Gimbal Camera-Based Flight Guidance for Striking Illegal Drones

Hojun Lee[1] and Kyuman Lee[2]

[1] School of Computer Science and Engineering, Kyungpook National University, Daegu, 41566, Korea
[2] School of Space Engineering Sciences, Kyungpook National University, Daegu, 41566, Korea
{`hojunlee66, klee400`}`@knu.ac.kr`,
`https://knurobot.wixsite.com/arrf`

Abstract. As the illegal use of drones has emerged as a social issue, drone interception technology to directly strike a target has rapidly developed. However, existing methods using a fixed camera are often too slow to successfully strike the target, or may miss the target entirely. To address this issue, this paper proposes an interceptor flight guidance algorithm that tracks and strikes drones using a gimbal camera. First, artificial intelligence techniques detect the target on a real-time video stream from a gimbal camera. Next, the gimbal tracks the detected target so that it is centered in the image. Lastly, based on the pitch and yaw angles of the gimbal, the desired linear and angular velocity vector commands of the interceptor are generated to track and strike the target. To validate the proposed method's performance, simulation environments based on a physics engine are established. We test the proposed method in two scenarios with different starting distances between the interceptor and the drone. At a far distance from the target, the proposed method strikes 4.4 seconds faster than the existing method. Moreover, when the distance is too close and the field of view is too limited, methods using a fixed camera fail to strike the target, whereas our method has a greater than 80% success rate. These results demonstrate that the proposed method strikes targets more efficiently and successfully.

Keywords: Anti-Drone, Flight Guidance, Target Tracking

1 Introduction

Recent events such as the 2022 Ukraine war have underscored the active deployment and significance of drones in warfare [1, 2]. Particularly noteworthy are self-destructing drones armed with explosives, which are used to target and destroy objectives through methods involving throwing and collisions. These drones not only pose a military threat, but may also be used in terrorist and other malicious activities. Their destructive capabilities can lead to safety and security risks by enabling attacks on civilian facilities and social infrastructure.

Consequently, the development of anti-drone technology has become increasingly important.

The methods for striking these illegal drones can be broadly categorized into “soft kill” and ”hard kill” methods. Soft kill methods aim to render drones inoperable without physically destroying them, typically by interfering with radio signals through techniques such as jamming and spoofing. However, soft kill methods require precise drone identification and manipulation, leading to varying effectiveness based on factors like drone size, shape, and speed. On the other hand, hard kill methods involve destroying drone targets. Unlike soft kill methods which primarily disrupt target flight through radio interference, hard kill methods include deployment for direct interception and technologies such as capture nets, laser beams, or shotguns. These approaches provide more aggressive means of neutralizing drone threats, albeit with potential risks and legal considerations. In this paper, we propose a hard kill method utilizing interceptor drones equipped with a gimbal camera.

2 Related Work

Due to military secrecy, literature on interceptor drones is limited. However, information on similar methods can be found in competitions where non-threatening targets (such as balls) are identified and struck. Additionally, existing methodology for tracking human targets can also be applied to striking illegal drones. Beul, Marius, et al.[3] proposed an algorithm for locating and striking aerial targets in the MBZIRC 2020 challenge using CNNs (Convolutional Neural Networks) to detect and strike moving targets using the information of their size and color. A Rohan, et al.[4] suggested a method for detecting and tracking moving or stationary targets using drones. The CNNs were used for target detection and utilized a PID (Proportion-Integral-Derivative) drone controller to keep the target in the center of the image during tracking. Zhao, Moju, et al.[5] presented an algorithm for detecting and striking moving aerial targets in the MBZIRC 2020 challenge. They used MobileNet V2 for detection and applied HLS (Hue-Luminance-Saturation) color filters to the detected target regions, leveraging pre-existing information. They estimated the target’s position using a depth camera and used this information to strike the target. Anastasiou, et al.[6] proposed a robust detection and tracking algorithm for UAS (Unmanned Aircraft Systems) using computer vision algorithms and a combination of PID controllers to track moving vehicles. Karras, et al.[7] proposed an image-based visual servo control scheme to track moving targets using multi-rotor drones. Liu, Yisha, et al.[8] proposed an algorithm for tracking moving targets from the ground using a gimbal camera. Liu, Xuancen, et al.[9] tracked targets in video footage based on the KCF tracker and utilized a proportional navigation method. Bartak, Roman, and Adam Vykovský[10] introduced the FollowMe mode for tracking human targets using an AR.Drone, which tracked arbitrary user-selected targets in a video stream and controlled the drone using a PID controller.

