

Article

# Autonomous Target Tracking of UAV Using High-Speed Visual Feedback

Hsiu-Min Chuang , Dongqing He and Akio Namiki \* 

Graduate School of Science and Engineering, Chiba University, Chiba 263-8522, Japan; hsiumin.chuang@chiba-u.jp (H.-M.C.); katoukei1994@chiba-u.jp (D.H.)

\* Correspondence: namiki@faculty.chiba-u.jp

Received: 7 September 2019; Accepted: 22 October 2019; Published: 26 October 2019



**Featured Application:** Direct visual servoing of UAVs.

**Abstract:** Most current unmanned aerial vehicles (UAVs) primarily use a global positioning system (GPS) and an inertial measurement unit (IMU) for position estimation. However, compared to birds and insects, the abilities of current UAVs to recognize the environment are not sufficient. To achieve autonomous flight of UAVs, like birds, the UAVs should be able to process and respond to information from their surrounding environment immediately. Therefore, in this paper, we propose a direct visual servoing system for UAVs, using an onboard high-speed monocular camera. There are two advantages of this system. First, the high image sampling rates help to improve the ability to recognize the environment. Second, the issue of control latency can be effectively solved because the position control signals are transmitted to the flight controller directly. In the experiment, the UAV could recognize a target at update rates of about 350 Hz, and a target tracking task was successfully realized.

**Keywords:** UAV; high-speed vision; visual servoing

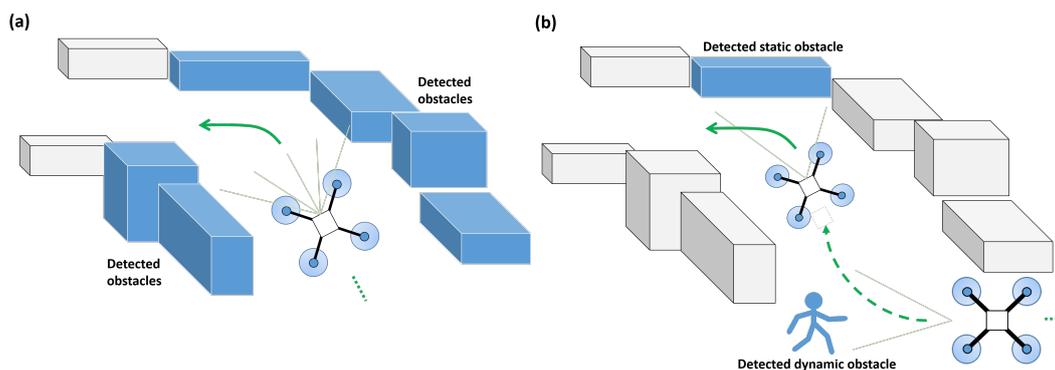
## 1. Introduction

Unmanned aerial vehicles (UAVs), which are often called drones or multicopters, have become a popular research topic because of their high autonomy and flexibility. To achieve autonomous flight of a UAV, it is important to obtain its position and attitude in real time. In most current UAVs, an inertial measurement unit (IMU) is used to acquire the attitude, and the Global Positioning System (GPS) is used to acquire the position. These signals are then fused by a Kalman filter. However, GPS is not suitable in indoor environments. Even outdoors, it is also unsuitable if there are many obstacles. When the GPS signal is blocked, the UAV easily loses its locational information. On the other hand, flying creatures such as birds and insects use mainly visual information for their flight control [1]. In particular, in environments with many obstacles, visual information plays an important role in collision avoidance [2,3]. In order to realize completely autonomous flight of a UAV in any environment, a robust and stable visual feedback flight control method is indispensable.

In conventional research on vision systems for UAVs, visual simultaneous localization and mapping (Visual SLAM) and visual odometry (VO) for vision-based navigation are the main topics. The V-SLAM technique constructs a live map of the surrounding environment passed by a UAV, and the UAV localizes itself relative to this constructed map [4,5]. VO estimates the egomotion of a UAV using successive captured images instead of constructing a complete environmental map of the UAV [6]. These studies focused on the recognition of a 3D environment, and UAVs were controlled based on the recognition result. However, the computational cost of V-SLAM is high, and the processing speed is not very high.

On the other hand, in flying organisms, real-time visual information is used to avoid collisions and fly in narrow spaces. In such cases, it is not necessary to obtain a detailed 3D environment map in world coordinates, and a UAV can fly by visual servoing even if only the relative position and orientation between the UAV's body and the obstacle is obtained. The process of accurate 3D environment recognition is omitted, enabling responsive and quick flight. We think that the performance of such responsive visual servoing in conventional UAV research is still not high enough. There have been studies using simple optical flow sensors [7,8], ultrasonic sensors [9,10], and laser imaging detection and ranging (LIDAR) sensors for self-navigation [11,12]. However, there are not many examples using visual servoing control for UAVs. This is due to the low frame rate of conventional vision systems and the heavy computational cost of visual processing.

Our group has developed various types of high-speed vision systems with processing speeds of around 500 Hz to 1000 Hz [13]. In this paper, we developed a new lightweight high-speed vision system for UAVs and visual servo control for quick target tracking. The proposed vision system could recognize the surrounding environment in a short time and could immediately transmit control signals to the flight controller. Figure 1 shows the difference between the relative-position-based visual servoing used in this paper and the conventional approach. The relative-position-based visual servoing estimates the relative position and orientation to the target for visual servoing instead of collecting complete environmental information for making a control strategy.



**Figure 1.** Concepts of conventional approach (a) and relative-position-based visual servoing (b).

This paper is organized as follows: In Section 2, research on UAV control using visual information is described. In Section 3, we introduce the system configuration of our UAV and the moving target used in the experiment. In Section 4, we explain the process of image processing for estimating the relative pose between a target and the UAV. In Section 5, we introduce the principle of the relative-position-based visual servoing and the controller design for a position hold task. In Section 6, first we explain the experimental method and present the experimental results of the position hold task and target tracking in real flight, and then we discuss the experimental results. Finally, we conclude our work in Section 7.

## 2. Related Works

A number of studies on controlling UAVs using visual sensors have been conducted in recent years [14]. In particular, studies on visual servoing for UAVs are reported in [15]. Watanabe et al., proposed a multi-camera visual servo system for controlling unmanned micro helicopters by using visual information [16]. Yu et al., presented a visual control method for a small UAV, in which binocular stereo vision and an ultrasonic system are fused to implement an obstacle avoidance strategy [17]. Guenard et al., proposed an eye-in-hand visual servo control system for stationary flight of a UAV [18]. Teuliere et al., utilized a color-based object recognition method to chase a moving target from a flying UAV [19]. Eberli et al., proposed a pose estimation algorithm using one single circular landmark for controlling a quadrotor [20]. Jung et al., presented a visual navigation strategy that enables a UAV

to fly through various obstacles in complex environments [21]. Lee et al., proposed a method of autonomous landing of a VTOL UAV on a moving platform using image-based visual servoing (IBVS) instead of calculating the position of the target [22]. Jabbari et al., proposed an adaptive dynamic IBVS scheme for underactuated UAV control [23]. Araar et al., proposed and compared two visual servoing schemes based on Hough parameters to track linear structured infrastructures [24]. Falanga presented a quadrotor system capable of autonomously landing on a moving platform using an onboard vision sensor for localization and motion estimation of the moving platform [25]. Thomas et al., proposed vision-based localization and servoing for a quadrotor to realize autonomous dynamic grasping and perching [26,27]. In addition to these studies, for improving the flight performance of UAVs, a number of robust visual servo controller designs have also been presented [28–32].

In previous studies, the highest reported onboard processing rate for the controller strategy was 75 Hz, which is not high enough to realize high-speed autonomous flight of a UAV. To achieve this goal, it is necessary to integrate a high-speed vision system into the UAV. A high-speed vision system is effective for controlling a robot at high speed, and various applications of high-speed vision have been developed [33–36]. Moreover, a high-speed tracking system [37] and a high-speed velocity estimator for UAVs [38] have also been proposed.

Considering the merit of the high noise tolerance of high-speed vision systems and the possibility for high-speed control, in this paper we propose a relative-position-based visual servoing method using an onboard high-speed camera to realize an autonomous target tracking task.

