaptive extended Kalman filter were combined to estimate 6-DoF pose of target. However, this method was merely evaluated under...
easy simulated configuration where noncooperative object possesses discriminative image features and simple motion patterns.

Sun [11] and Liu [12], respectively, presented robust adaptive PBVS controllers with sliding-mode theory, assuming that the pose of noncooperative target is accurately measured during approaching stage. In real application, it is hard to achieve this ideal precondition that makes most of PBVS algorithms weak and vulnerable. We, therefore, introduce a simple but powerful PBVS method as baseline algorithm that adopts state-of-the-art 2-D monocular tracker (e.g., KCF [13], SiamRPN [14]) with RGB-D images to predict 3-D position of any non-cooperative target.

All the methods mentioned above, either PBVS or IBVS algorithms, decompose active visual tracking task into many subproblems including features extraction, features matching, pose estimation, control law design. Many researchers [15], [16], [17] have proposed that method following the idea of task decomposition is often trivial and suboptimal while it can not adapt to complex and dynamic environment.

With the rapid development of deep learning, deep reinforcement learning (DRL) that trains an end-to-end neural network with reinforcement learning algorithm has achieved great successes in a variety of fields. Minh et al. [18] utilized DRL to achieve high performances beyond human-level in Atari series of video games. AlphaGo [19] developed by DeepMind also surpassed the best Go player. In the context of robotic manipulation, Levine et al. [20] proposed a DRL-based approach to achieve hand-eye coordination for robotic grasping from monocular images. This provides a novel perspective for solving active visual tracking task.

According to whether the agent involves environment model, DRL algorithms can be divided into the following two categories [21]: model-based and model-free methods. Only if the environment model is correct, the model-based reinforcement learning algorithm can significantly outperform model-free methods with high sampling efficiency. However, it is often unavailable or difficult to be learned, especially for space noncooperative object. To this end, this article prefers model-free methods that estimates the state or action value function through plenty of trial-and-errors to optimize the policy $\pi(a|s, \theta)$.

One class of model-free reinforcement learning algorithms is value-based method that uses deep neural network to approximate action-value function with Q-learning. The most famous value-based DRL algorithm is DQN proposed by Minh [18], which is adopted in this work. To handle the overestimate problem of vanilla DQN, diverse variant algorithms like Double DQN [22], Dueling DQN [23], and Prioritized DQN [24] emerge in later. In addition, policy-based DRL algorithms [25], [26], [27] as another major type of model-free methods also attract extensive concerns, but it will not be involved in this article.

In recent, DRL has also made plenty of achievements in aerospace domain. High-level mission planning and decision-making of spacecraft based on the combination of partially observable Markov decision process with DQN were presented in paper [28]. Hovell [29], [30] adopted D4PG algorithm [31] as deep guidance of conventional controller for spacecraft approaching and docking, which alleviated the simulation-to-reality gap problem [32], [33] and achieved comparable performance on planar gravity-offset testbed. Wang et al. [34] also proposed self-tuning controller based on reinforcement learning for tracking noncooperative objects in space.

However, most related works based on DRL in aerospace domain have not considered learning policy directly from raw images of which information is abundant, convenient and lossless. It is mainly because the cost of this manner is unaffordable that samples extensive experimental images by thousands of trial-and-errors for training. To this end, we construct a virtual environment to simulate the active visual tracking scenarios in real-world space, which involves a chasing spacecraft mounted with vision camera and 18 types of space noncooperative targets.

Inspired by the method proposed in [35] that can track simple translational target with discriminative features in normal 2-D planar environment, we propose an end-to-end space noncooperative object active visual tracking algorithm, named as DRLAVT, which learns 3-D visual servoing policy straightforward by interactions with our simulation environment. Comparing to previous work in [35], the dynamic model of environment adopted in this article becomes more complex and randomized. Multiple targets and 3 types of disturbances are also considered evaluating the transferability and robustness of the DRL agent. Furthermore, our method can achieve an excellent tracking performances with two types of neural network architecture (i.e., ConvNet and ResNet) either for color image or RGB-D image.

The contributions of our work in this article are summarized as following:

1) An end-to-end active visual tracker based on deep Q-learning algorithm, named as DRLAVT, is presented. It provides interesting and pioneering methodology for space noncooperative object tracking, which learns optimal policy from raw color images or RGB-D images directly.

2) As comparison, we propose a novel and robust PBVS baseline method that adopts state-of-the-art 2-D monocular tracking algorithm with RGB-D images to predict 3-D position of target.

3) To train and evaluate active visual trackers, we construct simulation environment that consists of 18 space noncooperative object models and a chaser spacecraft mounted with vision sensors, which is available and open-source on https://github.com/Dongzhou-1996/SNCOAT.

The rest of this article is organized as follows. Section II formulates the problem of active visual tracking for space noncooperative object and introduces the configurations of simulation environment. We propose PBVS baseline algorithm in Section III and our DRLAVT algorithm in
Section IV. Furthermore, extensive experiments are implemented in Section V to show the effectiveness and advancement of DRLAVT. Finally, Section VII concludes this article.

