applied sciences

MDPI

Article

Real-Time Visual Tracking of Moving Targets Using a Low-Cost Unmanned Aerial Vehicle with a 3-Axis Stabilized Gimbal System

Xuancen Liu [1], Yueneng Yang [1,2], Chenxiang Ma [3], Jie Li [4] and Shifeng Zhang [1,*]

[1] College of Aerospace Science and Engineering, National University of Defense Technology, Changsha 410073, China; liuxuancen14@nudt.edu.cn (X.L.); yangyueneng@163.com (Y.Y.)
[2] College of Intelligence Science and Engineering, National University of Defense Technology, Changsha 410073, China
[3] School of Aeronautics and Astronautics, University of Electronic Science and Technology of China, Chengdu 611731, China; chenxiangma@std.uestc.edu.cn
[4] College of Systems Engineering, National University of Defense Technology, Changsha 410073, China; tclijie@nudt.edu.cn
* Correspondence: zhang_shifeng@hotmail.com

Received: 4 July 2020; Accepted: 20 July 2020; Published: 23 July 2020

Featured Application: A complete solution for visual detection and autonomous tracking of a moving target is presented, which is applied to low-cost aerial vehicles in reconnaissance, surveillance, and target acquisition (RSTA) tasks.

Abstract: Unmanned Aerial Vehicles (UAVs) have recently shown great performance collecting visual data through autonomous exploration and mapping, which are widely used in reconnaissance, surveillance, and target acquisition (RSTA) applications. In this paper, we present an onboard vision-based system for low-cost UAVs to autonomously track a moving target. Real-time visual tracking is achieved by using an object detection algorithm based on the Kernelized Correlation Filter (KCF) tracker. A 3-axis gimbaled camera with separate Inertial Measurement Unit (IMU) is used to aim at the selected target during flights. The flight control algorithm for tracking tasks is implemented on a customized quadrotor equipped with an onboard computer and a microcontroller. The proposed system is experimentally validated by successfully chasing a ground and aerial target in an outdoor environment, which has proven its reliability and efficiency.

Keywords: target tracking; UAV; onboard vision; gimbal system; low-cost sensors

1. Introduction

The past decade has witnessed an explosive growth in the utilization of unmanned aerial vehicles (UAVs), attracting more and more attention from research institutions around the world [1,2]. With a series of significant advances in technology domains like micro-electro-mechanical system (MEMS), many UAV platforms and mission-oriented sensors limited to military affairs in the past are now widely applied to industrial and commercial sectors [3–7]. UAV-based target tracking is one of the most challenging tasks, which is closely related to applications such as traffic monitoring, reconnaissance, surveillance, and target acquisition (RSTA), search and rescue (SAR), inspection of power cables, etc. [8–13]. The tracking system is an important part of UAVs which detects a target of interest rapidly in a large area, and then performs continuous surveillance of the selected target in the tracking phase [14]. In order to achieve target acquisition and localization, military UAVs are usually equipped with airborne radars [15] or guided seekers [16]; however, they are too heavy and unaffordable for most civilian used UAVs. As one of the most popular UAV platforms, quadrotors are more stable and have a lower manufacturing cost than helicopters. For portability and flexibility, the takeoff weight of most quadrotors is less than 15 kg, so payload and battery endurance for onboard equipment are very limited. In this situation, precisely and robustly tracking a moving target by using a small-scale UAV platform is still a challenging task because the onboard computational capability is poor and sensors are low-cost [17].

Compared with many other detection sensors, cameras seem to have an inherent potential for UAV-based target tracking tasks, for they passively receive the environmental information and visual features, while still being low-cost and lightweight [18]. In recent years, various methods are studied in many research works to achieve target tracking by using cameras. Wenzel et al. [19] presented a visual tracking system for autonomous landing of a Hummingbird quadrotor by using an infrared (IR) camera that was extremely cheap. In [20], a vision-based landing control algorithm for an autonomous helicopter was designed and implemented by using a downward-pointing charge coupled device (CCD) camera. The onboard system was integrated with an algorithm for visual acquisition of the target (a moving helipad) and state estimation, which calculated the six degrees of freedom (DOF) pose with respect to the landing pad. The helipad which consists of a letter "H" surrounded by a circle is a typical structured pattern widely used in vision-based tracking and landing tasks because it can be easily detected and recognized in cluttered environments [21]. The research on square-based fiducial markers, such as AprilTag [22] and ArUco [23], has also aroused increasing interests toward marker-based visual tracking. Ref. [24] presented a fully autonomous flight control system for tasks of target recognition, geo-location, following and finally landing on a AprilTag marker which is attached to a high-speed moving car. Ref. [25] proposed a vision-based swarming approach for employing three or more Parrot Ar.Drone 2.0 quadrotors without any extra positioning sensors, while the ArUco markers fixed on the obstacles were used for localization and mapping. However, artificial markers or cooperative targets can only be applied to some specific scenarios such as autonomous landing and indoor navigation, in which the actual size and main features of the pattern are already known [26]. Since the vision system may be requested to track an arbitrary target with unknown size, shape, and motion, it is impossible to have already acquired information and know how to recognize it.

Object tracking is one of the most fundamental fields in computer vision and has a wide range of applications in many areas. For this task, many algorithms for visual tracking have been proposed, which can be generally categorized into generative and discriminative methods according to their appearance models. Generative methods typically search for the image region which is most similar to the target, such as incremental subspace tracker (IVT [27]), L1 tracker [28], real-time compressive sensing tracking (RTCST [29]), superpixel tracking [30], mean shift algorithm [31–33], and the continuously adaptive mean shift (CAMSHIFT) algorithm [34]. Discriminative methods or tracking-by-detection methods deal with the tracking problem as a binary classification task and separate the target object from the background, such as multiple instance learning (MIL) tracker [35], Struck [36], on-line AdaBoost (OAB) tracker [37], and Tracking-Learning-Detecting (TLD) [38]. The potential of correlation filters for visual tracking has aroused tremendous research interest because it reduces the overhead time through fast Fourier transformation (FFT) [39]. Bolme et al. [40] presented a minimum output sum of squared error (MOSSE) filter, which produced stable correlation filters when initialized using a single frame. It is able to quickly adapt to variations in scale, rotation, and lighting while operating at 669 frames per second. Henriques et al. [41] proposed the correlation filter of the circulant structure with kernel (CSK), and used the kernel trick to learn the appearance model. After that, the Kernelized Correlation Filter (KCF) tracker that Henriques proposed in [42] further improved the CSK method by using the histogram of oriented gradients (HOG) feature instead of gray feature to represent the object, which showed an amazing speed on the OTB2013 dataset [43]. However, the original KCF tracker uses a fixed size template and is not able to handle scale changes and occlusions, leading to bad performance in some scenarios. To alleviate theses drawbacks, Montero et al. [44] proposed fast scalable solution based on KCF framework which used an adjustable Gaussian window function and a keypoint-based model for scale estimation to deal with the fixed size limitation in the Kernelized Correlation Filter. Zhang et al. [45] presented an improved KCF tracker by adopting a cascade classifier composed of multi-scale correlation filter and NN (Neighbor Nearest) classifier, which showed favorable performance in accuracy and robustness.

To actually track a moving target in real time with a vision-based system onboard a multi-rotor UAV platform, three steps are required. The first step is to detect the target and localize it in each image frame as described above. The second step involves getting the line-of-sight (LOS) angle or the position of the target and maintaining the target in the center of the camera's field of view (FOV). The third step is to use this information obtained by the vision system to define the control task of an autonomous UAV when flying around the target. Generally, a downward-facing strap-down camera has advantages with its small size, light weight, and simplicity [46]. However, the output of the strap-down camera cannot be directly used in the flight control system, for it couples with the UAV body angular motion. In addition, the restricted FOV of the camera makes it a challenging task to keep the fast-moving target in the image, which also means the FOV constraints must be fully considered in the guidance law [47]. To solve these problems, a gimbal system is widely used to provide inertial stability to the camera by isolating it from the UAV motion and vibration. A gimbaled camera now available on the market provides a decoupling along the roll, pitch, and yaw attitudes, which has become an important unit of the UAV system. In [48], a 3-axis gimbaled camera (DJI Zenmuse X3) is used to solve the problem, which enabled autonomous landing of a quadrotor on a high-speed ground vehicle. The system is experimentally implemented and validated by successfully landing a commercial quadrotor on the roof of a car moving at speeds of up to 50 km/h. Ref. [49] presented a field-tested mini gimbal mechanism to produce an estimation of the target position, which allowed a flying-wing UAV to fly around the target. Jakobsen et al. [50] presented the architecture of a pan/tilt/roll camera system implemented on the Georgia Tech's UAV research helicopter. Each axis is driven by a servo and optical encoders are utilized to measure the gimbal orientation. Whitacre et al. [51] performed flight tests using a SeaScan UAV with a gimballing camera to track ground targets and studied the effect of altitude, camera FOV, and orbit center offsets within the geo-location tracking performance for both stationary and moving targets.

In this paper, motivated by the existing challenges, an onboard vision-based system for tracking arbitrary 3D objects moving at unknown velocities is proposed by utilizing a 3-axis gimbal system. A KCF tracker is used to detect and localize the target of interest from images acquired by the gimbaled camera. The 3-axis stabilized gimbal system is driven by brushless direct current motors (BLDCMs) and a magnetic rotary encoder is attached to each axis with a high-resolution output of the angular position. Then, the results of the KCF tracker are put into a proportional-derivative (PD) controller, which aligns the optical axis of the gimbaled camera with the LOS joining the camera and the target. A tracking strategy for multi-rotor UAVs is proposed based on the proportional navigation (PN) method, which makes it possible to keep following the target, although the target position is not known. A low-cost experimental quadrotor is customized for real-time flight tests, which is equipped with a microcontroller using consumer sensors and an onboard computer for image processing. The presented system is experimentally implemented and validated by successfully tracking a commercial quadrotor flying along an unknown path. By analyzing the flight data of both the target and the interceptor, the system is demonstrated to be reliable and cost-effective.

The rest of this paper is organized as follows. The vision system and control of the gimbaled camera are described in Section 2. The visual tracking algorithm used to detect and localize the target in image stream is presented in Section 3. Section 4 established the dynamic model of the quadrotor, and the target tracking strategy as well as the control law is presented. Section 5 describes preparation for flight experiments, and the results of the flight tests are given in this part. Finally, Section 6 concludes this paper and presents the future work.

2. Gimbal System

2.1. Problem Formulation and System Architecture

Although there are many algorithms capable of tracking targets from the video streams, techniques reported in the field of computer vision cannot be easily extended to airborne applications because of high dynamic UAV-target relative motion. In this section, a small commercial drone (41 cm in diameter) is considered as the target with unknown speed. If this unwanted aerial visitor flies into places such as airports, prisons, and military bases where consumer drones are not allowed, it can cause a big problem. Therefore, a lot of anti-UAV defense systems are being introduced to combat the growing threats of malicious UAVs. One way is to use a rifle-like device which sends a high-power electromagnetic wave to jam the UAV control systems and force them to land immediately. Another option is to capture the target in mid-air, using a UAV platform that carries a net gun.

Consider Figure 1, which depicts the UAV-Target relative kinematics and defines the coordinate systems. Let B denote the body frame that moves with the UAV, C the camera frame that is attached to the UAV but rotates with respect to frame B, N a North-East-Down (NED) coordinate system taken as an inertial reference frame, I, the image frame. The origin of the camera frame C is the optical center and Z_c coincides with the optical axis of the camera. Following the notation introduced in [52], let $p_C = [x_c, y_c, z_c]^T$ denote the position of the target in frame C. The rotation transformation ${}^n_cR$ from frame C to frame N is:

$$ {}^n_cR = {}^n_bR{}^b_cR \tag{1} $$

where the transformation ${}^n_bR$ is calculated by the roll, pitch, and yaw angles of the UAV given by the flight controller. ${}^b_cR$ can be computed by the gimbal system using the relative angular position measured by the encoders. Let p be the position of the target with respect to the optical center of camera resolved in N, which is given as:

$$ p = {}^n_bR{}^b_cR \cdot p_C \tag{2} $$

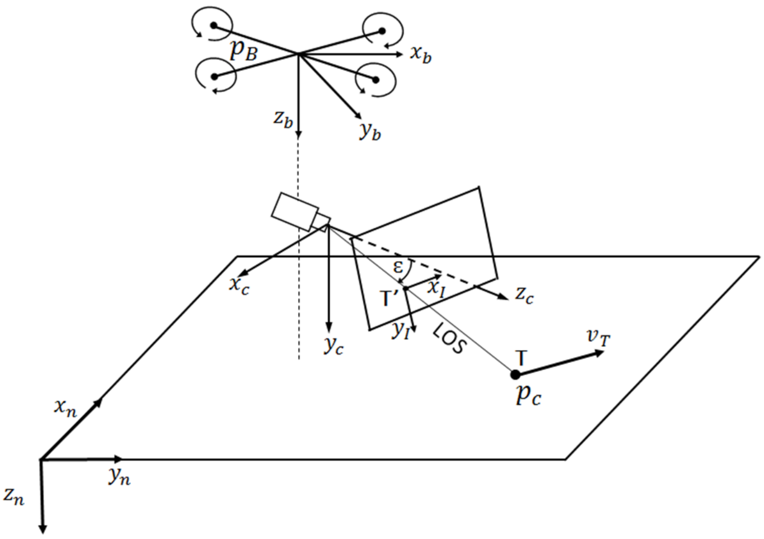


Figure 1. UAV-Target relative kinematics.