3 Methodology

This section describes the method of recognizing targets within a video and calculating the distance between the recognized target and the center of the video. It explains the method of target interception using a fixed camera and our proposal to improve target interception method using a gimbal camera that to addresses the drawbacks of the method. In this paper, ”the term “target” will refer to illegal drones.

3.1 Illegal Drone Detection

YOLO(You Only Look Once)[11] is a real-time object detection system that identifies objects in images or videos in a single pass. In the context of YOLO, ”single pass” refers to the process of analyzing an image to detect objects in one evaluation. Unlike other detection systems that scan an image multiple times at different scales or locations, YOLO divides the image into a grid and simultaneously predicts bounding boxes and class probabilities for each grid cell. This approach allows YOLO to achieve high speed and performance in real-time target detection due to the single pass required through the neural network to detect objects without repeated processing or complex pipelines. For targets detected via AI from images captured by an interceptor-mounted camera, the system calculates the pixel distance between the coordinates of the target and the central coordinates of the image.

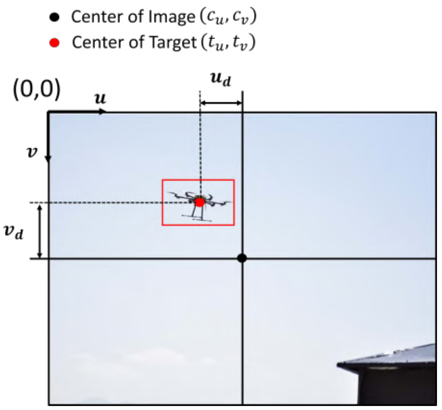


Figure. 1. Pixel distance between target and image centers.

First, the distance between the center of the target and the center of the image is calculated as shown in Fig. 1 Using AI-based object detection algorithms, the area of the detected object is represented by the red bounding box. The black circle represents the center of the image, while the red circle represents the center of the target. The center coordinates of the target are represented as $(t_u,\, t_v)$,and the center coordinates of the image are represented as $(c_u,\, c_v)$. Then, in the 2D pixel coordinate system, we calculate the distance along the u-axis and v-axis between the center of the image (black circle) and the center of the target (red circle) in pixel units.

$$\begin{aligned} u_d &= c_u - t_u \\ v_d &= c_v - t_v \end{aligned} \tag{1}$$

Eq.(1) represents the formula for calculating pixel distance along each axis, where u_d represents the distance along the u-axis between the center of the image and the center of the target and v_d along the v-axis.

3.2 Striking Using a Fixed Camera

This section explains the methodology for tracking and striking targets using a fixed camera attached to an interceptor.

In this method, it is assumed that the camera is attached in the direction of the interceptor heading. First, the pixel distances u_d and v_d between the target and the image center for each frame are calculated. Then, using the calculated u_d and v_d, a velocity vector for tracking and striking the target is created.

$$\vec{V}_{f,\mathrm{linear}} = \left[V_{f,x}\ V_{f,y}\ V_{f,z}\right]^\mathrm{T}, \tag{2}$$

where

$$\begin{aligned} V_{f,x} &= V_\mathrm{max} \\ V_{f,y} &= 0.0 \\ V_{f,z} &= K_{p,fz}\, v_d + K_{i,fz} \int v_d\, dt + K_{d,fz}\, \frac{dv_d}{dt} \end{aligned} \tag{3}$$

In Eq.(2), $\vec{V}_{f,\mathrm{linear}}$ represents a vector with three components, where $V_{f,x}$ represents the drone's forward velocity, $V_{f,y}$ represents lateral velocity, and $V_{f,z}$ represents vertical velocity. In addition, V_max represents the maximum speed of the drone, which can be changed to the speed desired to track and strike.

