Proceedings of the 38th Chinese Control Conference
July 27-30, 2019, Guangzhou, China

A New Image-Based Visual Servo Control Algorithm for Target Tracking Problem of Fixed-Wing Unmanned Aerial Vehicle

Lingjie Yang[1], Zhihong Liu[1], Xiangke Wang[1]
1. College of Intelligence Science and Technology, National University of Defense Technology, Changsha 410073, P. R. China
E-mail: zhliu@nudt.edu.cn

Abstract: This paper presents a new image-based visual servo (IBVS) control algorithm for the tracking of a static target by a fixed-wing unmanned aerial vehicle (UAV). The sensor used is a monocular camera fixed on the drone. The centroid coordinates of the target are extracted to be the inputs of the algorithm, while the control output is obtained to adjust the heading direction of the drone by the image Jacobian model. Besides, the problem that the shifts of the centroid caused by the changes of the UAV's attitude during tracking can be solved by a compensation mechanism. Through experiments by the hardware-in-the-loop (HIL) simulation, we evaluate the proposed algorithm. The results show that the proposed algorithm can effectively track the static target by a fixed-wing UAV.

Key Words: IBVS, fixed-wing UAV, fixed camera, HIL simulation

1 Introduction

The past ten years have seen a growing applications for UAVs in both civilian fields and military fields [1–4]. Considering the fact that UAVs are better suited for dull, dirty, and dangerous missions than manned aerial vehicles, they have been widely used for intelligence, survelliance, reconnaissance, and ground target tracking.

The studies about tracking of the targets for UAVs have been widely carried out. According to the types of UAVs, the fixed-wing UAV, rotor UAV and helicopter UAV can be used to achieve the tracking of ground targets [5–7]. In terms of tracking strategies, there exist model predictive control (MPC) [8], adaptive control [9], synthesis control [10] and so on. Towards the difference of the state of the target, [11] used a pairs of UAVs to keep a certain distance from the moving target, while [12] used quadrotor to track the target that is static by model-based method to locate itself. Considering the different ways of mounting camera, [13] used UAVs with gimbaled cameras to achieve the tracking of target, and [14] implemented it with fixed cameras.

Visual servo control can be used to solve the problem of tracking for UAV with camera. According to the differences of feedback information, it can be divided into position visual servo (PBVS), IBVS and 2.5D visual servo [15]. For PBVS, the camera coordinate system needs to be transformed with 3D Cartesian coordinate system. However, it separates visual reconstruction problems from control. PBVS is mainly used to estimate the relative pose from the target to the camera [16–18]. While IBVS directly deals with the feature points on images, the errors caused by the sensor model and calibration are small, and the positioning accuracy is high. More specifically, Jabbari et al. [19] used IBVS method for controlling the 3D translational motion and the yaw rotation of a quadrotor, Xie et al. [20] proposed a dynamic IBVS control to make a rotary wing UAV follow parallel lines. For 2.5D visual servo, it decouples rotation and translation that allows translation control to be done on a 2D image while the attitude control to be implemented with 3D information. In particular, Lippiello et al. [21] presented a hybrid image- and position-based visual servoing for the control of unmanned aerial manipulator. Considering the advantages of the fixed-wing UAVs in the mission execution, Peliti et al. [22, 23] studied the tracking of ground target for a fixed-wing UAV with a gimbaled camera based on IBVS. Mills et al. [24] presented an IBVS control design for fixed-wing UAVs tracking locally linear infrastructure with a fixed camera. Actually, the target tracking problem for the drone with a fixed camera is more challenging than that with a gimbaled camera, since the optic axis of the camera can not continuously point to the target without the gimbal.

In this paper, a new IBVS control algorithm is presented for a fixed-wing UAV equipped with a fixed camera to track a static target. From the RGB image, the target is detected and a feature point is extracted from it as the input. Since it is fixed and the changes of the drone's attitude will affect that of the camera, the camera's optic axis can not continuously point to the target. Thus, a compensation mechanism is proposed, which enables the determination for the position of desired feature point. After that, the image Jacobian matrix is adopted to build the connection of the velocity between the feature point and the UAV, and it is not necessary to calculate the pseudo-inverse of the matrix. Finally, the algorithm is evaluated by a HIL simulation, which is composed of Gazebo, off the shelf autopilot and the ground station. The results show that the fixed-wing with a fixed camera can effectively track the static target.