II. PROBLEM FORMULATION AND SIMULATION ENVIRONMENT

A. Active Visual Tracking Problem

An active visual tracking problem involves a chaser spacecraft mounted with vision camera and a noncooperative object (e.g., spacecrafts, asteroids, rockets) with complex 6-DoF motion, which is shown in Fig. 1. It is worth noting that we only focus on how to guide spacecraft approach to the target with images, that is, no relative attitude synchronization is considered in this article. Besides, we further simplify the dynamic models of the chaser and target [29], [30] by an assumption that both objects are free-floating under close range (less than 50 m), although more precise orbital dynamic models are adopted in other studies.

During active visual tracking, Four coordinate systems depicted in Fig. 1 are involved, as follows.

1) Reference frame, denoted by $\mathcal{F}_I$, of which origin is located at somewhere near both chaser and target.
2) Chaser body-fixed frame, denoted by $\mathcal{F}_B$, of which origin is the center of mass (CoM) of spacecraft.
3) Vision camera frame, denoted by $\mathcal{F}_C$, of which axis aligns with $\mathcal{F}_B$. The transformation matrix from $\mathcal{F}_C$ to $\mathcal{F}_B$ is denoted as $M_B^C$.
4) Target body-fixed frame, denoted by $\mathcal{F}_T$, of which origin is the CoM of target.

We first give the simplified dynamic model of chasing spacecraft:

$$\dot{X}_s^I = \frac{u}{m_s} + \Omega$$ (1)

in which, $X_s^I$ is the 3-D position of chaser (i.e., CoM of chaser) in $\mathcal{F}_I$. $u = \{F_x, F_y, F_z\}$ is the control force. In real application, the control force provided by the thruster is often bounded. To this end, we assume that $\|u\|_\infty \leq 50$ N.

$\Omega$ is the noise term introduced by actuator disturbance, computational time-delay, and image blur together. In addition, $m_s$ is the mass of spacecraft which is set to 113.9 kg.

And then, because of the noncooperative nature of target, we assume there are no control force and torque applied on it. Hence, the complex 6-DoF motion is reduced to uniform linear motion and rotation

$$\dot{X}_T^I = C_{pos}$$

$$\dot{\omega}_T = C_{ang}$$ (2)

where $X_T^I = \{x_T^I, y_T^I, z_T^I\}$ and $\omega_T = \{\alpha, \beta, \gamma\}$ are, respectively, the 3-D position and angular of target in $\mathcal{F}_I$. $C_{pos}$ and $C_{ang}$ are two constants denoted the translational and rotational speed.

The goal of active visual tracking can be formulated as follows:

$$e = \|r_B^T - r^*\|_2 = \|M_B^C \cdot r_N^T - r^*\|_2 = 0$$ (3)

in which, $r_B^T$ is the position of target in $\mathcal{F}_B$, $r^*$ denotes the expected distance between chaser and target. In this work, we set $r^* = \{0, 0, 5\}$.

B. Environment Specification

To study more powerful and general active visual tracking algorithm, especially for DRL-based methods, we construct a space simulation environment by using virtual physic engine, CoppeliaSim, which includes spacecraft mounted with a perspective vision camera and 18 types of noncooperative object models. All the models can be concluded into the following five categories: 1) asteroids; 2) return Capsules; 3) rockets; 4) satellites; 5) space stations.

It is clearly shown in Fig. 2 that the geometry, size, and texture of different types of targets are various. Therefore, traditional PBVS and IBVS algorithms based on handcrafted image features are impossible to be adapted to all the objects.

We suppose that the only visual sensor in simulated environment to perceive noncooperative target is a perspective vision camera installed at the front of chaser as shown in Fig. 1. However, it is worthwhile noting that both color image and depth map (see in Fig. 3) can be achieved by the camera. The model of vision camera is described in Fig. 4, which is important for active visual tracking. Since only perspective angle ($\alpha_x, \alpha_y$) and resolution ($W, H$) can be obtained in CoppeliaSim, we, therefore, compute camera intrinsic matrix $M_{intr}$ following (4):

$$M_{intr} = \begin{bmatrix} \frac{W}{2 \tan(\alpha_y/2)} & 0 & \frac{W}{2} \\ 0 & \frac{H}{2 \tan(\alpha_x/2)} & \frac{H}{2} \\ 0 & 0 & 1 \end{bmatrix}.$$ (4)

In this work, we set $\alpha_x = \alpha_y = 60^\circ$ and $W = H = 256$.

Furthermore, the initial state of noncooperative object is another important configuration for active visual tracking research, involving initial position, attitude, velocity, and angular velocity. There is a necessary condition that target should be observable at beginning stage of tracking.
Fig. 2. 18 types of space noncooperative target models, involving Asteroids, Return Capsules, Rockets, Satellites, and Space Station. Objects with green labels are utilized for training, the others are adopted to evaluate.

Fig. 3. Color image and depth map of Asteroid 01 simultaneously captured by simulated vision camera. (a) Color image. (b) Depth map.

Fig. 4. Schematic of perspective vision camera in simulated environment. We also depict the distribution of initial position of noncooperative target in $\mathcal{F}_C$.

To validate the robustness of active visual trackers, the following three types of perturbations are introduced into our simulated environment.

1) Actuator noise. In application, the margin between ideal controller output and real actuator output is inevitable. Therefore, we multiply action command with a noise factor $f_a$ which obeys $\mathcal{N}(1, 0.3)$.

2) Processing time-delay. Considering the variance of running speeds, we add a random time-delay $D$ to sampling time $\Delta t$ of simulation (original sampling time is set to 0.1 s).