### 3. System

#### 3.1. Developed UAV

Figure 2a shows the platform we used in the experiment. For fast motion, we selected a QAV250 fiber frame as the flying platform. For high-speed target recognition, we chose a XIMEA MQ003CG-CM high-speed camera with a Theia SY110M ultra-wide lens. Considering the need for fast image processing and pose estimation, as the onboard companion computer, we chose a Jetson TX2, which is a low-power embedded platform launched by NVIDIA, including a 256-core NVIDIA Pascal GPU, a hex-core ARMv8 64-bit CPU complex, and an 8-GB LPDDR4 memory with a 128-bit interface. As the flight controller, we chose a Pixhawk2.1 Cube as the basic hardware for developing our own flight controller.

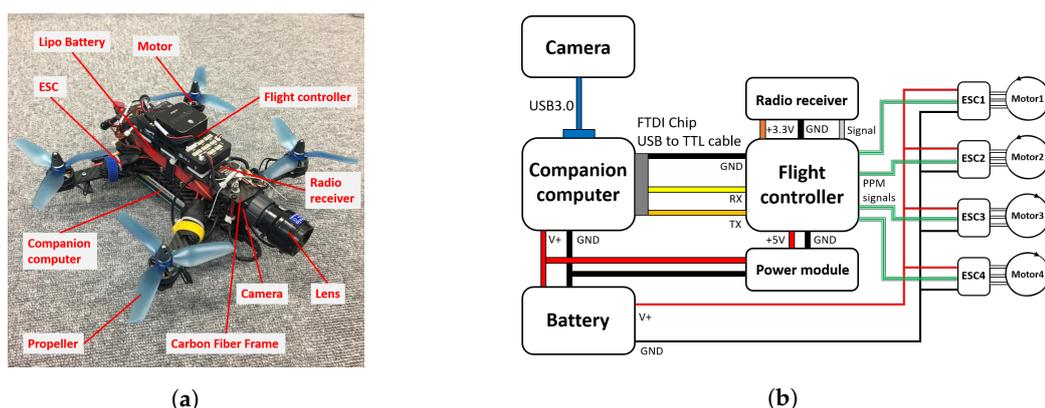


Figure 2. Developed UAV. (a) Platform (b) Wiring configuration.

Figure 2b shows the wiring diagram of our UAV. The high-speed camera transmitted a raw image to the companion computer via a USB3.0 cable. After the relative pose was calculated, the current position and setpoint were transmitted to the flight controller using the mavlink protocol via an FTDI USB to TTL cable. Finally, PPM signals were sent to ESCs for converting them to 3-phase signals to drive the brushless motors. The specifications of our platform are shown in Table 1.

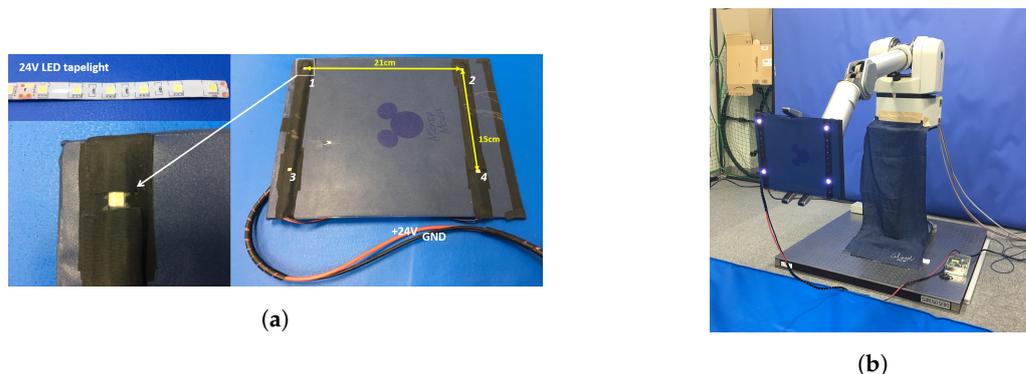
Moreover, the GPU-based Harris corner algorithm was used for detecting features. This algorithm was built on VisionWorks [39], which is a CUDA software development package for computer vision and image processing.

**Table 1.** Specifications of our platform.

Parameter	Value
Maximum diameter	250 [mm]
Total mass (with battery)	880 [g]
Companion computer	NVIDIA Jetson TX2
Lipo battery	2200 mAh 35C
Camera	XIMEA MQ003CG-CM (648 × 488 [pixels])
Lens	Theia SY110M (f:1.7 [mm])

### 3.2. Moving Target

In the experiment, we prepared a sample moving target as a reference for visual servoing to verify the performance quantitatively. For convenience of fast feature detection in the image processing, we attached four LED lights on a plate to serve as the moving target, as shown in Figure 3a. We define the coordinates of each LED as  $(0, 0, 0)$ ,  $(0.21, 0, 0)$ ,  $(0, 0.15, 0)$ ,  $(0.21, 0.15, 0)$  from points 1–4 respectively. For moving the target plate, we used a Barrett WAM 7-DoF robot manipulator, as shown in Figure 3b, and moved it with different speeds and accelerations [40].



**Figure 3.** Configuration of target object. (a) Target object. (b) Robot arm for moving target.

## 4. Vision-Based State Estimator

In this section, we explain the vision-based estimator of the pose of the UAV. This estimator outputs the relative position and orientation with respect to the target. By using the advantage of high-speed vision that objects appear to be stationary, simple high-speed image processing and state estimation is realized. The result is used in the controller explained in the next section.

The estimator is composed of the following four parts: (1) feature detection, (2) inlier extraction, (3) relative pose estimation, and (4) noise reduction.

### 4.1. Feature Detection

We use the Harris corner algorithm [41] for detecting features on the target. This is a general-purpose feature detection algorithm that can be easily applied to other recognition tasks. In addition, it is suitable for high-speed image processing because of its low computational load. However, the detection capability of the Harris corner algorithm is not good compared to other advanced image feature detection algorithms such as SURF [42] and FAST [43]. In the case of using high-speed vision, only slightly changes of the target are observed between frames, and temporal continuity can be assumed, so it is easy to prevent false detections.

If the following score  $\mathbf{R}$  is larger than a threshold value at an image point  $\mathbf{u}(x, y)$ , we define it as a corner:

$$\mathbf{R} = \det\mathbf{M} - k(\text{trace}\mathbf{M})^2 \tag{1}$$

$$\mathbf{M} \triangleq \sum_{x,y} w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \tag{2}$$

where  $I_x$  and  $I_y$  are image derivatives in the  $x$  and  $y$  directions, respectively,  $w$  is a window function for giving weights to the pixels underneath, and  $k$  is a constant for calculating  $\mathbf{R}$ .

#### 4.2. Image Centroid

At time  $t$ , we can classify features into some sections by using the center of all image features,  $C$ , as shown in Figure 4. Then the four features are computed as the centroid of all selected features by employing the image moment [44] in the following equation with known features in each section:

$$\mathbf{p} = \begin{bmatrix} \frac{m_{1,0}}{m_{0,0}} & \frac{m_{0,1}}{m_{0,0}} \end{bmatrix}^T \tag{3}$$

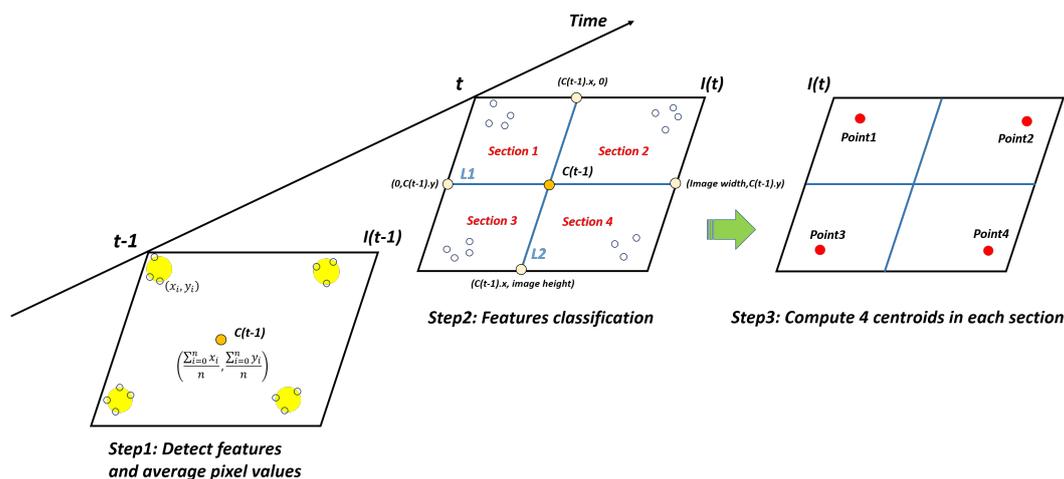


Figure 4. Image processing for selecting four inliers for pose estimation.