Thus, the target position p_T in NED coordinate system can be estimated, which is given as:

$$ p_T = p_B + {}^n_bRp_{BC} + p \tag{3} $$

where p_{BC} is the position of gimbaled camera relative to the UAV body in frame B, and p_B is the UAV position in N.

To deal with the task of moving target tracking, an autonomous quadrotor UAV system equipped with a 3-axis gimbaled camera is constructed to detect and follow the flying object in the pursuit-evasion scenario. The visual tracking system processes the images and drives the gimbal to search the target areas. Once an intruding drone is detected, the location of the target in each image frame is acquired, and this is utilized by the automated targeting module for aiming control of the gimbal. While the target is locked down, the camera pose can be used as input information to control the UAV flight. Figure 2 presents the proposed vision-based system.

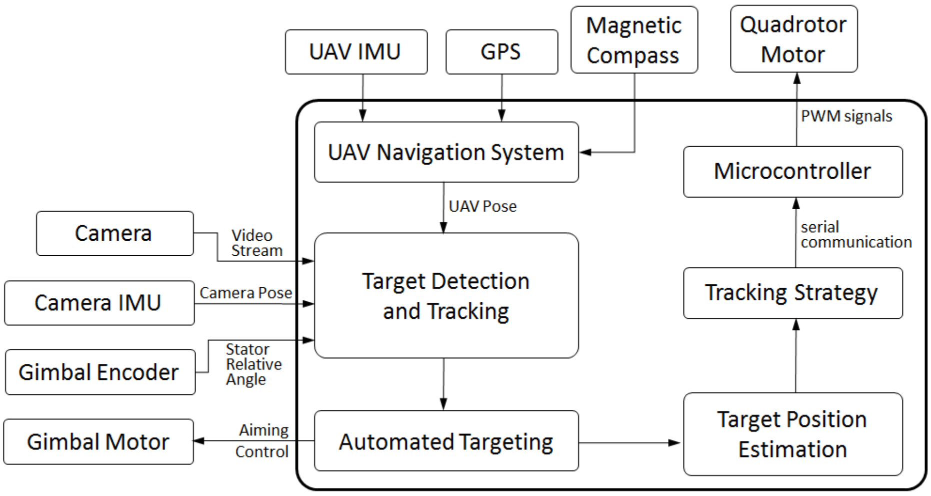


Figure 2. Architecture of the autonomous visual tracking system.

For most civil UAV applications, the gimbal system and the UAV flight control system are independent of each other. However, to achieve autonomous tracking of a moving target, the two systems are coordinated by the proposed vision algorithm which is implemented in the onboard computer running a Linux based system.

2.2. *Kinematics of the Gimbal System*

While the FOV of a single camera is limited, the gimbal systems are able to rotate the camera to a desired direction, which are widely applied to many fields such as filming and monitoring. When these systems are mounted onboard an UAV, the torque motors are activated by the IMUs and other angular sensors to compensate for all the rotations resulting from the UAV flight, which returns the stable member to its original attitude. As shown in Figures 3 and 4, the gimbal system which is used in this paper consists of direct current (DC) motors that balance the platform, magnetic rotary encoders that sense the relative rotation, embedded stabilization controller that process all the sensors information and output the control signals, the vibration damper that connects the outer gimbal to the UAV body, and the camera that captures the images.

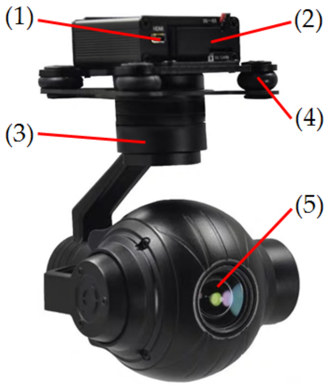


Figure 3. 3-axis gimbaled camera. (1) the output port of the video stream; (2) the embedded stabilization controller; (3) the brushless DC motor with magnetic rotary encoder; (4) the vibration damper; (5) the camera.

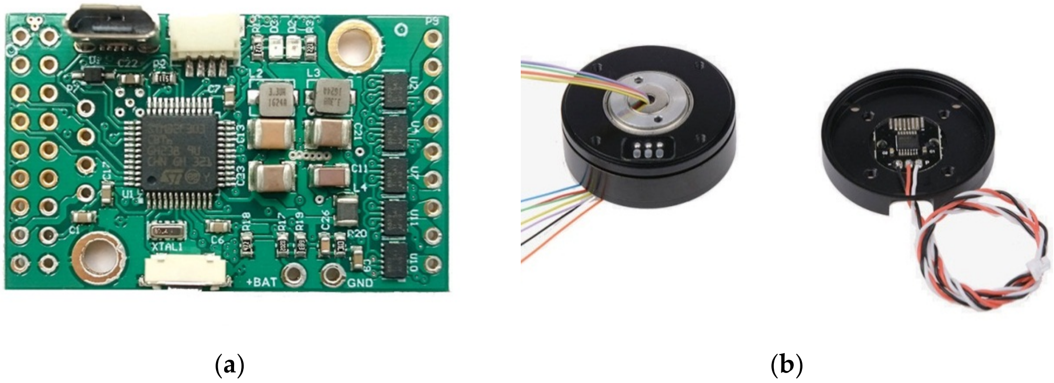


Figure 4. Details of the gimbal control system. (**a**) 3-axis stabilization controller (BaseCam SimpleBGC 32-bit Tiny); (**b**) HT3505 brushless DC motor with AS5048A magnetic rotary encoder.

The 3-axis gimbaled camera supporting structure consists of the case, outer frame, middle frame and inner frame as depicted in Figure 5. The kinematic relations are set as a yaw, roll, pitch sequence and four reference frames are introduced: the body-fixed frame F, the outer frame O, the middle frame M, the inner frame G connected by three revolute joints. Considering the common structure of gimbals in [53–55], relative angles are defined as yaw (θ_Y), roll (θ_R) and pitch (θ_P). The frame F is carried into frame O by rotation θ_Y around the axis z_F. Frame O is carried into frame M by rotation θ_R around the axis x_O. Finally, frame M is carried into frame G by rotation θ_P around the axis y_M. The coordinate systems of the gimbal (Figure 5) are placed parallel to each other as the initial state in the configuration (θ_Y,θ_R,θ_P) = (0, 0, 0).

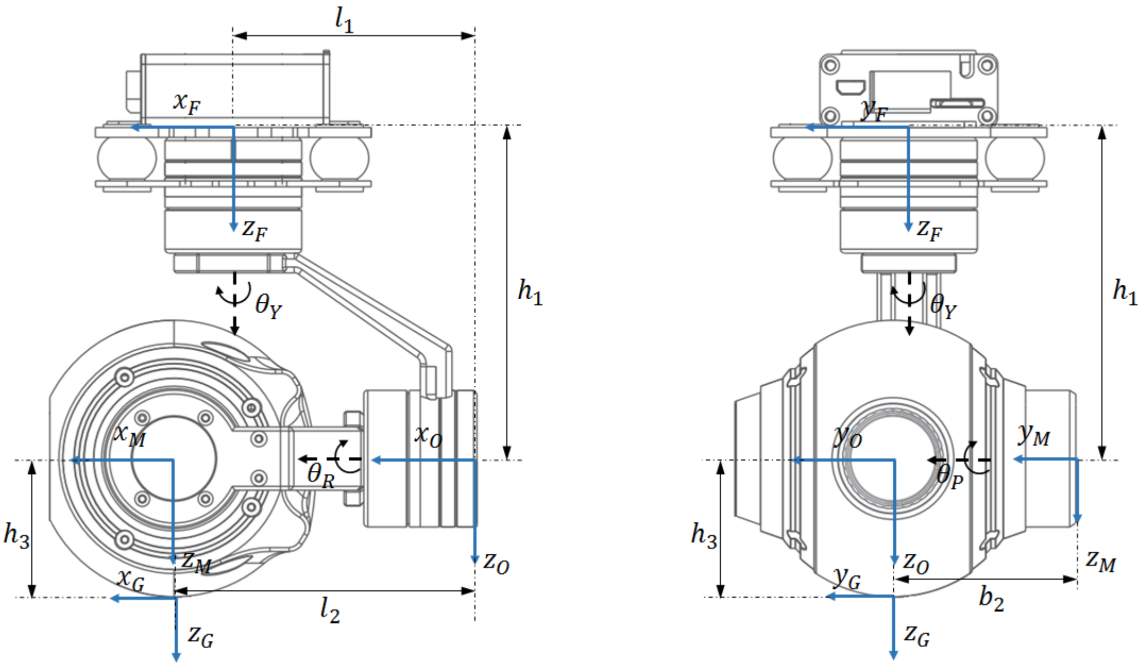


Figure 5. The gimbal in configuration (θ_Y,θ_R,θ_P) = (0, 0, 0) viewed from the side and the front with reference frames and relations between them. The direction of each joint is indicated by the dotted line.

The coordinate of an arbitrary point P in frame G denoted as the vector ${}^{G}r_{P}$ can be described in a different coordinate frame F using the rotation matrix ${}^{F}R_{G}$ and the translation vector ${}^{F}d_{G}$ between the frames according to the above relationship:

$$^{B}r_{P} = {}^{F}R_{G}{}^{G}r_{P} + {}^{F}d_{G} \tag{4}$$

where, in a 3D environment,

$$^{G}r_{P} = \begin{bmatrix} {}^{G}x_{P} \\ {}^{G}y_{P} \\ {}^{G}z_{P} \end{bmatrix} \tag{5}$$

A more convenient way to describe such transformation is to use homogeneous transformation matrices ${}^{F}T_{G}$ given as:

$$^{F}T_{G} = \begin{bmatrix} {}^{F}R_{G} & {}^{F}d_{G} \\ 0_{1\times3} & 1 \end{bmatrix} \tag{6}$$

Several intermediate transformations are required to get the final transformation in Equation (4) given as

$$\begin{bmatrix} {}^{B}r_{P} \\ 1 \end{bmatrix} = {}^{F}T_{G}\begin{bmatrix} {}^{G}r_{P} \\ 1 \end{bmatrix} \tag{7}$$

where

$$^{F}T_{G} = {}^{F}T_{O}{}^{O}T_{M}{}^{M}T_{G} \tag{8}$$

With the parameters l_1, l_2, h_1, h_3 and b_2 in Figure 5, the transformations between the frames are as follows.

