

Article

Quadcopter Drone for Vision-Based Autonomous Target Following

Wen-Chieh Chen, Chun-Liang Lin,^b Yang-Yi Chen * and Hsin-Hsu Cheng

Department of Electrical Engineering, National Chung Hsing University, Taichung 40227, Taiwan

* Correspondence: max10171124@yahoo.com.tw

Abstract: Unmanned aerial vehicles (UAVs) are becoming popular in various applications. However, there are still challenging issues to be tackled, such as effective obstacle avoidance, target identification within a crowd, and specific target tracking. This paper focuses on dynamic target following and obstacle avoidance to realize a prototype of a quadcopter drone to serve as an autonomous object follower. An adaptive target identification system is proposed to recognize the specific target in the complicated background. For obstacle avoidance during flight, we introduce an idea of space detection and use it to develop a so-called contour and spiral convolution space detection (CASCSD) algorithm to evade obstacles. Thanks to the low architecture complexity, it is appropriate for implementation on onboard flight control systems. The target prediction is integrated with fuzzified flight control to fulfill an autonomous target tracker. When this series of technical research and development is completed, this system can be used for applications such as personal security guard and criminal detection systems.

Keywords: UAV; vision-based identification; autonomous tracking; guidance

1. Introduction

Over the past few years, various innovative applications of unmanned aerial vehicles (UAVs) have emerged due to lower cost in hardware and appearance of powerful onboard microcomputers.

UAVs can conquer topographic barrier so that they are appropriate for conducting missions in dangerous zones, replacing manpower to transport goods, or even serving as a personal standby assistant.

Recently, artificial intelligence (AI)-based automatic control technology has received attention attributed to the significant improvement such as the generalization capability of deep learning. Deep learning drives many AI applications that improve automation, performing analytical and physical tasks without human intervention. For instance: the robot “pepper” in [1] is used for domestic and navigation purposes; the biped robot is used to replace soldiers [2]; there are medical hospitality robots [3], autonomous cars [4,5], and much more. In addition to land vehicles, there are various aerospace vehicles developed for specific purposes, such as the delivery UAVs [6–8], disaster relief UAVs, and multi-machine protocol flying UAVs [9].

UAVs that track targets need to adapt to various environments to fly. In [10], the authors proposed a robust adaptive recursive sliding mode attitude control, which can make the quadrotor effectively resist unknown interference. The work of [11] proposed an anti-jamming control scheme based on multi-observers to counteract multiple interferences so that the quadrotor can resist strong wind and load interference.

Focusing on the technical issues, while research efforts in target following and obstacle avoidance techniques have been quite mature in the field of robotics, the two issues are still challenging for the operational scenarios of miniature UAVs such as quadcopter drones



Citation: Chen, W.-C.; Lin, C.-L.; Chen, Y.-Y.; Cheng, H.-H. Quadcopter Drone for Vision-Based Autonomous Target Following. *Aerospace* **2023**, *10*, 82. <https://doi.org/10.3390/aerospace10010082>

Academic Editor: Hailong Huang

Received: 8 December 2022

Revised: 26 December 2022

Accepted: 3 January 2023

Published: 14 January 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

because of the relatively limited hardware resources available and the resulting neutral instability [12].

Improving the reliability of UAVs is one of the challenging issues in the market. Crash events of UAVs are not unusual. Moreover, static or dynamic target identification in a crowd is also a difficult issue for miniature UAV development due to the limited computational resources and budget cameras used [13]. In the academic research field, while there are research efforts dedicated to investigating target tracking [14–16] and obstacle avoidance [17–20], further improvement on functional enhancement, especially in target identification and tracking, is still highly demanded. For example, the proposed target tracking in [15] requires the target to be identified wearing a special T-shirt with a QR code on it. The study of [16] relies on color features to identify the specific target. The rough obstacle avoidance method adopted in [19] might cause the UAV to collide with obstacles when the sensor is not in parallel with the obstacles. Although the research tasks of [21,22] have combined the capability of target tracking and obstacle avoidance, their results were only verified via simulation study. The common difficulty of various research efforts is that the extremely large computational resources were unsuitable for implementation on the onboard microprocessor.

Considering the general weaknesses of the current approaches, this research task proposes the methods developed for target identification and movement prediction, the adaptive cruise control (ACC) system, and the Kalman filtering navigation to fulfill a mission-oriented flight control system. Our research motivations are listed as follows:

- The tracking capability of the commercial miniature UAV is still immature [12,13]. The methods developed in the papers might not be applicable in real-world applications [14,15]. These methods are feasible, but are slightly restricted from the viewpoint of practical applications. For human target following, this research proposes an adaptive target identification system to resolve the problem when the specific target is in a crowd.
- The current UAV obstacle avoidance algorithms revealed in the papers were difficult to implement in the small-scale embedded system because of the large computational sizes. Here, we propose a novel contour and spiral convolution space detection (CASCSD) algorithm to tackle the issue. Through the emulated expansion and etching of the image processing, we can filter out noises while enlarging imaging signals to indicate that if the obstacles might interfere with the flight path. This algorithm consumes less computational resources and is appropriate to be used in the current miniature UAV applications.
- Intelligent mobile assistants have recently become popular; however, UAVs moving inside a building, on stairs, or in rugged areas is still a challenge. A miniature UAV drone is one of the potential substitutes for work under these scenarios.

In addition, in order to improve the research quality, we also refer to other UAV and tracking-related literature, such as [23], which mainly discusses the deep learning application of UAVs in Internet of Things applications and investigates the related applications of UAVs in detail. Ref. [24] discusses the issues related to the bionic control of UAVs, which can provide more ideas for the control of UAVs in this paper. In the research of tracking, [25] uses a UAV to track the landing point on the roof of a car and allows the UAV to safely land at the target point. Ref. [26] proposed an adaptive lightweight UAV tracking algorithm, which allows UAVs to track various targets at high altitudes. Ref. [27] proposed a distance estimation method based on hybrid stereo vision, which can accurately calculate the distance between the robot and the object and can provide more ideas for this research on obstacle avoidance. Ref. [28] used deep learning, simulated tracking, and obstacle avoidance with multiple UAVs, and the results showed the algorithm's effectiveness. Synthesizing various research ideas, this study found that the ability of UAVs to fly indoors and outdoors is still lacking, and this research intends to complete a personal security guard system. The overall system of UAVs needs to have identification, tracking, and obstacle avoidance at the same time. The three functions and the algorithm need to be

as lightweight as possible. Only by fulfilling these demands can the experimental purpose of the research be achieved.

2. System Description

The dynamics of the UAV directly affect flight stability [29]; to enable the UAV to have intelligent tracking and obstacle avoidance functions, we first need to establish a quadrotor flight dynamic model and control equations. Secondly, the image and the entire coordinate system must be convertible so that we can import the image information and various algorithms into the flight control perform tasks.

2.1. Architecture

Figure 1 illustrates the quadcopter architecture implemented in this research task. The miniature UAV drone is designed to be a vision-based autonomous target follower. The functions of the key modules include the following modules: a flight control system including an inertial navigation system (INS), gyroscope, optical flow sensor using ground texture and visible features for indoor positioning, and barometer for attitude determination; an onboard AI processor is responsible for intelligent target identification and tracking, flight path computation with obstacle avoidance, and realization of all supporting algorithms; and a depth camera is responsible for transmitting external images to the AI processor. It also provides information of the relative distance between the target and the environment.

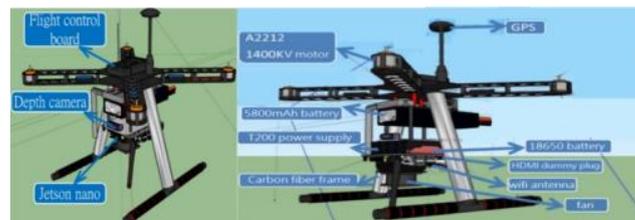


Figure 1. Architecture of the miniature quadcopter drone implemented in this research.

2.2. Modeling

The four-rotor mathematical model of the miniature quadcopter for the purpose of target identification and tracking is given in Appendix A.

Flight Dynamics Model (FDM) Description

The world frame and the quadcopter body frame of this study are shown in Figure 2, where (X_E, Y_E, Z_E) is the inertial coordinate system, O_E is the origin of the inertial coordinate, and (e_1, e_2, e_3) is the unit vector along the inertial coordinate axis. (X_{UB}, Y_{UB}, Z_{UB}) is the rigid body coordinate system of the quadcopter body, O_{UB} is the origin of the rigid body coordinate, and (b_1, b_2, b_3) is the unit vector along the rigid body coordinate axes.

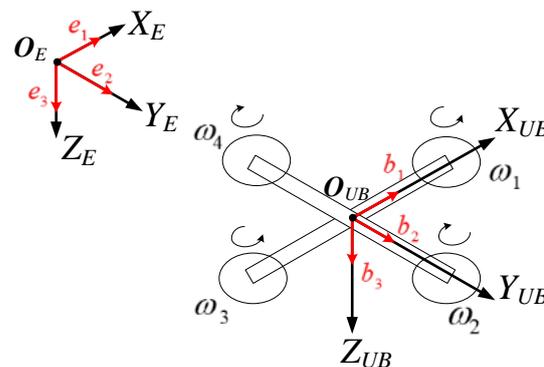


Figure 2. The world and body frames.

