Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 5, No. 1, 2023

Real-time UAV Position Control based on Visual Servo Control Algorithm

Yifeng Gao *

Kang Chiao International School East China Campus, Suzhou, Jiangsu, 215332, China
*** Corresponding author Email:** gyf18221031137@163.com

Abstract: Position control in UAVs has an important function in modern times UAVs. With the development of computer vision, UAV visual servo control became a research topic. In this paper, a visual servo control system based on color characteristic detection is proposed. By introducing the concept of the pixel of error and proportional control, the input of the PID controller can be adjusted to reach the desired sensitivity of the whole system. A simulation experiment is conducted in Simulink, and it shows that the visual servo control has achieved its goal of real-time position control of a UAV with a target object.

Keywords: UAV; Visual Servo Control; Characteristic Detection; Position Control; Real-time Target Detection.

1. Introduction

With the increasing application of UAVs in various industries in recent years, position control of UAVs is also an important research direction. One of the important application requirements of UAVs in various fields is the stable positioning of UAVs and one of the key technologies for UAVs to realize this goal is visual servo control of UAVs. Through research in computer vision and other related fields, researchers can control the attitude of the UAV through a visual servo linked to the UAV, so as to achieve the effect of position control.

The study on UAV visual servo control has found a potential for growth recently with the rapid increase of image processing techniques and computer speed, as visual servo control demands high real-time performance. Thus, it poses challenges on how to find out the position of the target with fast speed, in order to provide enough time for the algorithm to work. There are three types of target positioning methods: finding characteristics, object tracking, and object detection.

To find the characteristics of the image, color thresholding, shape recognition, and template matching can be used. Color thresholding is a type of image segmentation method, that divides pixels of an image into target pixels and non-target pixels. By adjusting the range of the thresholds, it is possible to find the position of the target objects in the image. This method is simple and easy to use but may be affected by lighting conditions. Shape recognition recognizes objects of a specific shape by computing shape characteristics in an image, such as edges, corners, contours, etc. This approach can recognize complex shapes but requires intensive computations. Template matching identifies objects similar to a template by computing the similarity between an image and one or more templates. This approach can match objects precisely but may require large amounts of training data.

The tracking method uses the bounding boxes to track the target object and thus achieves the goal of controlling the position of UAVs. According to Hao et al., traditional tracking techniques like Mean Shift, Kalman Filter, Particle Filter, etc., have been utilized extensively. However, these algorithms perform less well against complicated backdrops and noise that is nonlinear and non-Gaussian. As computer science advanced, additional categories of tracking methods were introduced. A solution to issues with long-term tracking of erroneous objects in video streams is the TLD algorithm. Objects are defined by their positions and range within a single frame. In each subsequent frame, the task is to determine the location and range of the object or to indicate the absence of the object [1]. The authors proposed a new tracking framework TLD, dividing long-term tracking tasks into three parts: tracking, learning, and detection. The tracker tracks objects from frame to frame, the detector locates all appearances observed so far and corrects the tracker if necessary. Some other tracking algorithms include KCF, TLD-KCF, GOTURN, etc [2]. These algorithms are able to better handle object changes and occlusions by learning object appearance. In particular, the deep learning-based GOTURN algorithm can realize real-time high-precision object tracking by learning a large number of video sequences. At the same time, they all can also better guarantee the real-time robustness of the operation.

Object detection is a similar method to object tracking. Both use bounding boxes to locate the target object in the image. YOLO, short for "You Only Look Once", is a real-time object detection algorithm. Different from traditional methods, YOLO treats object detection as a regression problem and predicts object types and bounding boxes from image pixels, instead of generating a pre-detected area and then classifying the object in it. This method greatly improves the detection speed, enabling YOLO to achieve real-time object detection [3]. The function of these algorithms is to help locate the coordinate of the target object in the image captured by a camera, and then apply it in the PID control to manipulate the position of UAVs.

Visual servo control systems can be divided into two categories: position-based visual servo control and image-based visual servo control. Position-based systems first need to recover the 3D structure of the scene from the image, and then control the UAVs based on this 3D structure. In contrast, image-based systems perform control directly on the image plane without restoring the 3D structure [4]. Using visual information to determine the position and pose of the observed object in Cartesian space and comparing it to the desired position and pose to determine the control deviation, the position-based visual servo control method creates a closed-loop control system in three-dimensional Cartesian

space. To achieve steady tracking of the target, a controller is created based on the control deviation to regulate the UAV's motion or visual load [5].