First, the interceptor forward speed($V_{f,x}$) is set to V_max. To to fly at the same height as the detected target, a PID controller is designed using v_d as the error variable, with its output represented as $V_{f,z}$. This means the drone moves forward at its maximum speed while being guided to fly at the same height as the target. For example, if the coordinates of a detected target within a 640x480 pixel image are (200, 150), it can be determined that the target is at a higher altitude than the interceptor. This means that by using only Equation 2, the height is increased while flying forward at maximum speed, utilizing the distance along the v-axis in pixels(denoted as v_d), between the detected target in the image and the center of the image. In this method, $V_{f,y}$ is zero because the heading of the interceptor rotates toward the target using Eq.(4).

$$\vec{V}_{f,\mathrm{angular}} = \left[p_f\ q_f\ r_f\right]^\mathrm{T}, \tag{4}$$

where

$$\begin{aligned} p_f &= 0 \\ q_f &= 0 \\ r_f &= K_{p,fr}\, u_d + K_{i,fr} \int u_d\, dt + K_{d,fr}\, \frac{du_d}{dt} \end{aligned} \tag{5}$$

$\vec{V}_{f,\text{angular}}$ is the angular velocity vector. p_f represents the angular roll velocity, q_f represents the angular pitch velocity, and r_f represents the angular yaw velocity. The term u_d refers to whether the detected drone in the image is located to the left or right of the interceptor. Therefore, the PID controller was designed using u_d as an error value so that the heading of the interceptor faces the target, and then allocates the result value of the controller to r as shown in Eq.(5). All PID gain values used in this equation are user-defined. This is explained in Algorithm 1. In summary, by sending the velocity vector generated using Eqs.(2) and (4) as a command to the Flight Control Unit (FCU), the interceptor's heading will rotate towards the target. The FCU is an important component for drone navigation and control that interprets these commands to adjust the drone's orientation and velocity, ensuring that it maintains the same height as the target while striking it at maximum speed.

Algorithm 1 Striking Algorithm Using a Fixed Camera

```
Require: $V_{\max}$                                   ▷ Interceptor's Maximum Speed
Require: $c_u, c_v$                                   ▷ Center of Target coordinate
Require: $K_{p,fz}, K_{i,fz}, K_{d,fz}$
Require: $K_{p,fr}, K_{i,fr}, K_{d,fr}$
Ensure: $\vec{V}_{f,\text{linear}}$, $\vec{V}_{f,\text{angular}}$
procedure FLIGHT_GUIDANCE_ALGORITHM
    while strike do
        Calculate $u_d$, $v_d$ by Eq.(1)
        Calculate $\vec{V}_{\text{linear}}, \vec{V}_{\text{angular}}$ by Eq.(2) and (4)
        UPDATE $\vec{V}_{\text{linear}}, \vec{V}_{\text{angular}}$ to FCU
    end while
end procedure
```

3.3 Striking Using a Gimbal Camera

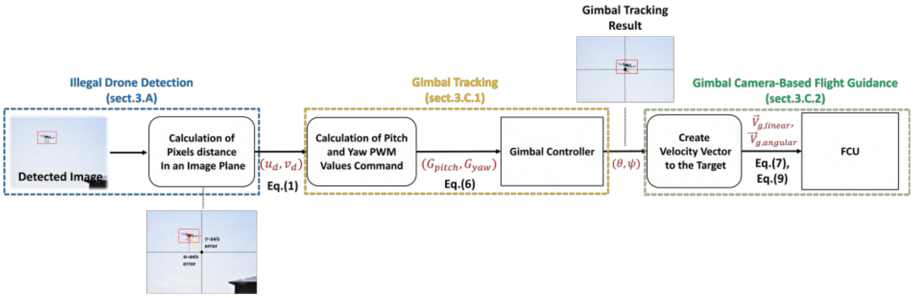


Figure. 2. System overview of the proposed method.

