Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 6902348, 10 pages
https://doi.org/10.1155/2022/6902348

Research Article

OTCS: An Online Target Close-Up Shooting Method Based on the UAV Image System

Wentao Wang,[1] **Huibin Wang**,[2] **Xuzhou Shi**,[1] **and Ming Chen**[1]

[1]*College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China*
[2]*College of Computer Science, Chuzhou University, Chuzhou 239099, China*

Correspondence should be addressed to Ming Chen; mingchennj@163.com

Received 27 October 2021; Revised 25 February 2022; Accepted 30 March 2022; Published 2 May 2022

Academic Editor: Yingjie Wang

Unmanned aerial vehicles (UAV) equipped with intelligent gimbaled cameras can take images and identify specific targets from them in real-time. However, the targets in the images are generally small and difficult to be seen. This paper proposes an Online Target Close-up Shooting (OTCS) method to solve the problem of taking high-definition (HD) close-up images of the targets. Firstly, we build an online target close-up shooting model, which uses deep learning (DL) algorithm to identify the targets from the images taken online. Then, an intelligent control algorithm is presented to control the gimbaled camera to take a close-up shot of the target, which considered the target being fixed on the ground statically and moving at an ununiform speed. Finally, we build a UAV-based prototype system and conduct a series of experiments to verify our proposal. Experimental results show that our proposed method is feasible and outperforms traditional methods in effectiveness.

1. Introduction

Aerial photography is an important function of unmanned aerial vehicles (UAV), which plays a crucial role in many applications, such as news reporting, rescuing, forest fire monitoring, transmission pipeline inspection, and traffic monitoring. Thanks to advantages including flexibility and high effectiveness, this image system technology enabled by UAV could expand the task coverage of the base stations on the ground, where UAVs are equipped with photography equipment [1, 2].

There is increasingly more research work on UAV image systems (e.g., [3–8]). The systems could currently use object detection algorithms to accurately identify specific targets and monitor their status. For example, [3] develops an advanced vehicle detection method to improve the original Viola-Jones object detection scheme for better vehicle detections from low altitude unmanned aerial vehicle imagery. [4] efficiently interleaves a fast keypoint tracker and presents a real-time approach for image-based localization within large scenes. [5] proposes a long-term tracking method on the basis of a multifeature coded correlation filter for vision-based UAV flocking control. However, the existing works on UAV image system overlook the target close-up shooting application to detect and take images of the specific targets. How to further obtain more pixel information of objects after detection is still a problem. Unlike UAV-based searching (e.g., [9, 10]), the purpose of this type of task is not only to check whether there are specific targets but also to take close-up images for more details of them.

On the other hand, traditional UAV aerial photography methods generally adopt the offline processing mode, i.e., the images taken by UAV are processed after the UAV landed. However, offline processing mode cannot meet the requirements of some urgent or real-time tasks, such as border patrol, surveillance, and power line inspection. These tasks usually adopt the online processing mode; that is, the images taken by UAVs during flights are processed in real-time. Online methods can be processed in ground stations, edge nodes, and on-board computers. Each of them has its own advantages and disadvantages. Ground stations and edge nodes could provide higher computing efficiency, but

the data links with the UAVs may be unstable in harsh environments. On-board computing, a form of computing that is done on site could minimize the data transmission delay at the expense of computing efficiency [11, 12].

Given the aforementioned requirements, we advance the research on the control method of UAV-based Online Target Object Close-up Shooting (OTCS). In addition to real-time object detection, the system is required to automatically adjust parameters such as the focal length of the camera, so as to take a high-definition (HD) of the target object with the largest possible area and as many pixels as possible (referred to as close-up images). We use YOLOv3 algorithm to detect the target object from the input image [13]. Gimbaled camera, an electronic camera supported by a three-axis pan-and-tilt, is mounted on the UAV to take images. People can control the direction of the camera by deflecting the pan-and-tilt and then adjusting the parameter of the camera such as focal length to change the camera screen [12, 14]. However, the gimbaled camera is mainly controlled by people, and there are still the following problems in intelligent object close-up: (1) lack of a system model that supports object close-up, (2) lack of pan-and-tilt deflection and camera zooming algorithm for static object close-up, and (3) lack of dynamic gimbaled camera control algorithm for dynamic object close-up.

In this backdrop, the technical route in this paper is as follows: first, the trained YOLOv3 algorithm is adopted to detect the target object from the taken images and build the relative positioning model of the object; the second is to study the intelligent control of the gimbaled camera under static scenarios and take HD images with more pixels of the object; the third is to study the close-up technology of intelligent control of the gimbaled camera under dynamic scenarios to further improve the efficiency of object searching. Moreover, a prototype system needs to be developed to verify the above theoretical results. Our main contributions in this paper are threefold: (1) A novel object close-up system model is put forward. (2) Intelligent gimbaled camera control algorithms in dynamic and static scenarios are presented to realize object close-up in a UAV-based system. (3) Extensive experiments are conducted to verify the effectiveness of our proposal. The remaining of this paper is organized as follows. Section 2 summarizes related work. Section 3 states the system on the basis of the YOLOv3 algorithm and introduces the gimbaled camera model. Section 4 presents our intelligent gimbaled camera control algorithms for objection detection and close-up in static and dynamic scenarios. Section 5 implements the prototype system and tests the gimbaled camera control algorithm. Section 6 concludes this paper briefly.