#### 4.3. Relative Pose Estimation— Perspective $n$ Points (PnP)

Once we have tracked features  $\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n \in \mathbf{R}^2$  on the image, with a known camera matrix and a set of 3D positions  ${}^O\mathbf{X}_1, {}^O\mathbf{X}_2, \dots, {}^O\mathbf{X}_n \in \mathbf{R}^3$ , we can derive the camera pose  ${}^O\mathbf{T}$  by employing the Perspective-n-Point (PnP) algorithm [45]. The general formulation of PnP is to find the homogeneous transformation matrix  ${}^O\mathbf{T}$  that minimizes the image reprojection error, as follows:

$$\arg \min \sum_k \| \mathbf{p}_n - \hat{\mathbf{p}}_n \|^2 \tag{4}$$

where  $\hat{\mathbf{p}}_n$  is the reprojection of the 3D point  ${}^O\mathbf{X}_n$  into the image according to the transformation  ${}^O\mathbf{T}$ .

By using iterative Levenberg-Marquardt optimization [46], we can find such a pose that minimizes the reprojection error of these four points on the target. The pose represents the relative position  ${}^O\mathbf{t}$  and attitude  ${}^O\mathbf{R}$  between the target's coordinate system and the camera's coordinate system.

Finally, since we use the relative position for the UAV’s position control directly, we applied a smoothing filter to the estimated position  ${}^O\mathbf{t}'$  as follows:

$${}^O\mathbf{t}'(i) = \frac{1}{m+1} \sum_{j=-m}^0 {}^O\mathbf{t}(i+j), \tag{5}$$

where  $i$  is the time step, and  $m$  is the order of smoothness. In the experiment, we set  $m$  to 7. On the other hand, the estimated orientation will be transformed to a quaternion and integrated with the IMU using the extended Kalman filter (EKF) algorithm for the attitude controller. Figure 5 shows the principle of PnP algorithm.

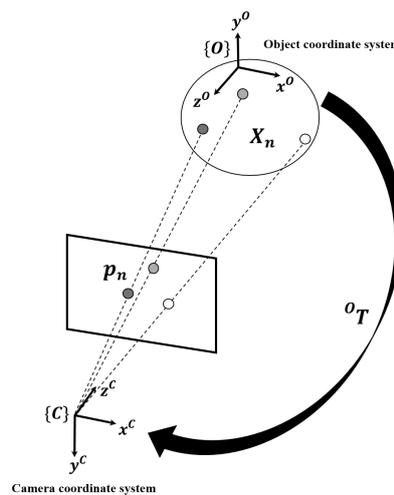


Figure 5. Principle of PnP.

Figure 6 shows a flowchart of the high-speed target recognition method in our system.

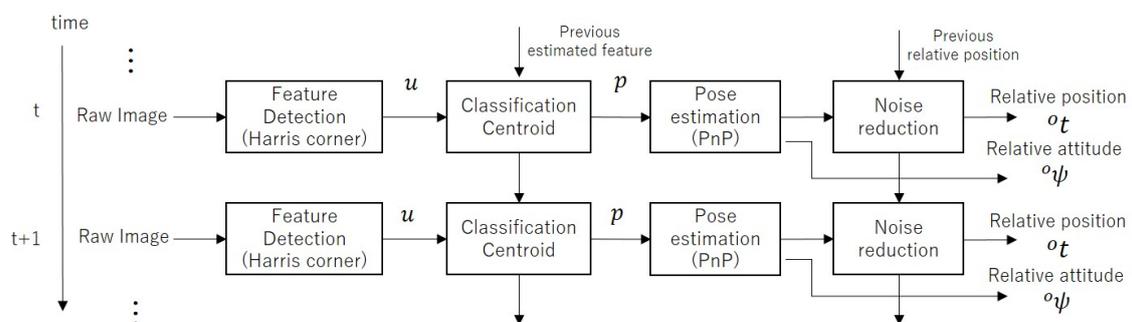


Figure 6. Flowchart of high-speed target recognition.

## 5. Visual Servoing Controller Design

### 5.1. Relative-Position-Based Visual Servoing

To realize responsive quick motion, we propose relative-position-based visual servoing, which calculates only the pose of the UAV relative to the target. Since only relative positions are necessary for our controller, the computational load of image processing on the onboard companion computer will be effectively decreased and we can get more available visual information in the same time duration. Figure 7 shows the relative-position-based visual servoing method we used in the experiment.

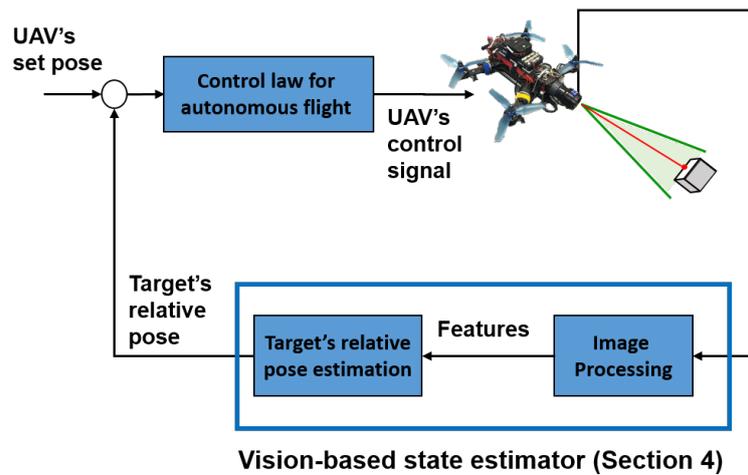


Figure 7. Control loop of relative-position-based high-speed visual servoing.

### 5.2. Coordinate Transformation

There are three coordinate systems in our system: the object coordinate system  $O$ , the camera coordinate system  $C$ , and the UAV's body coordinate system  $B$ , as shown in Figure 8a. We can obtain  ${}^Ox$  and  ${}^OR$  directly by PnP.

Considering the possibility of electromagnetic interference, we estimate the relative attitude by using a vision estimator instead of a magnetometer. The relative yaw angle  ${}^O\psi$  shown in Figure 8b is calculated by decomposing the relative rotation matrix derived from the PnP algorithm mentioned in the previous section. For the attitude control, the relative yaw angle estimation is used as an observation, and this is integrated with the IMU in the EKF state estimator.

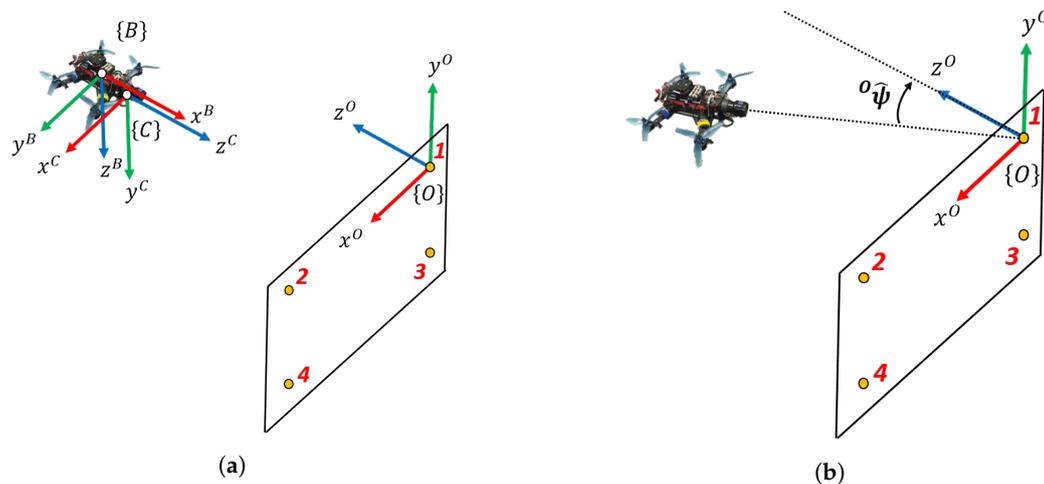


Figure 8. Coordinate systems and relative attitude. (a) Relationship between each coordinate system. (b) Relative yaw angle.