According to the deviation between the pixel coordinates of the target in the image calculated by the tracking algorithms, and the set expected pixel coordinates given by desired results, the researchers can adjust the PID instructions of the UAV, and let the UAV operate according to the instructions, such as tracking target objects or planning routes [5]. The desired pixels are compared with the actual pixels according to the error, and the controller is designed to control the UAV's motion or visual load to achieve stable tracking of the target. Compared with position-based visual servo control methods, image-based visual servo methods do not need to estimate and measure the attitude of the target and the UAV, do not need to reconstruct the 3D space, and are easy to apply in practice as they do not require very high calibration of the visual load. In this paper, this approach will also be taken to control the UAV.

2. Proposed Method

2.1. Target Positioning

In the controlling system, it needs to track these features in continuous frames. With TLD or YOLO, the position of the target will be acquired with the central point of the bounding box of the detected object, the approximate coordinate of the object can be determined. A bounding box is usually defined by the coordinate (x_1, y_1) of its upper left corner and the coordinate (x_2, y_2) of its lower right corner. Thus, the central position of the target object can be calculated.

$$x = \frac{x_1+x_2}{2} \tag{1}$$

$$y = \frac{y_1+y_2}{2} \tag{2}$$

A more straightforward method is characteristic detection. The computer can calculate certain traits of the target object in the image, such as color, edges, templates, etc. In the case of using color to detect the target object, a threshold can be applied to identify traffic lights by setting a point to filter different color channels of the HSV image. It requires less calculation than other methods and thus achieves the goal of real-time control. Meanwhile, with an appropriate threshold, the coordinate of the target object can be provided directly, without the need for calculating the central position of bounding boxes. Therefore, color detection will be used in this essay as a way to detect the characteristics of the target object.

2.2. Proportional-integral-derivative Control (PID)

PID control is a usual control algorithm widely used in various types of control systems. The PID controller works by calculating the error values between an expected value and the actual value. Then, it will use this error to adjust the input to let the actual value approach the expected value.

The PID controller consists of three parts: proportion, integral, and derivative of the error. The weighted sum of these three terms is the output of the PID controller.

This is the mathematical expression of a classic PID controller. $u(t)$ is the output of the system, K_p, K_i, and K_d is the gain of proportion, integral, and derivative, respectively. Adjusting these values will change the performance of the system [6].

$$u(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{de(t)}{dt} \tag{3}$$

2.3. Visual Servo Control

As a feedback control system, visual servo control uses visual information to control the position of UAVs. The general idea is that a camera will be used to capture images. Then, the characteristic features or pending targets will be recognized by various algorithms. In this case, the algorithm is color thresholding. Next, these targets can be processed as input signals to drive the UAV.

Next, the system will calculate the error values, which is the difference between expected values and actual values. In the case of image-based visual servo control, the error is referred to an offset of pixel, or error. The expression of the offset of pixel is:

$$e = p^* - p = (x^* - x, y^* - y) \tag{4}$$

The value of e describes the offset of the actual position of the pixel relative to the desired position of the target object in the images. If the actual position of target object in the image is at the pixel of x and y and the expected position of the object is x^* and y^*, the error will be the difference of these two values, respectively [4].

To utilize the error of pixel to control the UAV, a common method is proportional control, that is, the control signal is proportional to the error of pixel [4].

$$u = ke \tag{5}$$

k is a scaler factor manipulating the relationship between error of pixel and the input signal of UAV's PID controller.

In image-based visual servo control, the goal of the system is to minimize the value of e to achieve the ideal movement of the UAV [4].

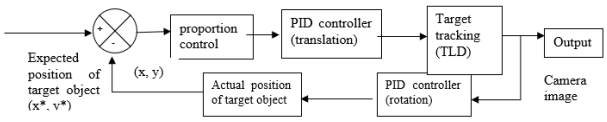


Figure 1. UAV visual servo control algorithm

In the Simulink environment, the camera's direction is [0, 0, 0], meaning that it is facing straight ahead. Thus, in order to control the UAV's position in the YOZ plane, the UAV must be able to move on the y-axis and z-axis. Two detailed formulae of PID control are given:

$$u_x(t) = K_{p_x} k e_x(t) + K_{i_x} \int_0^t k e_x(t)dt + K_{d_x} \frac{dk e_x(t)}{dt} \tag{6}$$

$$u_y(t) = K_{p_y} k e_y(t) + K_{i_y} \int_0^t k e_y(t)dt + K_{d_y} \frac{dk e_y(t)}{dt} \tag{7}$$

x and y refer to the coordinate of the image captured by camera. $e_x(t)$ is the error of pixel of x-axis in the image, and $e_y(t)$ is the error of pixel of y-axis in the image. K_{p_x}, K_{i_x}, K_{d_x} and K_{p_y}, K_{i_y}, K_{d_y} are the gain of the PID controller on these two axes. t is the time, the variable of the formulae. k is the scaler factor.