Proceeding to the development given in the Appendix A, we have the acceleration and angular acceleration equations of the quadcopter as:

$$\begin{aligned}
 \ddot{x} &= \frac{1}{M} k_2 G_{R^3}(\vec{b}_3, \vec{e}_1)(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2), \\
 \ddot{y} &= \frac{1}{M} k_2 G_{R^3}(\vec{b}_3, \vec{e}_2)(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2), \\
 \ddot{z} &= \frac{1}{M} (k_2 G_{R^3}(\vec{b}_3, \vec{e}_3)(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) - Mg), \\
 \dot{p} &= \frac{1}{I_{11}} \ell k_2 (\omega_1^2 + \omega_4^2 - \omega_2^2 - \omega_3^2) \\
 \dot{q} &= \frac{1}{I_{22}} \ell k_2 (\omega_3^2 + \omega_4^2 - \omega_1^2 - \omega_2^2), \\
 \dot{r} &= \frac{1}{I_{33}} \ell k_1 (\omega_1^2 + \omega_3^2 - \omega_2^2 - \omega_4^2)
 \end{aligned} \tag{1}$$

To proceed, we perform coordinate transformation from the body frame to the world frame along the yaw, roll, and pitch axes:

$$\begin{bmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & -s\phi \\ 0 & s\phi & c\phi \end{bmatrix} = \begin{bmatrix} c\psi c\theta & c\psi s\theta s\phi - s\psi c\phi & c\psi s\theta c\phi + s\psi s\phi \\ s\psi c\theta & s\psi s\theta s\phi + c\psi c\phi & s\psi s\theta c\phi - c\psi s\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \tag{2}$$

where θ, ϕ, ψ denote the angles of pitch, roll, and yaw respectively (see Figure 3). For completeness, the coefficients of the air resistance C_{D1}, C_{D2}, C_{D3} and coefficients of the rotary resistance $C_{1DR}, C_{2DR}, C_{3DR}$ are considered. It can then be further transferred into the state space representation by considering the gyro effect:

$$\begin{aligned}
 \ddot{x} &= \frac{1}{M} (k_2 (c\psi s\theta c\phi + s\psi s\phi) u_1 - C_{D1} \dot{x}), \\
 \ddot{y} &= \frac{1}{M} (k_2 (s\psi s\theta c\phi - c\psi s\phi) u_1 - C_{D2} \dot{y}), \\
 \ddot{z} &= \frac{1}{M} (k_2 (c\theta c\phi) u_1 - Mg - C_{D3} \dot{z}), \\
 \dot{p} &= \frac{1}{I_{11}} (\ell k_2 u_2 - C_{1DR} p) - \frac{I_{33} - I_{22}}{I_{11}} \dot{r}, \\
 \dot{q} &= \frac{1}{I_{22}} (\ell k_2 u_3 - C_{2DR} q) - \frac{I_{11} - I_{33}}{I_{22}} \dot{r}, \\
 \dot{r} &= \frac{1}{I_{33}} (\ell k_1 u_4 - C_{3DR} r) - \frac{I_{22} - I_{11}}{I_{33}} \dot{p}
 \end{aligned} \tag{3}$$

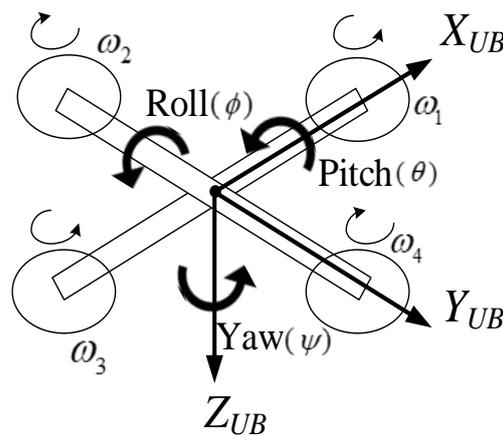


Figure 3. Schematic diagram illustrating the respective attitude angles.

3. Target Tracking and Obstacle Avoidance

A depth camera is utilized to detect the target and surrounding objects. Before conducting target identification, image calibration is conducted.

3.1. Calibration and Coordinate Conversion

The pinhole imaging model [30,31] is used to describe this process, and it is corrected by the principle of similar triangles (see Figure 4).

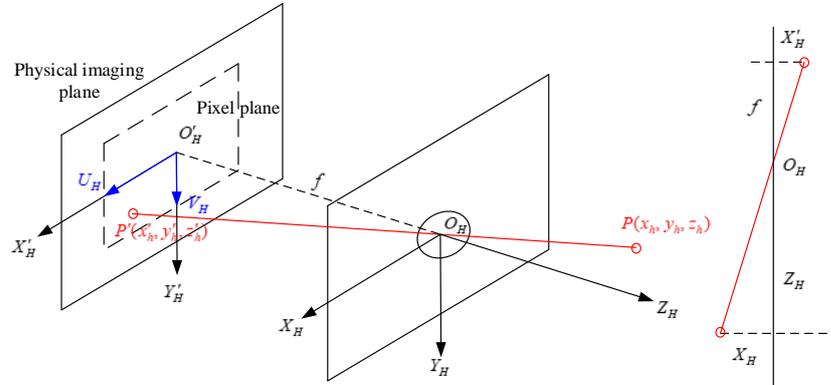


Figure 4. The pinhole photo picturing model.

Suppose that the camera coordinate system is denoted by (X_H, Y_H, Z_H) with the origin O_H . The pixel coordinates are X'_H, Y'_H, Z'_H and the origin is O'_H , as shown in Figure 4. Furthermore, there is a targeted point $P(x_h, y_h, z_h)$ in reality, and its new coordinate mapped to the pixel plane is $P'(x'_h, y'_h, z'_h)$. Let the focal length f be the distance from the pinhole to the pixel coordinate plane. This gives rise to the following ratio formula:

$$X'_H = f \frac{X_H}{Z_H}, Y'_H = f \frac{Y_H}{Z_H} \tag{4}$$

We define the origin of the pixel coordinate system c as being located at the upper left corner of the image; the U_H axis is parallel to the X_H axis, and the V_H axis is parallel to the Y_H axis and is at the same plane as the pixel coordinate system. Thus, there is a two-axis difference between the two-pixel coordinates.

Assume that the U_H and V_H axes are zoomed in by α and β times, respectively, and the origin is linearly shifted by the base unit $[c_x, c_y]^T$. One obtains:

$$U_H = \alpha X'_H + c_x, V_H = \beta Y'_H + c_y \tag{5}$$

Let $\alpha f = f_x$ and $\beta f = f_y$, then

$$U_H = f_x \frac{X_H}{Z_H} + c_x, V_H = f_y \frac{Y_H}{Z_H} + c_y \tag{6}$$

Transforming this into a matrix representation gives

$$Z_H \begin{bmatrix} U_H \\ V_H \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_H \\ Y_H \\ Z_H \end{bmatrix} = KP \tag{7}$$

where K is the camera's inner parameter matrix. It is used to compute the camera's outer parameter matrix.

Let P_w be the point P from the camera coordinate to the world coordinate, where R and t are the rotation and translation of the camera, respectively.

$$Z_H \begin{bmatrix} U_H \\ V_H \\ 1 \end{bmatrix} = K(RP_w + t) \tag{8}$$

Performing normalization gives

$$RP_w + t \rightarrow \begin{bmatrix} X_H & Y_H \\ Z_H & 1 \end{bmatrix}^T \quad (9)$$

The point P is at the normalized plane at the point $Z_H = 1$. The normalized coordinate that multiplies the camera intrinsic parameters results in the modified pixel coordinate.

3.2. Dynamic Target ID and Locking

The traditional YOLO model [32] can only identify objects that are already trained, it would be insufficient while being applied to the current objective. We updated the function of specific target identification by considering the limited onboard resources.

At first, the results b_w, b_h, b_x, b_y of the bounding boxes generated by the modified YOLO network given in [33] are used to determine the centroid position of the targeted person. The coordinates of the four vertices of each bounding boxes are $(b_x - \frac{b_w}{2}, b_y - \frac{b_h}{2})$, $(b_x + \frac{b_w}{2}, b_y - \frac{b_h}{2})$, $(b_x - \frac{b_w}{2}, b_y + \frac{b_h}{2})$, and $(b_x + \frac{b_w}{2}, b_y + \frac{b_h}{2})$. To be realized on the onboard computer, the modified YOLO is only used to frame possible objects in the image frame.