3) Image blur. Because of rapid camera motion, severe image blur often leads to tracking failure. In this article, we simulate different levels of image blur by the average of $N$ consecutive images ($N \geq 2$).

C. Evaluation Mechanism

Reasonable evaluation mechanism is important to show real ability of active visual trackers. In this section, more details of our evaluation configuration are provided.

At first, an one-shot protocol [36] is introduced into our evaluation, that is, the use of unseen classes of space noncooperative objects for evaluation. As we mentioned before, there are 18 different types of target models constructed by us (see in Fig. 2), in which two-thirds of models are used for training and the others are utilized for validation.

Second, we add delayed ending stage into each episode to explore recovery ability of active visual trackers, which can also balance the distribution of positive and negative samples during training. The length of delayed ending is recommended to be 10–20 frames.

Finally, the average episode length (AEL) and average episode reward (AER) are adopted as evaluation metrics. The higher AEL and AER are, the more accurate and robust active visual tracker is. AEL and AER are defined...
which demonstrated its generalization ability in aerospace domain [10].

At the same time, we generate 3-D point cloud \( P_i = \{(x_i^C, y_i^C, z_i^C) \in \mathbb{R}^3 | i = 1, 2, \ldots, r \} \) with RGB-D image, which can be easily calculated by following equation:

\[
\begin{bmatrix}
    x_i^C \\
    y_i^C \\
    z_i^C 
\end{bmatrix} = M_{\text{intr}} \begin{bmatrix}
    u_i \\
    v_i \\
    1
\end{bmatrix} 
\]

where \( M_{\text{intr}} \) is the intrinsic matrix of vision camera mentioned in (4) and \( z_i^C > 0 \).

After 2-D tracking result \( A_t \) and point cloud \( P_t \) are both available, the frustum proposal \( P_{\text{frustum}} = \{(x_i^C, y_i^C, z_i^C) \in \mathbb{R}^3 | i = 1, 2, \ldots, n \} \) can be extracted as shown in Fig. 6. In this work, we assume the extracted frustum proposal only contains the point cloud of object of interest. So the 3-D target position \( r_i^C \) in \( F_C \) is estimated by using frustum average operation which is formulated as

\[
r_i^C = \left[ \frac{1}{n} \sum_{i}^n x_i^C, \quad \frac{1}{n} \sum_{i}^n y_i^C, \quad \frac{1}{n} \sum_{i}^n z_i^C \right]^T.
\]

B. Controller Design

Although various control methods (e.g., optimal control, sliding-mode control, robust control) for active visual tracking have been proposed, in our opinion, the accurate and real-time vision perception algorithm described herein before will significantly alleviate complexity of control laws design. We, therefore, utilize classical PID controller to realize active visual tracking

\[
u(t) = K_P \cdot e(t) + K_I \cdot \int e(t) dt + K_D \cdot \dot{e}(t)
\]

where \( e(t) \) is defined in (3), \( K_P, K_I, \) and \( K_D \) are the coefficients of PID controller, respectively. To correctly evaluate PBVS algorithm, we make a deal that the coefficients of controller are prohibited to manually fine-tune.

IV. DRLAVT ALGORITHM

In this section, we propose an active visual tracker based on DQN, named as DRLAVT, which guides chaser spacecraft approach to the noncooperative target only relied on color or RGB-D images.
A. Deep Q-Learning

DQN [18] is one of the most famous value-based reinforcement learning algorithms, which leverages powerful representational ability of deep neural networks to approximate the optimal action value function $Q^*(s, a)$.

**Lemma 1** The action value function $Q_{i+1}(s, a)$ can approximate $Q^*(s, a)$ by iterative update with Bellman equation, when $i \to \infty$:

$$Q_{i+1}(s, a) = \mathbb{E} \left[ r_t + \gamma \max_{a'} Q_i(s', a') | s, a \right]$$ (10)

in which, $s$ is the environment state observed by the agent, $a$ is the action implemented by agent when it observed a state $s$, and $r_t$ is the reward that agent achieves from environment at $t$th timestep after it takes an action. $\gamma$ is the discount factor for future rewards.

According to Lemma 1, DQN can be trained with loss function $\mathcal{L}(\theta_i)$

$$\mathcal{L}(\theta_i) = \mathbb{E}_{(s, a, r, s') \sim D} \left[ (y_i - Q(s, a; \theta_i))^2 \right]$$ (11)

in which, $y_i = r_t + \gamma \max_{a'} Q(s', a'; \theta^-_i)$ is temporal-difference target, $\theta_i$ represents deep Q-network at $i$th iteration, and $\theta^-_i = \theta_i / N$ denotes target network that are updated periodically. The samples $(s, a, r, s')$ utilized to train Q-network are drawn uniformly at random from memory replay pool $D$, also named as experience replay mechanism, which decreases the correlations in observation sequence.

B. Action Space and Reward Function

At first, reasonable predefined action space in low dimension is necessary for DRLAVT algorithm. We, therefore, consider a simple action set included 11 types of translational actions in 3-D space without rotational operation, which makes DQN converge quickly during training stage. The details of action space defined in chaser body-fixed coordinate system $\mathcal{F}_B$ can be seen in Fig. 8. For example, to take an action like $(1, -1, 0)$, chaser spacecraft should move forward 1 step in $x$-axis and back 1 step in $y$-axis. In simulation environment, we assume one step equal to 0.5 m, which is realized by the low-level servoing controller.