The transformation between the frame F and frame O:

$$^{F}R_{O} = \begin{bmatrix} \cos\theta_Y & -\sin\theta_Y & 0 \\ \sin\theta_Y & \cos\theta_Y & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{9}$$

$$
{}^{F}d_{O} = \begin{bmatrix} -l_1 \cos\theta_Y \\ -l_1 \sin\theta_Y \\ h_1 \end{bmatrix} \tag{10}
$$

or

$$
{}^{F}T_{O} = \begin{bmatrix} \cos\theta_Y & -\sin\theta_Y & 0 & -l_1 \cos\theta_Y \\ \sin\theta_Y & \cos\theta_Y & 0 & -l_1 \sin\theta_Y \\ 0 & 0 & 1 & h_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{11}
$$

The transformation between the frame *O* and frame *M*:

$$
{}^{O}R_{M} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_R & -\sin\theta_R \\ 0 & \sin\theta_R & \cos\theta_R \end{bmatrix} \tag{12}
$$

$$
{}^{O}d_{M} = \begin{bmatrix} l_2 \\ -b_2 \cos\theta_R \\ -b_2 \sin\theta_R \end{bmatrix} \tag{13}
$$

or

$$
{}^{O}T_{M} = \begin{bmatrix} 1 & 0 & 0 & l_2 \\ 0 & \cos\theta_R & -\sin\theta_R & -b_2 \cos\theta_R \\ 0 & \sin\theta_R & \cos\theta_R & -b_2 \sin\theta_R \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{14}
$$

The transformation between the frame *M* and frame *G*:

$$
{}^{M}R_{G} = \begin{bmatrix} \cos\theta_P & 0 & \sin\theta_P \\ 0 & 1 & 0 \\ -\sin\theta_P & 0 & \cos\theta_P \end{bmatrix} \tag{15}
$$

$$
{}^{M}d_{G} = \begin{bmatrix} h_3 \sin\theta_P \\ b_2 \\ h_3 \cos\theta_P \end{bmatrix} \tag{16}
$$

or

$$
{}^{M}T_{G} = \begin{bmatrix} \cos\theta_P & 0 & \sin\theta_P & h_3 \sin\theta_P \\ 0 & 1 & 0 & b_2 \\ -\sin\theta_P & 0 & \cos\theta_P & h_3 \cos\theta_P \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{17}
$$

Thus, the total rotation matrix ${}^{F}R_{G}$ and translation vector ${}^{F}d_{G}$ between frame *F* and frame *G* is:

$$
{}^{F}R_{G} = \begin{bmatrix} \cos\theta_Y \cos\theta_P - \sin\theta_Y \sin\theta_R \sin\theta_P & -\cos\theta_R \sin\theta_Y & \cos\theta_Y \sin\theta_P + \cos\theta_P \sin\theta_Y \sin\theta_R \\ \cos\theta_P \sin\theta_Y + \cos\theta_Y \sin\theta_R \sin\theta_P & \cos\theta_Y \cos\theta_R & \sin\theta_Y \sin\theta_P - \cos\theta_Y \cos\theta_P \sin\theta_R \\ -\cos\theta_R \sin\theta_P & \sin\theta_R & \cos\theta_R \cos\theta_P \end{bmatrix} \tag{18}
$$

$$
\begin{aligned}
{}^{F}d_{G} &= {}^{F}R_{O}\,{}^{O}R_{M}\,{}^{M}d_{G} + {}^{F}R_{O}\,{}^{O}d_{M} + {}^{O}d_{F} \\
&= \begin{bmatrix} l_2 \cos\theta_Y - l_1 \cos\theta_Y + h_3 \cos\theta_Y \sin\theta_P + h_3 \cos\theta_P \sin\theta_Y \sin\theta_R \\ l_2 \sin\theta_Y - l_1 \sin\theta_Y + h_3 \sin\theta_Y \sin\theta_P - h_3 \cos\theta_Y \cos\theta_P \sin\theta_R \\ h_1 + h_3 \cos\theta_R \cos\theta_P \end{bmatrix}
\end{aligned} \tag{19}
$$

The ${}^{F}R_{G}$ is called a pitch-roll-yaw rotation matrix according to the order in which the rotation matrices are successively multiplied. In a similar way, the rotation between frame and inertial reference frame can be described as:

$$ {}^{N}R_{G} = \begin{bmatrix} \cos\alpha_Y\cos\alpha_P - \sin\alpha_Y\sin\alpha_R\sin\alpha_P & -\cos\alpha_R\sin\alpha_Y & \cos\alpha_Y\sin\alpha_P + \cos\alpha_P\sin\alpha_Y\sin\alpha_R \\ \cos\alpha_P\sin\alpha_Y + \cos\alpha_Y\sin\alpha_R\sin\alpha_P & \cos\alpha_Y\cos\alpha_R & \sin\alpha_Y\sin\alpha_P - \cos\alpha_Y\cos\alpha_P\sin\alpha_R \\ -\cos\alpha_R\sin\alpha_P & \sin\alpha_R & \cos\alpha_R\cos\alpha_P \end{bmatrix} \tag{20} $$

where α_Y, α_R, and α_P are derived using the information from the gyros and accelerometers on the IMU attached to the camera. The ${}^{N}R_{G}$ can also be derived using the rotation matrix ${}^{F}R_{G}$ and ${}^{N}R_{F}$, as below:

$$ {}^{N}R_{G} = {}^{N}R_{F}{}^{F}R_{G} \tag{21} $$

where ${}^{F}\omega_{NF}$, and the angular velocity of frame F respect to frame N introduced in frame F is measured and available given as:

$$ {}^{F}\omega_{NF} = \begin{bmatrix} p & q & r \end{bmatrix}^{T} \tag{22} $$

Angular velocities of frame O, M, and G respect to frame N introduced in its own frames are as follows:

$$ {}^{O}\omega_{NO} = {}^{O}\omega_{NF} + {}^{O}\omega_{FO} = \left({}^{F}R_{O}\right)^{T}{}^{F}\omega_{NF} + \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_Y \end{bmatrix} = \begin{bmatrix} p\cos\theta_Y + q\sin\theta_Y \\ -p\sin\theta_Y + q\cos\theta_Y \\ r + \dot{\theta}_Y \end{bmatrix} \tag{23} $$

$$ {}^{M}\omega_{NM} = {}^{M}\omega_{NO} + {}^{M}\omega_{OM} = \left({}^{O}R_{M}\right)^{T}{}^{O}\omega_{NO} + \begin{bmatrix} \dot{\theta}_R \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} p\cos\theta_Y + q\sin\theta_Y + \dot{\theta}_R \\ \cos\theta_R(-p\sin\theta_Y + q\cos\theta_Y) + \sin\theta_R(r + \dot{\theta}_Y) \\ \sin\theta_R(p\sin\theta_Y - q\cos\theta_Y) + \cos\theta_R(r + \dot{\theta}_Y) \end{bmatrix} \tag{24} $$

$$ \begin{aligned} {}^{G}\omega_{NG} &= {}^{G}\omega_{NM} + {}^{G}\omega_{MG} = \left({}^{M}R_{G}\right)^{T}{}^{M}\omega_{NM} + \begin{bmatrix} 0 \\ \dot{\theta}_P \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} \cos\theta_P(p\cos\theta_Y + q\sin\theta_Y + \dot{\theta}_R) - \sin\theta_P(\sin\theta_R(p\sin\theta_Y - q\cos\theta_Y) + \cos\theta_R(r + \dot{\theta}_Y)) \\ \cos\theta_R(-p\sin\theta_Y + q\cos\theta_Y) + \sin\theta_R(r + \dot{\theta}_Y) + \dot{\theta}_P \\ \sin\theta_P(p\cos\theta_Y + q\sin\theta_Y + \dot{\theta}_R) + \cos\theta_P(\sin\theta_R(p\sin\theta_Y - q\cos\theta_Y) + \cos\theta_R(r + \dot{\theta}_Y)) \end{bmatrix} \end{aligned} \tag{25} $$

The inertia matrices of the outer gimbal, middle gimbal, and inner gimbal are:

$$ J_O = diag\left\{ J_{Ox} \quad J_{Oy} \quad J_{Oz} \right\} \tag{26} $$

$$ J_M = diag\left\{ J_{Mx} \quad J_{My} \quad J_{Mz} \right\} \tag{27} $$

$$ J_G = diag\left\{ J_{Gx} \quad J_{Gy} \quad J_{Gz} \right\} \tag{28} $$

where J_{kx}, J_{ky}, J_{kz} $\quad (k = O, M, G)$ refer to the diagonal elements. For simplicity, it is assumed that the off-diagonal elements of inertia matrices can be neglected and only the moments of inertia are considered. The angular momentum of the pitch gimbal is:

$$ H_G = J_G{}^{G}\omega_{NG} \tag{29} $$

the roll gimbal is:

$$ H_M = J_M{}^{M}\omega_{NM} + {}^{M}R_{G}J_G{}^{G}\omega_{NG} \tag{30} $$

the yaw gimbal is:

$$ H_O = J_O{}^{O}\omega_{NO} + {}^{O}R_{M}(J_N{}^{M}\omega_{NM} + {}^{M}R_{G}J_G{}^{G}\omega_{NG}) \tag{31} $$

and each member of the gimbal system is treated as a rigid body and the moment equation can be written as:

$$\tau_k = \frac{dH_k}{dt} + {}^k\omega_{Nk} \times H_k \quad (k = O, M, G) \tag{32}$$

where external torques τ_O, τ_M and τ_G, about z_F, x_O and y_M, respectively, are applied to gimbals from motor and other external disturbance torques.

2.3. Stabilization and Aiming

The camera fitted on the innermost frame is inertially stabilized and controlled by the gimbal system. Furthermore, the system is required to align its optical axis in elevation and azimuth with a LOS joining the camera and target. Figure 6 describes the angular geometry of how the gimbaled camera aims at the target, where θ is the pitch angle of the UAV body, δ is the boresight angle, λ is the LOS angle, θ_P is the pitch angle of the gimbal frame, and ε is the boresight error angle.

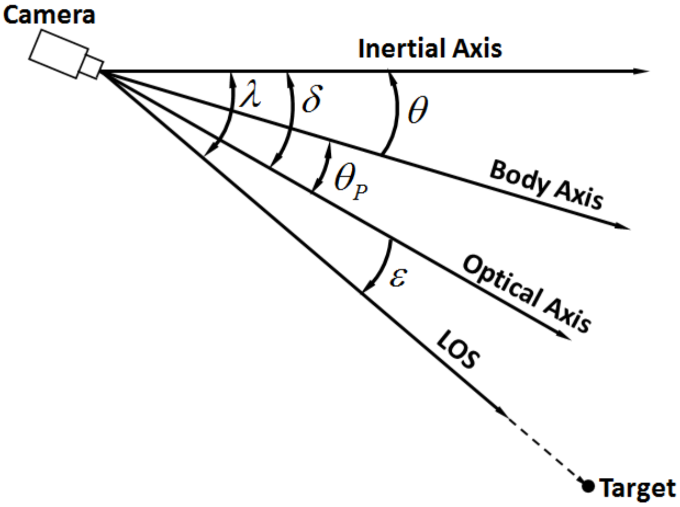


Figure 6. Angular geometry of the gimbal system.

There are four operation modes, namely, the preset angle mode, search mode, stabilize mode, and tracking mode as shown in Figure 7. When the gimbal system is powered on and initialized, its direction is kept at $\delta = 0$ in inertial space. In preset angle mode, the optical axis of the camera will be set to a given angle and the control system will maintain the desired direction despite disturbances. Then, the system may switch to search mode, in which the gimbal will rotate circularly between its minimum and maximum angle to search a larger range of area.

When a target is confirmed, the control system will switch to the tracking mode and keep the target in the center of the camera view. In addition, if the target is lost and cannot be recaptured in a few seconds, the system will return to the search mode and try to find it again. Figure 8 shows how the control system works which contains two loops: tracking loop and stabilizing loop.

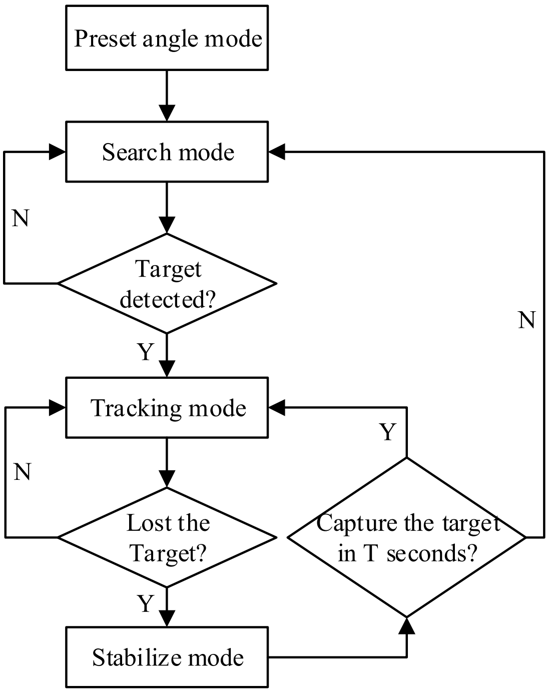


Figure 7. Diagram of the system operation modes.

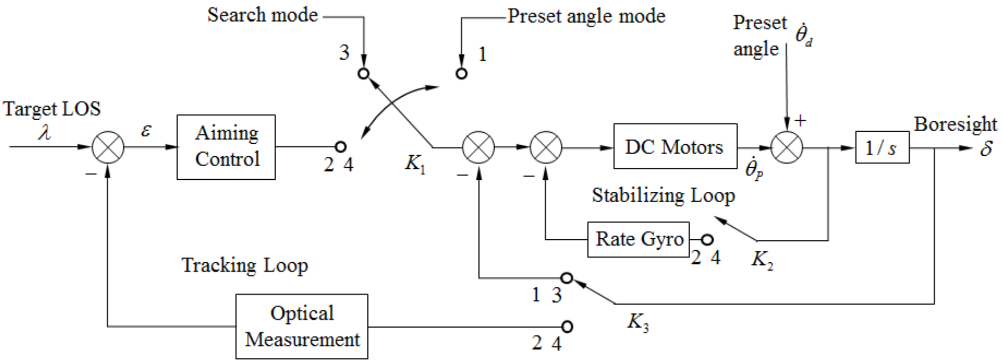


Figure 8. Architecture of the gimbal control system in elevation channel. 1 for the preset angle mode, 2 for the stabilize mode, 3 for the search mode, and 4 for the tracking mode.