To track the specific object with the limited computational resources, a lightweight algorithm is proposed in this research task for dynamic target identification. The algorithm involves three parts which are calculated per video frame, including: the color scores of the target's clothes, pants, and hair; the respective length-to-width ratio scores; and the tracking error of the dynamic target. The equations are described as follows:

$$\begin{aligned} CS &= \sum_{i=1}^n w_{1i} \cdot \text{component color}[i], \\ LS &= \sum_{i=1}^m w_{2i} \cdot \text{component length}[i] \end{aligned} \quad (10)$$

where *component color* [i] denotes the *i*th colored component with a Boolean value ("1" for matched and "0" for unmatched), *component length* [i] denotes the *i*th length-to-width ratio component ("1" for matched and "0" for unmatched), *m* and *n* represent the total number of colored components and length-to-width ratio components, respectively, and w_{1i} and w_{2i} are the *i*th weighted exponents. After obtaining scores of *CS* and *LS*, the position score is calculated according to the target locking schematic illustrated in Figure 5 with the score

$$PS = A \frac{1 - e^{-s(PA - bias)}}{1 + e^{-s(PA - bias)}} \quad (11)$$

where $A = CS + LS$, *PA* refers to the accuracy of calculating the target position of each frame. *bias* and *s* are the offset and slope of the sigmoid function, respectively, with their respective equations being as follows: $PA = \left(\sqrt{(x_p - x_c)^2 + (y_p - y_c)^2 + (z_p - z_c)^2} \right)^{-1}$, where (x_p, y_p, z_p) denotes the position of the current target frame, (x_c, y_c, z_c) denotes the position of the previous frame, and $bias = \frac{n_1}{IS}$, where n_1 represents the number of targets locked within the image space $IS = L \cdot W$, with *L* and *W* being the image length and width, respectively. The bias is used to adjust the maximum error size of the position and filter out the surrounding noise. For the value of *s*, when the target position falls within the gray area as shown in Figure 5, it means that the target position is not located in the central focused region. The closer the position is to the edge, the lower the score. It is designed by $s = \frac{RS}{m_1}$, where m_1 represents the number of people locked within the reconnaissance space $RS (= \pi ab) \subset IS$ with the semi-major axis of length $a = \alpha W$ and semi-minor axis of length $b = \beta L$ and $0 < \beta \leq \alpha \leq 1$. The rules are summarized as follows:

1. There are many people that appear in the image. The bias is increased to reflect the noisy background.

2. There are few people in the image, but they are tightly crowded. The bias is decreased and s is increased by increasing β, α to rise discriminative sensitivity.
3. There are many people, but they are widely dispersed. The bias is decreased to aid highlighting the target.
4. There are few people and they are widely dispersed. The bias is set to zero and a large s is suggested to boost the discriminative effect.

After obtaining $CS, LS,$ and $PS,$ the target ID score is characterized by

$$RR = (CS'W_1 + LS'W_2 + PS'W_3) \cdot 100\% \tag{12}$$

where the normalized individual scores $CS', LS',$ and PS' correspond to $CS, LS,$ and $PS,$ respectively, and $W_i, i = 1 \sim 3$ are the respective weighting factors satisfying $\sum_{i=1}^3 W_i = 1.$

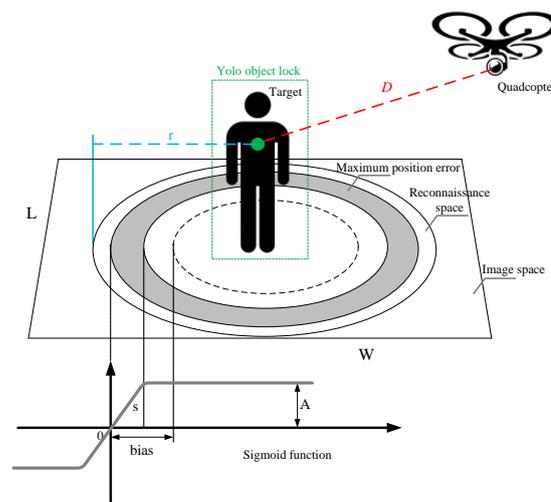


Figure 5. Target locking schematic framework.

Figure 6 illustrates the operational flow of the adaptive target identification system.

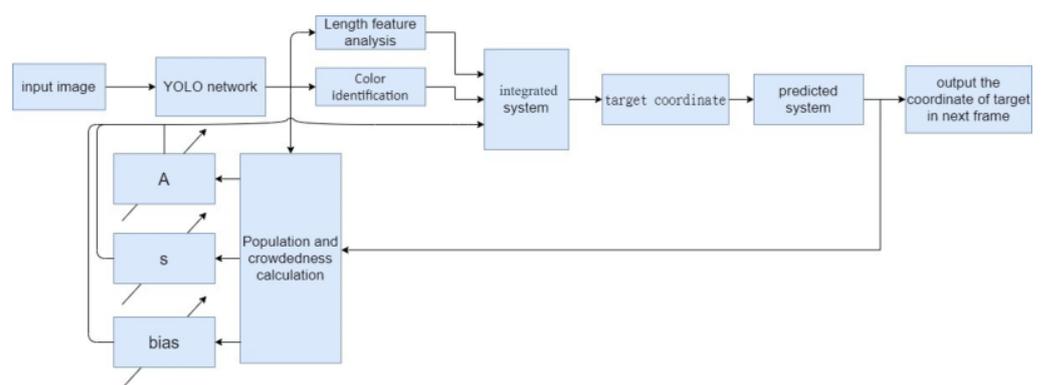


Figure 6. Operational flow of the adaptive target identification system.

3.3. Target Positioning in 3D Space

Once the target position in the pixel coordinates is obtained, it is transferred to the Cartesian plane. In addition, the relative distance D with respect to the target provided by the depth lens is integrated with the pixel coordinate representation to yield the target position in the spherical coordinates, as shown in Figure 7.

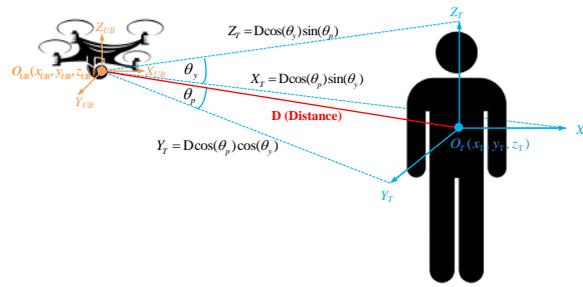


Figure 7. Coordinate transformation.

For the current experiment, the camera’s field of view (FOV) is $69.4 \times 42.5 \times 77$ degrees with 640×480 p (4:3) pixel resolutions. It is used to compute the degree of the pixels within a frame. We end up with the following equations to transform the spherical coordinate system to the Cartesian coordinates:

$$\begin{aligned} X_T &= D \cos(\theta_p) \sin(\theta_y), \\ Y_T &= D \cos(\theta_p) \cos(\theta_y), \\ Z_T &= D \cos(\theta_p) \sin(\theta_y) \end{aligned} \tag{13}$$

where θ_p and θ_y are the pitch and yawing plane line-of-sight (LOS) angles, respectively. The quadcopter rotates its body to move; thus, we have

$$P_{new} = RP_{origin} \tag{14}$$

where P_{new} is the quadcopter position after performing rotation and R is the coordinate rotational matrix.

3.4. Target Movement Estimation

Let the target state vector be x_t , including the target position p , the velocity v , and the acceleration a , which is random and conformed to the Gaussian distribution with mean μ and variance Σ . Consider external interference and measurement so that the prediction of measurement statistics is characterized by the estimated measurement $\hat{z}_t = H_t \hat{x}_t$, with the error covariance of estimation P_t and t denoting the current time step and the real measurement characterized by $(\mu_1, \Sigma_1) = (z_t, R_t)$, where R_t is the measurement noise covariance.

The prediction equations of the state and error covariance P_t are given by

$$\begin{aligned} \hat{x}_t &= F_t \hat{x}'_{t-1}, \\ P_t &= F_t P_{t-1} F_t^T + Q_t \end{aligned} \tag{15}$$

where the process noise covariance $Q_t = \varepsilon_q I_9$ and the system matrix F_t given by

$$F_t = I_9 + \begin{bmatrix} 0_3 & \delta t I_3 & 0.5 \delta t^2 \\ 0_3 & 0_3 & \delta t I_3 \\ 0_3 & 0_3 & 0_3 \end{bmatrix}$$

where δt is the sampling period. Including the two Gaussian distribution terms provides the state update equations:

$$\begin{aligned} \hat{x}'_t &= \hat{x}_t + K'(z_t - \hat{z}_t), \\ P'_t &= P_t(I_9 - K'H_t), \\ K' &= P_t H_t^T (H_t P_t H_t^T + R_t)^{-1} \end{aligned} \tag{16}$$

where $H_t = [I_3 \quad 0_{3 \times 6}]$ and $R_t = \varepsilon_r I_3$, with ε_r being a small positive constant.

3.5. Flight Path Planning

The imaging processing system uses an embedded microcomputer to realize deep learning and Kalman filtering computation. We first propose a contour and spiral convolution space detection algorithm (CASCSO) to handle obstacle avoidance and plan an appropriate flight path to track the target with its center of mass located at $O_S(x_s, y_s, z_s)$.

