Novel Reward Function for Autonomous Drone Navigating in Indoor Environment

Khuong G. T. Diep[1], Viet-Tuan Le[1], Tae-Seok Kim[1], Anh H. Vo[1], Yong-Guk Kim[1]

[1]Dept. of Computer Science and Engineering, Sejong University

khuongdiep@sju.ac.kr, tuanlv@sju.ac.kr, 22110617@sju.ac.kr, vohoanganh@sju.ac.kr, ykim@sejong.ac.kr

Abstract

Unmanned aerial vehicles are gaining in popularity with the development of science and technology, and are being used for a wide range of purposes, including surveillance, rescue, delivery of goods, and data collection. In particular, the ability to avoid obstacles during navigation without human oversight is one of the essential capabilities that a drone must possess. Many works currently have solved this problem by implementing deep reinforcement learning (DRL) model. The essential core of a DRL model is reward function. Therefore, this paper proposes a new reward function with appropriate action space and employs dueling double deep Q-Networks to train a drone to navigate in indoor environment without collision.

1. Introduction

Unmanned aerial vehicles (UAVs) commonly known as drones are aircraft that can fly without any human pilot on board and can operate under remote control by a human operator, or even fully autonomous with no provision for human intervention. They were originally developed for military purposes such as gathering and delivering information from dirty and dangerous places. However, with the rapid emergence of advanced accessories and artificial intelligence, UAVs nowadays have shown their great potential in dealing with a wide range of missions from aerial photography [1] to disaster investigation [2] , from surveillance to search and rescue (SAR) [3], from delivery of goods to relaying of internet connections [4], [5]. Despite the benefits of utilizing drones in real life, their operations are still limited because several obstacles will appear on the way of the drones to complete their missions. For this reason, an obstacle avoidance algorithm should be developed to support the drones to perform effectively in hazardous situations such as the appearance of trees, obstacles, humans or narrow corridors. Therefore, the primary goal of this study is to develop a drone obstacle avoidance model using deep reinforcement learning. The main contribution in this work is explained as below:

- Developing a deep reinforcement learning algorithm for drone obstacle avoidance;
- Proposing a reward function and an appropriate action space for drone obstacle avoidance;
- Evaluating the model in some unseen environments.

2. Related work

There are many works that implement reinforcement learning to solve obstacle avoidance task. The study [6] presents a novel framework in which the U-Net is trained using labels generated by optical flow and critic networks in a reward-driven manner. To determine the direction in which the drone chose to move, the optical flow technique was employed to construct raw optical vectors between two subsequent images. The resulting vectors were then sent into the critic network in order to create the label map for training the U-Net model. Finally, the U-Net model output was sent into the actor network to create control commands. Several policy gradient algorithms using the continuous action space have been tested in this work. As a result, ACKTR has shown the best performance among different algorithms. The proposed framework also showed great performance in both the trained environment and the unseen environments. Although the authors successfully tested the proposed model in real life environment, the drone is limited to moving in three linear axes and cannot change direction, resulting in an unusual behavior.

Another application of DRL algorithm was studied in [7], where drone operating in the indoor environments. This paper focused on training a drone to autonomously avoid obstacles in continuous action space using the soft-actor-critic algorithm (SAC). The goal was to train the UAV to select a correct and smooth action using only image data. The depth maps were used as input and SAC was paired with a variational autoencoder (VAE) to train the UAV to accomplish obstacle avoidance tasks in a simulation environment with various wall constraints. However, the reward function in this work is not general. It only can be used for an environment that has a hole on the wall and outputs of the actor network are the velocities in y and z direction.

3. Methodology

The overall concept of the algorithm is showed in this

section. The obstacle avoidance (OA) algorithm contains two phases. Firstly, the depth image is provided by Airsim simulator [8]. Then, stacking a sequence of depth images to consider the temporal information during navigation. Finally, the consecutive depth images are passed through the D3QN model, which output the control command. The available action space for drone is split into two categories. The linear x-velocity has 2 candidates: [0.8, 0.4] and the angular velocity has 5 actions: [±0.5, ±0.25, 0]. The zero angular velocity in action space decreases the chance of collision in the complex environments. Fig. 1 shows the flow diagram of the obstacle avoidance system.