Gimbal Tracking The tracking algorithm aims to control a gimbal camera so that the target's center aligns with the center of the image plane. To achieve this, the algorithm uses PID control to adjust the gimbal's pitch and yaw angles. Through this system, the angles can be dynamically and precisely regulated based on the deviation of the target from the center of the image. This method ensures smooth and accurate tracking of the target, compensating for any target movements or changes in the platform's orientation, thereby maintaining continuous surveillance. First, the values of u_{error} and v_{error} are calculated to generate PWM (Pulse Width Modulation) output for gimbal control. To generate these values, a PID controller is designed as described in Eq.(6):

$$\begin{aligned} G_{\text{pitch}} &= K_{p,\text{pitch}}\, v_d + K_{i,\text{pitch}} \int v_d\, dt + K_{d,\text{pitch}} \frac{dv_d}{dt} \\ G_{\text{yaw}} &= K_{p,\text{yaw}}\, u_d + K_{i,\text{yaw}} \int u_d\, dt + K_{d,\text{yaw}} \frac{du_d}{dt} \end{aligned} \tag{6}$$

The PID controller is used to continuously track the target detected on the 2D image plane. u_d and v_d, are calculated based on the 2D-pixel coordinate system and used as inputs to the PID controller. G_{pitch} represents the output for pitch control of the gimbal, while G_{yaw} corresponds to the output for yaw control. Each PID controller configured in this manner is used to determine the PWM output for gimbal control. The output is then transmitted to the gimbal controller to track the detected target in real-time. This approach ensures that the gimbal can adjust its orientation swiftly and accurately, compensating for any dynamic changes in target position or platform movement.

Gimbal Camera-Based Flight Guidance We describe the flight guidance algorithm based on gimbal camera Information to track and strike the target.

Fig. 2 represents the flowchart of the flight guidance algorithm using gimbal camera. The Pitch angle of the gimbal is defined as theta (θ), and the Yaw angle is defined as psi (ψ), as shown in Fig. 3.

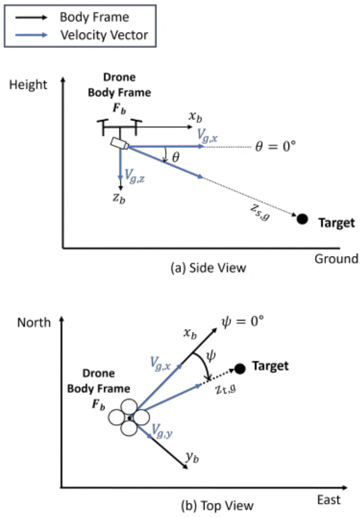


Figure. 3. Side View and Top View: Generated Velocity Vector

The drone's body is defined in the body coordinate system F_b, with the drone's heading direction represented as x_b, the lateral direction as y_b, and the vertical direction as z_b. Additionally, $z_{s,g}$ and $z_{t,g}$ represent the optical axis of the gimbal camera. In Fig. 3, θ is defined as the angle from x_b to $z_{s,g}$, allowing us to estimate the relative distance between the drone and the target. For example, when θ is greater than zero, the target is at a lower distance than the drone. Furthermore, in Fig. 3, ψ represents the relative direction to the target with respect to the drone's heading. For instance, when ψ is greater than zero, the target is located to the right of the drone's heading. In addition, based on these measurements, we can make informed decisions about the relative positioning of the drone and the target. This information serves as the foundation for generating guidance commands to effectively track and strike the target. The resulting striking algorithm operates based on the current pitch and yaw angles of the gimbal. First, the pitch and yaw angles of the gimbal are extracted through the gimbal controller. This provides the real-time angle of the gimbal camera, which is essential for the next step. After the pitch and yaw angle are extracted, a velocity command for target striking is generated. The velocity command for striking a target using the gimbal's pitch and yaw is described in Eqs.(7) and (8):

$$\vec{V}_{g,\text{linear}} = \left[V_{g,x} \; V_{g,y} \; V_{g,z}\right]^{\mathrm{T}}, \tag{7}$$

where

$$\begin{aligned} V_{g,x} &= V_{\max} \cdot \cos(\theta) \cdot \cos(\psi) \\ V_{g,y} &= V_{\max} \cdot \cos(\theta) \cdot \sin(\psi) \\ V_{g,z} &= V_{\max} \cdot \sin(\theta) \end{aligned} \tag{8}$$

In Eq.(7), $\vec{V}_{g,\text{linear}}$ represents a vector with three components in which $V_{g,x}$ represents the drone's forward velocity, $V_{g,y}$ represents lateral velocity, $V_{g,z}$ represents vertical velocity, and $V_{\max}$ represents the maximum velocity of the drone. This vector is utilized for controlling the linear motion of the drone.