2. Related Work

In recent years, UAV technology has made great progress; there have been many studies on UAV image applications. The existing research on UAV image system focus on detection optimization, object tracking, and vision positioning. Firstly, many efforts for visual object detection have been conducted [15]. Rozantsev et al. investigated the problem of detecting flying objects with a single moving camera. And a regression-based approach for object-centric motion stabilization of image patches is proposed, which can achieve effective classification on spatio-temporal image cubes [9]. Dasgupta proposed a multiagent-based prototype system that uses swarming techniques inspired from insect colonies to perform automatic target recognition using UAVs. They presented algorithms for the different operations performed by the UAVs in the system and for different swarming strategies, which are embedded within software agents located on the UAVs [16]. Currently, deep learning (DL) technology has developed greatly. Compared with traditional object detection algorithms, Convolutional Neural Networks (CNN) have shown great advantages in accuracy and speed [17]. Therefore, CNN has been widely adopted in the field of computer vision [13]. Zhao et al. have adopted the on-board computing to meet real-time application requirements and used YOLOv3 algorithm to identify the vehicle in the images, but it does not consider further obtaining more object information [10]. Zhao et al. proposed a UAV inspection system, composed of a splicing module and a detection module. The splicing module obtains the video collected by the UAV camera, selects the frames in the video to splice into a panoramic image, and transmits it to the detection module. The detection module runs the Faster-RCNN algorithm to detect the object and returns a panoramic image with the detected object highlighted using the bounding box [18]. Furthermore, Zhang et al. presented SlimYOLOv3 with fewer trainable parameters and floating point operations (FLOPs) in comparison of original YOLOv3 as a promising solution for real-time object detection on UAVs [19]. To accomplish reliable pedestrian detection using unmanned aerial vehicles (UAVs) under night-time conditions, Wang et al. developed an image enhancement method to improve the low-illumination image quality [20].

Related to the real-time object tracking and vision positioning, there are only a few works. Tang et al. developed an integrated framework of tracking-learning-detection on the basis of multifeature coded correlation filter has been developed. To achieve long-term tracking, a redetector is trained online to adaptively reinitialize target for global sensing [5]. Liang et al. used the detection results to initialize the object tracker based on kernelized correlation filtering and go on to modify the tracking results. In the tracking process, a camera motion compensation strategy that adapts to the texture of the observation scenario is introduced to achieve target relocation, which enhances the robustness of the detection algorithm in complex scenarios [21]. Chen et al. put forward an object tracking control method, which divides the target tracking problem into three modes: object searching, object tracking, and object loss. For each mode, the corresponding control strategy is designed to realize the switch between different modes [22]. Zou et al. establish an on-board pan-tilt camera control system based on biomimetic eye, which can compensate the deflection caused by the UAV rotation and the movement of ground relative to the UAV [23]. However, the above methods about UAV image systems cannot apply to our target close-up problems, where the identification algorithm should be used to detect

the specific object and the gimbaled camera control should take into account object positioning and tracking. Compared with existing efforts listed above, our proposal in this paper could obtain close-up images with more information while detecting the target. Moreover, the computing modes of the UAV image system are compared and analyzed.

3. System Model

3.1. OTCS System. As shown in Figure 1, the OTCS system includes the following components: controller, pan-and-tilt, camera, and analyzer. The controller is the core of the OTCS, which coordinates the operation of various components of the system. Pan-and-tilt is used to carry the camera and control the direction of the camera. The camera is used to take images, and the analyzer is a computing device to detect whether an image contains specific objects. The OTCS system is scheduled by controller, which can call the controlling pan-and-tilt (PTC) and the controlling camera (CC) algorithm. The analyzer can run the object detection algorithm based on DL and output position parameters, which is used to control the pan-and-tilt camera. Gimbaled camera, i.e., pan-and-tilt and camera are connected with controller. Therefore, controller can control pan-and-tilt deflection to change the direction of the camera and adjust its focal length. The camera transmits the taken image to analyzer, which analyzes the image and the results are then be fed back to controller.

The basic principle of OTCS is as follows. Firstly, detect the target object O and compute relative position based on the YOLOv3 algorithm [5]. Secondly, control pan-and-tilt deflection and camera and take images with more pixels of object O_s to obtain more information. As shown in Figure 2, the process of OTCS is as follows.

(1) Controller controls camera to take pictures of P_0, where pan-and-tilt and camera are controlled based on specific parameters

(2) Controller sends P_0 to analyzer. Analyzer analyzes P_0 by DL algorithm. If P_0 contains suspected object O_s, analyzer will output the position coordinates in the image and confidence ε or it will return

(3) Controller calls controlling pan-and-tilt algorithm according to the position coordinates to control pan-and-tilt deflection so that the object can appear in the center of the camera screen

(4) Controller calls controlling camera algorithm to adjust parameters including focal length of camera to maximize the object in the image and take image P_{sm}

(5) Controller sends P_{sm} to analyzer. P_{sm} will be kept if the confidence is greater than ε_2. Then, the system restores the original parameters and waits for new instructions

3.2. Object Detection. We adopt the YOLOv3 algorithm to analyze the original image P_0 taken by the gimbaled camera. YOLOv3 has been trained with a large number of samples before processing and can quickly detect the object contained in P_0. The Darknet-53 network is used as the feature extractor, and it improves the inability of previous versions to accurately identify small objects [13].

YOLOv3 predicts bounding boxes using dimension clusters as anchor boxes. The output parameters include IsExist, ε and P_{size}, and Q_{all}. IsExist = true indicates that P_0 contains the suspected object O_s, and ε is the confidence of detection. As shown in Figure 3, P_{size}, i.e., $X_{\max} \times Y_{\max}$, is the size of P_0, and $Q_{lu}(x_{lu}, y_{lu})$, $Q_{ld}(x_{ld}, y_{ld})$, $Q_{rd}(x_{rd}, y_{rd})$, and $Q_{ru}(x_{ru}, y_{ru})$ indicate the vertices of the bounding box. The offset error of gimbaled camera is μ. Thus, we have:

$$\begin{cases} \text{IsExist} = \text{true}, P_0 \text{ contains } O_s, \\ \text{IsExist} = \text{false}, P_0 \text{ not contains } O_s. \end{cases} \tag{1}$$

If O_s lies in the center of P_0, we have

$$\begin{cases} \dfrac{X_{\max}}{2} - \mu < x_{lu} + \dfrac{x_{ru} - x_{lu}}{2} < \dfrac{X_{\max}}{2} + \mu \\ \dfrac{Y_{\max}}{2} - \mu < y_{ld} + \dfrac{y_{lu} - y_{ld}}{2} < \dfrac{Y_{\max}}{2} + \mu \end{cases}. \tag{2}$$