Reward is the source of intelligence for an agent in reinforcement learning, like labeled data in supervised learning. What the agent learned from extensive trials-and-errors is heavily depended on the definition of reward function.
Therefore, we formulate a reward function for space non-cooperative object active visual tracking, which includes a visible reward term $r_{\text{vis}}$ and a distance penalty term $r_{\text{dist}}$:

$$r_t = r_{\text{vis}} - r_{\text{dist}}.$$ (12)

In our opinion, the first thing that an agent must learn is to keep the space noncooperative target always in the field of view. We, therefore, add the visible reward term into reward function. Considering the unbalance of visibility of samples, a higher penalty is given for the case when the target is out of view. So the final visible reward is formulated as

$$r_{\text{vis}} = \begin{cases} +1, & \text{in camera view} \\ -5, & \text{out of camera view} \end{cases}.$$ (13)

For active visual tracking, the agent should reduce 3-D tracking errors defined in (3). Therefore, distance penalty term $r_{\text{dist}}$ is also added into reward function

$$r_{\text{dist}} = e = \| r_T^p - r^* \|_2.$$ (14)

C. Q-Network Architecture

The core of deep Q-learning algorithm is to predict action values with observed image by Q-network. In original paper, Mnih [18] utilized a shallow neural network, which only consists of three convolutional layers and two fully connected layers. Considering the powerful representational ability of deeper convolutional networks [39], [40], we adopt two types of neural network architectures in this work, which are clearly shown in Fig. 7.

The first architecture proposed by us is ConvNet derived from original DQN. The main improvements include the following equation.

1) Additional convolutional layer with kernel size 1 and stride 1, which aims to merge the features in depth axis of tensor, especially for RGB-D images.
2) more max-pooling layer following each convolutional layers, which significantly reduce the dimension of tensor and learns high-level patterns from raw input image.
3) additional fully connected layer and Dropout regularization method. Although colorful tricks are used in ConvNet, it remains tiny and elegant.

To study the impact of deeper convolutional layers on active visual tracking task, we further utilize ResNet [40] as the second architecture of Q-network here, which can involve hundreds of convolutional layers without performance degradation due to the property of residual units. Therefore, we develop three ResNets (i.e., ResNet-18, ResNet-34, and ResNet-50) with different configurations. There are two types of residual units utilized to constructed ResNets, as shown in Fig. 9. With the depth of ResNet increasing, the second residual unit can significantly reduce the parameter numbers and computational costs.

V. EXPERIMENTS

In this section, we first evaluate PBVS baseline and DRLAVT algorithms on simulated environment with 20 repetitions, which demonstrates the effectiveness and advancement of our method. The influences of disturbances, reward functions, key components, and different DQN variants on DRLAVT performance are further studied. In addition, we also validate DRLAVT with multiple motion patterns to explore what it learned from hundreds of trial-and-errors. It is worthwhile noting that all the experiments are implemented on HPC server with Intel Xeon@E502650v4@2.2 GHz CPU and Nvidia Tesla P100 GPU. The detailed training configuration are provided in Table II.

A. Evaluation Results

1) PBVS Baseline: In this article, we, respectively, adopt two classical types of 2-D monocular trackers, SiamRPN [14] and KCF [13] into our PBVS framework and, respectively, evaluate them in simulated environment, where the initial state of 2-D monocular tracker is automatically generated by evaluation toolkit and reinitialization is prohibited when tracker fails. Experiment results listed at the first and third rows of Table 1 show that PBVS algorithms based on either SiamRPN or KCF only tracks the target in a short time, however, it achieves very high AER metric. It means that the tracking accuracy of PBVS baseline is excellent while its robustness is pretty weak.

In addition, we also find out the active tracking performances significant degrades 23.6% under AER metric after the replacement of 2-D monocular tracker, although there is a slight improvement on running speed. It proves that an accurate, fast, and robust 2-D monocular tracker is important for PBVS baseline.

Furthermore, we switch to control the velocity of chaser spacecraft in PBVS framework via the tracking error signals instead of direct force control. The kinematic model of
chaser spacecraft is formulated as

\[ X^I_s = u' + \Omega' \tag{15} \]

in which, \( X^I_s \) is the position of chaser, \( u' \) is the control command of chaser velocity, and \( \Omega' \) is noise term. We also assume that \( ||u'||_{\infty} \leq 5 \). The evaluation results in the second and fourth rows of Table I demonstrate PBVS baseline with velocity control reaches higher performance comparing to force control.

The active tracking results of PBVS baseline are illustrated in Fig. 10. We can clearly see that the 3-D tracking errors are in long-cycle fluctuation, which are jointly caused by the saturation of PID controller, random shift of 2-D monocular tracker, and inaccurate estimation of 3-D position when relative motion occurs between the chaser and noncooperative target.

2) DRLAVT: The DRLAVT based on ConvNet that directly adopts color image to select action is evaluated first, of which results are listed at the fifth row of Table I. Our DRLAVT greatly outperforms PBVS baseline algorithm under AEL metric and runs up to 743.68 Hz without tracking initialization. Although it achieves worse AER score comparing to baseline algorithm, because of the difficulty for the agent with reinforcement learning to learn depth information from color images. It is also clearly shown in Fig. 11, the oscillation of trajectory looks like...
severe and the maximum tracking error in $z$-axis is about 6 m, which is the main reason that DRLA VT acquired low AER score.