Using the flyable space detection process, the image is divided into two types of blocks, denoted by “1” or “0”, which refer to the safe-to-fly (STF) and non-STF regions, respectively. We use the infrared of the depth camera to measure the relative distance. As it is easily affected by environmental light, the image processing system first slightly erodes the scope of the space, that is, the expanded scope of the obstacle, which is performed to prevent smaller obstacles from being filtered. Next, it enhances the dilated space range to filter out the light pollution or other noise that infrared ray cannot measure. Then, the space is eroded back to its original size; see Figure 8 for the process. A safety buffering range to prevent the UAV from drifting is reserved at the edge of each detected obstacle.

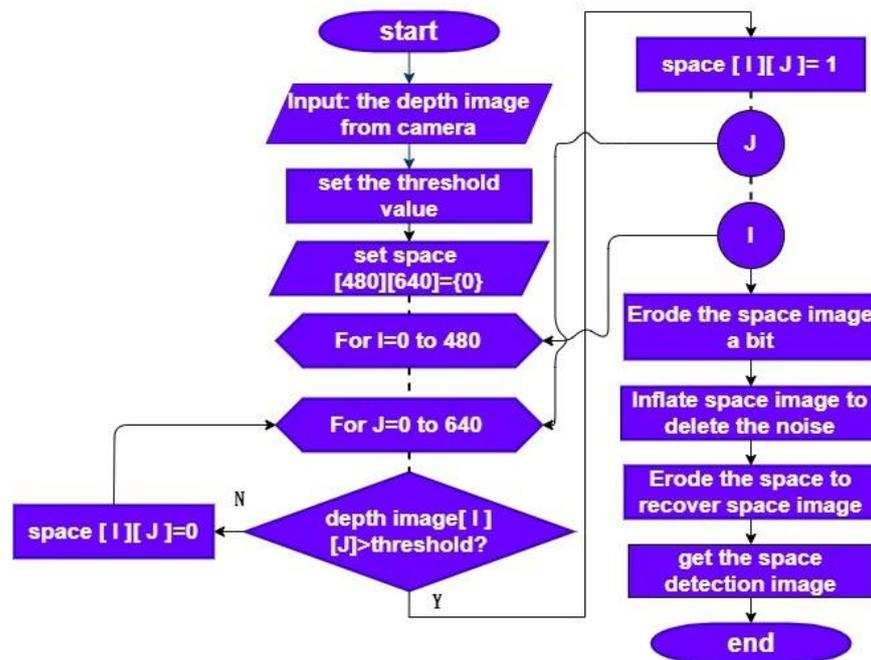


Figure 8. Noise removal and spatial detection flow.

To ensure that the target can be tracked with the shortest path, the two-dimensional pixel coordinates are inversely derived from the predicted coordinates generated by the Kalman filter. Let the predicted coordinates of the Kalman filter be $O_p = [x_p, y_p, z_p]^T$. The line-of-sight (LOS) angles θ_p and θ_y can be deduced from

$$\theta_p = \cos^{-1}\left(\frac{z_p}{D_p}\right), \theta_y = \cos^{-1}\left(\frac{x_p}{D_p}\right) \tag{17}$$

where $D_p = \|O_p\|$. Based on the current camera’s FOV, the center position of the target measured from the image are $c_x = 11.3\theta_p$ and $c_y = 9.22\theta_y$.

The spiral convolution from the centroid position outwards determines whether there is available space for the quadcopter to fly forward. The spiral convolution means that the filter matrix rotates from inside to outside and convolves the spatial detection feature map. The size of the filter depends on the coverage of the boundary on the imaging plane, as illustrated in Figure 9.

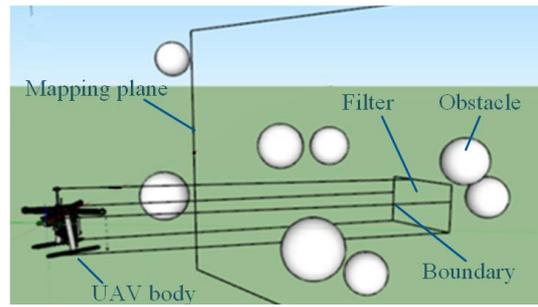


Figure 9. Graphical explanation of the CACSD distinguishing the safety zone.

As shown in Figure 9, the image projection will be zoomed in. Therefore, it is necessary to set up a threshold of the mapping frame from the FOV. For the current system, the lengths of x and y axes per pixel are determined by $image_{width} = \frac{threshold \tan(34.7)}{240}$ and $image_{height} = \frac{threshold \tan(21.25)}{320}$.

The control system measures the distance between the camera and the global positioning system (GPS), the left and right wings, and the feet and the ratio between them, and calculates the size of the quadcopter on the imaging plane at the critical distance.

For the spiral convolution, it starts from the target mass center on the pixel coordinates to find the space to move from the inside to the outside. If the target is not found, the next process is completed. Because the space around the target has already been searched during the process of spiral convolution, there is no need to continue searching for the shortest path around the target. Instead, it directly starts searching for the remaining part.

4. Adaptive Cruise Control (ACC)

The adaptive cruise control system aims at the distance between the quadcopter drone and the target by adaptive adjusting the speed of target tracking.

4.1. Thrust Force

The acceleration of the quadcopter is determined by the body inclination angle $a = g \tan(\theta)$. As the moment of inertia I_{xx} , arm length ℓ , and motor coefficient k_2 are all constants, thus the rotational speed of the motor is a major concern. The thrust force is computed by $\|F\| = 23 \left(\frac{length}{10}\right)^3 \frac{pitch}{10} \left(\frac{volt \cdot kV}{1000}\right)^2$, where $length$ is the propeller diameter, $pitch$ denotes the propeller pitch angle, and kV denote the voltage per revolution of the motor. The thrust force of the motor is 1392 gram-force. With the obtained thrust force, propeller diameter, and moment of inertia, the inclination is $\theta(t) = \theta_0 + 0.5\ddot{\theta}t^2$.

4.2. Fuzzy Control Implementation

Figure 10 illustrates the scenario of the target follower. When the quadcopter is far away from the target, the flight control command increases the angle of the pitch axis to speed up the task.

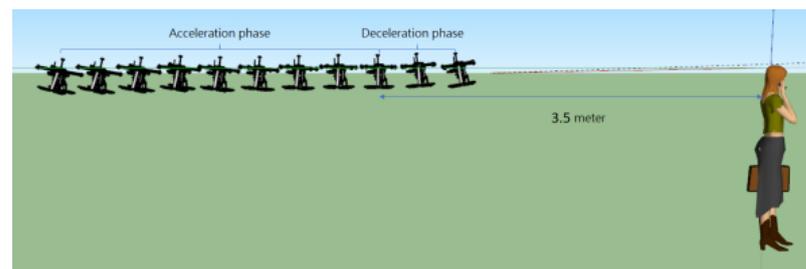


Figure 10. Graphical explanation of the target follower.

As the quadcopter drone gradually approaches the target with higher speed, the relative distance between the target and quadcopter would be gradually decreasing. When the quadcopter decelerates but the target person starts to move forward, the quadcopter would respond by speeding up again. The faster the target walks, the faster the quadcopter will follow.

Figure 11 displays the schematic diagram of the overall control system where the motor mixing algorithm (MMA) transfers the roll, yaw, and pitch control commands to the driving commands of four DC brushless motors.

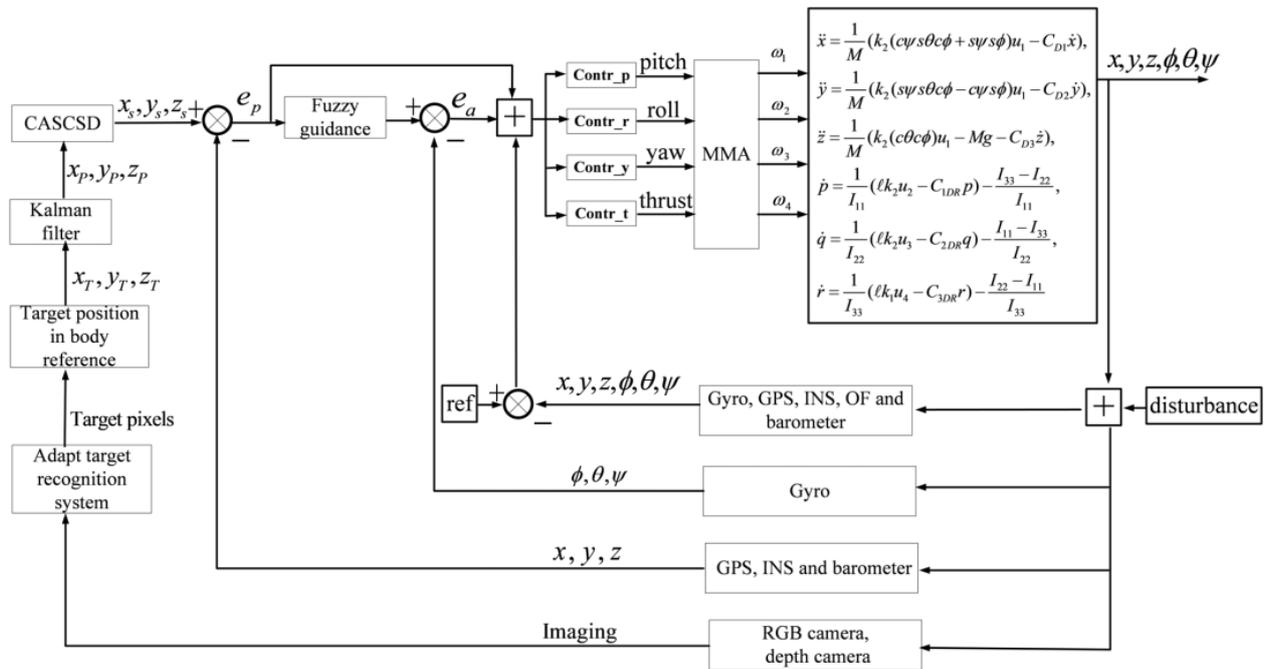


Figure 11. The flight control system.