Based on the image information and measurement data received from angular sensors, the tracking loop generates a rate command to direct the boresight towards the target LOS so that the pointing error can be kept near zero. On the other hand, the stabilizing loop isolates the camera from UAV motion and external disturbances, which would perturb the aim-point. The control loops in roll, elevation, and azimuth channels are related by the cross coupling unit based on the gimbal system dynamics, which may be defined as the impact on one axis with the rotation of another [56].

3. Target Tracking

3.1. KCF Tracker

The KCF tracker [42,57] that is used in this paper considers the process of sample training as a Ridge Regression problem, which is also a regular minimization problem with a closed solution.

Consider a $n \times 1$ vector $x = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}^T$ as the base sample, which represents a patch with the target of interest. A small translation of this vector is given as:

$$Px = \begin{bmatrix} x_n & x_1 & \cdots & x_{n-1} \end{bmatrix}^T \tag{33}$$

where P is the permutation matrix:

$$P = \begin{bmatrix} 0 & 0 & 0 & \cdots & 1 \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \tag{34}$$

and u shifts can be made to achieve a larger translation by using the matrix power $P^u x$. By cyclic shifting operations, we can use these vectors to constitute a circulant matrix as:

$$X = C(x) = \begin{bmatrix} (P^0 x)^T \\ (P^1 x)^T \\ (P^2 x)^T \\ \vdots \\ (P^{n-1} x)^T \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & x_3 & \cdots & x_n \\ x_n & x_1 & x_2 & \cdots & x_{n-1} \\ x_{n-1} & x_n & x_1 & \cdots & x_{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_2 & x_3 & x_4 & \cdots & x_1 \end{bmatrix} \tag{35}$$

and it is useful that all circulant matrices are diagonalized by the Discrete Fourier Transform (DFT), regardless of the base sample x, which can be expressed as:

$$X = Fdiag(\hat{x})F^H \tag{36}$$

where $\hat{x}$ is the DFT of base sample, $\hat{x} = F(x)$, and F is a constant matrix known as the DFT matrix that does not depend on x.

Based on Ridge Regression, the goal of training is to find a function $f(z)$ that minimizes the squared error over samples x_i and their regression targets y_i, as shown below:

$$E = \min_w \sum_i (f(x_i) - y_i)^2 + \lambda \|w\|^2 \tag{37}$$

where the regularization parameter λ is used to control over fitting, as in the Support Vector Machines (SVM) [57] and w represents the filter coefficients.

Consider a linear regression function $f(z) = w^T z$, the minimizer has a closed-form solution given as:

$$w = (X^T X + \lambda I)^{-1} X^T y \tag{38}$$

where X is the circulant matrix with one sample per row x_i, each element of y is a regression target y_i, and I is an identity matrix. By utilizing the diagonalization of the circulant matrices, Equation (38) can be expressed in Fourier domain as:

$$\hat{w} = \frac{\hat{x}^* \odot \hat{y}}{\hat{x}^* \odot \hat{x} + \lambda} \tag{39}$$

where $\hat{w}$, $\hat{x}$, $\hat{y}$ are the DFT of w, x, y, respectively, and $\hat{x}^*$ is the complex-conjugate. In addition, w can be easily recovered in the spatial domain with Inverse Discrete Fourier Transform (IDFT).

When regression function $f(z)$ is nonlinear, the kernel trick is used to map the inputs of a linear problem to a nonlinear and high-dimensional feature space $\varphi(x)$:

$$w = \sum_i \alpha_i \varphi(x_i) \tag{40}$$

Then, the variables under optimization are α instead of w. The kernel function k is used to compute the algorithm in terms of dot-products, as shown below:

$$\varphi^T(x)\varphi(x') = k(x, x') \tag{41}$$

where all the dot-products between samples are stored in a $n \times n$ kernel matrix K, with elements:

$$K_{ij} = k(x_i, x_j) \tag{42}$$

and the regression function $f(z)$ can be expressed as:

$$f(z) = w^T z = \sum_{i=1}^{n} \alpha_i k(z, x_i) \tag{43}$$

The solution of this regression function can be given as:

$$\alpha = (K + \lambda I)^{-1} y \tag{44}$$

where K is the kernel matrix and α is the vector of coefficients α_i, which express the solution in the dual space. By making K circulant, Equation (44) can be diagonalized as in the linear case, obtaining:

$$\hat{\alpha} = \frac{\hat{y}}{\hat{k}^{xx} + \lambda} \tag{45}$$

where $\hat{k}^{xx}$ is the correlation kernel of x with itself in Fourier domain and $\hat{\alpha}$, $\hat{y}$ are the DFT of vector α, y.

For the kernel matrix K^z between all training samples (cyclic shifts of x) and candidate image patches (cyclic shifts of base patch z), each element of K^z is given by $k(P^{i-1}z, P^{j-1}x)$. From Equation (43), the regression function can be computed for all candidate patches with:

$$f(z) = (K^z)^T \alpha \tag{46}$$

where $f(z)$ is a vector, containing the output of all cyclic shift of z, which is the full detection response. The position where the output response takes the maximum value is the position of the target in a new frame. To compute Equation (46) efficiently, it is diagonalized as shown below:

$$\hat{f}(z) = \hat{k}^{xz} \odot \hat{\alpha} \tag{47}$$

where a hat $\wedge$ denotes the DFT of a vector. In this paper, given the nonlinear Gaussian kernel $k(x, x') = \exp(-\frac{1}{\sigma^2}\|x - x'\|^2)$, we get:

$$k^{xx'} = \exp\Big(-\frac{1}{\sigma^2}\Big(\|x\|^2 + \|x'\|^2 - 2F^{-1}(\hat{x}^* \odot \hat{x}')\Big)\Big) \tag{48}$$

where the kernel correlation can be computed by using a few DFT/IDFT and element-wise operations in $O(n \log n)$ time. Henriques et al. [42] proved that the conversion from inverse operation of matrix in spatial domain to matrix multiplication in Fourier domain would greatly reduce the computational complexity and shorten the computation time.

In the tracking process, considering the target variations, such as illumination, scale, occlusion, and deformation factors, the target apparent model and coefficient vector are updated after each frame [58], as shown below:

$$x_t = (1 - \eta_t)x_{t-1} + \eta_t x \tag{49}$$

$$\alpha_t = (1 - \eta_t)\alpha_{t-1} + \eta_t \alpha \tag{50}$$

where x_{t-1}, x_t are the target model updated after the $t-1$ and the t frame. α_{t-1}, α_t are the coefficient vector updated after the $t-1$ and the t frame. η_t is the learning rate.

3.2. Target Localization

Consider a monocular camera model as shown in Figure 9.

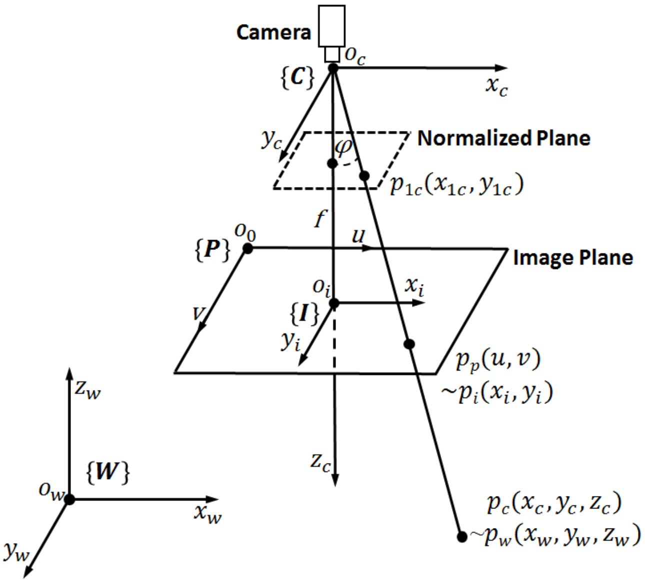


Figure 9. Coordinate system of a monocular camera.

When an arbitrary point $p(x_w, y_w, z_w)$ in world frame W is detected by the camera, its 2D position $p_i(x_i, y_i)$ on the image plane can be given as:

$$x_i = \frac{f x_c}{z_c} = (u - u_0)dx \tag{51}$$

$$y_i = \frac{f y_c}{z_c} = (v - v_0)dy \tag{52}$$

where (x_c, y_c, z_c) is the position of p in the camera frame, f is the focal length, (u, v) is the target position in pixel values, (u_0, v_0) is the intersection of the optical axis, and the image plane, dx and dy are the physical length per pixel in the x_i and y_i axis directions. Equations (51) and (52) can be integrated as:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = M_{in} \begin{bmatrix} x_c/z_c \\ y_c/z_c \\ 1 \end{bmatrix} = M_{in} \begin{bmatrix} x_{1c} \\ y_{1c} \\ 1 \end{bmatrix} \tag{53}$$

where $[x_{1c}, y_{1c}, 1]^T$ is the point on the normalized plane and M_{in} is the intrinsic parameter matrix of the camera, which is given as:

$$M_{in} = \begin{bmatrix} f/dx & 0 & u_0 \\ 0 & f/dy & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{54}$$

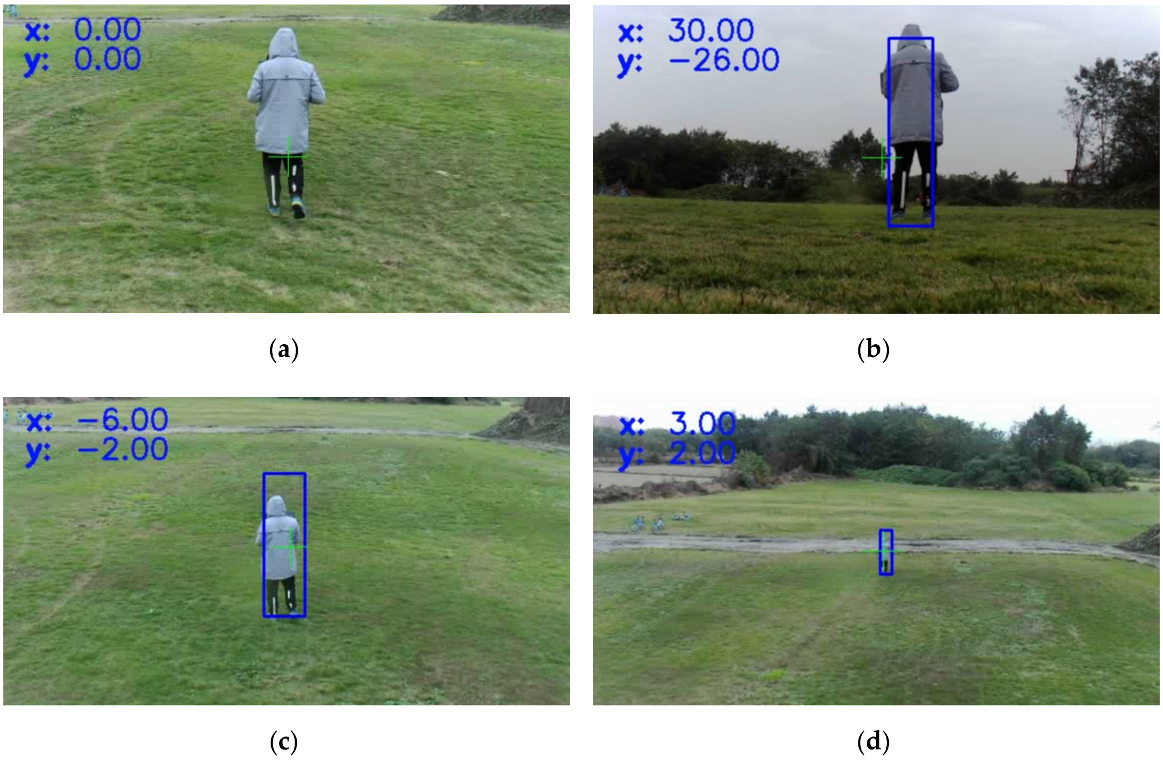
With a calibrated camera, the pointing error angle ε in elevation and azimuth can be computed by using the above relations, as shown below:

$$\varepsilon_{\chi} = \arctan(x_c/z_c) \tag{55}$$

$$\varepsilon_{\gamma} = \arctan(y_c/z_c) \tag{56}$$

where ε_{χ} and ε_{γ} are the pointing error or boresight error in azimuth and elevation, respectively, which can be put into the target aiming system as input information to control the gimbal.

A scalable KCF tracker is used in this paper, with the scale changes of the target taken into consideration. The tracker not only updates the centroid position of the target in the image frame, but also outputs the target size in pixel values. This can be used to control the distance between the interceptor and the target while tracking, though the physical size of the target is unknown. The running results of tracking a pedestrian with the proposed tracker are shown in Figure 10. The bounding box changes with adaption to the target variations in the video streams.



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

Figure 10. Tracking of a human target moving at pedestrian speed. (**a**) video streams with no target selected; (**b**) tracking of the selected target with its error pixels displayed on screen; (**c**) autonomously lock on the target in close range using the gimbal system; (**d**) real-time adjustment to the scale changes of the target.