The paper is organized as follows. Section II presents the problem of tracking a static target for the fixed-wing UAV. Section III proposes a compensation mechanism for the circumstance that camera is fixed and design a control law. Section IV describes the setup of HIL simulation and analyzes the tracking effect by plotting data. Section V is a summary of our work and discussion of future work.

2 Problem Statement

This paper aims to use a new IBVS control algorithm for the fixed-wing UAV to track a static target, and the object of our research is a fixed-wing UAV with a fixed camera. During the flight of tracking, the fixed-wing UAV flies at uniform speed relative to the ground. We can divide the problem into three parts:

(1) Obtain the velocity of feature point ($\dot{S}_c$)

We are able to get the images of ground scene with camera

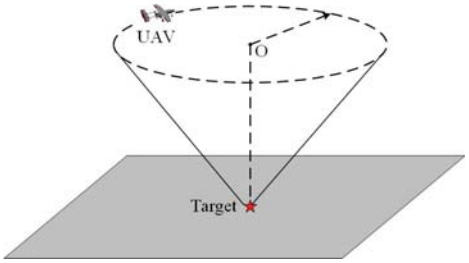

Fig. 1: The drone is tracking the target and circling around it finally.

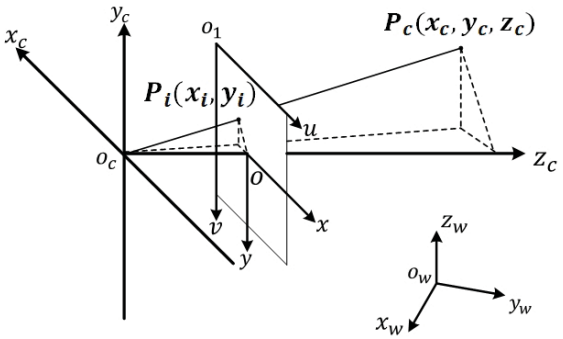

Fig. 2: The setup of four types of coordinate systems.

during the flight. Since the difference between foreground and background are remarkable in the images, we can use OSTU method to distinguish them. The method divides an image into two parts according to the grayscale characteristics of it, and the foreground is target. At the same time, the greater the variance between foreground and background is, the greater the difference between them is. However, once the target is mistakenly divided into background part, the variance would decrease. Therefore, the goal of OSTU method is to find an optimal segmentation making the variance between background and target maximal.

For the obtained image $I(m,n)$, we denote the pixel segmentation threshold of foreground and background by T_h, and the number of pixel belonging to the foreground is denoted by N_f, the other is denoted by N_b. Next we compute the proportion of pixels in the foreground ω_f and its average pixel value μ_f, and the same to background is ω_b and μ_b. Finally we get the average pixel value μ_t of the whole image $I(m,n)$, and the variance V_a between two parts can be expressed as follows:

$$\mu_t = \omega_f \cdot \mu_f + \omega_b \cdot \mu_b, \tag{1}$$

$$V_a = \omega_f(\mu_f - \mu_t)^2 + \omega_b(\mu_b - \mu_t)^2, \tag{2}$$

where

$$\omega_f = \frac{N_f}{m \cdot n},$$

$$\omega_b = \frac{N_b}{m \cdot n}.$$

When the value of V_a is maximal, the corresponding segmentation threshold T_h is optimal, then the foreground and the background can be represented by binarized image, thereby obtaining the target area.

After detecting the target in the image, we can get the coordinates of the feature point S_c, and $\dot{S}_c$ can be obtained by comparing it with the coordinates of desired feature point.

(2) Transform the velocity in different coordinate systems

Fig. 2 shows four coordinate systems, including the world coordinate system $\mathbb{F}_w$, camera coordinate system $\mathbb{F}_c$, image coordinate system $\mathbb{F}_i$ and pixel coordinate system $\mathbb{F}_p$. It's necessary to transfrom the velocity of the drone among them.

(i) $\mathbb{F}_w$ (o_w-$x_w y_w z_w$)

It's used to represent the position of the camera in 3-dimensional space, thereby reflecting its relative position to the target. The establishment of it satisfies the principle of convenience.

(ii) $\mathbb{F}_c$ (o_c-$x_c y_c z_c$)

It takes the camera's optical center as the origin, the $x_c o_c y_c$ plane is parallel to the imaging plane, and the $\vec{z}$ is along the camera's optic axis. There exist rotation transformation and translation transformation from $\mathbb{F}_w$ to $\mathbb{F}_c$.