To cope with the high nonlinearity of the quadcopter flight dynamics, a fuzzy logic control is incorporated here as the core of the ACC system. A fuzzy guidance controller with two inputs and one output is implemented. The adopted membership functions are shown in Figures 12–14, where the fuzzified inputs are $\mu(D)$ and $\mu(V)$ and the fuzzified output is $\mu(O)$, with $\mu(\cdot)$ representing the degree of membership of fuzzification; D and V are the relative distance and speed between the quadcopter and the target, respectively, and O is the acceleration output. Each group of the membership functions possesses seven fuzzy sets. The membership functions in Figure 12 specify the relative distance between the quadcopter and the target. Considering the capability of the camera, the fixed relative distance is set to be 3.5 m. This corresponds to the linguistic variable ZO.

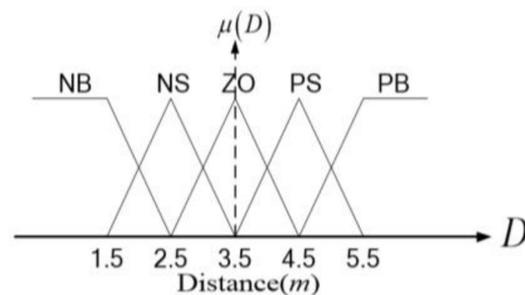


Figure 12. Membership functions for characterizing the target-quadcopter relative distance.

The membership functions in Figure 13 are used to fuzzify the relative speed. In the universe of discourse, zero means that the target and the quadcopter are in still status or they are synchronously moving; the interval from zero to two refers to the walking status of the target, and the interval from two to four refers to the running status. Refined fuzzification of the variables is achievable by increasing the number of membership functions.

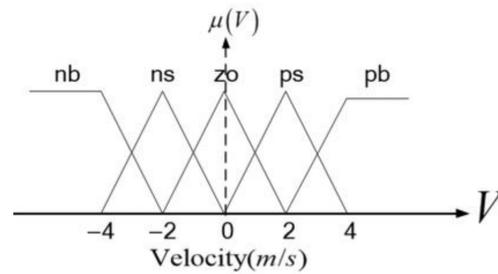


Figure 13. Membership functions for characterizing the target-quadcopter relative speed.

Figure 14 shows the membership functions characterizing the acceleration command. The goal is for the quadcopter to maintain a certain distance and quadcopter speed and LOS between to the target.

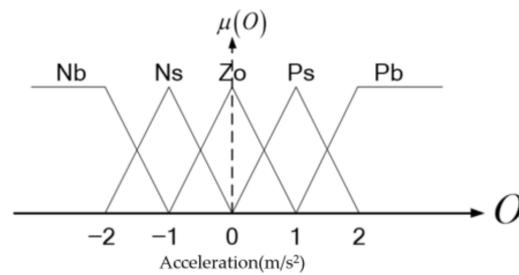


Figure 14. Membership functions for characterizing control command.

There are 25 guidance rules in total that are sorted in Table 1. The guidance law adopts proportional navigation guidance (PNG) commonly adopted by homing air target missiles based on the fact that the target and quadcopter are on a contact course when their direct LOS does not significantly change direction as the range closes. The guidance rules here are to keep a constant relative distance between the target and the quadcopter. The center of gravity method is adopted for the defuzzification of the inferred results.

Table 1. Fuzzy guidance rule base.

O		D				
		Nb	Ns	Zo	Ps	Pb
V	nb	Pb	Pb	Ps	Ps	Zo
	ns	Pb	Ps	Ps	Zo	Ns
	zo	Ps	Ps	Zo	Ns	Ns
	ps	Ps	Zo	Ns	Ns	Nb
	pb	Zo	Ns	Ns	Nb	Nb

5. Experimental Verification

In order to prove the feasibility of the theory, we conducted experiments such as outdoor tracking targets, indoor tracking targets, ladder tracking targets, and intelligent obstacle avoidance. The results are shown and explained as follows, and one can also refer to the following video URL: <https://www.youtube.com/watch?v=ko9BaSFTcwA> (accessed on 21 November 2021).

5.1. Autonomous Flight

Firstly, to examine the capability of a fundamentally stable flight, a 3D rectangular flight for the performance validation of the ACC was conducted. The real-world experimental results are shown in Figure 15 for the autonomous flight. The green solid line shows the planned flight path, the blue dotted line shows the flight result estimated by the extended Kalman filter (EKF), and the red dotted line presents the GPS measurement result. It verifies that under the interference of strong wind, the offset of the UAV flight path and the planned path is satisfactorily less than 0.2 m under the help of the real-time kinematic positioning (RTK) technique [23].

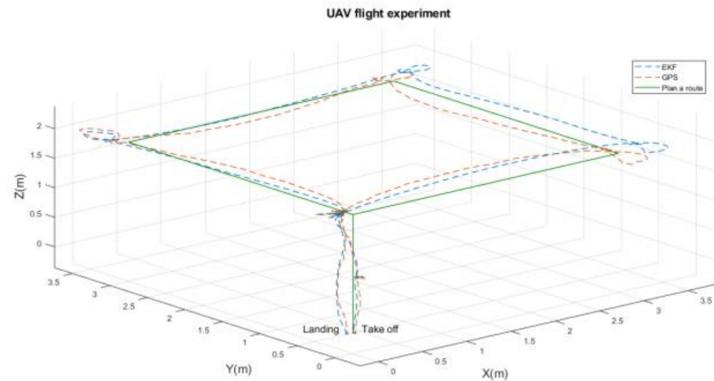


Figure 15. The experimentally autonomous rectangular flight.

5.2. Target Identification

A demonstrative image for locating the object based on the proposed YOLO algorithm is shown in Figure 16, where the matching scores are shown corresponding to the target feature, human face, clothing, and shorts.

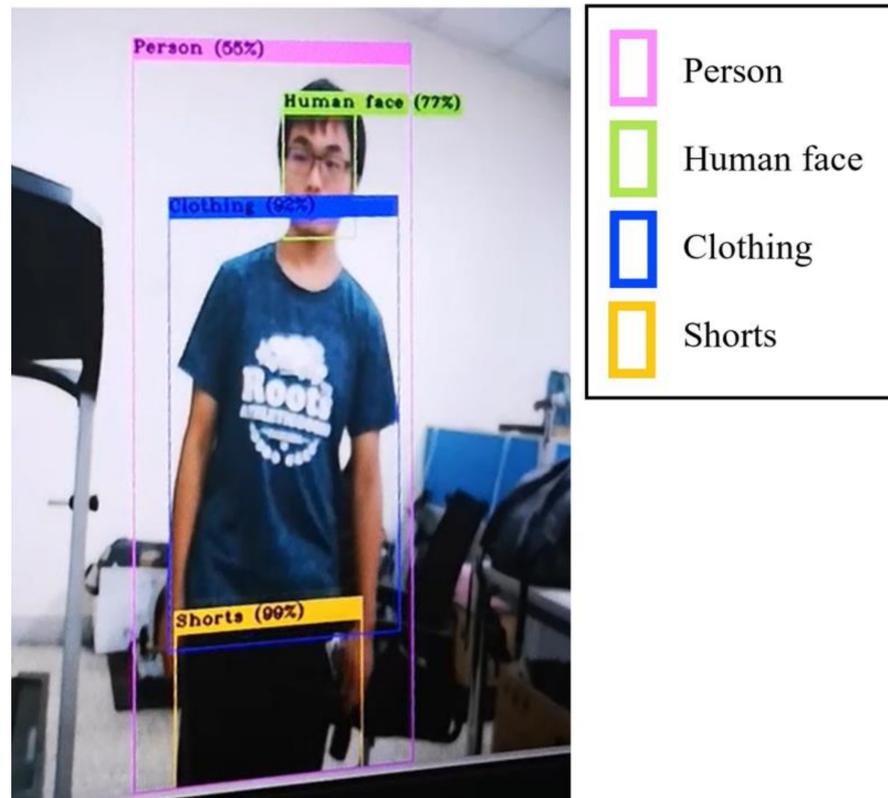


Figure 16. Demonstration of the locking the target.