4. Flight Control Algorithm

4.1. UAV Dynamic Model

The 6 DOF motion of a rigid quadrotor is described in Figure 11.

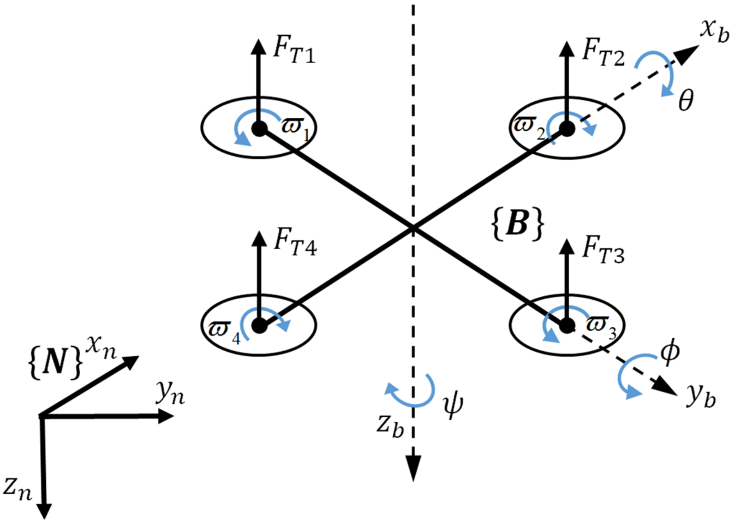


Figure 11. The quadrotor with corresponding frames.

Let m denote the mass of the quadrotor and J the moment of inertia. The external forces and torques which act on the quadrotor platform are primarily caused by propellers and gravity. A local NED frame N and body-fixed frame B are introduced to describe the motion of the quadrotor. ${}^np = \begin{bmatrix} {}^np_x & {}^np_y & {}^np_z \end{bmatrix}^T$ and ${}^nv = \begin{bmatrix} {}^nv_x & {}^nv_y & {}^nv_z \end{bmatrix}^T$ are the position and linear velocity of the quadrotor's mass center relative to N. $\Theta = [\phi\ \theta\ \Psi]^T$ is the roll/pitch/yaw angles, which represents the orientation of the quadrotor in N. The rotation matrix R_b^n from B to N is expressed as:

$$
{}^NR_B = \begin{bmatrix} \cos\theta\cos\psi & \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi \\ \cos\theta\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix} \tag{57}
$$

The equations of motion can be described as:

$$
{}^n\dot{p} = {}^nv \tag{58}
$$

$$
{}^n\dot{v} = gn_3 - \frac{f_b}{m}R_b^n n_3 \tag{59}
$$

$$
J \cdot {}^b\dot{\omega} = -{}^b\omega \times \left(J \cdot {}^b\omega\right) + G_a + \tau \tag{60}
$$

where f_b is the force applied to the quadrotor given in B, τ is the torque, g is the gravitational acceleration and ${}^b\omega$ is the angular velocity of the quadrotor in B. The gyroscopic moment G_a is mainly produced by propellers, which can be neglected. In addition, the translational dynamics shown in Equations (58) and (59) can be simplified as:

$$
{}^n\ddot{p}_x = -\frac{f_b}{m}(\sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi) \tag{61}
$$

$$
{}^n\ddot{p}_y = -\frac{f_b}{m}(-\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi) \tag{62}
$$

$$
{}^n\ddot{p}_z = g - \frac{f_b}{m}\cos\phi\cos\theta \tag{63}
$$

Furthermore, it can be assumed that $\sin\phi \approx \phi$, $\cos\phi \approx 1$, $\sin\theta \approx \theta$, and $\cos\theta \approx 1$ for small angle approximation, which leads to a simplified dynamic model as described in [59].

4.2. Tracking Strategy

The tracking strategy used in this paper is based on proportional navigation (PN), which is a well-known guidance law and has been widely used to enable a missile to catch its target in optimal time. The constant bearing approach considers that the missile will finally collide with the detected target if the LOS angle is kept constant. The PN method improves the constant bearing approach to accommodate for target maneuver by accelerating the missile in a direction lateral to the LOS with magnitude proportional to the rate change of the LOS angle. There are different types of PN methods according to their different mathematical formulations and their performances have been analyzed in [60], when applied to guidance of a quadrotor.

The desired acceleration $\ddot{u}^{des}$ obtained by the PN method can be expressed as:

$$\ddot{u}^{des} = NL \times \dot{\lambda} \tag{64}$$

where $\dot{\lambda}$ is rate change of the LOS angle, N is the navigation gain, and L is the normal direction of the acceleration command that is calculated for different ways as follows:

$$L_{RTPN} = u_T - u \tag{65}$$

$$L_{IPN} = \dot{u}_T - \dot{u} \tag{66}$$

$$L_{PPN} = -\dot{u} \tag{67}$$

$$L_{NGL} = \frac{|\dot{u}|u}{|u_T - u|} \cdot \frac{\sin\beta}{|\dot{\lambda}|} \tag{68}$$

where L_{RTPN}, L_{IPN}, L_{PPN}, and L_{NGL} represent for Realistic True Proportional Navigation (RTPN) [61], Ideal Proportional Navigation (IPN) [62], Pure Proportional Navigation (PPN) [63], and Nonlinear Guidance Law (NGL) [64], respectively. $u_T, \dot{u}_T, \ddot{u}_T \in R^2$ are the position, velocity, and acceleration of the target, respectively, $u, \dot{u}, \ddot{u} \in R^2$ are the position, velocity, and acceleration of the interceptor, respectively. $|\cdot|$ represents the magnitude of the vector and β is the angle between interception velocity and LOS of the target.

In addition, $\dot{\lambda}$ can be described as:

$$\dot{\lambda} = \frac{(u_T - u) \times (\dot{u}_T - \dot{u})}{|u_T - u|^2} \tag{69}$$

The aim of the research in this paper is to track a moving target with a quadrotor platform, and coordinated control of the UAV and gimbaled camera is considered. As shown in Figure 12a,b, it can be done in two directions: the longitudinal direction and the lateral direction [65].

The λ_χ, λ_γ are the lateral and longitudinal LOS angle of the target, respectively. The ε_χ, ε_γ are the boresight error angle in lateral and longitudinal direction, respectively, which are controlled to be zero ($\varepsilon_\chi, \varepsilon_\gamma \to 0$). The $\dot{u}_T^\chi$, $\dot{u}_T^\gamma$ are the velocity of the target in lateral and longitudinal directions, respectively. The $\dot{u}^\chi$ is the velocity of the UAV in lateral direction that is aligned with the axis x_b of the body frame B and the desired yaw angle of the UAV is $\psi^{des} = \lambda_\chi$. A pure pursuit guidance law is used for tracking in the lateral direction, which works on the principle that, if the interceptor persistently points towards the target ($\psi \to \lambda_\chi, \dot{\psi} \to \dot{\lambda}_\chi$), then it will ultimately intercept it.

When the tracking is initiated, the UAV follows the target by tracking its lateral LOS angle with a forward speed $\dot{u}^\chi$, which is the horizontal component of the approaching velocity $\dot{u}_{App}$. To keep following the target moving at unknown speed, the approaching velocity $\dot{u}_{App}$ is decided by the scale changes of the target in the image frame. Then, a PPN guidance law will activate and the acceleration command $\ddot{u}_{PPN}^{des}$ will be applied to the UAV according to the rate changes of λ_γ, as shown in Figure 12b.

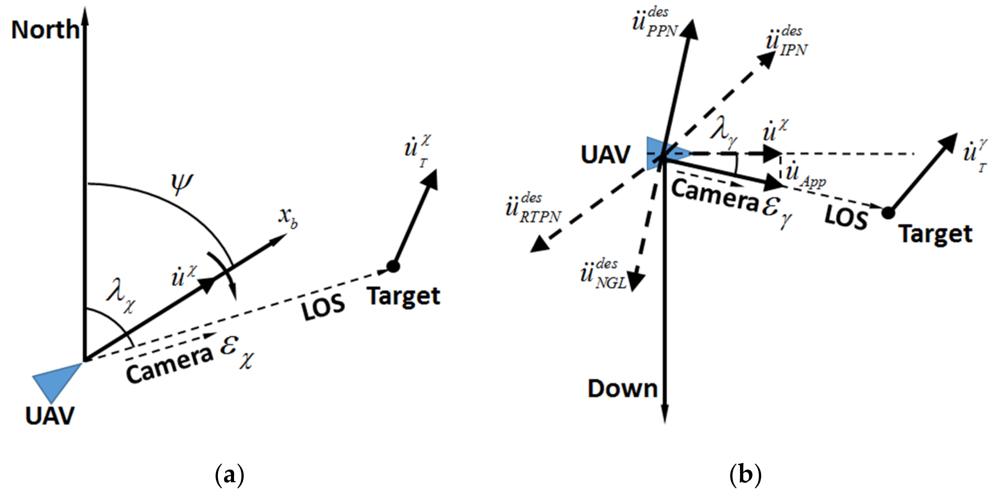


Figure 12. Schematic diagram showing the tracking strategy. (**a**) description of tracking the lateral LOS angle based on a pure pursuit method; (**b**) description of tracking the longitudinal LOS angle based on PN guidance law.

4.3. Flight Control System

The quadrotor is a typical underactuated system with only four independent inputs less than the degrees of freedom, so only the desired position and desired yaw angle can be directly tracked. The desired roll angle and the desired pitch angle are determined by the known ones. The flight control system of a quadrotor is described in Figure 13.

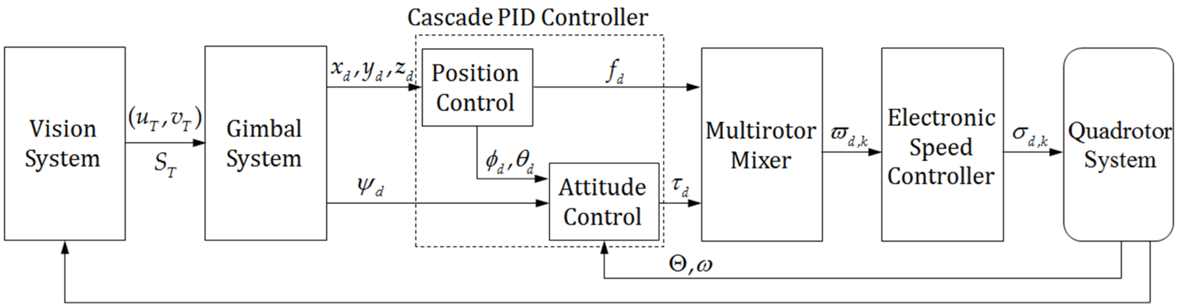


Figure 13. Hierarchical control architecture of the quadrotor for target tracking.

The pixel values u_T, v_T are the centroid of the target position in each image frame, and S_T is the scale change of the target, which is given by the proposed KCF tracker. The state of the quadrotor (Θ, ω) expressed in global NED frame is given by the autopilot using an Extended Kalman Filter (EKF). The desired position $\begin{bmatrix} x_d & y_d & z_d \end{bmatrix}^T$ and the desired yaw angle ψ_d are given by the gimbal system in the global NED using the above-mentioned tracking strategy. A cascade proportional-integral-derivative (PID) controller is designed to individually control the 6 DOF motion of the quadrotor. The attitude control loop is implemented on the microcontroller, while the outer loop for position control is implemented on the onboard computer. All PID gains have been preliminarily tuned in hovering flight tests. The outputs of the cascade PID controller are the desired force f_d and the desired torque τ_d, which are applied to the UAV body. The mixer gives the desired angular velocity of each motor to the electronic speed controller (ESC), which is expressed as a Pulse Width Modulation (PWM) signal.

It is worth mentioning that the thrust value is not only determined by the desired position, but also by the takeoff weight of the quadrotor platform. Thus, the height control can be considered as two parts: a slightly changed base value for hovering control and a fast controller for position control.

When the target is selected, the quadrotor will keep the distance relative to the target based on the estimation of its scale changes in the image frame, which is shown in Supplementary Materials.

5. Experiments and Results

5.1. Experimental Setup

Most experimental UAVs are equipped with expensive sensors and devices, such as high-precision IMUs, 3D light detection, and ranging (Lidar) sensors and differential global positioning system (DGPS), which will definitely improve the control accuracy but are unaffordable in many practical applications. To test the proposed tracking system in this paper, a customized quadrotor platform (65 cm in diameter) is used to perform all the flight experiments, which weighs 4.2 kg including all the payloads, as shown in Figure 14. The cost is much lower than the other platforms (e.g., DJI Matrice 200). The 3-axis gimbal at a dimension of 108 × 86.2 × 137.3 mm^3, weighs only 409 g, as shown in Figure 15a. The IMUs and encoders are consumer sensors at very low prices. The camera with focal lengths ranging from 4.9–49 mm has a maximum resolution of 1920 × 1080 at 60 frames per second (fps). The cost of the gimbaled camera is less than $400, which makes it very attractive considering its great performance. The AS5048A magnetic rotary encoder used in the gimbal system measures the angular position of each axis, which has a 14-bit high resolution output (0.0219 deg/LSB).