(iii) $\mathbb{F}_i$ (xoy)

It takes symmetry point of the camera's focus relative to the optical center as the origin, plane xoy is the symmetry of the imaging plane. The coordinates of points expressed by $\mathbb{F}_c$ satisfy the triangle similarity with those expressed by $\mathbb{F}_i$.

$$\frac{x_i}{x_c} = \frac{y_i}{y_c} = \frac{-f}{z_c} \tag{3}$$

As shown in Fig. 2, f is $|o_c o|$ and represents focal length, (x_i, y_i) and (x_c, y_c, z_c) represent the coordinates of target in $\mathbb{F}_i$ and $\mathbb{F}_c$, respectively.

(iv) $\mathbb{F}_p$ (uo_1v)

It's coplanar with $\mathbb{F}_i$, origin o_1 is located in the top left corner of the image. Besides, its axes are parallel to those in $\mathbb{F}_i$, and directions are the same as well. The correspondences among them are shown as follows:

$$\begin{cases} u = u_0 + \frac{x_i}{du} \\ v = v_0 + \frac{y_i}{dv} \end{cases}, \tag{4}$$

where u_0 and v_0 represent the coordinates of origin (o) in $\mathbb{F}_p$, du and dv represent the width and height of unit pixel respectively.

(3) Construct the image Jacobian model

The image Jacobian matrix is used to set up the connection between $\dot{S}_c$ and the velocity of the drone V, which includes the linear velocity T and the angular velocity Ω. The relationship is shown as follows:

$$\dot{S}_c = J_v \cdot V, \tag{5}$$

in which J_v is the image Jacobian matrix, and $J_v \in R^{2\times 6}$.

After acquiring the motion velocity V of the target in $\mathbb{F}_c$ by the velocity $\dot{S}_c$ of the feature point, it needs to be converted into corresponding motion velocity of the drone. At this time the transformation relationships between $\mathbb{F}_c$ and $\mathbb{F}_b$ (body coordinate system) needs to be taken into account, which are the rotation matrix of different attitude angles. $\mathbb{F}_b$ of fixed-wing UAV is shown in Fig. 3, $\vec{x_b}$ points to the head of the drone, $\vec{z_b}$ is perpendicular to the drone's body and points to the ground, $\vec{y_b}$ is perpendicular to the plane $x_b o_b z_b$ and points to the right wing.

Rotations around axes in Fig. 3 $(\dot{\psi}, \dot{\theta}, \dot{\phi})$ correspond to three different rotation matrices respectively. With them,

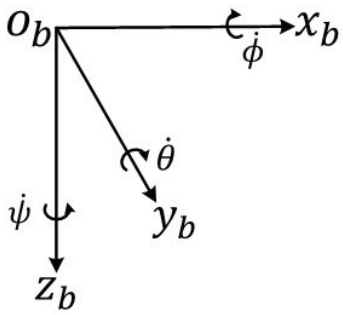

Fig. 3: Body coordinate system and rotation clockwise around the axes respectively.

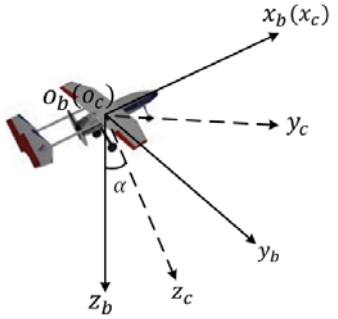

Fig. 4: Relationship between $\mathbb{F}_c$ and $\mathbb{F}_b$.

the corresponding linear velocity and angular velocity of the drone in different coordinate systems can be obtained from the current coordinate system.

However, there mainly exist two challenges when using the fixed-wing UAV with a fixed camera to track ground target.

(1) Since the camera is fixed on the drone, the optic axis will shift when the attitude of the drone changes, which causes that the camera can not always point to the target.

(2) It's necessary to invert the image Jacobian matrix for purpose of obtaining the motion velocity V through image-based visual servo control. Nevertheless, it is underactuated so that we need to obtain the pseudo-inverse of it, which may produce singular value and the process is time-consuming.

3 Control Approach

We present a new IBVS control algorithm to solve the problem of target tracking for fixed-wing UAV with a fixed camera. In our work , we use RGB image as the single input of the system, and propose a compensation mechanism for the shifts of the feature point. After that, we design a control law by adopting the image Jacobian matrix to obtain desired velocity of the drone . Finally with the function of coordinate transformation, the yaw control can be obtained.

3.1 Compensation Mechanism

We choose the feature point $s(u,v)$ in $\mathbb{F}_p$, and denote desired feature point by s^* (u^*,v^*), then the velocity of the feature point is

$$\dot{s} = \frac{s^* - s}{T_s}, \tag{6}$$

where T_s represents the sampling period.