### 5.3. Position Hold Controller Design

We built our visual servoing controller based on the PX4 flight controller architecture [47]. Figure 9 shows the control diagram for the position hold flight. To achieve position hold control, first the current vehicle states  ${}^O\hat{x}$  and  ${}^O\hat{\psi}$  are estimated by a vision-based state estimator. Then, the position controller obtains the estimated relative position directly with a known relative attitude setpoint  ${}^Bq_{sp}$  and thrust setpoint  ${}^Bt_{sp}$ .

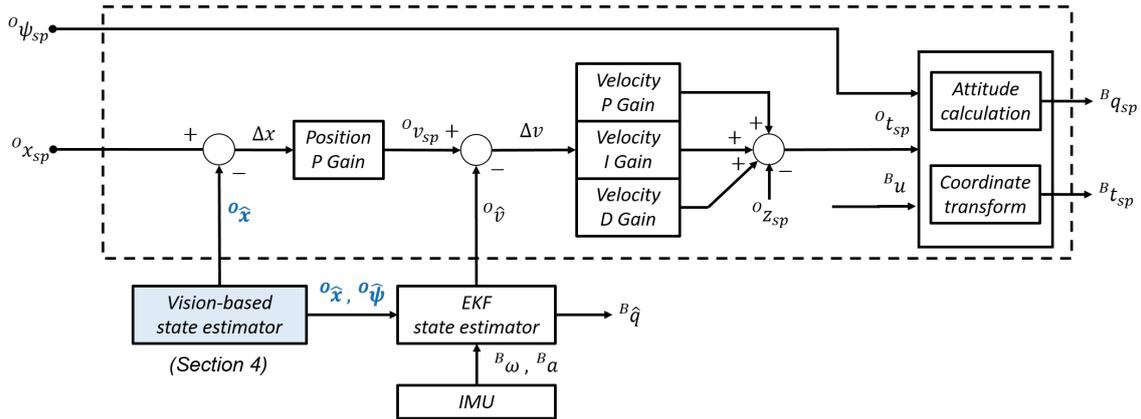


Figure 9. Design of direct vision-based position controller.

Equations for calculating the attitude setpoint  ${}^B q_{sp}$  and thrust setpoint  ${}^B t_{sp}$  are as follows: The velocity setpoint  ${}^O v_{sp}$  is calculated by

$${}^O v_{sp} = K_{px} \left( {}^O x_{sp} - {}^O \hat{x} \right), \tag{6}$$

where  $K_{px}$  is the position proportional parameter, and  ${}^O x_{sp}$  and  ${}^O \hat{x}$  represent the desired position and the estimated relative position. Then, the thrust setpoint  ${}^O t_{sp}$  is calculated by

$${}^O t_{sp} = K_{pv} \Delta \mathbf{v} + K_{dv} \frac{d}{dt} \Delta \mathbf{v} + {}^O z_{ini} - {}^O z_{sp}, \tag{7}$$

where the velocity error  $\Delta \mathbf{v} = {}^O v_{sp} - {}^O \hat{v}$ , and  $K_{pv}$  and  $K_{dv}$  are velocity P and D parameters in the PD controller,  ${}^O z_{sp}$  is the desired position in the z direction, and  ${}^O z_{ini}$  is the initial position in the z direction, which is calculated by

$${}^O z_{ini} = {}^O z_{ini} + \Delta \mathbf{v} K_{iv} dt, \tag{8}$$

where  $K_{iv}$  is the velocity integrative parameter. With a known thrust setpoint  ${}^O t_{sp}$  and yaw angle setpoint  ${}^O \psi_{sp}$ , we have the desired 3-axis unit vectors:

$$\mathfrak{B}_z = -{}^O \hat{t}_{sp}, \tag{9}$$

$$\mathfrak{B}_x = \mathfrak{B}_z \times \psi_{xy}, \tag{10}$$

$$\mathfrak{B}_y = \mathfrak{B}_z \times \mathfrak{B}_x, \tag{11}$$

where  $\mathfrak{B}_x$ ,  $\mathfrak{B}_y$ , and  $\mathfrak{B}_z$  are the 3-axis unit vectors of the UAV's body frame,  $\psi_{xy} = (-\sin \psi_{sp}, \cos \psi_{sp}, 0)$  is the vector of the desired yaw direction in the body frame's xy plane, and  ${}^O \hat{t}_{sp}$  is the normalized vector of  ${}^O t_{sp}$ . By stacking  $\mathfrak{B}_x$ ,  $\mathfrak{B}_y$ ,  $\mathfrak{B}_z$  together we obtain the desired rotation matrix  ${}^B R_{sp}$ :

$${}^B R_{sp} = \begin{bmatrix} \mathfrak{B}_x \\ \mathfrak{B}_y \\ \mathfrak{B}_z \end{bmatrix}. \tag{12}$$

By transforming  ${}^B R_{sp}$  to the quaternion form, we obtain attitude setpoints  ${}^B q_{sp}$ . With a known attitude setpoint  ${}^B q_{sp}$  and estimated attitude  ${}^B \hat{q}$  from the EKF state estimator, the angular rates setpoints  $\dot{\phi}_{sp}$ ,  $\dot{\theta}_{sp}$ , and  $\dot{\psi}_{sp}$  are calculated with tuned proportional parameters:

$$\dot{\phi}_{sp} = K_{p\phi} e_{\phi}, \tag{13}$$

$$\dot{\theta}_{sp} = K_{p\theta} e_{\theta}, \tag{14}$$

$$\dot{\psi}_{sp} = K_{p\psi} e_{\psi}, \tag{15}$$

where  $K_{pn}$ ,  $n \in \phi, \theta, \psi$  are tuned proportional parameters of roll, pitch, and yaw angle, respectively, and  $e_m = m_{sp} - \hat{m}$ ,  $m \in \phi, \theta, \psi$  are the angle errors between the desired attitude and the estimated attitude. In the rate controller, simple PID control is used, and its outputs and  ${}^B t_{sp}$  will be translated to the motor PWM signals in the mixer module. Figure 10 shows the flowchart of the relative-position-based visual servoing system.

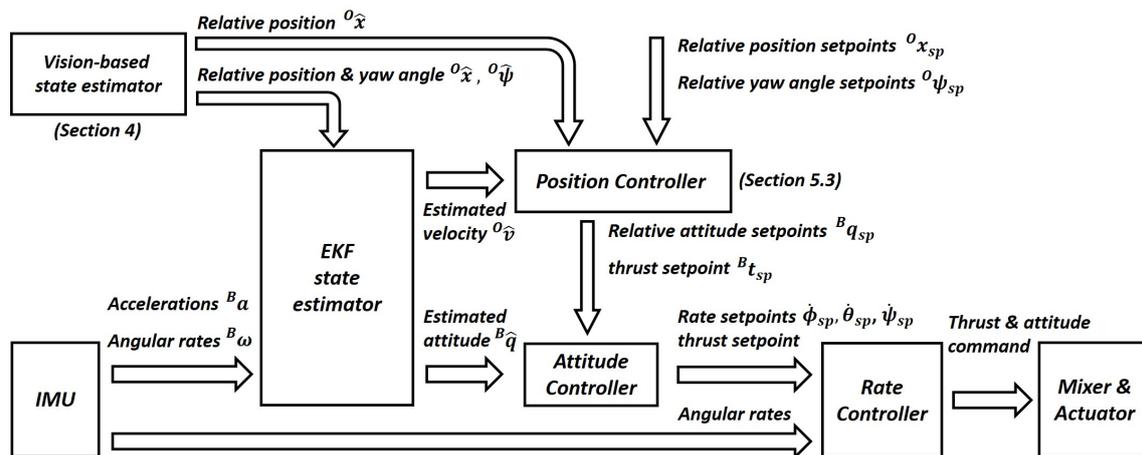


Figure 10. Flowchart of direct vision-based position controller.