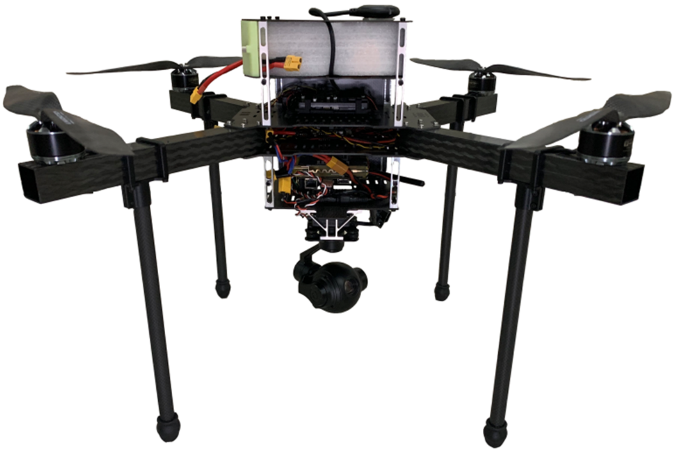
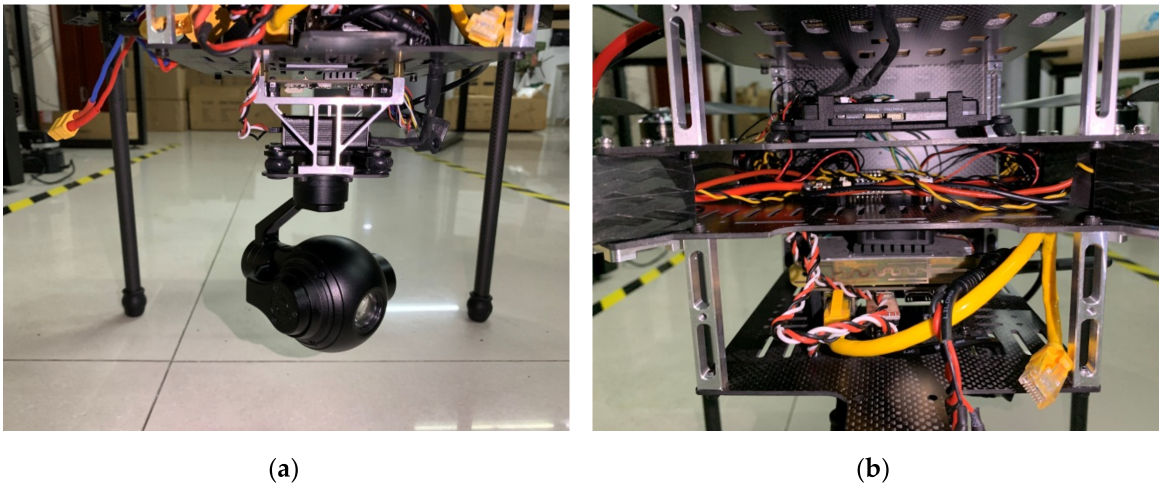


Figure 14. The customized quadrotor platform.



(**a**) (**b**)

Figure 15. The gimbaled camera (**a**) and onboard equipment (**b**).

As shown in Figure 15b, the quadrotor is equipped with an embedded microcontroller developed by the Pixhawk team at ETU Zürich [66]. The selected firmware version is 1.9.2.

To achieve real-time image processing, a NVIDIA Jetson TX2 module is used as an onboard computer to implement the tracking algorithm, which is almost the fastest and most power-efficient embedded AI computing device. The output data of the vision system can be transferred from the onboard computer to the Pixhawk flight controller using serial communication, which is based on a MAVLINK [67] extendable communication node for the Robot Operation System (ROS) [68]. The control rate is at 30 Hz limited by the onboard processing speed. By using a 2.4 GHz remote controller (RC), the tracking process can be initiated by switching to the offboard mode when a target is selected.

5.2. Experimental Results and Analysis

To evaluate the performance of the proposed tracking and targeting system, we test it in different situations. After selecting a target in the video streams, the gimbal system is activated and rotates the camera to point at the selected target, which can be regarded as a step response. The boresight error pixels are plotted in azimuth and elevation, respectively, which are also printed on the top left of the screen.

As shown in Figure 16, the system responds rapidly and the steady-state error is about ±3 pixels, while the initial errors are hundreds of pixels. This test is usually used to tune all the control parameters of the gimbal system, which can be completed on the ground.

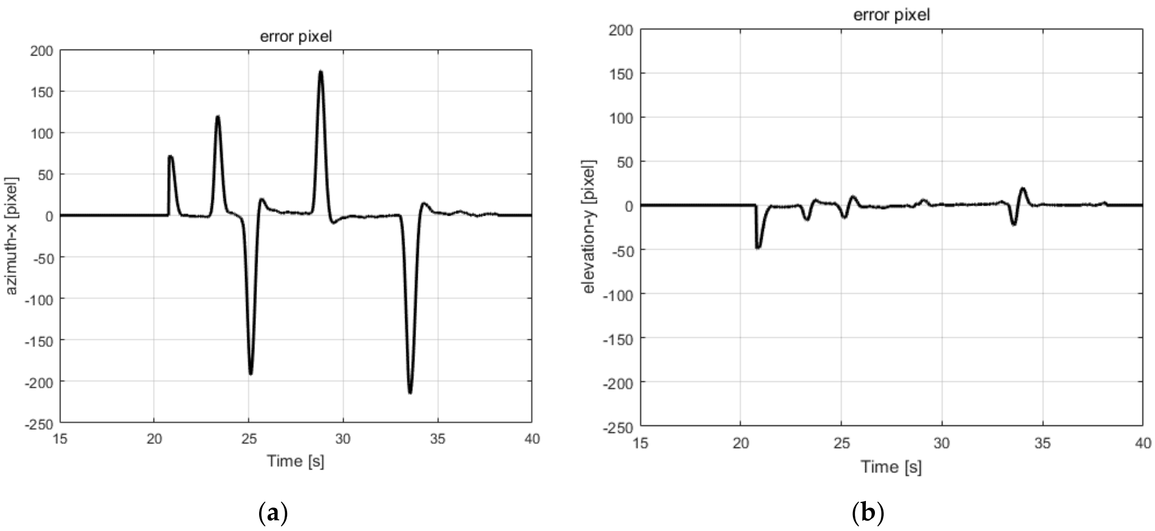
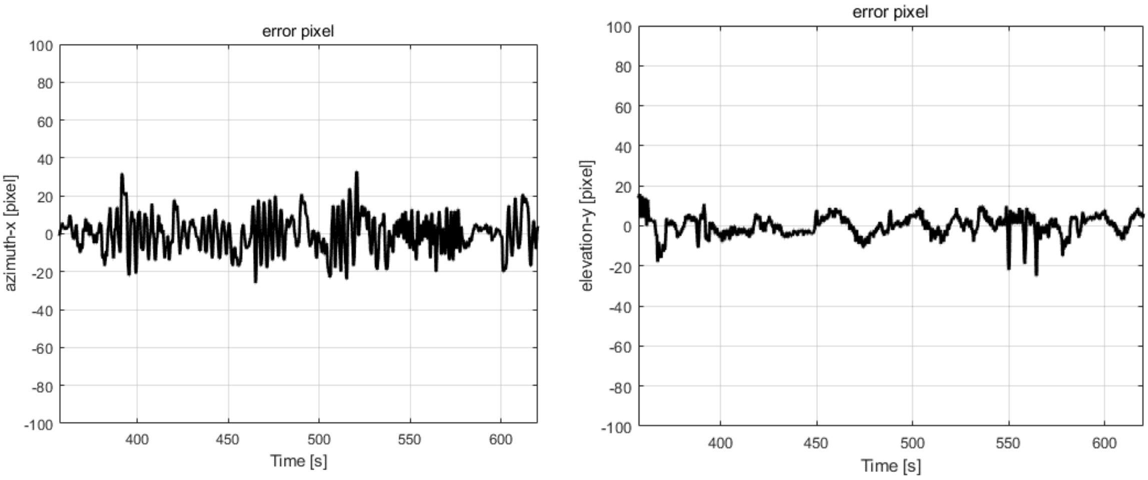


Figure 16. Boresight error in pixels during ground step tests. (**a**) the response of the gimbal system in azimuth; (**b**) the response of the gimbal system in elevation.

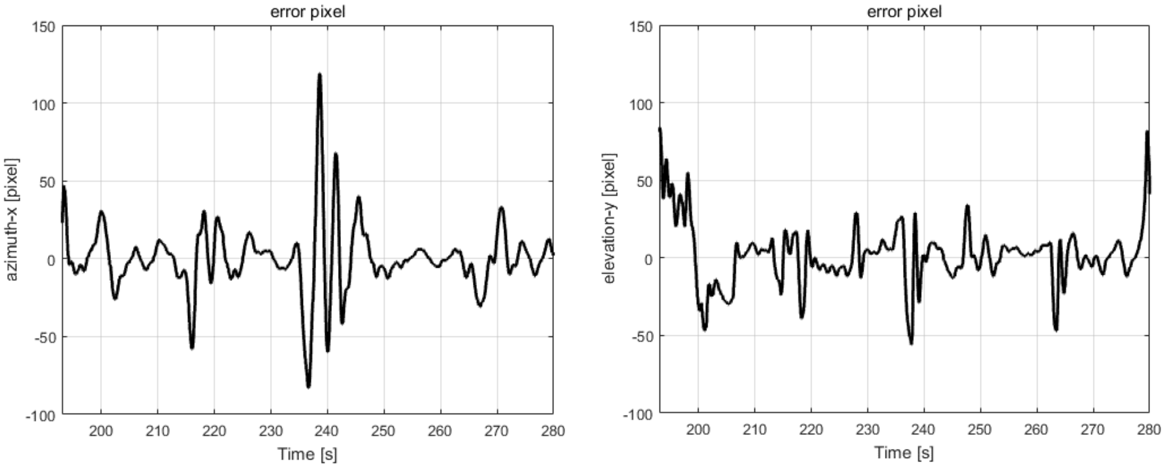
Then, the gimbal system and the onboard computer are fixed on the experimental quadrotor platform for further tests. To aim at a target from the UAV, the motion and vibration of the platform should be isolated. Once the target is locked, the UAV is in a fully autonomous mode controlled by the integrated system. Figure 17 shows the results of the boresight error while the UAV is following a pedestrian moving at 0.9–2 m/s. In about 250 seconds' flight, an accuracy of ±9.34 pixels in the azimuth and ±5.07 pixels in the elevation was achieved.



(**a**) (**b**)

Figure 17. Boresight error in pixels while the UAV is following a human target moving at pedestrian speed. (**a**) the response in azimuth; (**b**) the response in elevation.

While tracking a ground moving pedestrian, the altitude change of the target can be ignored in most situations, which makes the task less difficult. However, tracking a flying drone is much more complicated. The drones are able to change its position and velocity in a very short time, which may cause tracking errors or failures. During the flight experiment, autonomous tracking of an intruded drone has been achieved. Figure 18 shows the results of the boresight error while the UAV is tracking a flying drone.



(**a**) (**b**)

Figure 18. Boresight error in pixels while the UAV is tracking a flying drone. (**a**) the response in azimuth; (**b**) the response in elevation.

Some oscillations still remain in the current configuration, which occurred when the flight path of the target suddenly changed. It is a great challenge for the system to catch up with the target in such a short time. The deviations caused by the image transmission delay cannot be ignored, which is about 220 milliseconds for the current system. Other factors such as image noises and illumination changes may also have an impact on the tracking accuracy to some extent. The root mean square errors (RMSEs) of boresight errors in drone tracking experiment are listed in Table 1, compared with the other two tests.

Table 1. Root mean square errors (RMSEs) of boresight errors in different cases.

RMSE (Pixel)	Azimuth	Elevation	2D
Step test	31.57	4.83	31.94
Pedestrian following	9.34	5.07	10.63
Drone tracking	21.50	19.28	28.88

Figure 19 shows the process of a successful drone tracking experiment. During the flight, the roll/pitch/yaw angle of the UAV, camera, and gimbal frames are plotted, respectively, in Figure 20a–c. The actual approaching speed of the UAV has tracked the setpoint changes accurately as shown in Figure 20d. The trajectories of the intruded drone and the interceptor are plotted in a local NED frame as shown in Figure 21.

Higher control rate and less image transmission delay would significantly improve the response speed and the accuracy, if better hardware configuration were used. However, considering that all the sensors and onboard devices are low-cost, the performance of the proposed tracking system is very attractive in practical applications. A video of the experiments is available in the Supplementary Materials section (Video S1).