Fig. 4 shows the setup of $\mathbb{F}_b$ (body coordinate system) and $\mathbb{F}_c$, where α represents the angle between the camera's optic axis and the vertical direction. When the drone is circling around the target, the desired position of the target is expected to be the intersection B of the optic axis and ground. Thus we have

$$tan\alpha = \frac{|AB|}{|o_cA|}. \tag{7}$$

And it needs a rotation matrix R_α to implement parallelism between $\mathbb{F}_b$ and $\mathbb{F}_c$.

$$R_\alpha = \begin{pmatrix} 1 & 0 & 0 \\ 0 & cos\alpha & sin\alpha \\ 0 & -sin\alpha & cos\alpha \end{pmatrix} \tag{8}$$

Since the camera's optic axis will shift with the changes of the drone's attitude, we need to compensate the effect that it brings and determine the coordinates of feature point after compensation. We mount the camera in the same position as the centroid of the drone, and consider the effects of three attitude angles in turn.

(1) Roll angle (ϕ_1)

It's necessary to rotate α angle around $\vec{x}$ of $\mathbb{F}_c$ in state S_0 to make it transformed into $\mathbb{F}_c$ in state S_1. We denote the roll angle by ϕ_1, then corresponding transformation matrix is

$$R_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & cos\phi_1 & -sin\phi_1 \\ 0 & sin\phi_1 & cos\phi_1 \end{pmatrix}. \tag{9}$$

(2) Pitch angle (θ_1)

After the roll angle changes to zero, we need to compensate the effect caused by the pitch angle θ_1 to make it zero as well (state S_2). It's unnecessary to do translation transformation as the camera is fixed on the centroid of the drone, thus we need a rotation transformation matrix R_2 from state S_1 to state S_2:

$$R_2 = \begin{pmatrix} cos\theta_1 & 0 & sin\theta_1 \\ 0 & 1 & 0 \\ -sin\theta_1 & 0 & cos\theta_1 \end{pmatrix}. \tag{10}$$

(3) Yaw angle (ψ_1)

The yaw angle determines the flying direction of the drone, and a rotation matrix R_3 is needed if we want the yaw angle of the drone to be zero (state S_3) as well.

$$R_3 = \begin{pmatrix} cos\psi_1 & -sin\psi_1 & 0 \\ sin\psi_1 & cos\psi_1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{11}$$

Actually, ψ_1 is the control variable, the flying direction of the drone needs to be adjusted to make it circle around the static target at last, thus it is unnecessary to compensate the effect of ψ_1. However, ϕ_1 and θ_1 are the disturbance variables for the determination of the feature point, since the center of the image is hoped to be the desired position of the target when the drone is circling around the target. Therefore, it is necessary to compensate the effect caused by ϕ_1 and θ_1 to obtain the desired feature point.

We denote by $R^{(i-j,\cdot)}$ from the i-th row to j-th row of matrix R, by $R^{(\cdot,i-j)}$ from the i-th column to j-th column of matrix R. By means of compensating ψ_1 and θ_1, the corresponding coordinates of the feature point in state S_2 can be obtained.

According to the pinhole imaging principle, the map of the target on the image is the intersection of the imaging plane and the line passing through the target and the optical center. We select a random point $P_c(X_P, Y_P, Z_P)$ from the line, and denote the feature point of the target on the image by $P_i(x_P, y_P)$, then there exist the following relationships:

$$\frac{X_P}{x_P} = \frac{Y_P}{y_P} = \frac{Z_P}{-f} = k, \tag{12}$$

where k is a non-positive constant, x_P, y_P are the coordinates of the feature point in $\mathbb{F}_i$ of state S_0, then the coordinates of P can be represented by $(kx_P, ky_P, -kf)$.