$$\vec{V}_{g,\text{angular}} = \begin{bmatrix} p_g \ q_g \ r_g \end{bmatrix}^{\mathrm{T}}, \tag{9}$$

where

$$\begin{aligned} p_g &= 0 \\ q_g &= 0 \\ r_g &= K_{p,gr}\,\psi + K_{i,gr} \int \psi\, dt + K_{d,gr}\, \frac{d\psi}{dt} \end{aligned} \tag{10}$$

Eq.(9) is formulated as an angular velocity vector, consisting of the angular roll velocity denoted as p_g, the angular pitch velocity represented by q_g, and the angular yaw velocity indicated by r_g. Eq.(10) is designed to incorporate a PID controller that adjusts the heading of the interceptor to control the gimbal's yaw angle, ψ, to converge to 0. This controller is responsible for aligning the heading of the interceptor with the target. In Fig. 3, blue lines present the results of the generated velocity vectors $\vec{V}_{g,\text{linear}}$ and $\vec{V}_{g,\text{angular}}$, while the overall algorithm is summarized in Algorithm 2.

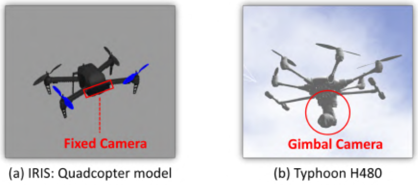
Algorithm 2 Target Striking Using a Gimbal Camera

```
Require: $V_{\max}$                                    ▷ Interceptor Maximum Speed
Require: $c_u, c_v$                                    ▷ Center of Target coordinate
Require: $K_{p,pitch}, K_{i,pitch}, K_{d,pitch}$
Require: $K_{p,yaw}, K_{i,yaw}, K_{d,yaw}$
Require: $K_{p,gr}, K_{i,gr}, K_{d,gr}$
Ensure: $G_{\text{pitch}}, G_{\text{yaw}}$             ▷ PWM Output for gimbal control
Ensure: $\vec{V}_{g,\text{linear}}, \vec{V}_{g,\text{angular}}$
procedure GUIDANCE_ALGORITHM
    while strike do
        Calculate $u_d$, $v_d$ by Eq.(1)
        Calculate $G_{\text{pitch}}, G_{\text{yaw}}$ by Eq.6
        UPDATE $G_{\text{pitch}}, G_{\text{yaw}}$      ▷ Send values to gimbal controller
        READ $\theta, \psi$                            ▷ Read from gimbal controller
        Calculate $\vec{V}_{g,\text{linear}}, \vec{V}_{g,\text{angular}}$ by Eq.(7),Eq.(9)
        UPDATE $\vec{V}_{g,\text{linear}}, \vec{V}_{g,\text{angular}}$      ▷ Send command to FCU
    end while
end procedure
```

4 Experiments and Results

In this section, we describe the experimental methods within the simulation environment, including the designs of two scenarios and the results obtained. For each experimental scenario, the results of two target strike algorithms are compared and analyzed: one using a fixed camera and the other as our proposed method using a gimbal camera. This comparison is intended to verify that the gimbal camera striking algorithm is faster and more accurate than the traditional fixed camera approach.

4.1 Experimental Setup



(a) IRIS: Quadcopter model (b) Typhoon H480

Figure. 4. Experimental drones in the GAZEBO environment.

The proposed method is divided into two primary modules: gimbal tracking and target striking. These modules were developed in Python 3. For target detection, we used the YOLOv8[11]. Subsequently, to track the detected target, gimbal control is executed through MAVSDK within the gimbal tracking module. The final stage involves the implementation of the target tracking and striking algorithm, which was performed utilizing the ROS noetic alongside the MAVROS package. For the Flight Control Unit (FCU) firmware, we used PX4 to enhance our system's capabilities by leveraging its advanced flight control algorithms and broad hardware support. We used GAZEBO to validate the performance of the algorithms for tracking and striking targets. As shown in Fig. 4, we used IRIS(a) for the interceptor model using a fixed camera and Typhoon(b) for the interceptor model using a gimbal camera. The experimental targets also used Typhoon(b).