There is no need to zoom to enlarge O_s if it can fill the entire image, which is expressed as follows:

$$\begin{cases} \mu < x_{lu} < 2\mu, Y_{\max} - 2\mu < y_{lu} < Y_{\max} - \mu \\ \mu < x_{ld} < 2\mu, \mu < y_{ld} < 2\mu \\ X_{\max} - 2\mu < x_{rd} < X_{\max} - \mu, \mu < y_{rd} < 2\mu \\ X_{\max} - 2\mu < x_{ru} < X_{\max} - \mu, Y_{\max} - 2\mu < y_{ru} < Y_{\max} - \mu \end{cases}. \tag{3}$$

4. Gimbaled Camera Control Algorithm

4.1. Object Positioning Model. The gimbaled camera control of OTCS is divided into two parts, i.e., pan-and-tilt deflection and camera zooming control. The pan-and-tilt deflection determines the orientation of the camera, and the zooming determines the range of the camera [24]. Pan-and-tilt deflection is determined by course angle and pitch angle. As Figure 4 depicts, the course shaft is perpendicular to the ground and the rotation angle, i.e., course angle is in the range of 0 to 360 degrees. Correspondingly, the pitch shaft is perpendicular to the course shaft, and the pitch angle is in the range of -90 to 90 degrees.

After detecting the object, the relative position is computed according to the output. As shown in Figure 5, P_0's size is $X_{\max} \times Y_{\max}$, and its center coordinate m is $(X_{\max}/2, Y_{\max}/2)$. n is the center of detection box of YOLOv3, whose coordinate is $(x_{ld} + ((x_{ru} - x_{ld})/2), y_{ld} + ((y_{ru} - y_{ld})/2))$. The pixel distance between two center points is d_{mn}. The system obtains the relative position by computing the offset of the object in the camera screen. d_{mn} can be decomposed into

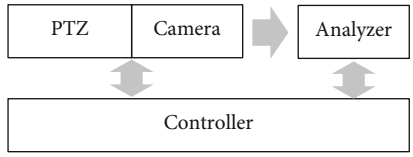


Figure 1: Main components of OTCS and relationships.

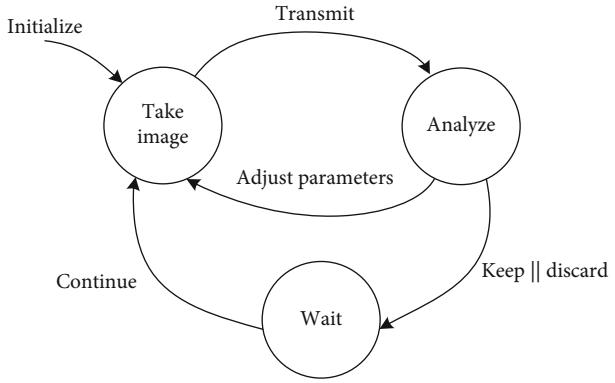


Figure 2: The finite state automaton of OTCS.

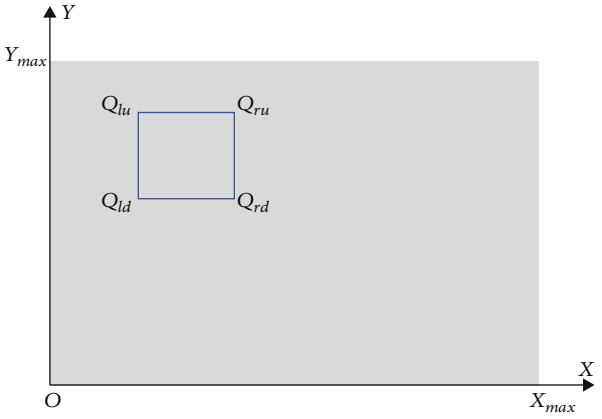


Figure 3: The pixel coordinate of the image.

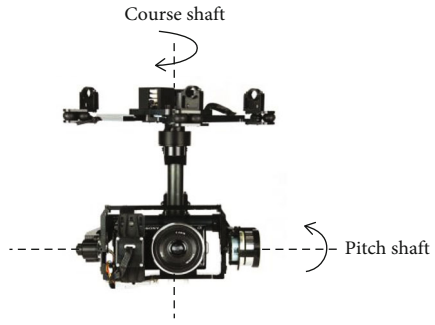


Figure 4: Pan-and-tilt posture.

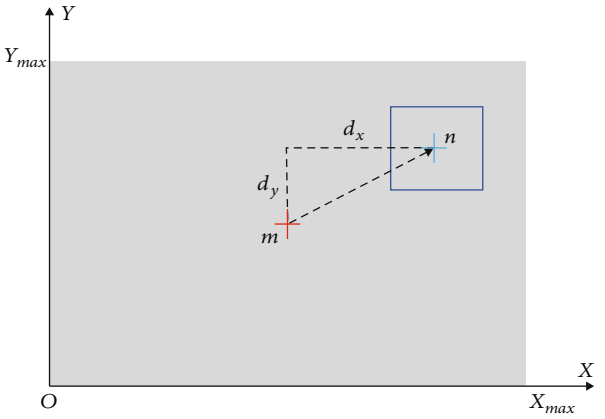


Figure 5: The location of the target object in the image.

two components, i.e., d_x and d_y, which correspond to the deflection of the course and pitch angle, respectively.