3. Experiment

3.1. Simulation Environment

The author designed an experiment to test the effectiveness of the visual servo control algorithm in MATLAB. The algorithm was tested with a Simulation 3D Scene in Simulink. The US city block is chosen as the 3D scene. The input of the image is provided by the Simulation 3D Camera model. The vehicle is a Simulation 3D UAV Vehicle with the type of quadrotor. We implemented a PID controller in Simulink to

regulate the UAV's flight based on the input from the color thresholding algorithm. The controller was tested under one initial translation point to evaluate its performance in maintaining the desired position. The output of the image recognition algorithm (i.e., the identified target's location) was fed into the PID controller as the reference input. The controller then calculated the error between the current and desired paths and adjusted the UAV's flight path accordingly. The expected results are that the controller should be able to effectively adjust the UAV's position in real time.



Figure 2. 3D scene and experiment vehicle

The initial translation is [0, 0, 0] m, the initial rotation is [0, 0, 0]°, and the initial velocity is [0, 0, 0] m/s. The translation of the camera is initial rotation is [0, 0, 0] m and rotation of the camera is initial rotation is [0, 0, 0]°. A traditional PID controller is utilized in the simulation, controlling the vehicle. In the 3D scene, a traffic light is chosen as the target object. The goal is controlling the position of the vehicle to locate the traffic light in the middle of the view of the camera, as it is a reference to the performance of visual servo control. The value of k is 1/11 in this case.

An image captured by the simulation 3D camera is analysed by a color thresholding algorithm. The image is described in HSV, where the ranges of the threshold of three channels are:

$$0.2 \leq H \leq 0.4 \quad (8)$$

$$0.2 \leq S \leq 1 \quad (9)$$

$$0.7 \leq V \leq 1 \quad (10)$$

H is normalized in MATLAB; 0.2-0.4 refers to the 72°-144°.

3.2. Simulation Results

An image is captured by the 3D camera at the initial state as shown in Figure 3. Then the image is imported into the color thresholding algorithm. From the calculations, the coordinates of the traffic light and central point, and the error of pixel can be obtained in Table 1. The result of HSV characteristic detection is shown in Figure 4.

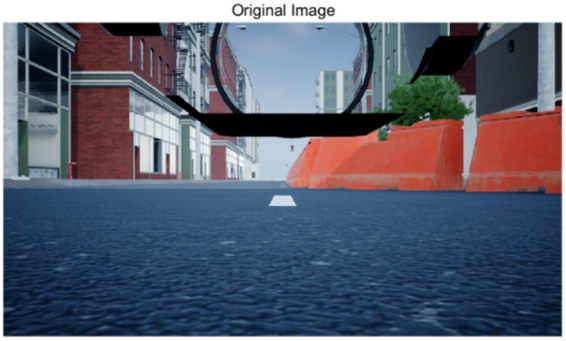


Figure 3. POV camera and traffic light (initial)

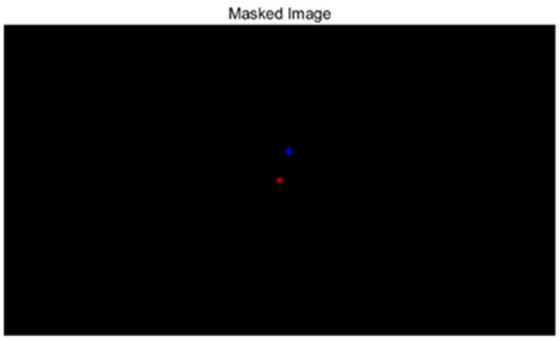


Figure 4. HSV characteristic detection (initial)

Table 1. Initial values of error of pixel

Coordinate of traffic light	$(662, 293)$
Coordinate of the centre	$(640.2, 359)$
Error in x-axis e_x	$-21.5\ pixel$
Error in y-axis e_y	$66\ pixel$

With the formulae for separate axes discussed in 2.3, the output of the PID controller can be obtained. The result of the movement of the three axes is shown in Table 2 and Figure 5.

Table 2. Results of visual servo control with PID controller

Displacement in x-axis	$0\ m$
Displacement in x-axis	$-1.95\ m$
Displacement in x-axis	$6\ m$

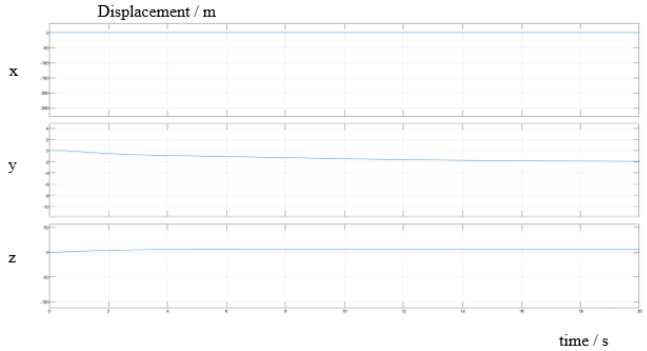


Figure 5. Result of visual servo control

Adjusting the movement in the y and z axes of the UAV, the visual servo control achieves the goal to reduce the error of pixels in the x and y axes in the image. Figure 6 and Figure 7 show the image captured by the 3D camera and the image with color thresholding. Table 3 shows the final values of the error of pixel.