The setting of weighting factors for target features is $W_1 : W_2 : W_3 = 0.3 : 0.2 : 0.5$. IS is set to 4×3 , $a = 0.8$, and $b = 0.6$. The position accuracy is normalized via the sigmoidal function, and (x_p, y_p, z_p) are the prediction coordinates from the previous frame of the target. It is based on the prediction of the position from the last frame compared with the measured position in the current frame. If the target achieves a high score with an accuracy level higher than, for example, 70%, the subject is recognized. We aimed to track the target with an orange jacket, blue trousers, and short black hair, as illustrated in Figure 17. The experimental result of the recognition rate (RR) of each subject within the video frame is listed in Table 2.

Table 2. RR of the objects that appear in Figure 17.

Object No.	1	2	3
RR (%)	99.7	41.0	7.5
Object No.	4	5	6
RR (%)	41.0	15.1	5.2

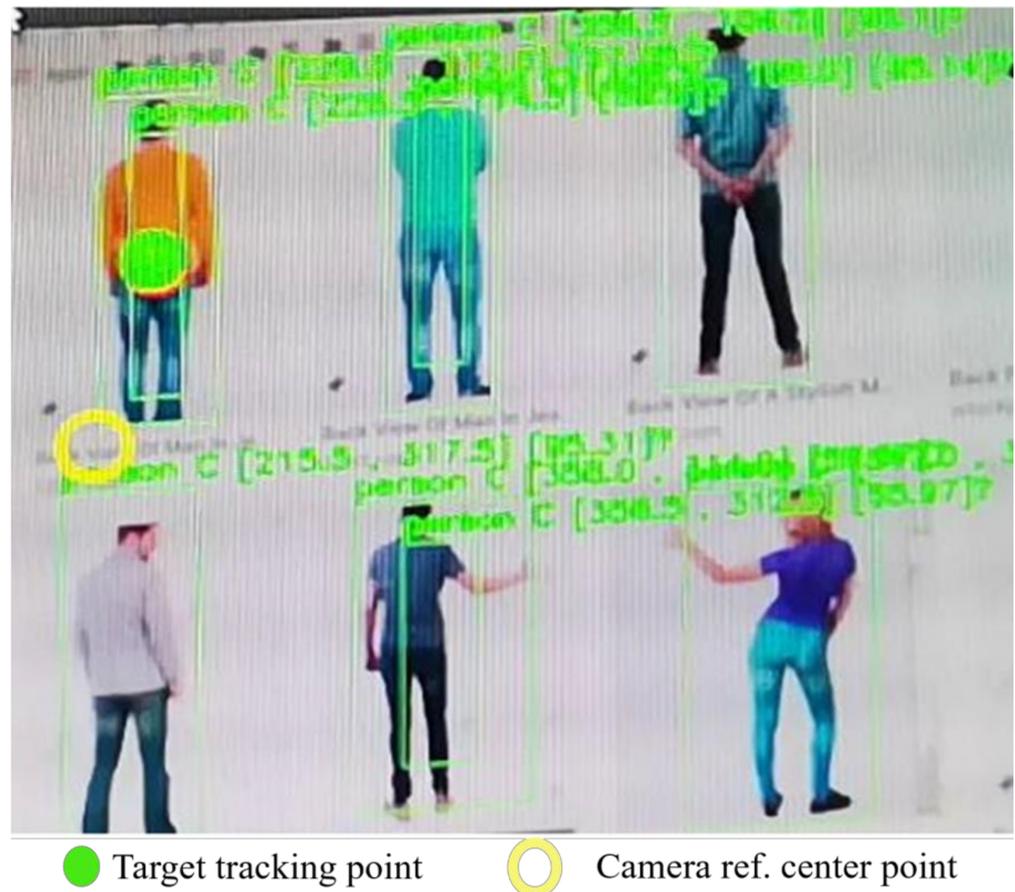


Figure 17. Experimental test of the adaptive target identification.

5.3. Estimation of Target Movement

Figure 18 illustrates the experimental result of the estimated target trajectory. The specific scenario is an inclined stair. The red line denotes the target trajectory and the green line denotes the predicted trajectory. This scenario represents a kind of extremal situation for the quadcopter drone during flight. When the target moves back and forth along the stair, the Kalman filter can still accurately predict the moving target trajectory.

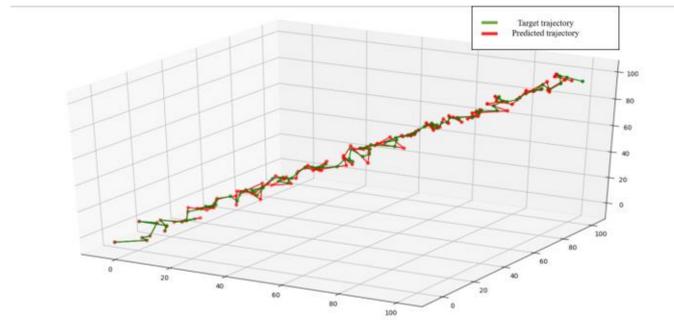


Figure 18. Kalman filtering estimation of the moving target movement along an inclined stair.

Figure 19 displays the real-world experimental results, which shows that even when the target is walking past a shelter, the state estimator can keep tracking the target. In the photo, the red dot denotes the human target center and the green dot shows the predicted point generated by the Kalman filter.



Figure 19. The experimental results of the target trajectory estimation.

5.4. Dynamic Target Tracking

The test environment includes outdoor grass, indoor corridors, and staircase. Figure 20 demonstrates the outdoor tracking test where the image tracking system, based on the deep learning algorithm proposed, has precisely locked target and started tracking.



Figure 20. Front tracking of the target from the view of the quadcopter.

As shown in Figure 21, there are two difficulties when the quadcopter is tracking the target under a windy environment. It is difficult for a lightweight quadcopter to maintain a stable altitude because the air pressure sensor might be affected by the environmental disturbance. The second is that the quadcopter itself is susceptible to the influence of wind, which always destabilizes the body attitude. The fuzzy control tuning proposed here has reserved a margin for tracking performance under the severe operating environments.



Figure 21. UAV tracking target test outdoors (The sequence of target movement and UAV tracking is shown in 1–4).

Figure 22 demonstrates the quadcopter tracking a target who is climbing the stairs. The difficulty in this case is that the target may simultaneously move horizontally and vertically. Thus, the quadcopter speed has to be accurately controlled to avoid the loss of locking.

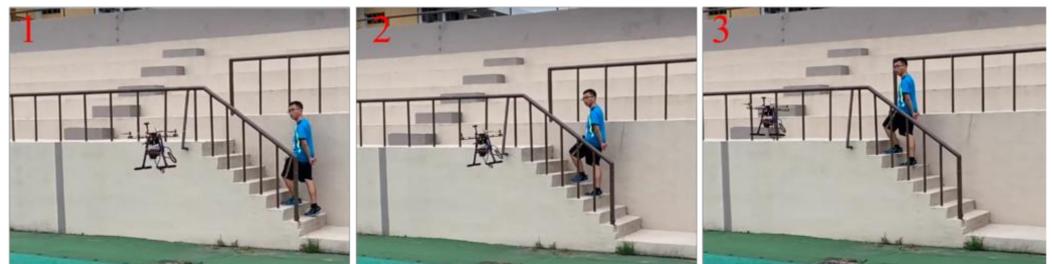


Figure 22. UAV tracking target test on stairs (The sequence of target movement and UAV tracking is shown in 1–3).

The case of target tracking in a narrow corridor is shown in Figure 23. This scenario tests the stability of the quadcopter without GPS assistance. For this kind of constricted environment, the quadcopter may suffer from the self-generated wind turbulence while the propelled air rebounds from the wall. Moreover, the indoor sharp corner movement becomes another critical issue to challenge the quadcopter's capability in space detection and agility of attitude control.



Figure 23. UAV tracking target test at the corridor (The sequence of target movement and UAV tracking is shown in 1–5).

5.5. Feature Comparison

The smart tracking and obstacle avoidance functions of this study are mainly based on the UAV personal security guard system. In order to prove the significance of the research, we cite several documents to show that the functions do not meet the research requirements. The comparison is as follows in Table 3.

Table 3. Feature comparison table.