To this end, we feed color image along with extra depth map (i.e., RGB-D image) to DRLA VT and retrain it from scratch. Note that before fetched into network, the depth map $I_{depth}$ has been normalized to $[0, 1]$ by the following equation:

$$I_{depth} = \frac{I_{depth}}{z_{max}}$$  \hspace{1cm} (16)

in which, $z_{max}$ is the maximum distance that simulated vision sensors can perceive. As we expected, DRLA VT with RGB-D images reaches the maximum AEL measurement and improve the AER about 950.98 scores, benefitted from the directly provided depth map and the channel-wise fusion of feature tensor using $1 \times 1$ convolution. The chaser can quickly approach to the target at the start of tracking and reduce tracking errors to 0 with small oscillation under the guidance of DRLA VT, which can be seen from the active tracking trajectory in Fig. 12(a) and tracking errors in Fig. 12(b).

DRLA VT that utilizes different ResNets as Q-network are also evaluated, including ResNet-18, ResNet-34, and ResNet-50. The results summarized at the final 6 rows in Table I clearly shows ResNet can further improve the performance of DRLA VT, because of its large model capacity and powerful representational ability. Particularly, DRLA VT with ResNet-34 can achieve 1001 AEL and 530.70 AER.

However, performance degradation inevitably occurs with the increasing depth of ResNet.

In addition, we have counted the parameter number and computational complexity of 4 neural networks utilized in our DRLA VT algorithm, which are summarized in Table III. The statistic results of computational overhead are consistent with average speed metrics (see the last column in Table I) in evaluation results. And it also demonstrates the DRLA VT based on ConvNet is an elegant and powerful method, which can be further deployed on spacecraft in future even the computational resource are limited.

3) Impact of Noise: The robustness of active visual trackers is of significance in application. Therefore, we
evaluate both PBVS baseline and DRLAVT algorithm under different types of perturbations, such as actuator noise, processing time-delay, and image blur. The experiment results are summarized in Table IV. We find out that actuator noise and time delay severely decrease the AEL and AER metrics of both active trackers, but the degradation of PBVS algorithm is larger than DRLAVT. Meanwhile, it is evident that DRLAVT is more sensitive to image blur than PBVS baseline which benefits from the complexity architecture of SiamRPN, as blurring level increases. Although the influence of image blur is relatively insignificant comparing to other perturbations, and it does not interrupt the active tracking process of DRLAVT. We think the agent is puzzled by image blurring to infer the actual depth information of the target, while it can react correctly to the translation in \( X \) and \( Y \) axes.

On the whole, our DRLAVT algorithm is more robust to different disturbances, of which AEL metric only degrades 1.7% even with the presence of three perturbations together. As comparison, the AEL measurement of PBVS baseline dramatically shrinks 72.1%.

B. Further Study on DRLAVT

1) Multiple Targets: Paper [17] presented that extensive simulated data generated by domain randomization method can make the agent more robust to environment variations and reach zero-shot simulation-to-real transfer on the task of drone racing. To this end, our DRLAVT is also trained about 300 episodes with 12 space noncooperative objects, which aims to make agent to learn various motion patterns of targets and optimal guidance policy, rather than specific geometries or textures. For comparison, an extreme case is introduced that only one target (Asteroid02) is used to train DRLAVT agent with the same configurations. The experiments are summarized in Table V, which include detailed evaluation measurements for five types of unseen targets (i.e., Asteroid06, Capsule03, Rocket03, Satellite03, Station03). It clearly shows that DRLAVT trained with Asteroid01 only achieves comparable results on the similar targets, such as Asteroid06 and Capsule03. For other three types of evaluated targets, the AER score achieved by DRLAVT even decreases to \(-3155.57\) while the AEL metric remains in high level. In contrast, DRLAVT trained with multiple targets can successfully track all the
targets with the high performance. Therefore, we are convinced that the multiple targets training indeed guaranteed the transferability of DRLAVT.

2) Reward Functions: We carry out ablation study on reward function defined in (12) to explore how it impacts the performance of DRLAVT. At first, we remove the visible term in original reward function

\[ r_1^t = r_{\text{dist}}. \]  

(17)

In addition, we modify the penalty of the visible term when the target is out of view, which is formulated as follows:

\[ r_{\text{vis}}' = \begin{cases} +1, & \text{in camera view} \\ -1, & \text{out of camera view} \end{cases} \]  

(18)

Therefore, we define the second new reward function

\[ r_2^t = r_{\text{vis}}' + r_{\text{dist}}. \]  

(19)

The evaluation results of DRLAVT retrained from scratch by using (17) and (19) are listed in Table VI. We also depict the episode length and action values curves during training in Fig. 13, which clearly proves that the visible term is of significance to accelerate the training process of DRLAVT and improve the performance of active visual tracker. In addition, the higher penalty in the visible terms can also improve the action values that DRLAVT predicts, as we expected.

3) Architecture Components: Another ablation study is also implemented on our proposed method, in which we,
Table VII
Influences of Different Key Components of ConvNet on DRLA VT Performance

| Components       | AEL   | AER   |
|------------------|-------|-------|
| original         | 1001  | 391.74|
| without BN       | 961.67| −796.62|
| without Layer4   | 1001  | −482.89|
| without Maxpool  | 992.20| −263.91|

Fig. 14. Training curves of DRLA VT without different key components. (a) Episode length. (b) Action value.

