Proceedings of the 2nd International Conference on Functional Materials and Civil Engineering
DOI: 10.54254/2755-2721/98/20241092

Autonomous Cargo Grabbing System for Drones Using Cameras and ROS

Kangrui Fang[1,2,*], Qigong Huang[1,3], Yilin Hao[1,4]

[1]College of Mechanical and Electrical Engineering, Central South University, Changsha, China
[*]corresponding author

[2]8203210526@csu.edu.cn
[3]8203221703@csu.edu.cn
[4]haoyilin@csu.edu.cn

Abstract. With the rapid development of low-altitude economy, the functional requirements for logistics UAVs are constantly improving. In this paper, an unmanned aerial vehicle system (UAV) is proposed to realize autonomous grasping of goods. First of all, a new type of grasping device is proposed, which can be easily mounted on the UAV. The camera image stream mounted on the UAV was obtained through camera calibration and corrected. Subsequently, the object recognition algorithm was used to process the image frames, successfully identify the goods to be captured, and calculate the position and direction relationship between the UAV and the target cargo. Next, the speed of the drone is adjusted by the PID controller to achieve a stable landing above the target cargo. Then, the drone is equipped with a robotic claw innovatively designed by our team to complete the gripping of the goods. Subsequently, the drone lifted off and began to deliver cargo. Throughout the grabbing process, a laptop equipped with Ubuntu and ROS systems communicates with the drone via Wi-Fi. The feasibility of the whole system was proved by experiments.

Keywords: logistics drone, autonomous gripping, PID controller, mechanical claw, position and direction relationship.

1. Introduction

In recent years, Unmanned Aerial Vehicles (UAVs), particularly multi-rotor drones, have garnered significant attention from research teams and are increasingly critical in the logistics sector. A pivotal challenge in achieving fully automated drone logistics is the autonomous grasping of cargo by UAVs. To enable this capability, it is essential to precisely monitor the quadrotor drone's position, orientation, and kinematic data. Typically, such data can be acquired through the Global Positioning System (GPS) and Inertial Navigation System (INS). However, in indoor environments where GPS signals are unavailable, visual sensors like tracking cameras are employed to gather the necessary information.

In this paper, while the tracking camera is used for the indoor positioning of the drone, the Intel® RealSense™ depth camera D435i is also used to identify the yellow cargo and measure the relative position of the cargo relative to the aircraft. The camera captures images of various resolutions in depth and color at a frame rate of 30Hz. Since the depth and color images are provided by different camera sensors, the two images need to be aligned. This can be achieved through the Intel RealSense OpenCV support library and the libalsense wrapper. After acquiring the RGB and depth images, object segmentation is performed through color filtering methods[1], followed by calculating the object's position relative to the camera using depth data. Once the object's position relative to the camera is determined, the homogeneous coordinate transformation formula[2] is applied to convert the object's coordinates from the camera's frame to the drone's frame. Finally, a PID control algorithm is employed to guide the drone in grasping the cargo.

2. The mechanical structure of the drone, the software and hardware of the system

Our team designed the logistics drone "KFR" as shown in figure 1. The robot serves as an algorithm demonstration and testing platform, mainly used for automatic grasping of goods as well as delivery of goods. It has the functions of automatic path planning, precise positioning, effective obstacle avoidance, and automatic grasping of goods.

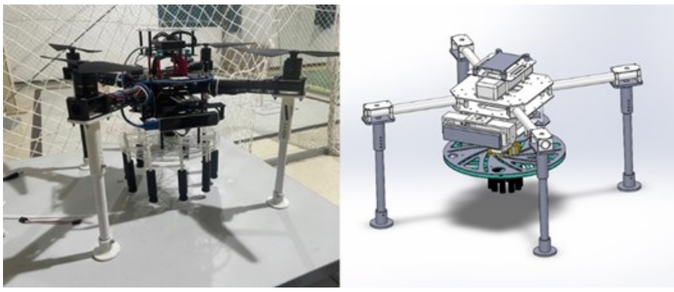

Figure 1. Mock-up and 3D models of drones

2.1. A new type of mechanical claw on board

The logistics UAV proposed in this paper mounts the mechanical claw on the bottom of the aircraft, and the aircraft acts as the carrier of the mechanical claw and drives the mechanical claw to move. When the aircraft needs to grab an object, it will land above the object that needs to be grabbed and then command the mechanical claw to grab the object.

The traditional mechanical claw is often mounted at the end of the mechanical arm, and the aircraft has the characteristics of unstable position and attitude relative to the mechanical arm, so the accuracy of the aircraft equipped with the mechanical claw to grasp the object is far less than that of the mechanical arm. In order to solve this problem, a new type of mechanical claw is proposed, as shown in figure 2.