Comparator	Application Field	Functional Narrative
This article	Indoor, outdoor (low altitude), stairwell	Identify by clothing; available for crowd, tracking, and obstacle avoidance
[15]	Indoor, outdoor (low altitude)	QR-code identification (strict conditions), tracking but no obstacle avoidance
[23]	Outdoor (high altitude)	Recognition and tracking functions are used for drones to park on the landing zone on the roof of the car
[24]	Outdoor (high altitude)	UAVs identify and track targets at a high altitude
[26]	Indoor	Use 3D vision to measure the distance between UAVs and objects (can be used for obstacle avoidance)
[28]	Outdoor	Using deep learning and simulating multiple UAV tracking and obstacle avoidance (simulation only)

6. Conclusions

This paper proposes a lightweight and intelligent UAV tracking and obstacle avoidance system. First, we used the YOLO model with fewer layers to recognize humanoid objects. Secondly, we proposed a lightweight adaptive target identification algorithm, which can extract personal clothing features and filter out complex image backgrounds other than the targets for target identification among crowds. Next, it was necessary to determine the movable space of the UAV, so we proposed the contour and spiral convolutional space detection (CASCSD) algorithm to search the moving area with the shortest distance between the UAV and the object and combine it with the Kalman filter, which estimates the displacement path when obstacles block the tracked object. Finally, a fuzzy adaptive cruise control (ACC) system keeps the drone at an optimal distance while tracking targets. As the selected experimental sites of this study are indoors, outdoors, and in stairwells, it can be applied to the personal security guard scenario.

Because the YOLO model is used for human recognition, it accounts for most of the computing power of the embedded system, resulting in only about ten frames of images

during tracking, so the tracking efficiency is not good. However, we have also tested the sum of the other algorithms; it only takes 0.03 s to complete all calculations. We believe that more experts and scholars will propose more lightweight human recognition algorithms in the future. By then, with the lightweight algorithm of this research, the personal security guard system will have higher flexibility and stability.

Author Contributions: Conceptualization, W.-C.C. and C.-L.L.; methodology, W.-C.C. and Y.-Y.C.; software, W.-C.C.; validation, Y.-Y.C., C.-L.L. and H.-H.C.; formal analysis, W.-C.C.; investigation, Y.-Y.C.; resources, C.-L.L.; data curation, H.-H.C.; writing—original draft preparation, W.-C.C.; writing—review and editing, Y.-Y.C.; visualization, W.-C.C.; supervision, C.-L.L.; project administration, C.-L.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Ministry of Science and Technology, Taiwan under the grant MOST 110-2221-E-005-084-MY2.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

The following development for quadcopter dynamics is standard in [34], in which necessary modifications been considered to fit in the current purpose. Firstly, for the upward thrust force of four motors shown in Figure 2, we define

$$\|F_i\| = k_2 \dot{\omega}_i^2 \quad (\text{A1})$$

where ω_i is the rotational speed of each motor and k_2 is a proportional constant.

The quadcopter heads move in the direction of b_3 . The four propellers generating downward thrust are standard and given by

$$\begin{aligned} F_1 &= k_2 |\dot{\omega}_1| \dot{\omega}_1 \vec{b}_3, \\ F_2 &= -k_2 |\dot{\omega}_2| \dot{\omega}_2 \vec{b}_3, \\ F_3 &= k_2 |\dot{\omega}_3| \dot{\omega}_3 \vec{b}_3, \\ F_4 &= -k_2 |\dot{\omega}_4| \dot{\omega}_4 \vec{b}_3 \end{aligned} \quad (\text{A2})$$

Assume that the magnitude of the generated torque is proportional to the square of the rotation speed. That is,

$$\|\tau_i\| = k_1 \dot{\omega}_i^2, \quad i = 1, \dots, 4 \quad (\text{A3})$$

where k_1 is a positive gain. The torques of the four motors can then be written as

$$\begin{aligned} \vec{\tau}_1 &= -k_1 |\dot{\omega}_1| \dot{\omega}_1 \vec{b}_3, \\ \vec{\tau}_2 &= -k_1 |\dot{\omega}_2| \dot{\omega}_2 \vec{b}_3, \\ \vec{\tau}_3 &= -k_1 |\dot{\omega}_3| \dot{\omega}_3 \vec{b}_3, \\ \vec{\tau}_4 &= -k_1 |\dot{\omega}_4| \dot{\omega}_4 \vec{b}_3 \end{aligned} \quad (\text{A4})$$

One can obtain the relationship between the above-mentioned motor thrusts F_1 – F_4 and the arm length of the quadcopter. The induced torque generated by the outer product of the force arm vector and the thrust is given by

$$\vec{\tau}_{Fi} = \vec{\ell}_{bi} \times F_i = \ell \|F_i\| \vec{b}_i \quad (\text{A5})$$

where ℓ refers to the arm length and $\vec{\ell}_{bi}$ denotes the i th force arm vector. The induced torques of four motors are, respectively, given by

$$\begin{aligned} \vec{\tau}_{F1} &= -\ell(k_2|\dot{\omega}_1|\dot{\omega}_1)\vec{b}_2, \\ \vec{\tau}_{F2} &= \ell(-k_2|\dot{\omega}_2|\dot{\omega}_2)\vec{b}_1, \\ \vec{\tau}_{F3} &= \ell(k_2|\dot{\omega}_3|\dot{\omega}_3)\vec{b}_2, \\ \vec{\tau}_{F4} &= -\ell(-k_2|\dot{\omega}_4|\dot{\omega}_4)\vec{b}_1 \end{aligned} \tag{A6}$$

The rigid body momentum and angular momentum are described by

$$\frac{dp_i}{dt} = \vec{F}_{i,ex}, \quad \frac{dL_i}{dt} = \vec{\tau}_{i,ex} \tag{A7}$$

where p_i is the rigid body momentum, L_i denotes the angular momentum, and the subscript ex represents the external force applied to the rigid body.

Lagrangian mechanics are adopted for the analysis. The configuration space of the quadcopter is denoted as $Q = (R^3 \times SO(3)) \times (R^3 \times SO(3))^4$, where $R^3 \times SO(3)$ and $(R^3 \times SO(3))^4$ refer to the position and direction of the quadcopter and motor frames, respectively. Because the four motors on the quadcopter are fixed on the frame and can only rotate around their respective axis, this means that they have only one degree of freedom relative to the quadcopter frame. Therefore, it can be further simplified to $Q = (R^3 \times SO(3) \times SO(2))^4$.

The quadcopter movement is controlled by the motor rotational speed. $R = [x_b \ y_b \ z_b]$ is in the $SO(3)$ group. It has the following property:

$$-\dot{R}^T R = (\dot{R}^T R)^T \tag{A8}$$

Next, define the angular speed matrix $\hat{\Omega} = -\dot{R}R^T$, where $\Omega = [p \ q \ r]^T$ and \cap denotes the antisymmetric matrix operation.

The kinetic energy of the quadcopter is given by

$$KE = KE_{tran} + KE_{rot} \tag{A9}$$

where the translational kinetic energy is

$$KE_{tran} = \frac{1}{2}M\|v\|_{R^3}^2 \tag{A10}$$

with M being the quadcopter mass and the rotational kinetic energy:

$$\begin{aligned} KE_{rot} &= \frac{1}{2}\int_V |\Omega \times (X_i - X_c)|^2 \rho dv \\ &= \frac{1}{2}G_{R^3}(I_C(\Omega), \Omega) \end{aligned} \tag{A11}$$

where $G_{R^3}(\cdot)$ is the standard inner product in R^3 and $I_C(\Omega)$ is the inertia tensor, ρ is the atmospheric density, X_i is each mass point on the quadcopter, $i = 1, 2, 3 \dots n$, X_c is the center of mass, and $X_i - X_c$ is the position of the small mass dm in the C_{xyz} coordinate system.

Let $V = [\dot{x} \ \dot{y} \ \dot{z}]^T$. We calculate the kinetic energy metric for the manifold by the following equation:

$$G_{ij} = \frac{\partial^2 KE}{\partial v^i \partial v^j} \tag{A12}$$

where v^i and v^j are the velocity parameters of $\dot{x}, \dot{y}, \dot{z}, p, q, r$ on the manifold. That is,

$$G = \text{diag}(MI_3, I_{11}, I_{22}, I_{33}) \tag{A13}$$

where I_{ii} denotes the moment of inertia of the corresponding axis and $\dot{x}, \dot{y}, \dot{z}$ are the respective linear speed of the center of mass along the corresponding axes. As $SO(3)$ is the smooth function, it is a pseudo-Riemannian manifold because of its characteristics. Therefore, one can use a tensor to derive the Levi-Civita connection coordinate and adopt the Christoffel symbols in the holonomic equation:

$$\Gamma_{ij}^k = \frac{1}{2} G^{k\ell} \left(\frac{\partial G_{\ell i}}{\partial x^j} + \frac{\partial G_{j\ell}}{\partial x^i} - \frac{\partial G_{ij}}{\partial x^\ell} \right) \quad (\text{A14})$$

As G is a constant matrix, it represents a flat manifold.