4.2 Simulation Results

The experimental scenario is shown in Fig. 5. Initially, the interceptor hovered at a height of 50m and waited for the target to be detected in the image. At the same time, the target crossed the front of the interceptor (shown as a red line in Fig. 5) at a height of 50m at a speed of 2.5 m/sec. The experimental case is divided into two scenarios based on the distance between the target and the interceptor to demonstrate the distinct effectiveness of the strike method using a gimbal camera. In the first scenario, this distance (d) was set at 12m, and in the second, the distance was set at 6m. When the target was detected in the camera image attached to the interceptor, the interceptor attempted to strike using each algorithm described in Chapter 3. Each striking algorithm was tested for five trials in each scenario, with analysis focusing on the time taken to strike the target, the striking accuracy, and the success rate.

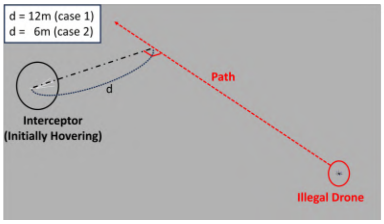


Figure. 5. Experimental scenario overview of Case 1 (distance d = 12 m), and Case 2 (d = 6 m). The red line represents the target flight path and the black line represents the interceptor path.

case 1(distance 12 m) Case 1 is defined by the scenario in which the distance between the interceptor and the target is 12m. The interceptor's velocity was 3.0 m/sec and the target velocity was 2.5 m/sec, with both operating at the same altitude. Fig. 6. shows the results of striking the target using a gimbal camera versus using a fixed camera.

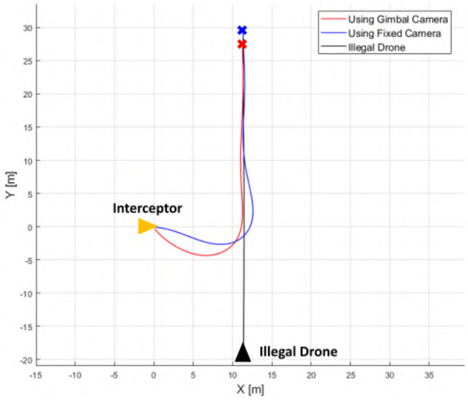
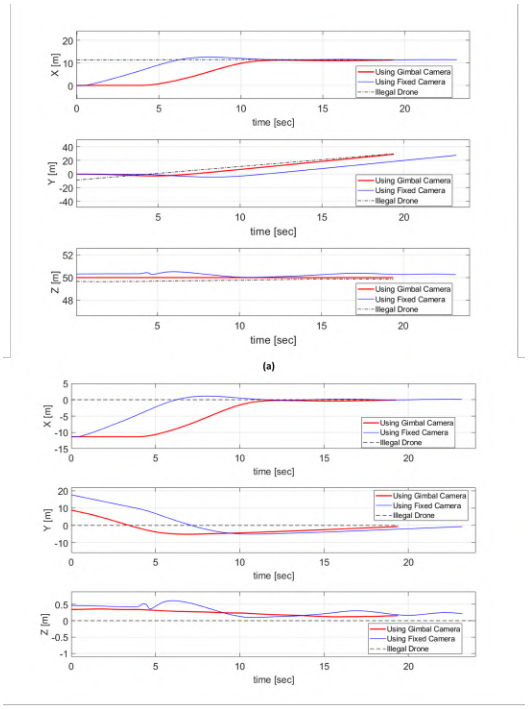


Figure. 6. The results of Case 1: Trajectories of the Interceptor with gimbal camera(red), Interceptor with Fixed Camera And Illegal Drone(black).

In Fig 6, the yellow triangle represents the interceptor, and the black triangle represents the target. The red line illustrates the path to the target using a gimbal camera, while the blue line shows the path to the target using a fixed camera. The red 'X' signifies the point at which the target was struck using gimbal camera guidance, and the blue 'X' represents the target strike point using fixed camera guidance. In this experiment, since the interceptor's speed is 0.5 m/sec faster than the target, the target can be struck by both methods when the target is detected within the image.



(a)

(b)

Figure. 7. The Position of Interceptor in Case 1 : (a) x,y,z position of the Interceptor; (b) x,y,z position distance between interceptor and Illegal Drone.