$$d_x = x_{ld} + \frac{x_{ru} - x_{ld}}{2} - \frac{X_{\max}}{2}, \tag{4}$$

$$d_y = y_{ld} + \frac{y_{ru} - y_{ld}}{2} - \frac{Y_{\max}}{2}. \tag{5}$$

Since d_{mn}, d_x, and d_y are pixel distances, and we need an actual distance between the object and the camera. It is necessary to reduce the computation error. We place an object with a known height of A vertically at the distance L in front of the camera; the center of the object is on the vertical line between the optical center and the object plane. We take an image of the object, and the pixel height of the object in the image is a. As shown in Figure 6, the focal length f of the camera is known; then, D_L can be computed according to the camera model:

$$D_L = \frac{a \times f}{A}. \tag{6}$$

As shown in Figure 6(b), D_L is the distance between the center of object O_s. Therefore, the actual distance D_{mn}, D_X, and D_Y, corresponding to d_{mn}, d_x, and d_y, are expressed as follows:

$$\begin{cases} D_{mn} = \dfrac{d_{mn} \times D_L}{f}, \\ D_L = \sqrt{D^2 - D_{mn}^2}, \end{cases}$$

$$D_L = \sqrt{\frac{D^2}{1 + \left(d_{mn}^2 / f^2\right)}}, \tag{7}$$

$$\begin{cases} D_X = \dfrac{d_x \times D_L}{f}, \\ D_Y = \dfrac{d_y \times D_L}{f}. \end{cases}$$

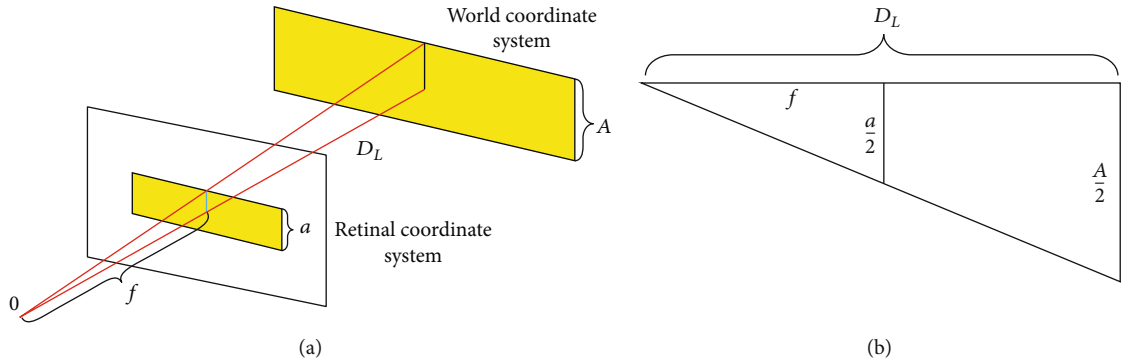


FIGURE 6: Image principle of the object.

4.2. Static Object Close-Up. In order to obtain HD images of the target object, we first analyze the scenario of static object close-up while the UAV is hovering at a certain height in the air. After the analyzer recognizes the target, controller controls the pan-and-tilt deflection according to its output information so that the camera can aim at O_s. Controller calls PTC and CC algorithm to control gimbaled camera according to the computed parameters.

According to the relative position of the object, adjusting the pan-and-tilt and camera focal length is needed to take HD images. As shown in Figure 7, D_L is perpendicular to D_X, and the angle α is the deflection angle.

$$\alpha = \arctan \frac{D_X}{D_L}. \tag{8}$$

Similarly, pitch angle deflection can be computed:

$$\beta = \arctan \frac{D_Y}{D_L}. \tag{9}$$

Then, the object is in the center of the screen after pan-and-tilt deflection. In order to obtain images with larger object and improve definition, the object in the image needs to be enlarged to the appropriate size via zooming. If the object, i.e., O_s, meets the Equation (5), the largest image can be obtained. The principle of camera imaging is shown in Figure 8.

f and D are the initial focal length and the distance between the camera and the object, respectively. $Y'/2$ is half of the Y-axis direction of the recognition frame, corresponding to half the height of the object, i.e., $h/2$. Considering the error of deflection, the minimum length of the detection box in Y-axis direction is

$$Y' - 2\mu - \mu. \tag{10}$$

The desired focal length can be expressed as follows:

$$f' = \frac{\left(\left(Y' - 3\mu\right)/2\right) \times D}{h/2}. \tag{11}$$

During the process of static object close-up, the relative position between the gimbaled camera and the object is fixed. After detecting the target object, controller calls the PTC algorithm and CC algorithm based on the parameters computed based on Equations (6)–(11). Algorithm 1 depicts this process.

4.3. Dynamic Object Close-Up. In order to improve the efficiency of object close-up, it is required that the system complete the close-up while the object is moving, not after the object has stopped. Since the target object can be detected by the analyzer when it appears on the camera screen, we use Kalman filtering to predict the motion state of the target and control gimbaled camera for a dynamic close-up. The Kalman filter is an algorithm that estimates the state of a system from measured data, which can optimally predict the state of the target by using a minimum-variance estimator [25, 26]. The Kalman filtering algorithm includes statement equation and measurement equation:

$$\begin{cases} x(k+1) = \Phi(k+1) \bullet x(k) + \Gamma(k+1) \bullet w(k), \\ y(k+1) = H(k+1) \bullet x(k+1) + v(k+1), \end{cases} \tag{12}$$

where $x(k)$ is the state of the system at time k, y is measurement vector, and w and v are state and measurement noise. Φ, Γ, and H are defined as the state transition matrix, noise transition matrix, and measurement matrix, respectively.