In order to obtain the coordinates of the feature point in $\mathbb{F}_i$ of state S_2, the transformation process is divided into four parts: (1) from $\mathbb{F}_c$ of state S_0 to $\mathbb{F}_b$ of state S_0; (2) from $\mathbb{F}_b$ of state S_0 to $\mathbb{F}_b$ of state S_1; (3) from $\mathbb{F}_b$ of state S_1 to $\mathbb{F}_b$ of state S_2; (4) from $\mathbb{F}_b$ of state S_2 to $\mathbb{F}_i$ of state S_2. The corresponding transformation matrices are R_α, R_1, R_2, R_α^{-1} respectively, where R_α^{-1} represents the inverse of R_α. Then we have

$$\begin{pmatrix} X'_P \\ Y'_P \\ Z'_P \end{pmatrix} = R \begin{pmatrix} X_P \\ Y_P \\ Z_P \end{pmatrix}, \tag{13}$$

in which

$$R = R_\alpha^{-1} R_2 R_1 R_\alpha.$$

When the value of Z'_P in Eq. (13) is $-f$, the value of k can be calculated, and the (X'_P, Y'_P) at this time is the corresponding coordinates of the target (x'_P, y'_P)in $\mathbb{F}_i$ of state S_2, it is

$$\begin{pmatrix} x'_P \\ y'_P \end{pmatrix} = k(R)^{(1-2,\cdot)} \begin{pmatrix} x_P \\ y_P \\ -f \end{pmatrix}.$$

3.2 Control Law Design

We denote the error of current feature point and desired feature point by $e = (x'_P, y'_P)$ after obtaining the coordinates of the feature point (x'_P, y'_P). And we adopt the exponential convergence $\dot{e} = -\lambda e$, where λ is a positive definite matrix of 2×2.

Eq. (5) illustrates the relationship between the velocity of the feature point and the motion velocity of the target in $\mathbb{F}_c$. Since the objective is to implement the control of fixed-wing UAV, it is necessary to transform the velocity of the drone in $\mathbb{F}_c$ to that in $\mathbb{F}_b$, and get the control at last. As Fig. 5 shows, we denote by $\mathbb{F}_{b_2}$ the body coordinate system in state S_2, by ${}^e\Omega = (\dot{\phi} \quad \dot{\theta} \quad \dot{\psi})^T$ the angular velocity of the Euler angles, and by ${}^{b_2}\Omega = (\omega_x, \omega_y, \omega_z)$ the angular velocity of the drone in $\mathbb{F}_{b_2}$. The velocity of the drone in $\mathbb{F}_c$ of state S_2 needs to be transformed to $\mathbb{F}_{b_2}$, then the latter is transformed to ${}^e\Omega$.

(1) From $\mathbb{F}_c$ of state S_2 to $\mathbb{F}_{b_2}$

There exists a rotation around $\vec{x}$ of $\mathbb{F}_c$ to make it coincide with $\mathbb{F}_{b_2}$, and R_α plays the role.

(2) From ${}^{b_2}\Omega$ to ${}^e\Omega$

The transformation is actually the projection of ${}^{b_2}\Omega$ on the axes that ${}^e\Omega$ corresponds to, and the relationship between them is

$${}^e\Omega = \begin{pmatrix} cos\theta & 0 & sin\theta \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} {}^{b_2}\Omega = R_e {}^{b_2}\Omega. \tag{14}$$

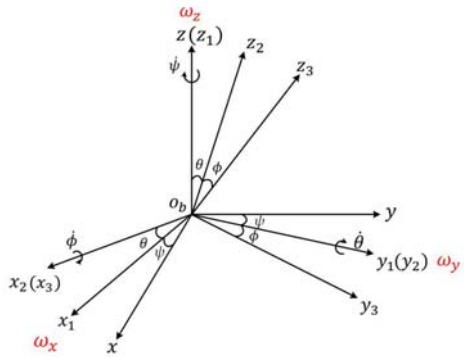

Fig. 5: Transformation among different angular velocity. ω_x,ω_y and ω_z represent angular velocity of body coordinate system; $\dot{\phi}$, $\dot{\theta}$ and $\dot{\psi}$ represent angular velocity of Euler angles.

Both the pitch angle and roll angle of the drone in state S_2 are always zero, thereby ${}^{b_2}\omega_y$ and ${}^{b_2}\omega_x$ remain zero all the time. Besides, the drone is flying at uniform speed of V_t along $\vec{x}$ in state S_2. Then there have

$${}^{b_2}V = \begin{pmatrix} V_t \\ 0 \\ 0 \end{pmatrix}, \quad {}^{b_2}\Omega = \begin{pmatrix} 0 \\ 0 \\ \dot{\psi} \end{pmatrix}. \tag{15}$$

Since the image Jacobian model is set up in $\mathbb{F}_c$, it is necessary to transform the velocity in $\mathbb{F}_{b_2}$ to that in $\mathbb{F}_c$ by R_α^{-1}.

$${}^cV = R_\alpha^{-1} \cdot {}^{b_2}V, \quad {}^c\Omega = R_\alpha^{-1} \cdot {}^{b_2}\Omega. \tag{16}$$

Combined with Eq. (5), we have

$$\begin{pmatrix} \dot{x}'_P \\ \dot{y}'_P \end{pmatrix} = J_v \begin{pmatrix} {}^cV \\ {}^c\Omega \end{pmatrix}. \tag{17}$$

As the variables in both ${}^cV$ and ${}^c\Omega$ just include V_t and $\dot{\psi}$, and when V_t is constant, the variable is unique. Therefore, we can use the least square method to obtain the solution, and the process does not involve the pseudo-inverse of the image Jacobian matrix.