## 6. Experimental Results

### 6.1. Method

In the experiment, first we made the quadrotor take off manually, and once the target was in view, the flight mode was automatically switched to the position hold mode. The quadrotor will fly to a setpoint relative to the target original point and keep its position. Since the relative position between the target and quadrotor is used for position-based visual servoing, the quadrotor will keep flying to a setpoint during the target's movement.

### 6.2. Position Hold

Before the experiment for dynamic target tracking, we tested the performance of direct visual servoing in a position holding experiment. The experimental configuration is shown in Figure 11. The setpoint position  ${}^o x$  was set to  $[0.6, 0.1, 0.1]^T$  [m], and the setpoint yaw angle  ${}^o \psi_{sp}$  was set to 0 [deg]. Note that they were set relative to the original point on the target.

Figure 12 shows the result of an autonomous position hold flight with a target recognition rate of about 350 Hz. The experimental results show that the positioning errors were [30.2, 63.2, 19.0] [mm] with standard deviations [0.64, 1.4, 0.43] [mm] in the x, y, and z directions, respectively. The angular tracking mean error was 2.3291° with a standard deviation 0.1689° in the yaw direction.

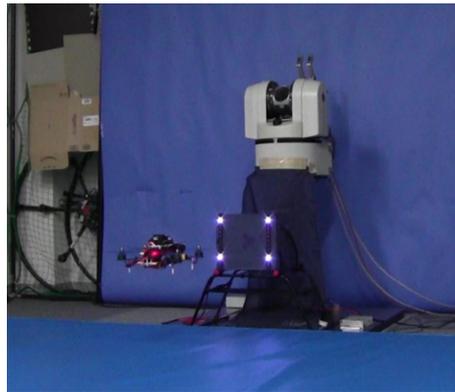


Figure 11. Configuration of position hold flight.

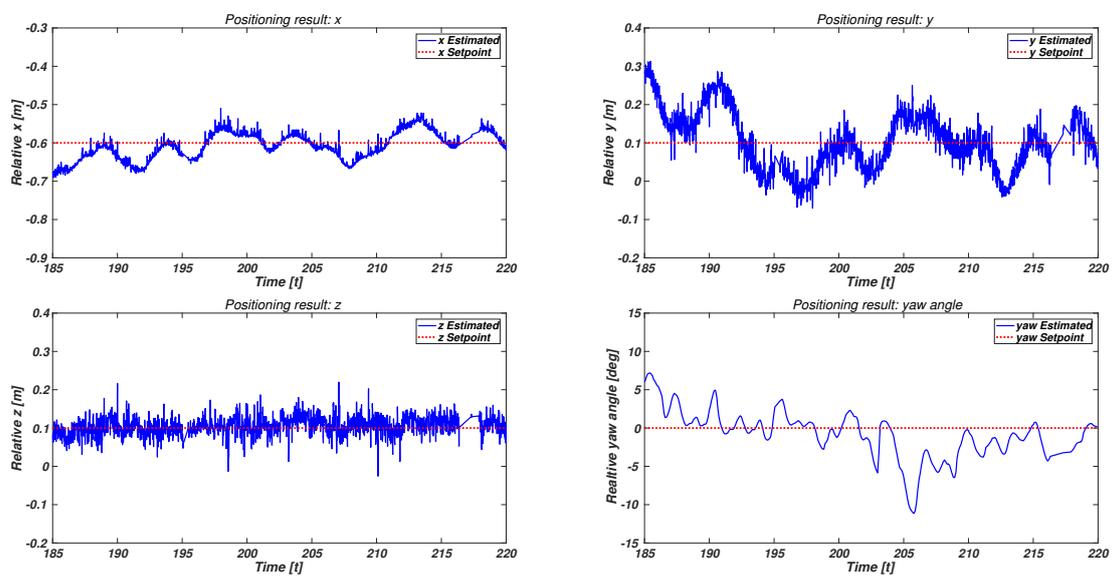


Figure 12. Positioning result of position hold flight.

### 6.3. Target Tracking

#### 6.3.1. Trajectory Simulation

For the dynamic target tracking, the trajectory of the target was set beforehand for the convenience of setting the moving speed of the target. In the experiment, the target was moved back and forward between two postures of the robot arm with different moving speeds to verify the tracking results in different conditions. Figure 13 shows the positions set for the tracking tasks.

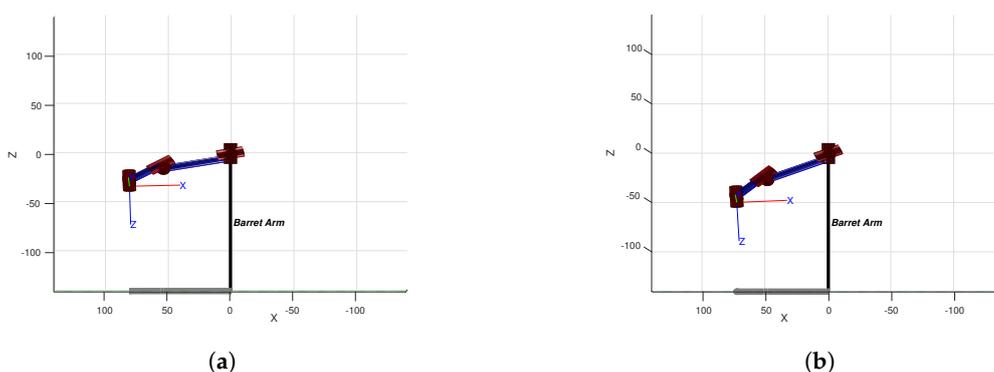


Figure 13. Simulation of target trajectory. (a) Target's 1st position. (b) Target's 2nd position.

### 6.3.2. Results

The tracking trajectories were transformed to the world coordinate system (Figure 14) located on the robot arm, as shown in Figure 15, with the target moving at a linear velocity of 0.5 [m/s] and a linear acceleration of 0.1 [m/s<sup>2</sup>]. The setpoint position  ${}^o x$  was set to [0.6, 0.1, 0.1]<sup>T</sup> [m], and the setpoint yaw angle  ${}^o \psi_{sp}$  was set to 0 [deg] relative to the original point on the target.

The experimental results showed that the positional tracking mean errors were [41.4, 51.2, 34.5] [mm] with standard deviations [1.2, 1.3, 1.0] [mm] in the  $x$ ,  $y$ , and  $z$  directions, respectively. The angular tracking mean error was 0.7006° with a standard deviation 0.0574° in the yaw direction. Since the relative setpoint was [0.6, 0.1, 0.1]<sup>T</sup> [m] in the  $x$ ,  $y$ , and  $z$  directions with respect to the object, we can see the position deviation between the UAV's trajectories and the robot end effector during the tracking.

To test the performance of our direct visual servoing system while tracking a high-speed moving target, we repeated the above experiment with higher acceleration. Figure 16 shows the tracking results while the target was moving backward and forward with a linear velocity of 0.5 [m/s] and a linear acceleration of 1.0 [m/s<sup>2</sup>].

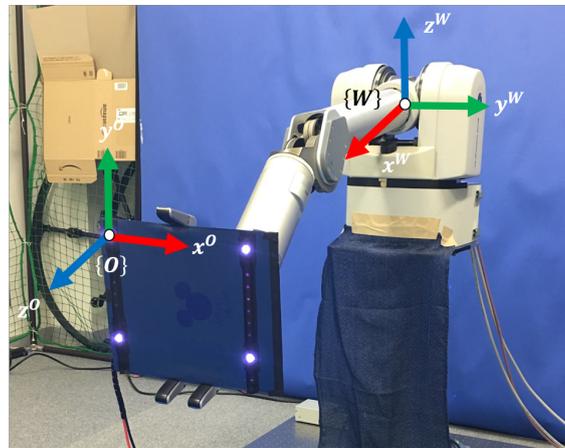


Figure 14. Relationship between object coordinate system and world coordinate system.