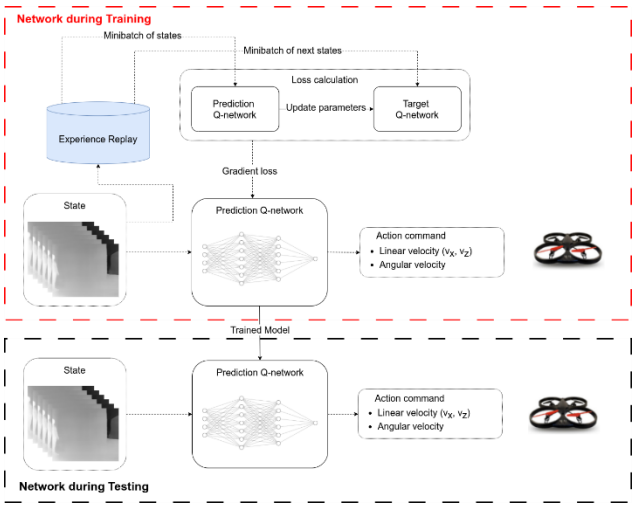

Figure 1. Diagram of training and testing for obstacle avoidance algorithm

2.1 Q Learning algorithm

The Deep Q-Network algorithm (DQN) has demonstrated exceptional performance in handling complicated problems. Deep Q-Learning, crucially, replaces the regular Q-table with a neural network. A neural network maps input states to (action, Q-value) pairs rather than mapping a state-action pair to a q-value. The DQN contains two value function: an online network and a target network. For the update process. The online network is updated at each iteration, whereas the target network is fixed for a set number of iterations before being modified. Because the target network is fixed, the online network may undergo consistent training. The target network function is expressed as below:

$$Q_{s,a} = r(s,a) + \gamma \sum_{s' \in S} P_{ss'} V_{s'} = r(s,a) + \gamma \max_a Q'(s',a')$$

However, according to reference [9], the original DQN tends to overestimate Q values, it might affect negatively to the training process. An algorithm called Double DQN was introduced to overcome this problem. Double DQN shows a great improvement upon DQN, especially, on the Atari 2600 domain. the expression of double Q-function:

$$Q_{s,a}^{Double} = r(s,a) + \gamma \max_a Q'(s', \arg\max_a Q(s',a))$$

There is another enhanced algorithm called Dueling DQN. B y dividing the network into two different streams, the dueling architecture can learn which states are or are not useful, without having to know the effect of each action at each time step. This algorithm computes a new quantity called the advantage A(s,a), which is the subtraction between Q-value and state-value:

$$A(s,a) = Q(s,a) - V(s)$$

To take advantage of the Double DQN and Dueling DQN methods, a hybrid model known as Double Dueling DQN (D3QN) was developed. This algorithm outperformed other techniques in the DQN variants, especially in discrete action space problems. The target Q-value is shown in equation below:

$$Q_{s,a}^{D3QN} = r(s',a') + \gamma Q(s', \arg\max Q(s',a;\theta,\alpha,\beta))$$

When compared with other DQN variants. D3QN shows a greater performance over 65% problems in the paper [10], [11]. Therefore, this paper adopted the D3QN for our algorithm.

2.2 Reward Function

Designing a well reward function is a crucial task in reinforcement learning algorithm. In navigation, choosing to move straight as much as possible is an ideal movement for drone to save battery. Therefore, a reward function that help the drone avoiding obstacles is developed to resolve this problem.

In this algorithm, the reward function is divided into three cases to cover the entire surrounding environment such as High-speed reward function, Turning reward function, Collision reward function. The minimum distance is calculated by finding the minimum distance values from the nearest obstacle to the drone. This distance is measured by Lidar sensor in Airsim. If the distance from the closest obstacle is larger than 1.5m, the obstacle is in the High-speed zone and vice versa. The ideal motion of the drone is to move straight as much as possible to reduce the consumption of battery. Therefore, it is vital to define an efficient reward that force the drone moving forward. The proposed reward function is defined as below:

$$R = \begin{cases} |\theta| - v_x & \min_{dis} < 1.5 \\ v_x - |\theta| & \min_{dis} \geq 1.5 \\ -10 & collision \end{cases}$$

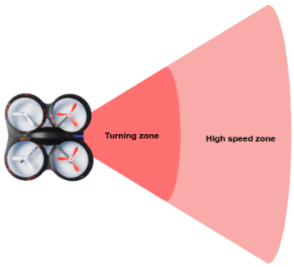

Figure 2. Illustration of Turning zone and High-speed zone

Where v_x is the linear speed in x direction and θ is the angular velocity. When the nearest obstacle is in the High-speed zone, the drone should choose to move straight forward with high speed and should not choose to turn in order to maximize the reward. On the other hand, when the obstacle is in the Turning zone, drone should turn to receive better reward and to avoid colliding with obstacles.

4. Experiments and results

4.1 Training settings

Figure 3. The designed training environment

A complex training map containing different kind of objects such as cylinders, walls, pillars and people is created. The drone in training process is initialized at the center of the map with random orientation. A training episode is stopped when the drone collides with obstacle or the number of steps is more than 500.

4.2 Experimental results

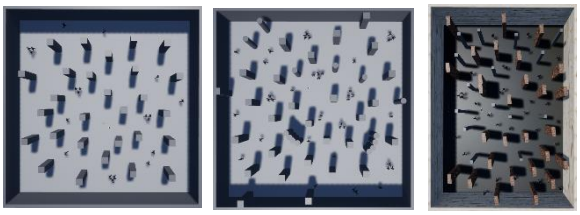

Figure 4. Testing maps for evaluation. From left to right are map a, b and c

To test the performance of the OA model, three new environments are created as shown in Fig. 4. The drone was tested with four different environments. The first environment is the environment that was trained and the other three ones are unseen environments. Each configuration is tested with 20 trials. An episode is considered to be successful when the drone can navigate more than 500 steps. The experimental results are described in Table 1.