After obtaining the value of measurements, considering the estimation error, we have the Kalman filter equation in iterative form:

$$\begin{cases} \hat{x}(k|k) = \Phi(k) \bullet x(k-1|k-1) + K(k) \bullet [y(k) - H(k) \bullet \Phi(k) \bullet x(k-1|k-1)], \\ K(k) = P(k, k-1) \bullet H^T(k) \bullet \left[H(k) \bullet P(k, k-1) \bullet H^T(k) + R(k)\right]^{-1}, \\ P(k|k-1) = \Phi(k) \bullet P(k-1|k-1) \bullet \Phi^T(k) + \Gamma(k, k-1) \bullet Q(k-1) \bullet \Gamma^T(k, k-1), \\ P(k|k) = [I - K(k) \bullet H(k)] \bullet P(k|k-1)\,; k = 0, 1, \cdots \end{cases} \tag{13}$$

Equation (13) fall into two groups: time update equations and measurement update equations. The time update

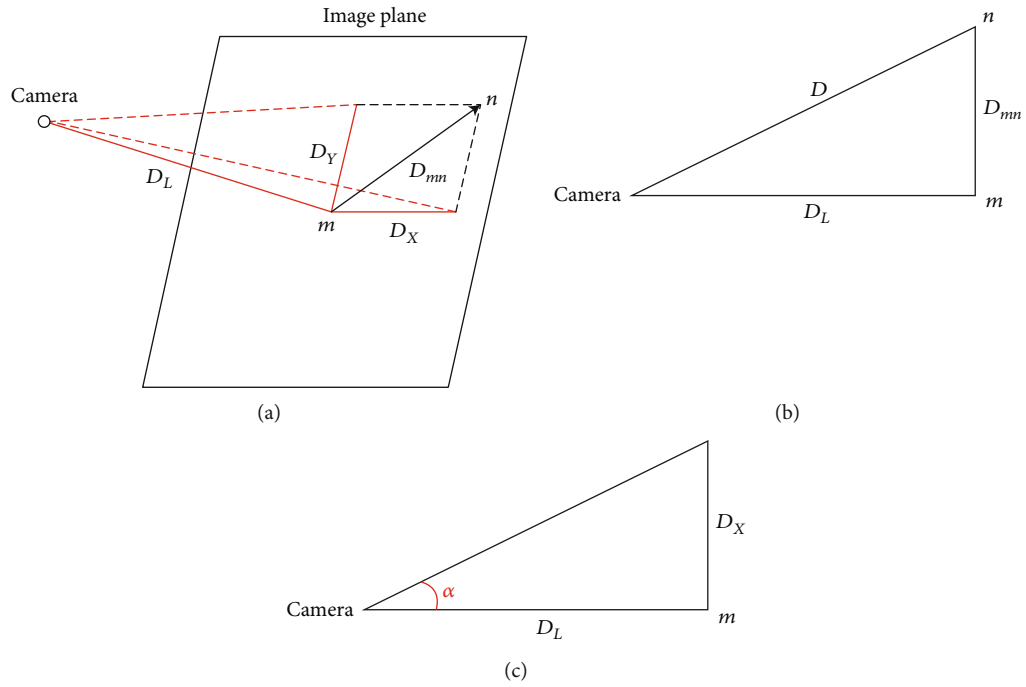


Figure 7: The computation of angles.

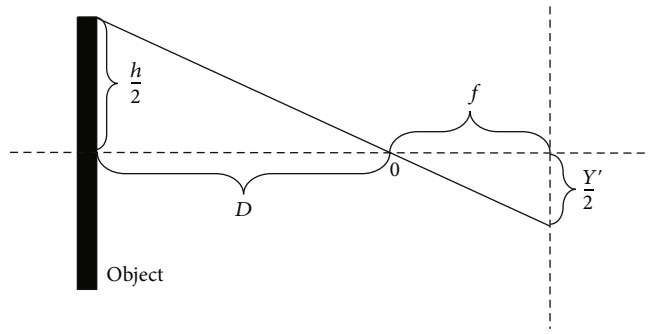


Figure 8: Camera imaging.

Input: Original image, Camera parameters
Output: Close-up image of the object
1: Run YOLOv3
2: **while** *IsExist* = *true* **do:**
3: Compute relative position parameter D_X, D_Y, D_L according to (9)
4: Compute α, β according to (10), (11)
5: PTC α, β
6: Compute f' according to (12)
7: CC zoom f' and focus
8: Take image P_{sm}
9: **if** $P_{sm}\, confidence > \varepsilon_2$ **do:**
10: Output P_{sm}
11: **end if**
12: **end while**

Algorithm 1: Static object close-up algorithm.

equations are responsible for projecting forward the current state and error covariance estimates to obtain the a priori estimates for the next time step. The Kalman filter computes a Kalman gain for each new measurement that determines how much the input measurement will influence the system state estimate. In other words, when a noisy measurement comes in to update the system state, the Kalman gain will trust its current state estimate more than this new inaccurate information, where $K(k)$ is the Kalman gain matrix and $P(k|k-1)$ is the prediction error covariance matrix. $\hat{x}(k|k)$ denotes the estimator based on given measurements; thus, we can predict the location of the target at time $k+1$:

$$\hat{x}(k+1|k) = \Phi(k+1) \bullet \hat{x}(k|k). \tag{14}$$

The state of the target is set as $x = (Q_{lu}, Q_{rd}, v_p)$, where Q_{lu} and Q_{rd} are the upper left and lower right vertices of the bounding box, respectively. $v_p = (v_x, v_y)$ is the pixel speed of the target in the screen. Then, we can use the Kalman filter to predict the location $\hat{x}$ and compute the deflection of the pan-and-tilt, i.e., $\widehat{\alpha}$ and $\widehat{\beta}$ based on (8)–(9). In order to keep the target tracking, we set angular velocity to control the pan-and-tilt. Let the interval of taking images be Δt, the optimal deflection velocity is

$$\begin{cases} \omega_\alpha = \dfrac{\Delta\alpha}{\Delta t} = \dfrac{\widehat{\alpha} - \alpha_0}{\Delta t}, \\ \omega_\beta = \dfrac{\Delta\beta}{\Delta t} = \dfrac{\widehat{\beta} - \beta_0}{\Delta t}, \end{cases} \tag{15}$$

where α_0 and β_0 are current deflection angle. The process of camera zooming may lead to the loss of the target. Therefore, we present a detection-tracking-shooting policy, which is to control the camera to take close-up image when the speed of the target is lower than 2 meters/second. The speed $\hat{v}$ will be computed through the relative position.

The dynamic close-up control algorithm is illustrated in Algorithm 2. Steps 1-2 get the measurements value through YOLOv3 algorithm. Steps 2-5 predict and track the target through the Kalman filter. Steps 6-11 compute control parameters and control the gimbaled camera to take close-up images. Finally, the image will be analyzed whether it can meet the requirement.