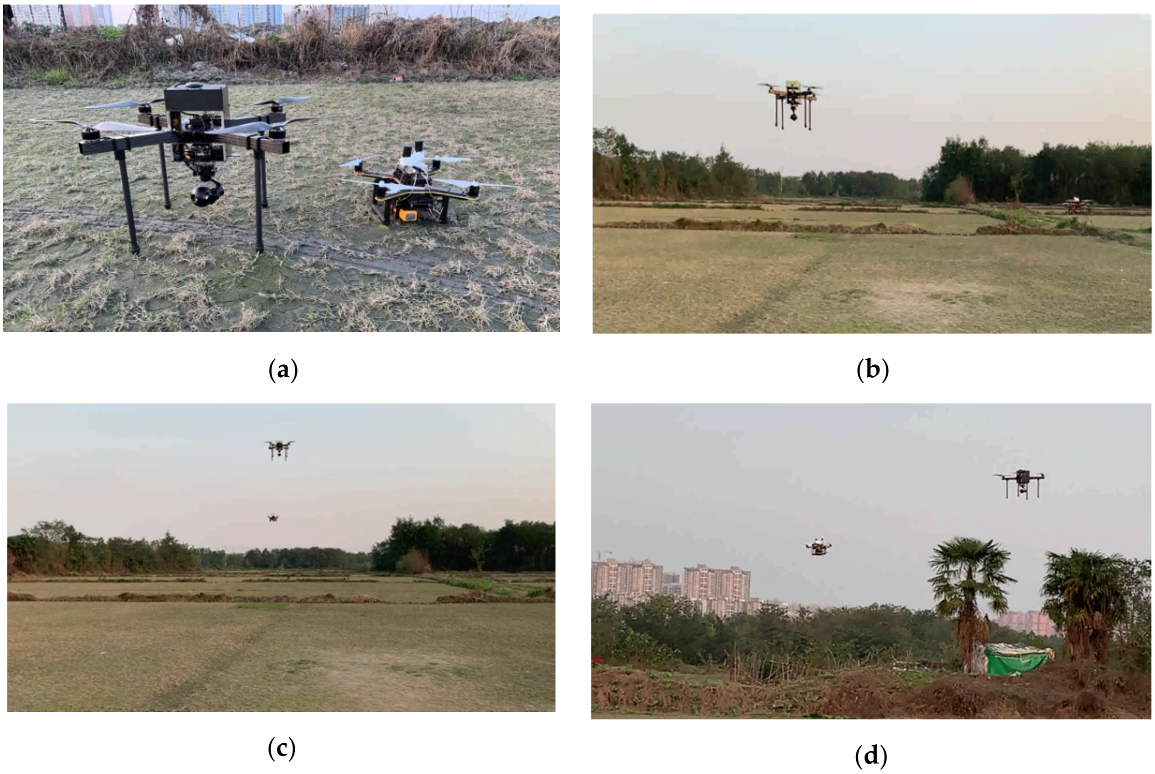
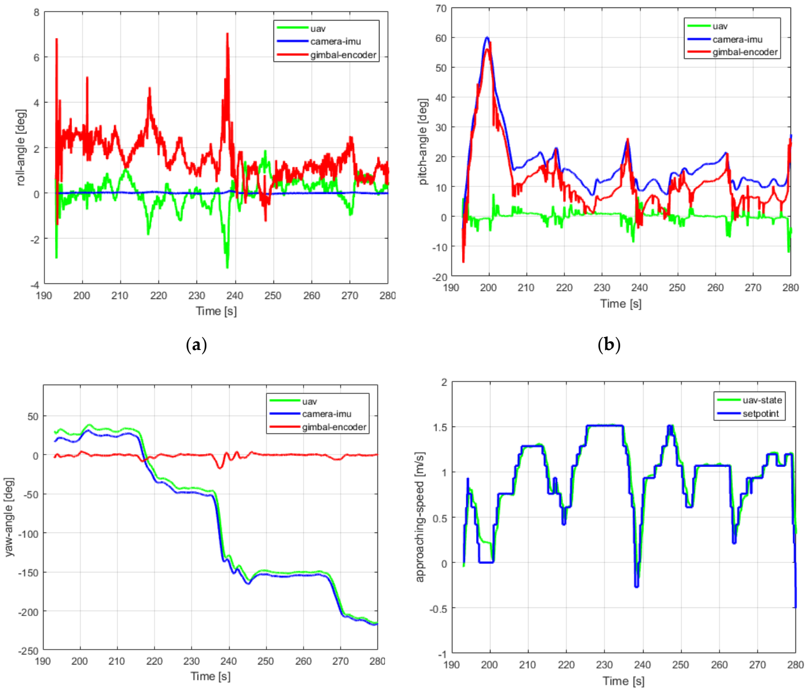

(**a**)
(**b**)
(**c**)
(**d**)

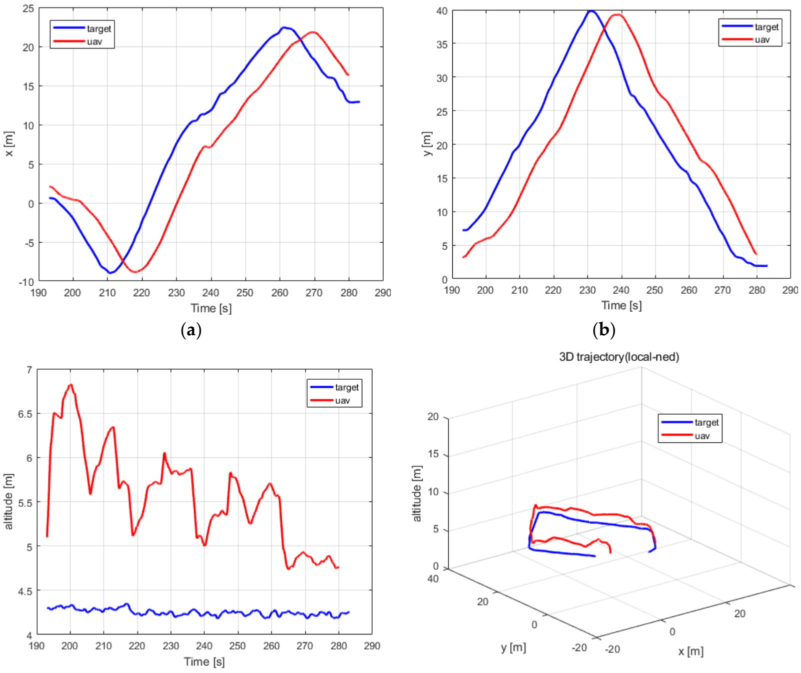
Figure 19. Autonomously tracking a flying drone with the experimental platform. (**a**) the target drone and the interceptor (the quadrotor platform); (**b**) the searching phase; (**c**) start the tracking phase when the target is recognized; (**d**) keep following the target drone.



(**a**) (**b**)

(**c**) (**d**)

Figure 20. The attitude and approaching speed of the UAV while autonomously tracking a flying drone. (**a**) roll angle measured by the camera IMU, encoder and UAV navigation system; (**b**) pitch angle measured by the camera IMU, encoder, and UAV navigation system; (**c**) yaw angle measured by the camera IMU, encoder, and UAV navigation system; and (**d**) the desired and actual approaching speed of the UAV.



(**a**) (**b**)

(**c**) (**d**)

Figure 21. The actual position (**a**–**c**) and 3D trajectory (**d**) of the UAV and the target drone in the local NED frame.

6. Conclusions

In the presented work, we proposed an onboard visual tracking system which consists of a gimbaled camera, an onboard computer for image processing and a microcontroller to control the UAV to approach the moving target. Our system used a KCF-based algorithm to detect and track an arbitrary object in real time, which has proved its efficiency and reliability in experiments. With the visual information, the 3-axis gimbal system autonomously aims at the selected target, which has achieved good performance during real flights. The proposed system has been demonstrated through real-time target tracking experiments, which enabled a low-cost quadrotor to chase a flying drone as shown in the video.

Future work may include using a laser ranging module attached to the camera, which is able to provide an accurate distance of the UAV with respect to the target. Even though this will increase the cost of the system, we look forward to its potential applications such as target geo-location and autonomous landing. Performance improvements could also be achieved by using deep learning-based detection algorithms combined with a large number of sample images. The CMOS sensors used in this paper are low-cost, which could lead to the effect of rolling shutter [69]. If the error is dramatic, compensation should be made to handle this issue.

Supplementary Materials: The following are available online at http://www.mdpi.com/2076-3417/10/15/5064/s1, Video S1: Gimbal-based Visual Tracking System for UAV.

Author Contributions: Conceptualization, X.L. and S.Z.; methodology, X.L. and Y.Y.; software, X.L. and C.M.; validation, X.L. and Y.Y.; formal analysis, J.L.; investigation, C.M.; data curation, J.L.; funding acquisition, Y.Y. and S.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Science and Technology Committee of China (02-ZT-005-021-01), National Science Key Lab Project of the Technology of Space Intelligent Control (No. HTKJ2019KL5022016), the Research Projects of Equipment (No. 2018-824), and the Support Program of Young Talents of Huxiang (No. 2019RS2029), Chinese Postdoctoral Science Foundation (No. 47661), The sixth Youth Fund Project of High Resolution Earth Observation System.

Acknowledgments: The authors would like to thank Rongwei Li, Qi Xiao, Xin Yi, and Ren Jin for their contribution to the experiments.

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

References

1. Cai, G.; Dias, J.; Seneviratne, L. A Survey of Small-Scale Unmanned Aerial Vehicles: Recent Advances and Future Development Trends. *Unmanned Syst.* **2014**, *2*, 175–199. [CrossRef]
2. Tomic, T.; Schmid, K.; Lutz, P.; Domel, A.; Kassecker, M.; Mair, E.; Grixa, I.L.; Ruess, F.; Suppa, M.; Burschka, D. Toward a Fully Autonomous UAV: Research Platform for Indoor and Outdoor Urban Search and Rescue. *Robot. Autom. Mag.* **2012**, *19*, 46–56. [CrossRef]
3. Judy, J.W. Microelectromechanical systems (MEMS): Fabrication, design and applications. *Smart Mater. Struct.* **2001**, *10*, 1115–1134. [CrossRef]
4. Zhao, Z.; Yang, J.; Niu, Y.; Zhang, Y.; Shen, L. A Hierarchical Cooperative Mission Planning Mechanism for Multiple Unmanned Aerial Vehicles. *Electronics* **2019**, *8*, 443. [CrossRef]
5. Laliberte, A.S.; Jeffrey, E.H.; Rango, A.; Winters, C. Acquisition, Orthorectification, and Object-based Classification of Unmanned Aerial Vehicle (UAV) Imagery for Rangeland Monitoring. *Photogramm. Eng. Remote Sens.* **2015**, *76*, 661–672. [CrossRef]
6. Hausamann, D.; Zirnig, W.; Schreier, G.; Strobl, P. Monitoring of gas pipelines - a civil UAV application. *Aircr. Eng. Aerosp. Technol.* **2005**, *77*, 352–360. [CrossRef]
7. Bristeau, P.J.; Callou, F.; Vissière, D.; Petit, N. The Navigation and Control technology inside the AR.Drone micro UAV. In Proceedings of the 18th World Congress of the International Federation of Automatic Control (IFAC), Milano, Italy, 28 August–2 September 2011; pp. 1477–1484.
8. Tai, J.C.; Tseng, S.T.; Lin, C.P.; Song, K.T. Real-time image tracking for automatic traffic monitoring and enforcement applications. *Image Vis. Comput.* **2004**, *22*, 485–501. [CrossRef]