Table VIII
Comparison of DRLA VT Based on Different DQN Variants

| Name               | AEL   | AER   |
|--------------------|-------|-------|
| Vanilla DQN [18]   | 1001  | 391.74|
| Double DQN [22]    | 1001  | 116.48|
| Dueling DQN [23]   | 1001  | 446.09|
| Prioritized DQN [24]| 1001  | 532.44|

Fig. 15. Training curves of DRLA VT based on different DQN variants. (a) Episode length. (b) Action value.

The evaluation results in Table VIII clearly show that Dueling DQN and prioritized DQN can significantly improve the performance of DRLAVT, which are also consistent with their training curves. Double DQN unexpectedly achieves worse AER, even though its action values are always higher than vanilla DQN during training. We think it is caused by the instability of double Q-learning when the tracked target is changing. In addition, all three types of DQN variants can speed up the learning progress as shown in Fig. 15.

 Exploration Inner Mechanism of DRLA VT: Although we have proven the effectiveness and advancement of our method herein before, the most interesting point is what agent actually learns from large-scale trial-and-errors with DRL. To this end, we initialize the noncooperative target at the desired position (i.e., $X_{B}^{T} = \{0, 0, 5\}$) with respect to chaser and simplify its motion with six types of patterns, involving left, right, up, down, forward, and backward. Then, we keep chaser stationary and make the statistics of actions selected by DRLA VT to each motion pattern, which are plotted in Fig. 16. It is obvious that our method takes near-optimal actions for corresponding motion pattern. For example, when the Satellite03 only move to left, the DRLA VT agent takes the 1st action (i.e., turn left) about 63% timesteps, the sixth action (i.e., backward) about 28% timesteps, and no action in the rest of timesteps. The reason why the chaser sometimes chooses to go backward is due to the wrong inference of DRLA VT for the depth of target. In addition, we find out that the geometry, texture, and category of the target just have negligible influences on DRLA VT, which benefits from the multiple targets training as mentioned before. Therefore, we think our method learns the motion patterns of space noncooperative target from images indeed.

VI. OUTLOOK

DRLA VT has achieved excellent performances for tracking general space noncooperative object as shown in previous section, however, it can still be further studied and improved from different aspects in future work.

We have adopted two types of neural network architectures, ConvNet and ResNet, into our method, which demonstrates that a deeper neural network can better approximate optimal state-action value function $Q^{*}(s, a)$. Therefore, we naturally consider it is a promising work to replace the...
backbone of DRLAVT with another state-of-the-art but complex architecture, Transformer, of which advancement has been proven in plenty of computer vision tasks, such as image classification [41], object detection [42], object tracking [43], and so on. Some concurrent works [44], [45], [46] were devoted to implement DRL based on Transformer, however, experimental results remained unsatisfactory.

There is no doubt that DQN-based methods are limited for continuing control problem. We, therefore, will introduce some policy gradient methods (e.g., DDPG [25], A3C [26], and soft-AC [27]) to better handle active visual tracking task of space noncooperative objects, which can also help our method to be applied in real-world and extended to more complex applications with high-dimension action space, for example, using space robotic manipulator to capture general targets.

Moreover, we think adversarial reinforcement learning [47] is another interesting point for space noncooperative object active visual tracking, which assumes both chaser and escaper are intelligent agents and can mutually enhance each other with dueling. In a word, we believe DRLAVT has bright prospects for space debris removal, asteroid exploration, autonomous rendezvous, and docking.

And the idea of our method that only relies on raw images can also be referenced to other space activities.

VII. CONCLUSION

In this article, we provide an open-source benchmark for space noncooperative object active visual tracking, which includes the simulation environment and evaluation toolkit. We think this can significantly facilitate the progress of related research, especially for those DRL-based methods. For comparison, we also present a novel PBVS baseline algorithm that adopts state-of-the-art 2-D monocular tracker, SiamRPN with RGB-D images to predict real-time position of the target. To the best of authors knowledge, the first end-to-end active visual tracker in aerospace, DRLAVT is proposed in this article, which guides the chasing spacecraft to approach an arbitrary target by merely using color or RGB-D images. The effectiveness and robustness of DRLAVT have been proved by extensive experiments. Compared with the PBVS baseline algorithm, the current DRL-based active visual tracker still has a large room for improvement in tracking accuracy. However, benefitted from the end-to-end pipeline, that is, directly maps input image to control action, our DRLAVT algorithm achieves
excellent robustness and real-time performance. We found out that more complex neural network, reasonable architecture design, and effective reward function can improve the active tracking performance of an agent. At final, we prove that the transferability of DRLAVT can be guaranteed by multiple targets training that forces the agent to learn an optimal control policy with respect to the object motion patterns. However, our DRLAVT remains limited with respect to continuous velocity/force control. And the tracking accuracy that DRLAVT achieved can be further improved with more complex neural network architecture. In addition, the real performance of our method in actual application is unknown. We will evaluate it with ground physical simulation on planar gravity-offset testbed.

REFERENCES

[1] P. Huang, F. Zhang, J. Cai, D. Wang, Z. Meng, and J. Guo, “Dexterous tethered space robot: Design, measurement, control, and experiment,” IEEE Trans. Aerosp. Electron. Syst., vol. 53, no. 3, pp. 1452–1468, Jun. 2017.