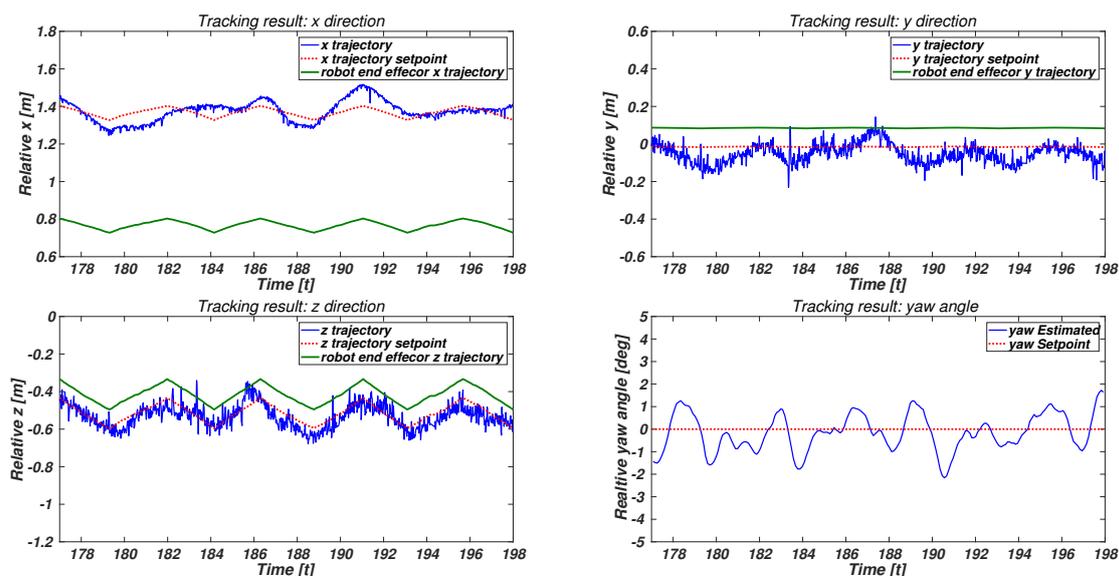
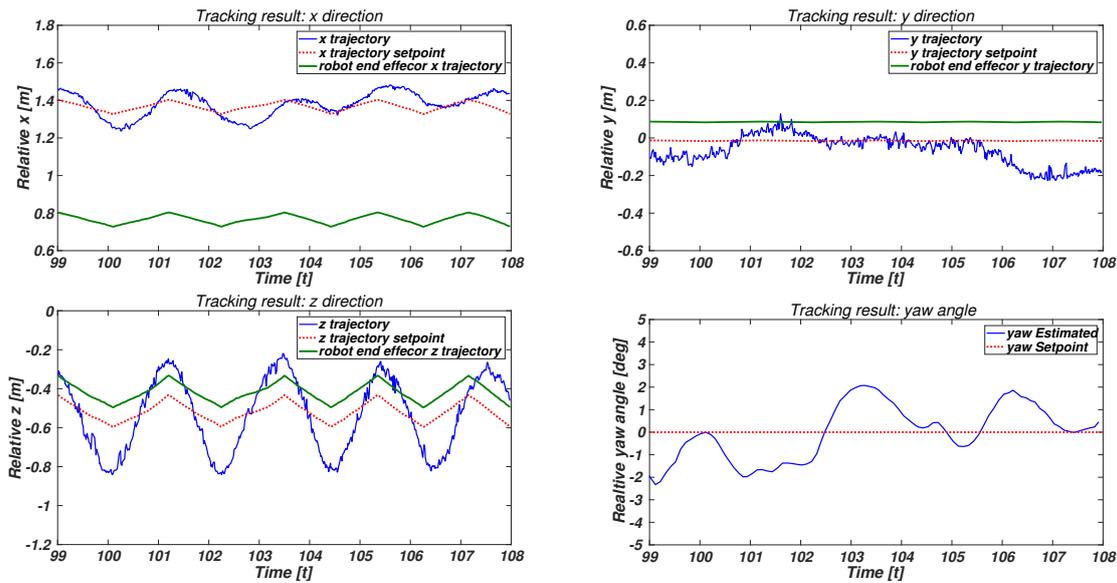


Figure 15. Trajectory of autonomous target tracking (Moving speed:  $v = 0.5$  [m/s],  $a = 0.1$  [m/s<sup>2</sup>], Target recognition rate: 350 Hz).



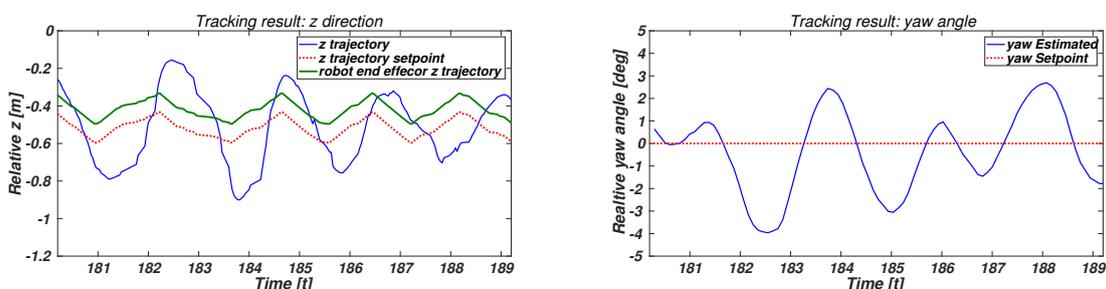
**Figure 16.** Trajectory of autonomous target tracking (Moving speed:  $v = 0.5$  [m/s],  $a = 1.0$  [m/s<sup>2</sup>], Target recognition rate: 350 Hz).

The experimental results showed that the positional tracking mean errors were [50.5, 80.1, 129.1] [mm] with standard deviations [1.8, 2.6, 4.9] [mm] in the  $x$ ,  $y$ , and  $z$  directions, respectively. The angular tracking mean error was  $1.0101^\circ$  with a standard deviation  $0.0019^\circ$  in the yaw direction. As shown in Figure 16, we found that there was an overshoot phenomenon while the target was moved with a higher linear acceleration. This may have resulted from an excessively high mass of inertia of our flying platform, which was originally used for a racing drone.

We also compared the tracking performance in the  $z$  direction and yaw angle estimation with different target recognition rates, as shown in Table 2. The experimental results showed that higher target recognition rates not only effectively improved the tracking performance but also allowed the accuracy of the attitude estimation to be evaluated. Figure 17 shows the tracking results with a 30 Hz target recognition rate. Compared with Figure 16, a tracking delay in the  $z$  direction was obvious due to the larger control latency resulting from insufficient position update rates.

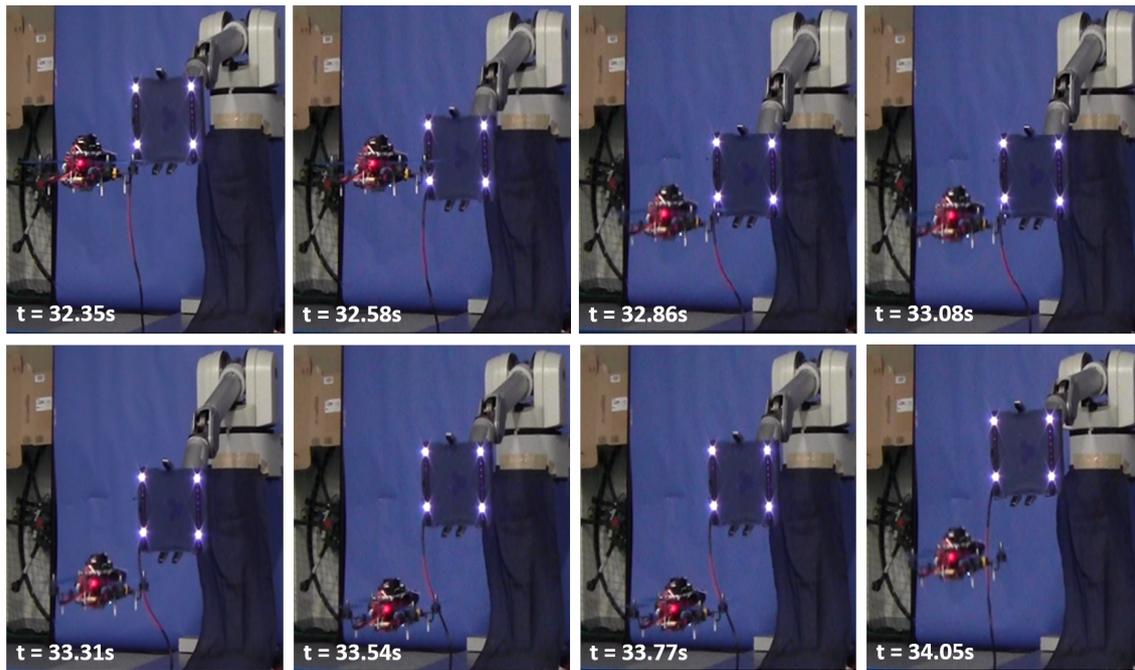
**Table 2.** Tracking performance comparison with different target recognition rates.