Table 1. Experimental results of 20 trials for each testing map

Environment	Map a	Map b	Map c
Complete/Total	19/20	17/20	11/20
Average steps	490	472	370

As shown in Table 1, the trained drone completed 19 episodes out of 20 with the simple configuration (map a). The successful rate on the second map (map b) is lower than in map a, but it is still relatively good, with 17 successful trials. Finally, in the most challenging map (map c), the drone only completes 11 of 20 trials. However, while the success ratio in the last configuration is low, the average number of steps is 370, indicating that the drone can still avoid most of the obstacles on the map.

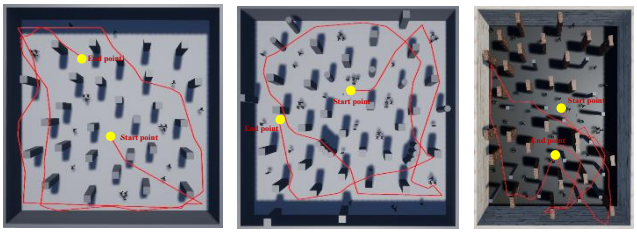

Figure 5. Examples of trajectories during testing for each map. From left to right are map a, b and c

5. Conclusion and future work

In summary, this paper has proposed a new reward function with an appropriate action space that help the drone safely navigate without collision. The obstacle avoidance performance of the model is evaluated by three unseen environments and the results show that the drone has a great ability to avoid obstacle even in the untrained environments. Future work should compare the performance with state-of-the-arts model and consider the ability to move up and down in order to use up all the drone's degrees of freedom.

References

[1] A. Puttock, A. Cunliffe, K. Anderson, and R. Brazier, "Aerial photography collected with a multirotor drone reveals impact of eurasian beaver reintroduction on ecosystem structure," Journal of Unmanned Vehicle Systems, vol. 3, no. 3, pp. 123–130, 2015. DOI: 10.1139/juvs-2015-0005.eprint: https://doi.org/10.1139/juvs-2015-0005.[Online].Available: https://doi.org/10.1139/juvs-2015-0005.

[2] S. M. S. Mohd Daud, M. Y. P. Mohd Yusof, C. C. Heo, et al., "Applications of drone in disaster management: A scoping review," Science Justice, vol. 62, no. 1, pp. 30–42, 2022, ISSN: 1355-0306. DOI: https://doi.org/10.1016/j.scijus.2021.11.002. [Online]. Available: https : / / www. sciencedirect . com / science / article / pii /S1355030621001477

[3] B. Mishra, D. Garg, P. Narang, and V. Mishra, "Drone-surveillance for search and rescue in natural disaster," Co

mputer Communications, vol. 156, pp. 1–10, 2020, ISSN: 0140-3664. DOI: https : / / doi . org / 10 . 1016 / j .comcom . 2020 . 03 . 012. [Online]. Available: https :/ / www . scienc edirect . com / science / article / pii / S0140366419318602.

[4] A. T. Azar, A. Koubaa, N. Ali Mohamed, et al., "Drone deep reinforcement learning: A review," Electronics, vol. 10, no. 9, 2021, ISSN: 2079-9292. DOI: 10.3390/electronics10090999. [Online]. Available: https://www.mdpi.com/2079-9292/10/9/999.

[5] A. Devo, J. Mao, G. Costante, and G. Loianno, "Autonomous single-image drone exploration with deep reinforcement learning and mixed reality," IEEE Robotics and Automation Letters, vol. 7, no. 2, pp. 5031–5038, 2022. DOI: 10 . 1109 / LRA . 2022 .3154019.

[6] S.-Y. Shin, Y.-W. Kang, and Y.-G. Kim, "Reward-driven u-net training for obstacle avoidance drone," Expert Systems with Applications, vol. 143, p. 113 064, 2020, ISSN: 0957-4174. DOI: https : / / doi . org / 10 . 1016 / j . eswa . 2019 . 113064. [Online]. Available: https : / / www. sciencedirect . com / science / article / pii / S095741741930781X.

[7] Z. Xue and T. Gonsalves, "Vision based drone obstacle avoidance by deep reinforcement learning," AI, vol. 2, no. 3, pp. 366–380, 2021, issn: 2673-2688. doi: 10.3390/ai2030023. [Online]. Available: https://www.mdpi.com/2673-2688/2/3/23.

[8] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in Field and Service Robotics, M. Hutter and R. Siegwart, Eds., Cham: Springer International Publishing, 2018, pp. 621–635, isbn: 978-3-319-67361-5.

[9] ZM. Lapan, Deep Reinforcement Learning Hands-On, Apply Modern RL Methods, with Deep Q-Networks, Value Iteration, Policy Gradients, TRPO, AlphaGo Zero and more. Birmingham, UK: Packt Publishing, 2018, 546 pp., ISBN: 978-1-78883-424-7.

[10] Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., et al. (2016).Asynchronous methods for de ep reinforcement learning. In International conferenceon machine learning (pp. 1928–1937)

[11] Wang, Z., Schaul, T., Hessel, M., Van Hasselt, H., Lan ct ot, M., & De Freitas, N. (2015).Dueling network architect ures for deep reinforcement learning. arXiv preprintarXiv :1511.06581.