9. Chow, J.Y.J. Dynamic UAV-based traffic monitoring under uncertainty as a stochastic arc-inventory routing policy. *Int. J. Transp. Sci. Technol.* **2016**, *5*, 167–185. [CrossRef]
10. Silvagni, M.; Tonoli, A.; Zenerino, E.; Chiaberge, M. Multipurpose UAV for search and rescue operations in mountain avalanche events. *Geomat. Nat. Hazards Risk* **2017**, *8*, 18–33. [CrossRef]
11. Bejiga, M.B.; Zeggada, A.; Nouffidj, A.; Melgani, F. A Convolutional Neural Network Approach for Assisting Avalanche Search and Rescue Operations with UAV Imagery. *Remote Sens.* **2017**, *9*, 100. [CrossRef]
12. Li, Z.; Liu, Y.; Walker, R.; Hayward, R.; Zhang, J. Towards automatic power line detection for a UAV surveillance system using pulse coupled neural filter and an improved Hough transform. *Mach. Vis. Appl.* **2010**, *21*, 677–686. [CrossRef]
13. Sa, I.; Hrabar, S.; Corke, P. Inspection of Pole-Like Structures Using a Visual-Inertial Aided VTOL Platform with Shared Autonomy. *Sensors* **2015**, *15*, 22003–22048. [CrossRef] [PubMed]
14. Chuang, H.-M.; He, D.; Namiki, A. Autonomous Target Tracking of UAV Using High-Speed Visual Feedback. *Appl. Sci.* **2019**, *9*, 4552. [CrossRef]
15. Wang, F.; Cong, X.-B.; Shi, C.-G.; Sellathurai, M. Target Tracking While Jamming by Airborne Radar for Low Probability of Detection. *Sensors* **2018**, *18*, 2903. [CrossRef] [PubMed]
16. Wang, Y.; Lei, H.; Ye, J.; Bu, X. Backstepping Sliding Mode Control for Radar Seeker Servo System Considering Guidance and Control System. *Sensors* **2018**, *18*, 2927. [CrossRef]
17. Liu, X.; Zhang, S.; Tian, J.; Liu, L. An Onboard Vision-Based System for Autonomous Landing of a Low-Cost Quadrotor on a Novel Landing Pad. *Sensors* **2019**, *19*, 4703. [CrossRef]
18. Yang, S.; Scherer, S.A.; Schauwecker, K.; Zell, A. Autonomous Landing of MAVs on an Arbitrarily Textured Landing Site Using Onboard Monocular Vision. *J. Intell. Robot. Syst.* **2014**, *74*, 27–43. [CrossRef]
19. Wenzel, K.E.; Rosset, P.; Zell, A. Low-cost visual tracking of a landing place and hovering flight control with a microcontroller. *J. Intell. Robot. Syst.* **2010**, *57*, 297–311. [CrossRef]
20. Saripalli, S.; Montgomery, J.F.; Sukhatme, G.S. Visually guided landing of an unmanned aerial vehicle. *IEEE Trans. Robot. Autom.* **2003**, *19*, 371–380. [CrossRef]
21. Yang, S.; Scherer, S.A.; Zell, A. An onboard monocular vision system for autonomous takeoff, hovering and landing of a micro aerial vehicle. *J. Intell. Robot. Syst.* **2013**, *69*, 499–515. [CrossRef]
22. Olson, E. AprilTag: A robust and flexible visual fiducial system. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, 9–13 May 2011; pp. 3400–3407.
23. Garrido-Jurado, S.; Muñoz-Salinas, R.; Madrid-Cuevas, F.J.; Marín-Jiménez, M.J. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognit.* **2014**, *47*, 2280–2292. [CrossRef]
24. Kyristsis, S.; Antonopoulos, A.; Chanialakis, T.; Stefanakis, E.; Linardos, C.; Tripolitsiotis, A.; Partsinevelos, P. Towards Autonomous Modular UAV Missions: The Detection, Geo-Location and Landing Paradigm. *Sensors* **2016**, *16*, 1844. [CrossRef] [PubMed]
25. Pestana, J.; Sanchez-Lopez, J.L.; de la Puente, P.; Carrio, A.; Campoy, P. A Vision-based Quadrotor Swarm for the participation in the 2013 International Micro Air Vehicle Competition. In Proceedings of the 2014 International Conference on Unmanned Aircraft Systems (ICUAS), Orlando, FL, USA, 27–30 May 2014; pp. 617–622.
26. Ma, L.; Hovakimyan, N. Vision-based cyclic pursuit for cooperative target tracking. In Proceedings of the 2011 American Control Conference, San Francisco, CA, USA, 29 June–1 July 2011; pp. 4616–4621.
27. Ross, D.A.; Lim, J.; Lin, R.S.; Yang, M.H. Incremental Learning for Robust Visual Tracking. *Int. J. Comput. Vis.* **2008**, *77*, 125–141. [CrossRef]
28. Mei, X.; Ling, H. Robust visual tracking using $\ell 1$ minimization. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Kyoto, Japan, 29 September–2 October 2009; pp. 1436–1443.
29. Li, H.; Shen, C.; Shi, Q. Real-time visual tracking using compressive sensing. In Proceedings of the 24th IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Colorado Springs, CO, USA, 20–25 June 2011; pp. 1305–1312.
30. Wang, S.; Lu, H.; Yang, F.; Yang, M.H. Superpixel tracking. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Barcelona, Spain, 6–13 November 2011; pp. 1323–1330.
31. Cheng, Y. Mean Shift, Mode Seeking, and Clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* **1995**, *17*, 790–799. [CrossRef]

32. Comaniciu, D.; Meer, P. Mean Shift: A Robust Approach Toward Feature Space Analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 603–619. [CrossRef]
33. Collins, R.T. Mean-shift blob tracking through scale space. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Madison, WI, USA, 16–22 June 2003; pp. 234–240.
34. Bradski, G.R. Computer Vision Face Tracking for Use in a Perceptual User Interface. *Intel Technol. J.* **1998**, *2*, 12–21.
35. Babenko, B.; Yang, M.H.; Belongie, S. Robust Object Tracking with Online Multiple Instance Learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 1619–1632. [CrossRef]
36. Hare, S.; Golodetz, S.; Saffari, A.; Vineet, V.; Cheng, M.M.; Hicks, S.L.; Torr, P.H. Struck: Structured Output Tracking with Kernels. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 2096–2109. [CrossRef]
37. Grabner, H.; Bischof, H. Online Boosting and Vision. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), New York, NY, USA, 17–22 June 2006; pp. 260–267.
38. Kalal, Z.; Mikolajczyk, K.; Matas, J. Tracking-Learning-Detection. *IEEE Trans. Softw. Eng.* **2011**, *34*, 1409–1422. [CrossRef]
39. Wei, J.; Liu, F. Coupled-Region Visual Tracking Formulation Based on a Discriminative Correlation Filter Bank. *Electronics* **2018**, *7*, 244. [CrossRef]
40. Bolme, D.S.; Beveridge, J.R.; Draper, B.A.; Lui, Y.M. Visual object tracking using adaptive correlation filters. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, FL, USA, 13–18 June 2010; pp. 2544–2550.
41. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. Exploiting the Circulant Structure of Tracking-by-Detection with Kernels. In Proceedings of the 12th European conference on Computer Vision, Florence, Italy, 7–13 October 2012; Springer: Berlin, Germany, 2012; pp. 702–715.
42. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. High-Speed Tracking with Kernelized Correlation Filters. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 583–596. [CrossRef] [PubMed]
43. Wu, Y.; Lim, J.; Yang, M.H. Online Object Tracking: A Benchmark. In Proceedings of the Computer Vision and Pattern Recognition (CVPR), Portland, OR, USA, 23–28 June 2013; pp. 2411–2418.
44. Montero, A.S.; Lang, J.; Laganière, R. Scalable Kernel Correlation Filter with Sparse Feature Integration. In Proceedings of the IEEE International Conference on Computer Vision Workshop, Santiago, Chile, 7–13 December 2015; pp. 587–594.
45. Zhang, Y.H.; Zeng, C.N.; Liang, H.; Luo, J.; Xu, F. A visual target tracking algorithm based on improved Kernelized Correlation Filters. Proceedings of 2016 IEEE International Conference on Mechatronics and Automation, Harbin, China, 7–10 August 2016; pp. 199–204.
46. Santos, D.A.; Gonçalves, P.F.S.M. Attitude Determination of Multirotor Aerial Vehicles Using Camera Vector Measurements. *J. Intell. Robot. Syst.* **2016**, *86*, 1–11. [CrossRef]
47. Falanga, D.; Zanchettin, A.; Simovic, A.; Delmerico, J.; Scaramuzza, D. Vision-based Autonomous Quadrotor Landing on a Moving Platform. In Proceedings of the 15th IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), Shanghai, China, 11–13 October 2017.
48. Borowczyk, A.; Nguyen, D.T.; Phu-Van Nguyen, A.; Nguyen, D.Q.; Saussié, D.; Le Ny, J. Autonomous Landing of a Multirotor Micro Air Vehicle on a High Velocity Ground Vehicle. *arXiv* **2016**, arXiv:1611.07329. Available online: http://arxiv.org/abs/1611.07329 (accessed on 26 May 2020).
49. Quigley, M.; Goodrich, M.A.; Griffiths, S.; Eldredge, A.; Beard, R.W. Target Acquisition, Localization, and Surveillance Using a Fixed-Wing Mini-UAV and Gimbaled Camera. In Proceedings of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain, 18–22 April 2005; pp. 2600–2605.
50. Jakobsen, O.C.; Johnson, E. Control Architecture for a UAV-Mounted Pan/Tilt/Roll Camera Gimbal. In Proceedings of the Infotech@Aerospace, AIAA 2005, Arlington, VA, USA, 26–29 September 2005; p. 7145.
51. Whitacre, W.; Campbell, M.; Wheeler, M.; Stevenson, D. Flight Results from Tracking Ground Targets Using SeaScan UAVs with Gimballing Cameras. In Proceedings of the 2007 American Control Conference, New York, NY, USA, 11–13 July 2007; pp. 377–383.
52. Dobrokhodov, V.N.; Kaminer, I.I.; Jones, K.D.; Ghabcheloo, R. Vision-based tracking and motion estimation for moving targets using small UAVs. In Proceedings of the 2006 American Control Conference, Minneapolis, MN, USA, 14–16 June 2006.

53. Johansson, J. Modelling and Control of an Advanced Camera Gimbal. Teknik Och Teknologier, 2012. Available online: http://www.diva-portal.org/smash/get/diva2:575341/FULLTEXT01.pdf (accessed on 26 May 2020).
54. Kazemy, A.; Siahi, M. Equations of Motion Extraction for a Three Axes Gimbal System. *Modares J. Electr. Eng.* **2014**, *13*, 37–43.
55. Barnes, F.N. Stable Member Equations of Motion for a Three-Axis Gyro Stabilized Platform. *IEEE Trans. Aerosp. Electron. Syst.* **1971**, *7*, 830–842. [CrossRef]
56. Otlowski, D.R.; Wiener, K.; Rathbun, B.A. Mass properties factors in achieving stable imagery from a gimbal mounted camera. In *Airborne Intelligence, Surveillance, Reconnaissance (ISR) Systems and Applications 2008*; International Society for Optics and Photonics: Orlando, FL, USA, 2008; Volume 6946.
57. Rifkin, R.; Yeo, G.; Poggio, T. Regularized Least-Squares Classification. In *Nato Science Series Sub Series III Computer and Systems Sciences*; IOS Press: Amsterdam, The Netherlands, 2003; Volume 190, pp. 131–154.
58. Islam, M.M.; Hu, G.; Liu, Q. Online Model Updating and Dynamic Learning Rate-Based Robust Object Tracking. *Sensors* **2018**, *18*, 2046. [CrossRef]
59. Pestana, J.; Mellado-Bataller, I.; Sanchez-Lopez, J.L.; Fu, C.; Mondragon, I.F.; Campoy, P. A General Purpose Configurable Controller for Indoors and Outdoors GPS-Denied Navigation for Multirotor Unmanned Aerial Vehicles. *J. Intell. Robot. Syst.* **2014**, *73*, 387–400. [CrossRef]
60. Tan, R.; Kumar, M. Tracking of Ground Mobile Targets by Quadrotor Unmanned Aerial Vehicles. *Unmanned Syst.* **2014**, *2*, 157–173. [CrossRef]
61. Guelman, M. The Closed-Form Solution of True Proportional Navigation. *IEEE Trans. Aerosp. Electron. Syst.* **1976**, *12*, 472–482. [CrossRef]
62. Yuan, P.J.; Chern, J.S. Ideal proportional navigation. *Adv. Astronaut. Sci.* **1992**, *95*, 81374. [CrossRef]
63. Mahapatra, P.R.; Shukla, U.S. Accurate solution of proportional navigation for maneuvering targets. *IEEE Trans. Aerosp. Electron. Syst.* **1989**, *25*, 81–89. [CrossRef]
64. Park, S.; Deyst, J.; How, J. A new nonlinear guidance logic for trajectory tracking. In Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit, Providence, RI, USA, 16–19 August 2004.
65. Mathisen, S.H.; Gryte, K.; Johansen, T.; Fossen, T.I. Non-linear Model Predictive Control for Longitudinal and Lateral Guidance of a Small Fixed-Wing UAV in Precision Deep Stall Landing. In Proceedings of the AIAA SciTech, San Diego, CA, USA, 4–8 January 2016.
66. Pixhawk. Available online: https://pixhawk.org/ (accessed on 20 May 2020).
67. Mavros. Available online: https://github.com/mavlink/mavros/ (accessed on 20 May 2020).
68. Ros. Available online: http://www.ros.org/ (accessed on 20 May 2020).
69. Mathisen, S.H.; Gryte, K.; Johansen, T.; Fossen, T.I. Cross-Correlation-Based Structural System Identification Using Unmanned Aerial Vehicles. *Sensors* **2017**, *17*, 2075.