4 Simulation

To evaluate the feasibility of the algorithm, we conduct HIL simulation experiments based on Gazebo and Pixhawk, and present results obtained from simulations.

4.1 Simulation Setup

In the experiment, Gazebo is used as the simulation environment, and models of fixed-wing UAV and car (acted as the target) are added into it. For the model of fixed-wing UAV, it weights $2.65kg$, the wing surface of it is $0.47m^2$, the wingspan is $2.59m$ and the chord length is $0.18rad$. Besides, the tilt angle α is 45 degrees, the horizon of vision is 100 degrees and the image width and height are 960 pixels and 720 pixels respectively.

Besides, Pixhawk is used to communicate with Gazebo via mavlink protocol. Simultaneously, QGroundControl (QGC) is acted as the ground station for the purpose of visualizing the current pose of the drone, and displaying the trajectory of the drone in real time.

Algorithm 1 Framework of the new IBVS control algorithm

Require: (x_P, y_P)
Ensure: $\dot{\psi}$

1: **while** extract (x_P, y_P) from the image **do**
2: calculate the rotation matrices R_α, R_1, R_2;
3: $R \Leftarrow R_\alpha^{-1} R_2 R_1 R_\alpha$
4: obtain the desired coordinates:

$$\begin{pmatrix} x'_P \\ y'_P \end{pmatrix} = k(R)^{(1-2,\cdot)} \begin{pmatrix} x_P \\ y_P \\ -f \end{pmatrix}$$

5: calculate the velocity of the drone in $\mathbb{F}_c$:

$${}^cV = R_\alpha^{-1} \cdot \begin{pmatrix} V_t \\ 0 \\ 0 \end{pmatrix}, \quad {}^c\Omega = R_\alpha^{-1} \cdot \begin{pmatrix} 0 \\ 0 \\ \dot{\psi} \end{pmatrix}$$

6: construct the image Jacobian model:

$$\begin{pmatrix} \dot{x}'_P \\ \dot{y}'_P \end{pmatrix} = J_v \begin{pmatrix} {}^cV \\ {}^c\Omega \end{pmatrix}$$

7: use the least square method:

$$\dot{\psi} \Leftarrow \begin{pmatrix} x'_P \\ y'_P \end{pmatrix}$$

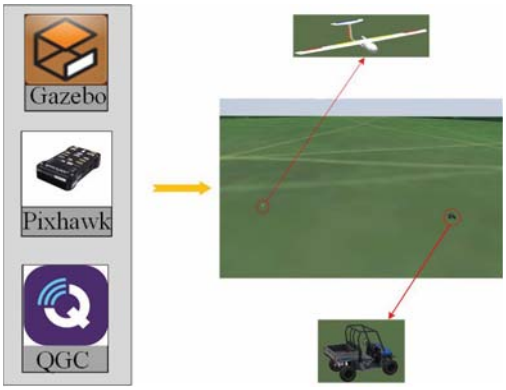

Fig. 6: Construction of hardware in the loop simulation. The left part includes the hardware (Pixhawk) and software (Gazebo, QGC) we used; the right part includes the simulation environment and the models of fixed-wing UAV and target (car).

The setup of simulation is shown as Fig. 6, the right part shows the simulation scene and the models of fixed-wing UAV and car, and the other part includes the hardware and software we use. The physical map of our simulation environment is shown as Fig. 7.

4.2 Simulation Description and Results

In the experiment, the car remains static on the ground to achieve tracking of the target for the drone, and the speed of the drone is changed to make it fly at uniform speed at constant altitude. In actual scenes, the static target corresponds to suspicious building or radar station, and the UAV performs monitoring task continuously after discovering it. Since the experiment aims at how to make the drone circle around the target after detecting it, the drone is set to fly to certain position at first so that it can detect the target and then

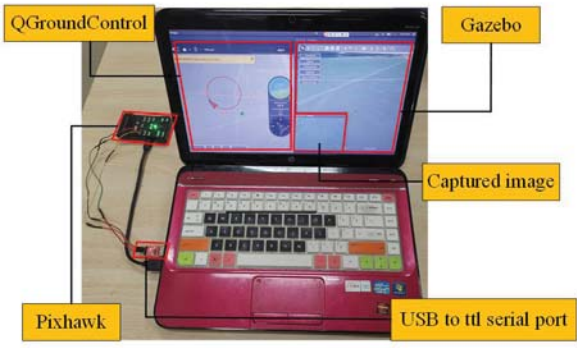

Fig. 7: Physical map of the simulation environment. The computer system we use is Ubuntu 14.04.