Lagrangian Mechanics

Recalling (A4)–(A6) and (A14), those are the conditions for the Newtonian mechanics that transfer to the Lagrangian mechanics. Let v be the element in the tangent bundle of the manifold of (8). $v_q \in TQ$ is the dual vector in the dual tangent space, $F_{(t,v_q)}$ is the Lagrangian force, and

$$F_{(t,v_q)}(w_q) = G_{R^3}(f_{(t,v_q)}, V_{(w_q)}) + G_{R^3}(\tau_{(t,v_q)}\Omega_{(w_q)}) \quad (\text{A15})$$

where $w_q \in T_qQ$, $\tau_{(t,v_q)}$ was defined in (A3) and $V_{(w_q)}$ and $\Omega_{(w_q)}$ are the velocity and angular speed, respectively. One can expand $F_{(t,v_q)}(\cdot)$ by linear projection:

$$F = F_x dx + F_y dy + F_z dz + F_p dp + F_q dq + F_r dr \quad (\text{A16})$$

Substituting (A1)–(A6) into (18) gives

$$\begin{aligned} F_x &= k_2 G_{R^3}(\vec{b}_3, \vec{e}_1)(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2), \\ F_y &= k_2 G_{R^3}(\vec{b}_3, \vec{e}_2)(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2), \\ F_z &= k_2 G_{R^3}(\vec{b}_3, \vec{e}_3)(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) - Mg, \\ F_p &= \ell k_2(\omega_1^2 + \omega_4^2 - \omega_2^2 - \omega_3^2), \\ F_q &= \ell k_2(\omega_3^2 + \omega_4^2 - \omega_1^2 - \omega_2^2), \\ F_r &= \ell k_1(\omega_1^2 + \omega_3^2 - \omega_2^2 - \omega_4^2) \end{aligned} \quad (\text{A17})$$

where g is the gravitational constant.

References

1. Karar, A.S.; Said, S.; Beyrouthy, T. Pepper Humanoid Robot as A Service Robot: A Customer Approach. In Proceedings of the 2019 3rd International Conference on Bio-engineering for Smart Technologies (BioSMART), Paris, France, 24–26 April 2019.
2. Cuthbertson, A. Watch: Is Google's New Two-Legged Robot the Soldier of the Future? In *Newsweek*; IBT Media: NY, USA, 2016.
3. Ou, S.Q. Vision-based Path Planning and Control of a Mobile Robot Based on DNN Object Recognition and ORB-SLAM2. Master's Thesis, National Chiao Tung University, Taiwan, 2019.
4. Kiss, G. External Manipulation of Autonomous Vehicles. In Proceedings of the 2019 IEEE Smartworld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation, Leicester, UK, 19–23 August 2019.
5. NTU. Singapore and Volvo Unveil World's First Full Size, Autonomous Electric Bus. 2019. Available online: <https://www.volvobuses.com/en/news/2019/mar/volvo-and-singapore-university-ntu-unveil-world-first-full-size-autonomous-electric-bus.html> (accessed on 8 June 2020).
6. Jeong, H.Y.; Song, B.D.; Lee, S. The Flying Warehouse Delivery System: A Quantitative Approach for the Optimal Operation Policy of Airborne Fulfillment Center. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 7521–7530. [CrossRef]
7. Palmer, A. Amazon Wins FAA Approval for Prime Air Quadcopter Delivery Fleet. In *Consumer News and Business Channel*; NBC Universal: Englewood Cliffs, NJ, USA, 2020.
8. Yang, N.K.; San, K.T.; Chang, Y.S. A Novel Approach for Real Time Monitoring System to Manage UAV Delivery. In Proceedings of the 2016 5th IIAI International Congress on Advanced Applied Informatics, Kumamoto, Japan, 10–14 July 2016.
9. Mechali, O.; Xu, L.; Xie, X.; Iqbal, J. Theory and practice for autonomous formation flight of quadrotors via distributed robust sliding mode control protocol with fixed-time stability guarantee. *Control Eng. Pract.* **2022**, *123*, 105150. [CrossRef]

10. Chen, L.; Liu, Z.; Gao, H. Robust adaptive recursive sliding mode attitude control for a quadrotor with unknown disturbances. *ISA Trans.* **2022**, *122*, 114–125. [[CrossRef](#)] [[PubMed](#)]
11. Guo, K.; Jia, J.; Yu, X.; Guo, L.; Xie, L. Multiple observers based anti-disturbance control for a quadrotor UAV against payload and wind disturbances. *Control Eng. Pract.* **2020**, *102*, 104560. [[CrossRef](#)]
12. Qin, X.; Wang, T. Visual-based Tracking and Control Algorithm Design for Quadcopter UAV. In Proceedings of the 2019 Chinese Control and Decision Conference (CCDC), Nanchang, China, 3–5 June 2019.
13. Zhang, W.; Song, K.; Rong, X.; Li, Y. Coarse-to-Fine UAV Target Tracking with Deep Reinforcement Learning. *IEEE Trans. Autom. Sci. Eng.* **2019**, *16*, 1522–1530. [[CrossRef](#)]
14. Wang, Z.; Liu, Z.; Wang, D.; Wang, S.; Qi, Y.; Lu, H. Online Single Person Tracking for Unmanned Aerial Vehicles: Benchmark and New Baseline. In Proceedings of the ICASSP 2019–2019 IEEE International Conference on Acoustics, Speech and Signal Processing, Brighton, UK, 12–17 May 2019.
15. Vasconcelos, F.; Vasconcelos, N. Person-Following Uavs. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision, Lake Placid, NY, USA, 7–10 March 2016.
16. Shen, Q.; Jiang, L.; Xiong, H. Person Tracking and Frontal Face Capture with UAV. In Proceedings of the IEEE 18th International Conference on Communication Technology, Chongqing, China, 8–11 October 2018.
17. Singla, A.; Padakandla, S.; Bhatnagar, S. Memory-Based Deep Reinforcement Learning for Obstacle Avoidance in UAV with Limited Environment Knowledge. *IEEE Trans. Intell. Transp. Syst.* **2019**, *22*, 107–118. [[CrossRef](#)]
18. Hou, J.; Zhang, Q.; Zhang, Y.; Zhu, K.; Lv, Y.; Yu, C. Low Altitude Sense and Avoid for MUAV Based on Stereo Vision. In Proceedings of the 2016 35th Chinese Control Conference, Chengdu, China, 27–29 July 2016.
19. Li, B.W. *Obstacle Detection and Collision Avoidance for Multicopters*; National Central University: Taichung, Taiwan, 2017.
20. Han, D.; Yang, Q.; Wang, R. Three-Dimensional Obstacle Avoidance for UAV Based on Reinforcement Learning and RealSense. *J. Eng.* **2020**, *13*, 540–544. [[CrossRef](#)]
21. Zheng, Z.; Yao, H. A Method for UAV Tracking Target in Obstacle Environment. In Proceedings of the 2019 Chinese Automation Congress, Hangzhou, China, 22–24 November 2019.
22. Wang, Y.; Wang, H.; Lun, Y. Shortest Path Planning of UAV for Target Tracking and Obstacle Avoidance in 3D Environment. In Proceedings of the Chinese Control Conference, Shenyang, China, 27–30 July 2020.
23. Heidari, A.; Navimipour, N.J.; Unal, M.; Zhang, G. Machine Learning Applications in Internet-of-Drones: Systematic Review, Recent Deployments, and Open Issues. *ACM Comput. Surv.* **2022**. [[CrossRef](#)]
24. Tanaka, S.; Asignacion, A.; Nakata, T.; Suzuki, S.; Liu, H. Review of Biomimetic Approaches for Drones. *Drones* **2022**, *6*, 320. [[CrossRef](#)]
25. Guo, K.; Tang, P.; Wang, H.; Lin, D.; Cui, X. Autonomous Landing of a Quadrotor on a Moving Platform via Model Predictive Control. *Aerospace* **2022**, *9*, 34. [[CrossRef](#)]
26. Tian, X.; Jia, Y.; Luo, X.; Yin, J. Small Target Recognition and Tracking Based on UAV Platform. *Sensors* **2022**, *22*, 6579. [[CrossRef](#)]
27. Solak, S.; Bolat, E.D. A new hybrid stereovision-based distance-estimation approach for mobile robot platforms. *Comput. Electr. Eng.* **2018**, *67*, 672–689. [[CrossRef](#)]
28. Li, B.; Wu, Y. Path Planning for UAV Ground Target Tracking via Deep Reinforcement Learning. *IEEE Access* **2020**, *8*, 29064–29074. [[CrossRef](#)]
29. Henkel, P.; Lamm, M.; Mittmann, U.; Fritzel, T.; Strauß, R.; Steinert, H.-J.; John, M. Verification of RTK Positioning of UAVs with High-Precision Laser Tracker. In Proceedings of the 2022 16th European Conference on Antennas and Propagation (EuCAP), Madrid, Spain, 27 March–1 April 2022.
30. Gao, S.; Zhang, T. *14 Lectures on Visual SLAM*; Publishing House of Electronics Industry: Beijing, China, 2017.
31. Zhang, Z. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 1330–1334. [[CrossRef](#)]
32. Redmon, J.; Farhadi, A. Yolov3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
33. Chen, W.C. Design and Implementation of an Intelligent UAV. Master's Thesis, National Chung Hsing University, Taiwan, 2021.
34. Huang, Q. *Mathematical Modeling of Quadcopter Dynamics*; Rose-Hulman Scholar: Terre Haute, IN, USA, 2016.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.