5. Experiments and Analysis

5.1. Experimental Prototype. In order to verify the feasibility of the OTCS and the performance of the gimbaled camera control algorithm presented in this paper, the prototype is built based on DJI Matrice 100 with an assembled communication module and gimbaled camera with network interfaces, and the algorithms are implemented on the onboard computer, i.e., NVIDIA Jetson Xavier NX. Moreover, the real-time images are transmitted to the ground station via the 4G network. The on-board computer controls the gimbaled camera via sending data messages. The camera supports 3.5x optical zoom and 5x digital zoom. A training dataset for object detection needs to be established for the efficiency of YOLOv3. To detect the object as accurately as possible from the image, we take a total of 2000 images from diverse angles and lightning conditions label them. Then, the labeled data set is divided into training set, validation set, and test set in a 6 : 2 : 2 scale.

In the test, the controller sends instructions to the camera through the HTTP protocol and saves the captured images on the web page. Then, controller extracts the feedback results from the website to obtain the captured pictures. The lens of the camera is first aligned at any position, and the object is placed to ensure that it can be photographed. The distance between the target object and the gimbaled camera is measured to simulate the GPS ranging function of the UAV. The test starts after placement. We use laptops and hosts equipped with high-performance graphics cards as edge nodes and ground stations, respectively.

5.2. Result Analysis. In the experiments, we first analyze the performance of three computing modes, in which data is processed in the UAV, edge node, and ground station. Since the delay of gimbaled camera control is decided by the mechanical structure, we only test the system delay. The default resolution of the images taken by the camera unit of this system is 4096×2160. Considering the needs of various real-time applications, images with 5 different resolutions, i.e., 640×480, 1280×960, 1600×1200, 2560×1920, and 4096×2160, are tested. Figure 9 shows images' processing delay of three different computing modes. As the definition of the photo increases, the system delay increases. Due to the increase of definition, YOLOv3 needs more time to detect. And it can be seen that the image resolution has a great effect on the processing time in the on-board computing. In edge computing, the resolution has less effect because ground station can provide more computing power. And it is almost unaffected due to the powerful computing power of the ground station in cloud computing.

Secondly, images need to be transmitted to edge node and ground station for processing in edge computing and cloud computing modes. We test the transmission delay of this process, which is shown in Figure 10. 4G communication is adopted in cloud computing, and we assume UAV moves within 4G coverage. Therefore, the transmission efficiency would not be affected by the distance between UAV and ground station. On the other hand, we adopt Wi-Fi in edge computing, and the delays at distances of 5 meters and 20 meters are given. It can be found that distance has a significant impact on Wi-Fi. When the distance reaches 20 meters, the delay is close to 4G, which weakens the advantages of edge computing over cloud computing.

Figure 11 shows the total computing delay of three computing modes, and the distance of Wi-Fi is 5 meters. We can see that the total delay of the on-board computing and edge computing using Wi-Fi transmission does not change much with the increase of the image resolution. However, the total computing delay of cloud computing using 4G increases rapidly with the increase of image resolution because of limited network bandwidth. Therefore, OTCS using on-

Input: Original image, Camera parameters
Output: Close-up image
1: Run YOLOv3 Algorithm // Get the measurements value
2: **while** *IsExist* = *true* **do:**
3: Compute optimal estimator according to (13)
4: Predict the state $\hat{x}$ according to (14)
5: Update Kalman filter
6: Compute $\omega_\alpha, \omega_\beta$ according to (15)
7: PTC $\omega_\alpha, \omega_\beta$
8: **if** $\hat{v} < 2m/s$ **do:**
9: CC zoom f' and focus
10: Take image P_{sm}
11: **end if**
12: **if** $P_{sm}\, confidence > \varepsilon_2$ **do:**
13: Output P_{sm}
14: **end if**
15: **end while**

Algorithm 2: Dynamic object close-up control.

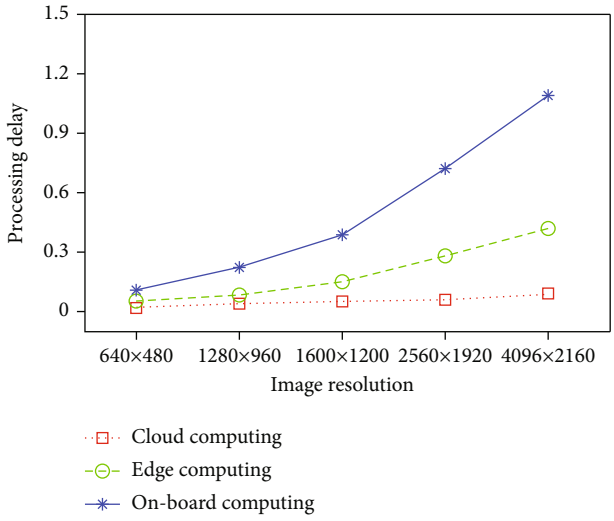


Figure 9: Processing delay.

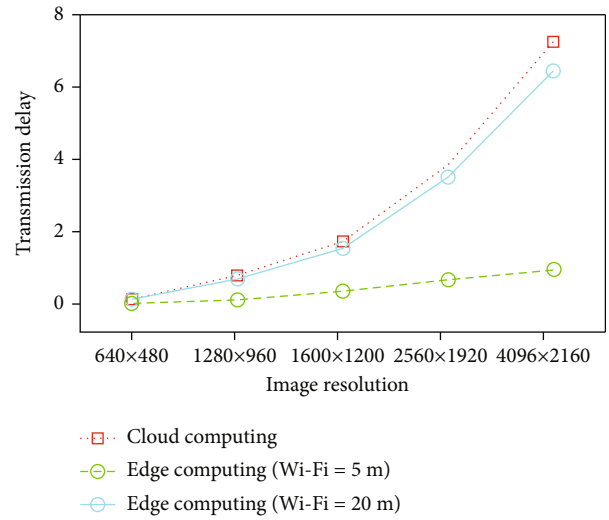


Figure 10: Transmission delay.