In Fig 7, the experimental results can be confirmed in more detail. Upon analyzing the results, it is found that striking the illegal drone using a gimbal camera is 3.7 seconds faster than using a fixed camera. The reason for this is that the method using a fixed camera requires rotating the heading of the interceptor to position the illegal drone at the center of the image. In Fig 7(a), which shows the change in the Interceptor's position during the process of striking the illegal drone, the method using a gimbal camera (represented by the red line) converges to the illegal drone's x position(black line). Similarly, Fig 7(b), which illustrates the distance between the Interceptor and the illegal drone in terms of x, y z positions, also confirms the convergence of the x position value to zero. In summary, striking the illegal drone using a fixed camera, which requires rotating the Interceptor's heading, results in slight overshooting and thus a slower strike compared to the method using a gimbal camera. Finally, TABLE 1 shows that there's hardly any difference in the distance between the interceptor and the illegal drone at the time of the strike, with a tiny gap of just 0.001m, showing they were almost exactly at the same results.

Table 1: Case 1 results, striking time and striking distance.

Method	Striking Time[s]	Accuracy[m]
Using Gimbal Camera	19.4	0.674
Using Fixed Camera	23.1	0.675

case 2(distance 6 m) Case 2 is the scenario in which the distance between the interceptor and the target was 6m. The difference from Case 1 is that as the distance between the interceptor and the target decreases, the target is perceived to be larger than Case 1 in the image from the interceptor camera. Although the target moves at the same speed, the target appears to move faster in the closer (6m) image. Case 2 is designed to validate that striking the target using gimbal camera guidance provides a more reliable method than a fixed camera for striking targets.

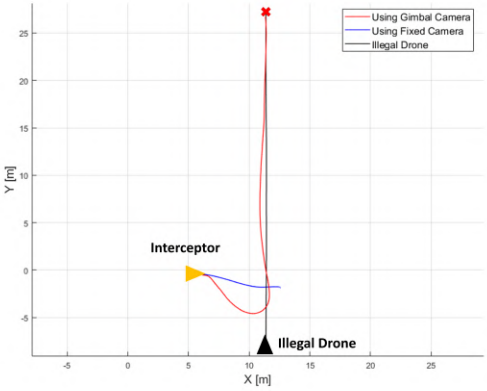


Figure. 8. The results of Case 2: Trajectories of the Interceptor with Gimbal Camera(red), Interceptor with Fixed Camera And Illegal Drone(black).

Fig. 8 shows the results of striking a target using both gimbal and fixed cameras. The meaning of each shape, line, and color used in this figure is consistent with those in Case 1. The gimbal camera method (red line) successfully struck the target in 19.22 seconds, whereas the fixed camera method (blue line) failed to strike. This occurred because with a fixed camera, the interceptor heading must be rotated to center the target in the camera image. In contrast, when using a gimbal camera, this rotation is not required; the gimbal camera's independent control structure allows for the target to be centered in the image without altering the interceptor's orientation. This results in a more agile response, improving the tracking and striking success rate. In addition, the distance between the interceptor and the target at the moment of the strike was 0.675m, confirming that the target was struck accurately.

Table 2: Case 2 results, striking time and striking distance.

Method	Striking Time[sec]	Distance[m]
Using Gimbal Camera	19.22	0.675
Using Fixed Camera	n/a	n/a

We performed five trials for each case, with the results shown in Table 3. First, in Case 1, when a gimbal camera was used to assist the target strike, the target was struck in an average of 19.8 seconds, 4.4 seconds faster than with a fixed camera. Additionally, upon striking, the distance between the interceptor and the target was 0.01 meters less when striking with the gimbal camera. Due to the interceptor's speed being faster than the target, both methods successfully achieved a strike in Case 1. In Case 2 using a gimbal camera, the target was struck in 20.1 seconds with a success rate of 80%. However, the method using a fixed camera failed to strike the target, since when the target detected in the image moves faster than the speed of the interceptor's rotation, tracking fails and the target is missed.

Table 3: The summary of experimental results.