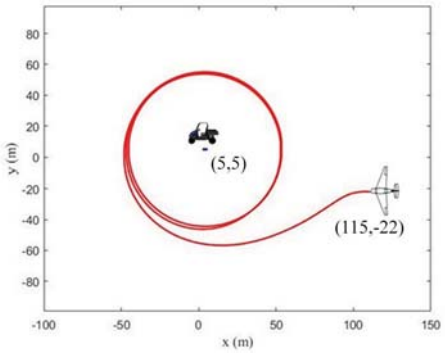

Fig. 8: Trajectory of the drone at the speed of 12 m/s. The blue point represents the location of car, and the red line represents the trajectory of the drone.

perform the task of tracking.

During the experiment, the flight height of the drone is set 50 meters, and $\lambda = diag\{0.5, 0.5\}$, $z = 43m$. The fixed-wing UAV needs to reach the minimum speed to take off, and it is $10m/s$ in the experiment. Simultaneously, the maximum speed does not exceed $17m/s$ due to the power limit of the fixed-wing UAV. Therefore, V_t is randomly chosen as 12 m/s.

In Fig. 8, the picture represents the trajectory that the drone circles around the target at the speed of $12m/s$. The result shows that the trajectory converges to circle at last.

Fig. 9 indicates that the distance between the target and the drone on a horizontal plane can converge to 50 m. The reason for the small error is that the height of the drone can not remain unchanged all the time, which caused the circling radius to change according to Eq. (7). Therefore, the results prove the stability of the algorithm.

Besides, Fig. 10 shows the trajectory of the drone in QGC. More specifically, the top center of the picture shows the state of the drone now is Offboard.

5 Conclusion and Future Work

This paper shows a new IBVS control algorithm that enables a fixed-wing UAV with a fixed camera to track a static target. The center of the target in the image can be used as the single input of the angular velocity control, and a compensation mechanism is proposed to solve the problem that the optic axis can not continuously point to the target. The

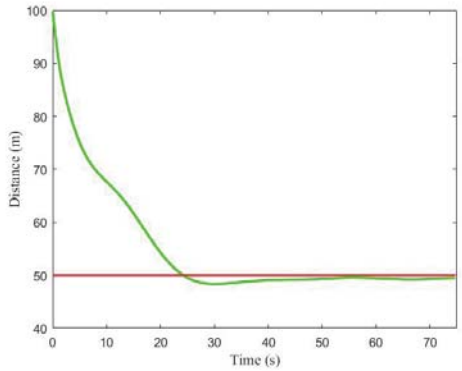

Fig. 9: Distance between the target and the drone on the horizontal plane.

Fig. 10: The trajectory of the drone when tracking the ground target in QGC.

results of the proposed algorithm show that the drone can circle around the target successfully. Future work will be focused on the improvement of the stability of the algorithm in interference environment, and the moving target will be considered.

References

[1] C. Deng, S. Wang, Z. Huang, Z. Tan, and J. Liu, "Unmanned aerial vehicles for power line inspection: A cooperative way in platforms and communications," *J. Commun*, vol. 9, no. 9, pp. 687–692, 2014.

[2] L. Tang and G. Shao, "Drone remote sensing for forestry research and practices," *Journal of Forestry Research*, vol. 26, no. 4, pp. 791–797, 2015.

[3] J. Colorado, I. Mondragon, J. Rodriguez, and C. Castiblanco, "Geo-mapping and visual stitching to support landmine detection using a low-cost UAV," *International Journal of Advanced Robotic Systems*, vol. 12, no. 9, p. 125, 2015.

[4] D. Orfanus, F. Eliassen, and E. P. de Freitas, "Self-organizing relay network supporting remotely deployed sensor nodes in military operations," in *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2014 6th International Congress on*, pp. 326–333, IEEE, 2014.