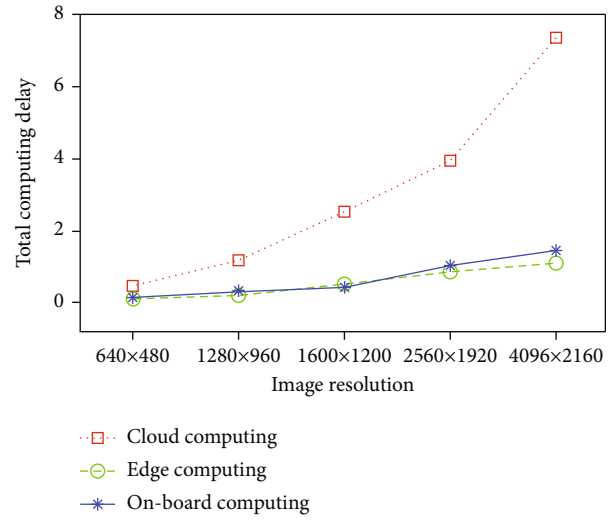
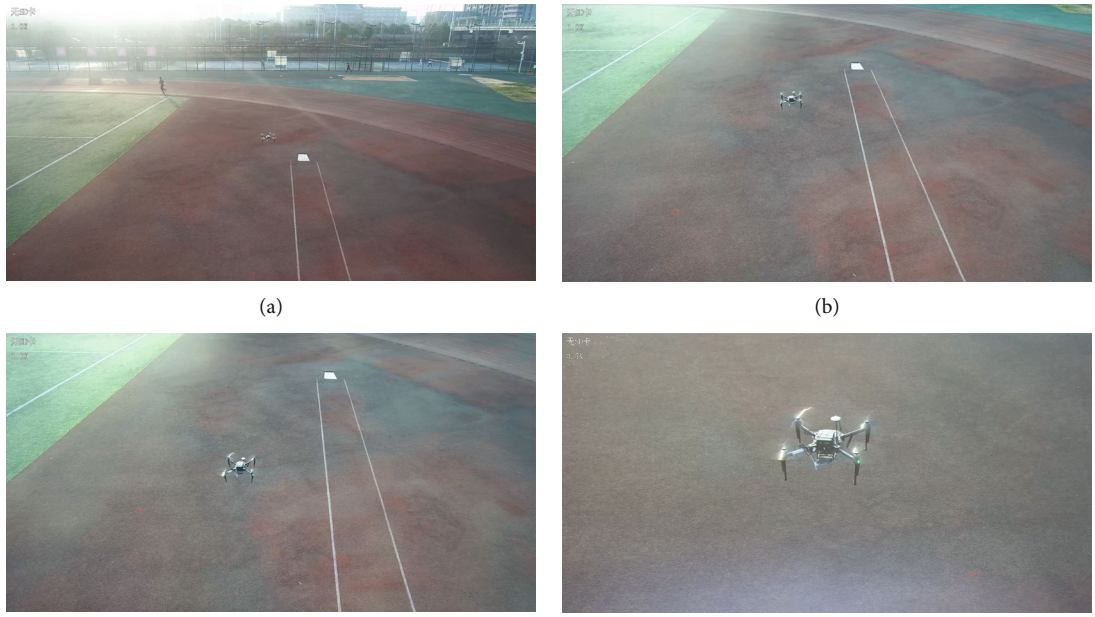


Figure 11: Total computing delay.

board computing can provide reliable computing power and is not affected by transmission distance. In contrast, the performance of edge computing and cloud computing depends more on UAV's communication capabilities.

We further test the dynamic control algorithm presented in this paper. Figure 12 shows the four snapshots during this process. The optical zoom of the camera is 3.5 times, and the image resolution is 4096 × 2160. The target object is a UAV, which moves at a nonuniform speed relatively. As shown in Figure 12(a), analyzer detects the object and computes the relative position of the object. Then, controller calls PTC algorithm so that pan-and-tilt starts deflecting. We can see that the gimbaled camera keeps track of the target object during its movement and takes a close-up image in Figures 12(b) and 12(c). Note that the screen is not zoomed in at first that is to avoid losing the target caused by its moving. In Figure 12(d), the object has been maximized in the image. Due to computing errors and the instability of the UAV flight process, the object does not appear in the center of the image.

We also compared our proposed prediction-based detection-tracking-shooting control method and traditional detection-shooting method, which take images after detection directly [11]. The UAV equipped with gambled camera is hovering at a height of 3 meters. The target object is initially placed where the camera can detect and then let it move. Tables 1 and 2 show the comparison results with different speeds. It can be seen that the success rate of the detection-shooting control is lower when the target moves fast. This is because camera zooming would cause the detection range to shrink, resulting in the loss of the target object. In contrast, our proposed algorithm could predict the target's position and take images when it moves slower, which could avoid object loss and increase the close-up rate.



(a) (b) (c) (d)

Figure 12: Image stream of dynamic close-up.

Table 1: Success rate of detection-tracking-shooting method.

Moving speed (m/s)	Repeating times	Close-up rate (%)	Object loss rate (%)
<1	20	100.0	0.0
1 ~ 3	20	85.0	5.0
>3	20	60.0	20.0

Table 2: Success rate of detection-shooting method.

Moving speed (m/s)	Repeating times	Close-up rate (%)	Object loss rate (%)
<1	20	80.0	20.0
1 ~ 3	20	15.0	85.0
>3	20	0.0	100.0

6. Conclusions

UAV-based object detection is a typical UAV image application. Obtaining additional image information of the target object after detection is of great practical importance. To achieve this goal, a UAV-carried gimbaled camera control model is presented in this paper. In the control model for the image systems, we analyze two scenarios, i.e., static and dynamic close-ups. Then, two control algorithms are put forward. The former could take static close-up images, while the latter takes close-up images in the latter scenario by estimating the relative position. A series of simulation experiments with diverse parameter settings is conducted based on the constructed prototype. Experiment results have shown that our gimbaled camera control algorithms could effectively obtain close-up images in diverse scenarios. In the future, we will study the application scenario of UAV-carried gimbaled camera to detect multiple objects and obtain close-up images.