	Method	Case	
		Case 1	Case 2
Time [sec]	Using Gimbal Camera	19.8	20.1
	Using Fixed Camera	24.2	n/a
Distance [m]	Using Gimbal Camera	0.67	0.67
	Using Fixed Camera	0.68	n/a
Success Rate [%]	Using Gimbal Camera	100%	80%
	Using Fixed Camera	100%	0%

5 Conclusion

In this paper, we proposed a flight guidance algorithm that detects, tracks, and strikes targets using a gimbal camera using the YOLOv8 system for target detection. In addition, we designed a gimbal camera-based tracker that follows targets detected in camera images. We also designed a flight guidance algorithm to strike targets based on the pitch and yaw angles of the gimbal camera. The algorithm was constructed based on ROS, and its performance was verified on GAZEBO. We performed a comparative analysis between the proposed method and the algorithm using a fixed camera for striking targets in two scenarios. In Case 1 (the first scenario), the proposed gimbal camera-based method struck 4.4 seconds faster than the fixed camera method and all strike attempts were successful. In Case 2, the proposed gimbal method successfully struck the target in 20.1 seconds. The proposed tracking method achieved an 80% success rate in striking the target, while the fixed camera method failed to strike in any of the attempts. In future research, we aim to improve the performance of the proposed method and implement the application of a zoom function supported by gimbal cameras to track and strike targets at longer distances.

6 Acknowledgement

Following are results of a study on the "Leaders in INdustry-university Cooperation 3.0" Project, supported by the Ministry of Education and National Research Foundation of Korea

References

1. Kunertova, Dominika: The war in Ukraine shows the game-changing effect of drones depends on the game. Bulletin of the Atomic Scientists 79(2), 95–102 (2023)
2. Eslami, Mohammad: Iran's drone supply to Russia and changing dynamics of the Ukraine war. Journal for Peace and Nuclear Disarmament 5(2), 507–518 (2022)
3. Beul, Marius, Schwarz, Max, Quenzel, Jan, Splietker, Malte, Bultmann, Simon, Schleich, Daniel, Rochow, Andre, Pavlichenko, Dmytro, Rosu, Radu Alexandru, Lowin, Patrick, et al.: Target chase, wall building, and fire fighting: Autonomous UAVs of team Nimbro at MBZIRC 2020. arXiv preprint arXiv:2201.03844 (2022)
4. Rohan, Ali, Rabah, Mohammed, Kim, Sung-Ho: Convolutional neural network-based real-time object detection and tracking for parrot AR drone 2. IEEE access 7, 69575–69584 (2019)
5. Zhao, Moju, Shi, Fan, Anzai, Tomoki, Takuzumi, Nishio, Maki, Toshiya, Kita, Ito, Kuromiya, Naoki, Okada, Kei, Inaba, Masayuki: Team JSK at MBZIRC 2020: Interception of fast flying target using multilinked aerial robot. Field Robotics 1, 70–101 (2021)
6. Anastasiou, A., Makrigiorgis, R., Kolios, P., Panayiotou, C.: Hyperion: A Robust Drone-based Target Tracking System. In: 2021 International Conference on Unmanned Aircraft Systems (ICUAS), 927–933 (2021)
7. Karras, George C., Bechlioulis, Charalampos P., Fourlas, George K., Kyriakopoulos, Kostas J.: Target Tracking with Multi-rotor Aerial Vehicles based on a Robust Visual Servo Controller with Prescribed Performance. n: 2020 International Conference on Unmanned Aircraft Systems (ICUAS), 480–487 (2020)
8. Liu, Yisha, Wang, Qunxiang, Hu, Huosheng, He, Yuqing: A novel real-time moving target tracking and path planning system for a quadrotor UAV in unknown unstructured outdoor scenes. IEEE Transactions on Systems, Man, and Cybernetics: Systems 49(11), 2362–2372 (2018)
9. Liu, Xuancen, Yang, Yueneng, Ma, Chenxiang, Li, Jie, Zhang, Shifeng: Real-time visual tracking of moving targets using a low-cost unmanned aerial vehicle with a 3-axis stabilized gimbal system. Applied Sciences 10(15), 5064 (2020)
10. Bartak, Roman, Vykovskỳ, Adam: Any object tracking and following by a flying drone. In: 2015 Fourteenth Mexican International Conference on Artificial Intelligence (MICAI), 35–41 (2015)
11. Redmon, Joseph, Divvala, Santosh, Girshick, Ross, Farhadi, Ali: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, 779–788 (2016)