Target Recognition Rates	$z$ Mean Error	$z$ Mean Error std.	Yaw Mean Error	Yaw Mean Error std.
350 Hz	0.1382 m	0.0057 m	$1.0246^\circ$	$0.1255^\circ$
30 Hz	0.1634 m	0.0129 m	$1.4859^\circ$	$0.1939^\circ$



**Figure 17.** Trajectory of autonomous target tracking (Moving speed:  $v = 0.5$  [m/s],  $a = 1.0$  [m/s<sup>2</sup>], Target recognition rates: 30 Hz).

Figure 18 shows the experimental result of dynamic target tracking within 1.7 s.



**Figure 18.** Autonomous target tracking (Moving speed:  $v = 0.5$  [m/s],  $a = 1.0$  [m/s<sup>2</sup>]).

#### 6.4. Discussion

In the experiments, the feasibility of direct visual servoing of a UAV was demonstrated by a position hold task and two target tracking tasks with different moving speeds. We used the PnP algorithm to obtain the relative pose of the target, and we assumed that the 3D positions of the features on the target coordinate system were known. However, in actual applications, detection of feature points and estimation of their 3D positions are required at the moment of entering the field of view. This is usually difficult with a monocular camera alone. Nevertheless, tracking and avoidance are often possible even if the relative position is not exactly accurate. In this case, a rough estimate is sufficient.

The measurement errors  $\Delta p$  of the vision-based position estimator can be described by the relationship of propagation of uncertainty:

$$\Delta p = f(\Delta u_1, \Delta u_2, \Delta u_3, \Delta u_4), \tag{16}$$

where  $\Delta u_n$  ( $n \in 1, 2, 3, 4$ ) are errors of four selected features on the image. (16) can be expanded as

$$\Delta p \approx \left(\frac{\partial f}{\partial u_1}\right)\Delta u_1 + \left(\frac{\partial f}{\partial u_2}\right)\Delta u_2 + \left(\frac{\partial f}{\partial u_3}\right)\Delta u_3 + \left(\frac{\partial f}{\partial u_4}\right)\Delta u_4 \tag{17}$$

From (17), we can calculate the errors of the estimated relative position  $\Delta p$  by propagating errors from  $\Delta u_1, \Delta u_2, \Delta u_3, \Delta u_4$ . In the experiment, we obtained estimation errors  $\Delta p$  of relative positions of about  $[57.3, 125.9, 142.7]^T$  [mm] in the  $x, y,$  and  $z$  directions, respectively. In addition, we found that the velocity estimation results of the EKF state estimator diverged as the sample rates of the estimated relative position became higher than that of the IMU (250Hz) with such estimation errors. Therefore, though we were able to recognize a target at about 350 Hz, its position information was sent to a flight controller operating at about 120 Hz to ensure a safe flight. To improve the accuracy and decrease the computational load of the state estimator, we will adopt an unscented Kalman filter (UKF) for velocity estimation in the future.

## 7. Conclusions

In the presented work, we proposed a relative-position-based visual servoing system employing a high-speed target recognition method and a novel controller design. Compared to previous research on visual servoing of UAVs, our system used visual position information for controlling the UAV directly with high rates, and the feasibility of the proposed control scheme was verified in a position hold flight. The advantages of the proposed method were demonstrated through real-time target tracking experiments. The experimental results showed that our proposed method can achieve better tracking performance than a low-speed target recognition setting.

In future work, we will aim to embark on improving the performance of the proposed vision-based state estimator, including the accuracy and robustness to the environment. Then we will focus on developing a pure vision-based flight controller, which means that the visual information will be used for controlling the UAV directly. Furthermore, we will perform real-time feature extraction and feature tracking of any actual targets. By extracting features in the surrounding environment directly, we aim to realize high-speed manipulation of UAVs by direct visual servoing, thereby attaining various tasks, like high-speed dynamic avoidance and high-speed target catching in any environment.

**Author Contributions:** Conceptualization, H.-M.C. and A.N.; methodology, H.-M.C. and A.N.; software, H.-M.C.; validation, H.-M.C. and D.H.; formal analysis, H.-M.C.; writing—original draft preparation, H.-M.C.; writing—review and editing, A.N.

**Funding:** This research received no external funding.

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

## References

1. Srinivasan, M.V.; Zhang, S.W.; Lehrer, M.; Collett, T.S. Honeybee navigation en route to the goal: Visual flight control and odometry. *J. Exp. Biol.* **1996**, *199*, 237–244. [[PubMed](#)]
2. Zhao, J.; Ma, X.; Fu, Q.; Hu, C.; Yue, S. An LGMD Based Competitive Collision Avoidance Strategy for UAV. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*; Springer: Cham, Switzerland, 2019; pp. 80–91.
3. Padhy, R.P.; Xia, F.; Choudhury, S.K.; Sa, P.K.; Bakshi, S. Monocular vision aided autonomous UAV navigation in indoor corridor environments. *IEEE Trans. Sustain. Comput.* **2018**, *4*, 96–108. [[CrossRef](#)]
4. Dissanayake, G.; Newman, P.; Durrant-Whyte, H.F.; Clark, S.; Csobza, M. A solution to the simultaneous localisation and mapping (SLAM) problem. *IEEE Trans. Robot. Autom.* **2001**, *17*, 229–241. [[CrossRef](#)]
5. Dowling, L.; Poblete, T.; Hook, I.; Tang, H.; Tan, Y.; Glenn, W.; Unnithan, R.R. Accurate indoor mapping using an autonomous unmanned aerial vehicle (UAV). *arXiv* **2018**, arXiv:1808.01940.
6. Fraundorfer, F.; Scaramuzza, D. Visual odometry: Part I: The first 30 years and fundamentals. *IEEE Robot. Autom. Mag.* **2011**, *18*, 80–92.
7. Kendoul, F.; Fantoni, I.; Nonami, K. Optic flow-based vision system for autonomous 3D localization and control of small aerial vehicles. *Robot. Autom. Syst.* **2009**, *57*, 591–602. [[CrossRef](#)]
8. Conroy, J.; Gremillion, G.; Ranganathan, B.; Humbert, J.S. Implementation of wide-field integration of optic flow for autonomous quadrotor navigation. *Auton. Robot.* **2009**, *27*, 189–198. [[CrossRef](#)]
9. Kim, J.; Kang, M.; Park, S. Accurate Modeling and Robust Hovering Control for a Quad-rotor VTOL Aircraft. *J. Intell. Robot. Syst.* **2010**, *57*, 9–26. [[CrossRef](#)]
10. Guanglei, M.; Haibing, P. The application of ultrasonic sensor in the obstacle avoidance of quad-rotor UAV. In *Proceedings of the 2016 IEEE Chinese Guidance, Navigation and Control Conference (CGNCC)*, Nanjing, China, 12–14 August 2016; pp. 976–981.
11. Suzuki, S. Integrated Navigation for Autonomous Drone in GPS and GPS-Denied Environments. *J. Robot. Mechatron.* **2018**, *30*, 373–379. [[CrossRef](#)]
12. Kumar, G.A.; Patil, A.; Patil, R.; Park, S.; Chai, Y. A LiDAR and IMU integrated indoor navigation system for UAVs and its application in real-time pipeline classification. *Sensors* **2017**, *17*, 1268. [[CrossRef](#)]