Data Availability

All data included in this study are available upon request by contact with the corresponding author.

Conflicts of Interest

We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant 61772271.

References

[1] B. Liu, Q. Zhu, and H. Zhu, "Trajectory optimization and resource allocation for UAV-assisted relaying communications," *Wireless Networks*, vol. 26, no. 1, pp. 739–749, 2020.

[2] T. Liu, M. Cui, G. Zhang, Q. Wu, X. Chu, and J. Zhang, "3D trajectory and transmit power optimization for UAV-enabled multi-link relaying systems," *IEEE Transactions on Green Communication and Networking.*, vol. 5, no. 1, pp. 392–405, 2021.

[3] Y. Xu, G. Yu, X. Wu, Y. Wang, and Y. Ma, "An enhanced Viola-Jones vehicle detection method from unmanned aerial vehicles imagery," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 7, pp. 1845–1856, 2017.

[4] H. Lim, S. N. Sinha, M. F. Cohen, and M. Uyttendaele, "Real-time image-based 6-DOF localization in large-scale

environments," *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2012, pp. 1043–1050, 2012.

[5] Y. Tang, Y. Hu, J. Cui et al., "Vision-aided multi-UAV autonomous flocking in GPS-denied environment," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 1, pp. 616–626, 2019.

[6] J. Biswas and M. Veloso, "Depth camera based indoor mobile robot localization and navigation," in *2012 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1697–1702, Saint Paul, MN, USA, 2012.

[7] X. Xiang, M. Zhai, N. Lv, and A. el Saddik, "Vehicle counting based on vehicle detection and tracking from aerial videos," *Sensors*, vol. 18, no. 8, pp. 2560–2576, 2018.

[8] T. Tang, Z. Deng, S. Zhou, L. Lei, and H. Zou, "Fast vehicle detection in UAV images," *International Workshop on Remote Sensing with Intelligent Processing (RSIP)*, vol. 2017, pp. 1–5, 2017.

[9] A. Rozantsev, V. Lepetit, and P. Fua, "Detecting flying objects using a single moving camera," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 5, pp. 879–892, 2017.

[10] X. Zhao, F. Pu, H. Wang, H. Chen, and Z. Xu, "Detection, tracking, and geolocation of moving vehicle from UAV using monocular camera," *IEEE Access*, vol. 7, pp. 101160–101170, 2019.

[11] X. Dou, M. Chen, B. Chen, and Y. Xu, "Research on computing models of systems for realtime image applications based on UAV," *Journal of Chinese Computer Systems*, vol. 30, 2020.

[12] C. Martínez, I. F. Mondragón, M. Olivares-Méndez, and P. Campoy, "On-board and ground visual pose estimation techniques for UAV control," *Journal of Intelligent and Robotic Systems*, vol. 61, no. 1–4, pp. 301–320, 2011.

[13] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, Las Vegas, 2016.

[14] L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys, "PIXHAWK: a system for autonomous flight using onboard computer vision," in *Proc. IEEE Int. Conf. Robot. Autom*, pp. 2992–2997, Shanghai, 2011.

[15] W. Diao, X. Sun, X. Zheng, F. Dou, H. Wang, and K. Fu, "Efficient saliency-based object detection in remote sensing images using deep belief networks," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 2, pp. 137–141, 2016.

[16] P. Dasgupta, "A multiagent swarming system for distributed automatic target recognition using unmanned aerial vehicles," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 38, no. 3, pp. 549–563, 2008.

[17] H. Lu, Y. Li, T. Uemura, H. Kim, and S. Serikawa, "Low illumination underwater light field images reconstruction using deep convolutional neural networks," *Future Generation Computer Systems*, vol. 82, pp. 142–148, 2018.

[18] Y. Zhao, T. Rui, Y. Li, and X. Zuo, "A UAV patrol system using panoramic stitching and object detection," *Computers and Electrical Engineering*, vol. 80, pp. 106473–106481, 2019.

[19] P. Zhang, Y. Zhong, and X. Li, "SlimYOLOv3: narrower, faster and better for real-time UAV applications," in *IEEE International Conference on Computer Vision*, Seoul, 2019.

[20] W. Wang, Y. Peng, G. Cao, X. Guo, and N. Kwok, "Low-illumination image enhancement for night-time UAV pedestrian detection," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5208–5217, 2021.

[21] D. Liang, S. Gao, H. Sun, and N. Liu, "UAV detection in motion cameras combining kernelized correlation filters and deep learning," *Acta Aeronautica et Astronautica Sinica*, vol. 42, no. 9, p. 323733, 2020.

[22] S. Chen, S. Guo, and Y. Li, "Real-time tracking a ground moving target in complex indoor and outdoor environments with UAV," in *2016 IEEE International Conference on Information and Automation (ICIA)*, pp. 362–367, Ningbo, China, 2016.

[23] H. Zou, Z. Gong, S. Xie, and W. Ding, "A pan-tilt camera control system of UAV visual tracking based on biomimetic eye," in *2006 IEEE International Conference on Robotics and Biomimetics*, pp. 1477–1482, Kunming, China, 2006.

[24] S. Chu, F. Zhang, N. Ji, Z. Jin, and R. Pan, "Pan-and-tilt self-portrait system using gesture interface," in *2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)*, pp. 599–605, 2017.

[25] W. Yuan, T. Hong, and M. Kadoch, "Improved Kalman filter variants for UAV tracking with radar motion models," *Electronics*, vol. 9, no. 5, p. 768, 2020.

[26] T. Basar, "A New Approach to Linear Filtering and Prediction Problems," in *Control Theory: Twenty-Five Seminal Papers (1932-1981)*, pp. 167–179, IEEE, 2001.