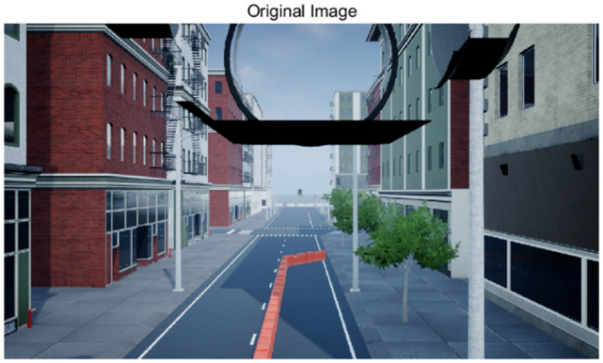


Figure 6. POV camera and traffic light (final)

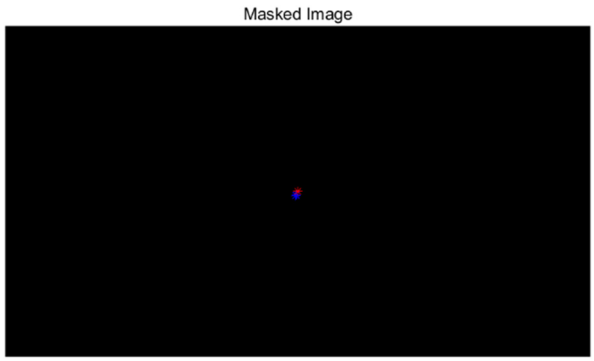


Figure 7. HSV characteristic detection (final)

Table 3. Final values of error of pixel

Coordinate of traffic light	$(639, 361)$
Coordinate of the center	$(635, 371)$
Error in x-axis e_x	$-2\ pixel$
Error in y-axis e_y	$8\ pixel$

4. Discussion

The article develops a visual servo control method to control the position of the UAVs with a target object. The procedures include locating the coordinate of the target object in the image with the color thresholding method, which detects the crucial characteristics of the traffic light in order to find its coordinate. Then, the error of the pixel is calculated by subtracting the coordinate of the target object and the coordinate of the center of the image. The result is errors in the x-axis and y-axis of the image, corresponding to the y-axis and z-axis in the simulation 3D scene. These errors are substituted into two separate PID controllers controlling the y and z axis of the UAV. A scalar factor k is introduced to manipulate the sensitivity of the PID system by changing the magnitude of the error. An experiment is conducted in Simulink. The goal of the experiment is to let the traffic light reach the center of the camera of the UAV with this visual servo control system. A conclusion can be drawn from the result that this visual servo control system has reached the goal, where the final values of error of pixel in Table 3 show smaller error values compared to those in Table 1.

In future studies, there are several potential research topics. First, an improvement of the PID controller can be introduced. Adding more parameters or using better fine-tuned controllers can bring higher sensitivity to the system, and therefore bring better performance. Second, object detection algorithms such as YOLO can be combined with visual servo control systems. By recognizing each object in the image, the UAV can conduct more tasks than merely finding the characteristic features in the images.

Acknowledgments

The author would like to thank Professor Peng Lu who provided invaluable guidance.

References

[1] Kalal, Z., Mikolajczyk, K., & Matas, J. (2012). Tracking-Learning-Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 34(7), 1409–1422.

[2] Liu, Y., Wang, Q., Hu, H., & He, Y. (2019). A Novel Real-Time Moving Target Tracking and Path Planning System for a Quadrotor UAV in Unknown Unstructured Outdoor Scenes. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 49(11), 2362–2372.

[3] Jiang, P., Ergu, D., Liu, F., Cai, Y., & Ma, B. (2022). A Review of Yolo Algorithm Developments. Procedia Computer Science, 199, 1066–1073.

[4] Hutchinson, S., Hager, G. D., & Corke, P. (1996). A tutorial on visual servo control. IEEE Transactions on Robotics and Automation, 12(5), 651–670.

[5] Zhao, M., Yan, J., Shi, D., Wang, Y., Pan, J., & Du, J. (2022). Research on Visual Servo Control of UAV Pod Based on Improved PID Algorithm. 2224(1), 012112–012112.

[6] Johnson, M. A., & Moradi, M. H. (2005). PID Control. Springer-Verlag London Limited.