[2] A. Flores-Abad, O. Ma, K. Pham, and S. Ulrich, “A review of space robotics technologies for on-orbit servicing,” Prog. Aerosp. Sci., vol. 68, pp. 1–26, Jul. 2014.

[3] G. Dong and Z. H. Zhu, “Autonomous robotic capture of non-cooperative target by adaptive extended Kalman filter based visual servo,” Acta Astronautica, vol. 122, pp. 209–218, 2016.

[4] H. Li and H. Baojin, “Optimization of multiple debris removal missions using an evolving elitist club algorithm,” IEEE Trans. Aerosp. Electron. Syst., vol. 56, no. 1, pp. 773–784, Feb. 2020.

[5] J. H. Ramos, T. D. Woodbury, and J. E. Hurtado, “Vision-based tracking of non-cooperative space bodies to support active attitude control detection,” in Proc. AIAA SPACE Astronautics Forum Expo., 2018, pp. 5535–5537.

[6] D. Zhou, G. Sun, and X. Hong, “3D visual tracking framework with deep learning for asteroid exploration,” 2021, arXiv: 2111.10737.

[7] D. Fourie, B. E. Tweddle, S. Ulrich, and A. Sanz-Otero, “Flight results of vision-based navigation for autonomous spacecraft inspection of unknown objects,” J. Spacecraft Rockets, vol. 51, no. 6, pp. 2016–2026, 2014.

[8] L. Zhang, F. Zhu, and Y. Hao, “An improved real-time visual tracking method for space non-cooperative target,” in Proc. Infrared Technol. Appl. Robot Sens. Adv. Control, 2016, vol. 10157, Art. no. 1015733.

[9] D. Volkov, G. B. Palmerini, and M. Sabatini, “A passive camera based determination of a non-cooperative and unknown satellite’s pose and shape,” Acta Astronautica, vol. 151, pp. 805–817, Oct. 2018.

[10] D. Zhou, G. Sun, J. Song, and W. Yao, “2D visual-based tracking algorithm for general space non-cooperative objects,” Acta Astronautica, vol. 188, pp. 193–202, Nov. 2021.

[11] L. Sun and Z. Zheng, “Adaptive relative pose control of spacecraft with model couplings and uncertainties,” Acta Astronautica, vol. 143, pp. 29–36, 2018.

[12] J. Liu, H. Li, Y. Luo, and J. Zhang, “Robust adaptive relative position and attitude integrated control for approaching uncontrolled tumbling spacecraft,” Proc. Inst. Mech. Engineers, Part G: J. Aerosp. Eng., vol. 234, no. 2, pp. 361–374, 2020.

[13] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, “High-speed tracking with kernelized correlation filters,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 37, no. 3, pp. 583–596, Mar. 2015.

[14] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, “High performance visual tracking with siamese region proposal network,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 8971–8980.

[15] S. Levine, N. Wageren, and P. Abbeel, “Learning contact-rich manipulation skills with guided policy search,” in Proc. IEEE Int. Conf. Robot. Auton., 2015, pp. 156–163.

[16] S. Levine, C. Finn, T. Darrell, and P. Abbeel, “End-to-end training of deep visuomotor policies,” J. Mach. Learn. Res., vol. 17, no. 1, pp. 1334–1373, Jan. 2016.

[17] A. Loquercio, E. Kaufmann, R. Ranftl, A. Dosovitskiy, V. Koltun, and D. Scaramuzza, “Deep drone racing: From simulation to reality with domain randomization,” IEEE Trans. Robot., vol. 36, no. 1, pp. 1–14, Feb. 2020.

[18] V. Mnih et al., “Human-level control through deep reinforcement learning,” Nature, vol. 518, no. 7540, pp. 529–533, Feb. 2015.

[19] D. Silver et al., “Mastering the Game of Go with deep neural networks and tree search,” Nature, vol. 529, no. 7587, pp. 484–489, Jan. 2016.

[20] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, “Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection,” Int. J. Robot. Res., vol. 37, no. 4/5, pp. 421–436, 2018.

[21] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction, Cambridge, MA, USA: MIT Press, 2018.

[22] H. Van Hasselt, A. Guez, and D. Silver, “Deep reinforcement learning with double q-learning,” in Proc. AAAI Conf. Artif. Intell., 2016, vol. 30, pp. 2094–2100.

[23] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas, “Dueling network architectures for deep reinforcement learning,” in Proc. 33rd Int. Conf. Mach. Learn., 2016, pp. 1995–2003.

[24] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, “Prioritized experience replay,” in Proc. Int. Conf. Learn. Representations (Poster), 2016.

[25] T. P. Lillicrap et al., “Continuous Control With Deep Reinforcement Learning,” in Proc. Int. Conf. Learn. Representations (Poster), 2016.

[26] V. Mnih et al., “Asynchronous methods for deep reinforcement learning,” in Proc. Int. Conf. Mach. Learn., 2016, pp. 1928–1937.

[27] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, “Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor,” in Proc. Int. Conf. Mach. Learn., 2018, pp. 1861–1870.

[28] A. Harris, T. Teil, and H. Schaub, “Spacecraft decision-making autonomy using deep reinforcement learning,” in Proc. 29th AAS/AIAA Space Flight Mechanics Meeting, Hawaii, 2019, pp. 1–19.

[29] K. Hovell and S. Ulrich, “On Deep Reinforcement Learning for Spacecraft Guidance,” in Proc. AIAA Scitech 2020 Forum, 2020, pp. 1600–1619.