13. Ishikawa, M.; Namiki, A.; Senoo, T.; Yamakawa, Y. Ultra highspeed robot based on 1 kHz vision system. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura, Portugal, 7–12 October 2012; pp. 5460–5461.
14. Suzuki, S. Recent researches on innovative drone technologies in robotics field. *Adv. Robot.* **2018**, *32*, 1008–1022. [[CrossRef](#)]
15. Kanellakis, C.; Nikolakopoulos, G. Survey on computer vision for UAVs: Current developments and trends. *J. Intell. Robot. Syst.* **2017**, *87*, 141–168. [[CrossRef](#)]
16. Watanabe, K.; Iwatani, Y.; Nonaka, K.; Hashimoto, K. A visual-servo-based assistant system for unmanned helicopter control. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, France, 22–26 September 2008; pp. 822–827.
17. Yu, Y.; Lont, C.; Weiwei, Z. Stereo vision based obstacle avoidance strategy for quadcopter UAV. In Proceedings of the 30th Chinese Control and Decision Conference, Shenyang, China, 9–11 June 2018; pp. 490–494.
18. Guenard, N.; Hamel, T.; Mahony, R. A practical visual servo control for an unmanned aerial vehicle. *IEEE Trans. Robot.* **2008**, *24*, 331–340. [[CrossRef](#)]
19. Teuliere, C.; Eck, L.; Marchand, E. Chasing a moving target from a flying UAV. In Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, CA, USA, 25–30 September 2011; pp. 4929–4934.
20. Eberli, D.; Scaramuzza, D.; Weiss, S.; Siegwart, R. Vision based position control for MAVs using one single circular landmark. *J. Intell. Robot. Syst.* **2011**, *61*, 495–512. [[CrossRef](#)]
21. Jung, S.; Cho, S.; Lee, D.; Lee, H.; Shim, D.H. A direct visual servoing-based framework for the 2016 IROS autonomous drone racing challenge. *J. Field Robot.* **2017**, *35*, 146–166. [[CrossRef](#)]
22. Lee, D.; Ryan, T.; Kim, H.J. Autonomous landing of a VTOL UAV on a moving platform using image-based visual servoing. In Proceedings of the IEEE International Conference on Robotics and Automation, Saint Paul, MN, USA, 14–18 May 2012; pp. 971–976.
23. Jabbari, H.; Oriolo, G.; Bolandi, H. An adaptive scheme for image-based visual servoing of an underactuated UAV. *Int. J. Robot. Autom.* **2014**, *29*, 92–104.
24. Araar, O.; Aouf, N. Visual servoing of a quadrotor uav for autonomous power lines inspection. In Proceedings of the 22nd Mediterranean Conference on Control and Automation, Palermo, Italy, 16–19 June 2014; pp. 1418–1424.
25. Falanga, D.; Zanchettin, A.; Simovic, A.; Delmerico, J.; Scaramuzza, D. Vision-based Autonomous Quadrotor Landing on a Moving Platform. In Proceedings of the International Symposium on Safety, Security and Rescue Robotics, Shanghai, China, 11–13 October 2017; pp. 11–13.
26. Thomas, J.; Loianno, G.; Sreenath, K.; Kumar, V. Toward image based visual servoing for aerial grasping and perching. In Proceedings of the IEEE International Conference on Robotics and Automation, Hong Kong, China, 31 May–7 June 2014; pp. 2113–2118.
27. Thomas, J.; Loianno, G.; Daniilidis, K.; Kumar, V. Visual Servoing of Quadrotors for Perching by Hanging From Cylindrical Objects. *IEEE Robot. Autom. Lett.* **2016**, *1*, 57–64. [[CrossRef](#)]
28. Zhang, X.; Fang, Y.; Zhang, X.; Jiang, J.; Chen, X. A Novel Geometric Hierarchical Approach for Dynamic Visual Servoing of Quadrotors. *IEEE Trans. Ind. Electron.* **2019**. [[CrossRef](#)]
29. Zheng, D.; Wang, H.; Wang, J.; Zhang, X.; Chen, W. Towards visibility guaranteed visual servoing control of quadrotor UAVs. *IEEE/ASME Trans. Mechatron.* **2019**, *24*, 1087–1095. [[CrossRef](#)]
30. Li, S.; van der Horst, E.; Duernay, P.; De Wagter, C.; de Croon, G.C.H.E. Visual model-predictive localization for computationally efficient autonomous racing of a 72-gram drone. *arXiv* **2019**, arXiv:1905.10110.
31. Li, S.; Ozoa, M.M.O.I.; De Wagtera, C.; de Croona, G.C.H.E. Autonomous drone race: A computationally efficient vision-based navigation and control strategy. *arXiv* **2018**, arXiv:1809.05958.
32. Zhao, W.; Liu, H.; Lewis, F.L.; Valavanis, K.P.; Wang, X. Robust Visual Servoing Control for Ground Target Tracking of Quadrotors. *IEEE Trans. Control. Syst. Technol.* **2019**. [[CrossRef](#)]
33. Namiki, A.; Matsushita, S.; Ozeki, T.; Nonami, K. Hierarchical processing architecture for an air-hockey robot system. In Proceedings of the 2013 IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 6–10 May 2013; pp. 1187–1192.

34. Kizaki, T.; Namiki, A. Two Ball Juggling with High-Speed Hand-Arm and High-Speed Vision System. In Proceedings of the IEEE International Conference on Robotics and Automation, Saint Paul, MN, USA, 14–18 May 2012; pp. 1372–1377.
35. Yamakawa, Y.; Namiki, A.; Ishikawa, M. Dynamic High-speed Knotting of a Rope by a Manipulator. *Int. J. Adv. Robot. Syst.* **2013**, *10*. [[CrossRef](#)]
36. Namiki, A.; Ito, N. Ball Catching in Kendama Game by Estimating Grasp Conditions Based on a High-Speed Vision System and Tactile Sensors. In Proceedings of the IEEE-RAS International Conference on Humanoid Robots, Madrid, Spain, 18–20 November 2014; pp. 634–639.
37. Hu, S.; Matsumoto, Y.; Takaki, T.; Ishii, I. Monocular stereo measurement using high-speed catadioptric tracking. *Sensors* **2017**, *17*, 1839.
38. Chuang, H.M.; Wojtara, T.; Bergstrom, N.; Namiki, A. Velocity Estimation for UAVs by Using High-Speed Vision. *J. Robot. Mechatron.* **2018**, *30*, 363–372. [[CrossRef](#)]
39. *VisionWorks Programming Tutorial*; NVIDIA: Santa Clara, CA, USA, 2016.
40. Available online: <https://github.com/BarrettTechnology/libbarrett> (accessed on 25 February 2016).
41. Harris, C.; Stephens, M. A combined corner and edge detector. In Proceedings of the Alvey Vision Conference, Manchester, UK, 31 August–2 September 1988; pp. 147–151.
42. Bay, H.; Tuytelaars, T.; Van Gool, L. Surf: Speeded Up Robust Features. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 404–417.
43. Rosten, E.; Porter, R.; Drummond, T. Faster and better: A machine learning approach to corner detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *32*, 105–119. [[CrossRef](#)]
44. Flusser, J. Moment invariants in image analysis. *Proc. World Acad. Sci. Eng. Technol.* **2006**, *11*, 196–201.
45. Lepetit, V.; Moreno-Noguer, F.; Fua, P. EPnP: An Accurate O(n) Solution to the PnP Problem. *Int. J. Comput. Vis.* **2009**, *81*, 155–166. [[CrossRef](#)]
46. Moré, J.J. The Levenberg-Marquardt algorithm: Implementation and theory. In *Numerical analysis*; Springer: Berlin/Heidelberg, Germany, 1978; pp. 105–116.
47. Meier, L.; Honegger, D.; Pollefeys, M. PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms. In Proceedings of the 2015 IEEE international conference on robotics and automation (ICRA), Seattle, WA, USA, 26–30 May 2015, pp. 6235–6240.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).