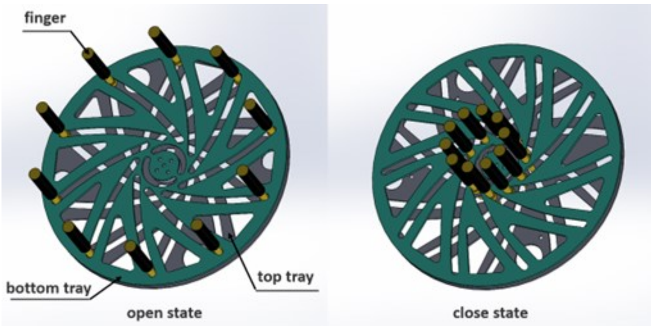

Figure 2. New mechanical gripper

Its grasping range is significantly improved over traditional mechanical claws and is well-suited for logistics drones.

This mechanical gripper consists of an upper tray, a lower tray, and multiple fingers, with the area enclosed by the fingers defining the gripping range. The design is based on the principle of cam

DOI: 10.54254/2755-2721/98/20241092

grooves[3]. The operational process is as follows: when the lower tray rotates relative to the upper tray, the fingers converge towards the center, pushing the object within the range towards the center of the tray, thereby achieving the grip. Due to its large gripping range, even if the drone does not land precisely above the object to be grasped, the gripper can still secure the object as long as the fingers cover it after landing.

2.2. Hardware systems for drones

The logistics drone hardware system includes a forward-facing Intel RealSense T265 tracking camera, which is used for the indoor positioning of the drone [4]; a downward-mounted Intel RealSense d435i camera, which measures the position of the aircraft relative to the object to be grasped; A Pixhawk 6C flight controller, which integrates a variety of sensors such as barometer, IMU and magnetometer, which can be used to obtain key information such as the attitude of the aircraft, and achieve precise control of the aircraft's flight through precise motor speed control; An Nvidia Jetson Nano edge computing device, which is the aircraft's on-board computer, used to run ROS, Open CV and other programs, in addition to the aircraft also has lithium batteries, servo drive boards, voltage regulator modules, etc., their connection architecture and connection methods are shown in figure 3.

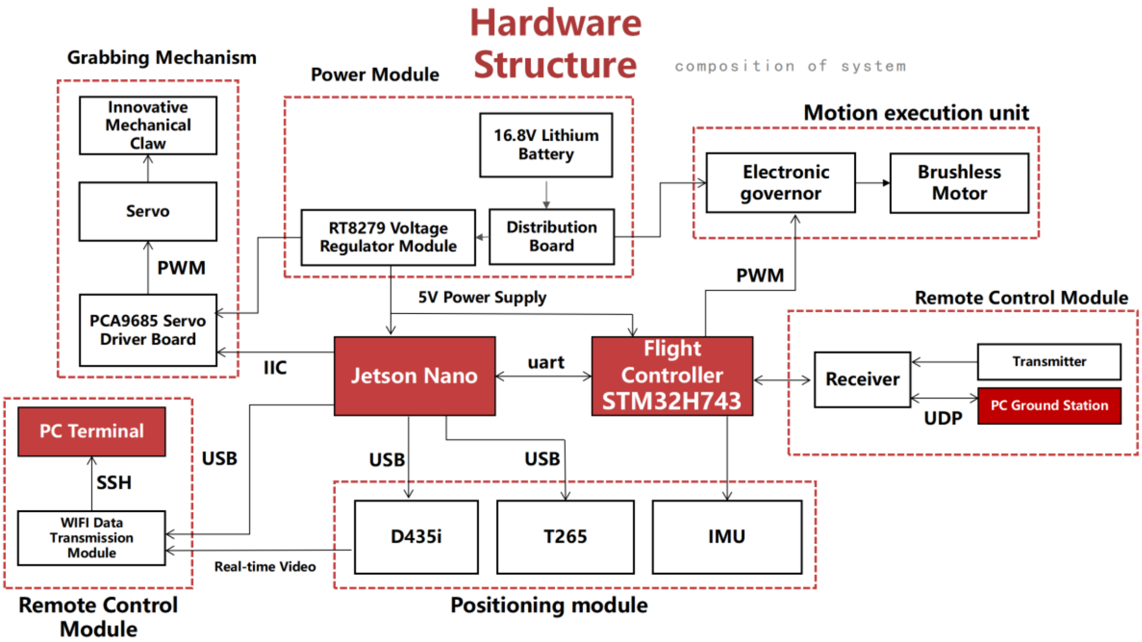

Figure 3. Aircraft hardware architecture

2.3. The software system of the drone

ROS, Robot Operating System is a flexible framework and toolbox for developing robotics applications. ROS can provide features similar to traditional operating systems, such as hardware abstraction, low-level device control, massage transfer between threads, and package management. In addition, it offers a plethora of utilities and libraries.

MAVROS (MAVLink ROS) is an open-source ROS (Robot Operating System) package, which is used to connect ROS and MAVLink protocols to achieve communication and control between ROS and UAVs. MAVLink is a lightweight communication protocol specifically designed to transmit information between drones and ground stations. In our project, the on-board computer is connected to the flight controller via Mavros to transmit flight control information to the flight controller.