[30] K. Hovell and S. Ulrich, “Deep reinforcement learning for spacecraft proximity operations guidance,” J. Spacecraft Rockets, vol. 58, pp. 254–264, Jan. 2021.

[31] G. Barth-Maron et al., “Using simulation and domain adaptation to improve efficiency of deep robotic grasping,” in Proc. IEEE Int. Conf. Robot. Autom., 2018, pp. 4243–4250.

[32] N. Stünderhauf et al., “The limits and potentials of deep learning for robotics,” Int. J. Robot. Res., vol. 37, no. 4/5, pp. 405–420, Apr. 2018.

[33] X. Wang, P. Shi, C. Wen, and Y. Zhao, “Design of parameter-self-tuning controller based on reinforcement learning for tracking non-cooperative targets in space,” IEEE Trans. Aerosp. Electron. Syst., vol. 56, no. 6, pp. 4192–4208, Jun. 2020.

[34] W. Luo, P. Sun, F. Zhong, W. Liu, T. Zhang, and Y. Wang, “End-to-end active object tracking and its real-world deployment via reinforcement learning,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 42, no. 6, pp. 1317–1332, Jun. 2020.

[35] L. Huang, X. Zhao, and K. Huang, “GOT-10k: A large high-diversity benchmark for generic object tracking in the wild,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 43, no. 5, pp. 1562–1577, May 2021.

[36] Y. Wu, J. Lim, and M.-H. Yang, “Object tracking benchmark,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 37, no. 9, pp. 1834–1848, Sep. 2015.

[37] M. Kristan et al., “The seventh visual object tracking VOT2019 challenge results,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. Work., 2019, pp. 2206–2241.
[39] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” Commun. ACM, vol. 60, no. 6, pp. 84–90, 2017.
[40] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 770–778.
[41] C.-F. R. Chen, Q. Fan, and R. Panda, “Crossvit: Cross-attention multi-scale vision transformer for image classification,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 357–366.
[42] Z. Sun, S. Cao, Y. Yang, and K. M. Kitani, “Rethinking transformer-based set prediction for object detection,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 3611–3620.
[43] X. Chen, B. Yan, J. Zhu, D. Wang, X. Yang, and H. Lu, “Transformer tracking,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 8126–8135.
[44] U. Upadhyay, N. Shah, S. Ravikanti, and M. Medhe, “Transformer based reinforcement learning for games,” 2019, arxiv:1912.03918.
[45] M. Janner, Q. Li, and S. Levine, “Offline reinforcement learning as one big sequence modeling problem,” in Proc. Adv. Neural Inf. Process. Syst., 2021, pp. 1273–1286.
[46] E. Parisotto et al., “Stabilizing transformers for reinforcement learning,” in Proc. 37th Int. Conf. Mach. Learn., 2020, pp. 7487–7498.
[47] F. Zhong, P. Sun, W. Luo, T. Yan, and Y. Wang, “AD-VAT: An asymmetric dueling mechanism for learning and understanding visual active tracking,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 43, no. 5, pp. 1467–1482, May 2021.

Dong Zhou was born in Hunan, China, in 1996. He received the B.S degree in automation from Harbin Engineering University, Harbin, China, in 2018. He is currently working toward the Ph.D. degree in control science and engineering from Harbin Institute of Technology, Harbin, China. His research interests include space noncooperative object visual tracking, 3-D computer vision, and deep reinforcement learning.

Guanghui Sun (Senior Member, IEEE) was born in Henan Province, China, in 1983. He received the B.S. degree in automation and the M.S. and Ph.D. degrees in control science and engineering from the Harbin Institute of Technology, Harbin, China, in 2005, 2007, and 2010, respectively. He is currently a Professor with the Department of Control Science and Engineering, Harbin Institute of Technology. His research interests include machine learning, computer vision, and aerospace technology.

Dong Zhou was born in Hunan, China, in 1996. He received the B.S degree in automation from Harbin Engineering University, Harbin, China, in 2018. He is currently working toward the Ph.D. degree in control science and engineering from Harbin Institute of Technology, Harbin, China. His research interests include space noncooperative object visual tracking, 3-D computer vision, and deep reinforcement learning.

Wenxiao Lei received the B.S. degree in automation and M.S. degree in control science and engineering in 2016 and 2019, respectively, from the Harbin Institute of Technology, Harbin, China, where he is currently working toward the Ph.D. degree in control science and engineering from Harbin Institute of Technology, Harbin, China.

His research interests include intelligent control of robotic manipulators, robotic motion planning, noncooperative target capture.

Ligang Wu (Fellow, IEEE) received the B.S. degree in automation from the Harbin University of Science and Technology, Harbin, China, in 2001 and the M.E. degree in navigation guidance and control and the Ph.D. degree in control theory and control engineering from the Harbin Institute of Technology, Harbin, China, in 2003 and 2006, respectively. His current research interests include switched systems, stochastic systems, computational and intelligent systems, sliding-mode control, and advanced control techniques for power electronic systems. Prof. Wu was the recipient of the National Science Fund for Distinguished Young Scholars in 2015, and China Young Five Four Medal in 2016. He was named as the Distinguished Professor of Chang Jiang Scholar in 2017, and was named as the Highly Cited Researcher from 2015 to 2021. In 2019, he was elevated to the IEEE Fellow grade for contributions to sliding-mode control and robust filtering.