[5] S. Zhu, D. Wang, and C. B. Low, "Ground target tracking using UAV with input constraints," *Journal of Intelligent & Robotic Systems*, vol. 69, no. 1-4, pp. 417–429, 2013.

[6] M. Odelga, P. Stegagno, and H. H. Bülthoff, "Obstacle detection, tracking and avoidance for a teleoperated UAV," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pp. 2984–2990, IEEE, 2016.

[7] D.-Z. Xu, B. Jiang, and P. Shi, "Global robust tracking control of non-affine nonlinear systems with application to yaw control of uav helicopter," *International Journal of Control, Automation and Systems*, vol. 11, no. 5, pp. 957–965, 2013.

[8] M. Kamel, M. Burri, and R. Siegwart, "Linear vs nonlinear mpc for trajectory tracking applied to rotary wing micro aerial vehicles," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 3463–3469, 2017.

[9] B. Zhao, B. Xian, Y. Zhang, and X. Zhang, "Nonlinear robust adaptive tracking control of a quadrotor UAV via immersion and invariance methodology," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 5, pp. 2891–2902, 2015.

[10] J.-J. Xiong and E.-H. Zheng, "Position and attitude tracking control for a quadrotor UAV," *ISA transactions*, vol. 53, no. 3, pp. 725–731, 2014.

[11] S. Kim, H. Oh, and A. Tsourdos, "Nonlinear model predictive coordinated standoff tracking of a moving ground vehicle," *Journal of Guidance, Control, and Dynamics*, vol. 36, no. 2, pp. 557–566, 2013.

[12] C. Teuliere, E. Marchand, and L. Eck, "3-D model-based tracking for UAV indoor localization," *IEEE Transactions on cybernetics*, vol. 45, no. 5, pp. 869–879, 2015.

[13] N. Farmani, L. Sun, and D. Pack, "An optimal sensor management technique for unmanned aerial vehicles tracking multiple mobile ground targets," in *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on*, pp. 570–576, IEEE, 2014.

[14] S. A. Quintero, D. A. Copp, and J. P. Hespanha, "Robust UAV coordination for target tracking using output-feedback model predictive control with moving horizon estimation," in *American Control Conference (ACC), 2015*, pp. 3758–3764, IEEE, 2015.

[15] F. Chaumette, "Visual servoing," in *Computer Vision*, pp. 869–874, Springer, 2014.

[16] M. M. Aref, R. Ghabcheloo, A. Kolu, M. Hyvonen, K. Huhtala, and J. Mattila, "Position-based visual servoing for pallet picking by an articulated-frame-steering hydraulic mobile machine," in *Robotics, Automation and Mechatronics (RAM), 2013 6th IEEE Conference on*, pp. 218–224, IEEE, 2013.

[17] G. Dong and Z. Zhu, "Position-based visual servo control of autonomous robotic manipulators," *Acta Astronautica*, vol. 115, pp. 291–302, 2015.

[18] M. G. Popova and H. H. Liu, "Position-based visual servoing for target tracking by a quadrotor UAV," in *AIAA Guidance, Navigation, and Control Conference*, p. 2092, 2016.

[19] H. Jabbari, G. Oriolo, and H. Bolandi, "An adaptive scheme for image-based visual servoing of an underactuated UAV," *International Journal of Robotics and Automation*, vol. 29, no. 1, pp. 92–104, 2014.

[20] H. Xie, A. F. Lynch, and M. Jagersand, "Dynamic ibvs of a rotary wing UAV using line features," *Robotica*, vol. 34, no. 9, pp. 2009–2026, 2016.

[21] V. Lippiello, J. Cacace, A. Santamaria-Navarro, J. Andrade-Cetto, M. A. Trujillo, Y. R. Esteves, and A. Viguria, "Hybrid visual servoing with hierarchical task composition for aerial manipulation," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 259–266, 2016.

[22] P. Peliti, L. Rosa, G. Oriolo, and M. Vendittelli, "Vision-based loitering over a target for a fixed-wing UAV," *IFAC Proceedings Volumes*, vol. 45, no. 22, pp. 51–57, 2012.

[23] F. R. Triputra, B. R. Trilaksono, T. Adiono, R. A. Sasongko, and M. Dahsyat, "A nonlinear camera gimbal visual servoing control using command filtered backstepping," *Journal of Unmanned System Technology*, vol. 3, no. 2, pp. 49–60, 2015.

[24] S. Mills, N. Aouf, and L. Mejias, "Image based visual servo control for fixed wing UAVs tracking linear infrastructure in wind," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pp. 5769–5774, IEEE, 2013.