Realsense-ros is a software package developed by Intel for the use of Intel RealSense depth cameras in ROS (Robot Operating System). In our project, we used RealSense-ROS to connect the T265 to the on-board computer, and the T265 sent the pose information to the on-board computer through RealSense-ROS to achieve indoor positioning of the aircraft.

Cv_bridge is a library commonly used in ROS (Robot Operating System) for the conversion and interaction of image formats between the ROS environment and the OpenCV image processing library. It allows for seamless conversion of image data in ROS with image data in OpenCV. In our project, we hand over the images to OpenCV for processing via cv_bridge.

3. Cargo Identification and Positioning

3.1. Sections, subsections and subsubsections

In the landing phase of the grabbing object, it is necessary to figure out the positional relationship between the drone and the grabbing object.

We use a yellow cylinder to simulate the goods, the picture of the cylinder and the model are shown in figure 4.

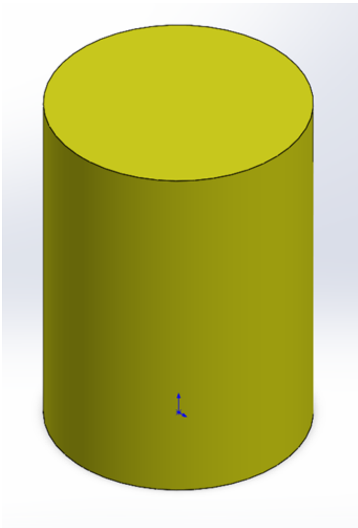

Figure 4. Grabbed goods

The Intel® RealSense™ depth camera D435i is used to recognize and locate objects. The camera captures images of various resolutions in depth and color at a frame rate of 30Hz. Since the depth and color images are provided by different camera sensors, the two images need to be aligned. This can be done through the Intel RealSense OpenCV support library and librealsense package implementation. After capturing RGB and depth images, the RGB images are used to segment the objects, and the position of the objects relative to the camera is calculated in combination with the depth map.

In this paper, an object detection and tracking method based on color segmentation and contour detection is used. Firstly, the area of the target object is extracted from the image by color segmentation technology, and then the contour detection algorithm is used to locate and track the target.

Color segmentation: Firstly, the color image is converted into HSV color space, and the area of the target object is extracted by color segmentation technology according to the preset color threshold range. In this paper, we have chosen yellow as the color of the target object and achieved accurate segmentation of the target by adjusting the threshold range[5].

Denoising and Contour Detection: In order to reduce the noise in the image and locate the target object more accurately, we perform median filtering on the segmented binary image, and then use the contour detection function cv2.findContours() provided by OpenCV to detect the contour of the target object.

Target Positioning & Tracking: Iterate through all detected contours, use the cv2.boundingRect() function to get the bounding box of each contour, and filter the target according to the set conditions

DOI: 10.54254/2755-2721/98/20241092

(e.g., width greater than or equal to 30 pixels). Finally, the cv2.circle() function is used to plot the detected target location in the original image.

3.2. Positioning of goods

In this paper, the purpose of cargo positioning is achieved by obtaining the three-dimensional coordinates of the center point of the cargo. On the basis of cargo recognition, the depth information extracted from the center point of the detected object by the Intel® RealSense™ depth camera D435i depth camera is used, and the standard program of the pixel to world transformation of the point is used to calculate the three-dimensional position of the target. Images of the color and depth channels of a camera using the dedicated software Intel RealSense Viewer are shown in .

Once the depth image is available, the depth information D_{x_0,y_0}, (C_{x_0}, C_{y_0}) can be achieved. And the three-dimensional spatial coordinates of the point P $(X_r Y_r Z)$ can be calculated by equations 5, 6, and 7. [6]

$$X_r = \frac{D_{x_0,y_0}\left(C_{x_0} - x\right)}{f_x} \tag{1}$$

$$Y_r = \frac{D_{x_0,y_0}\left(C_{y_0} - y\right)}{f_y} \tag{2}$$

$$Z = D_{x_0,y_0} \tag{3}$$

The variables "C_{x_0}", "C_{y_0}", "f_x", and "f_y" are the intrinsic parameters of the camera used to obtain information. Here, (f_x, f_y) represents the focal length components, and (C_{x_0}, C_{y_0}) represents the image center. This article utilizes the official calibration toolkit from Intel to calibrate the D435i camera, obtaining the intrinsic values of the camera. The intrinsic matrix of the camera obtained from the calibration results is as follows:

$$M_r = \begin{bmatrix} 610.244 & 0 & 320.619 \\ 0 & 610.244 & 246.028 \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

This means that the focal lengths of the RGB lens are $\left(f_x, f_y\right) = (610.244, 610.244)$, and the principal point is $(x, y) = (320.619, 246.028)$.

4. Control of the drone

4.1. The procedural flow of the drone to grab the goods

The whole process is shown in figure 6. The airborne computer carries out mark recognition and pose calculation, and sends the control information to the UAV flight control module to realize the control of the UAV.

The drone needs to reach the location where the goods can be visually captured, and then the drone will fly over the goods. The position deviation of the UAV will gradually decrease after reaching the threshold conditions in the X and Y directions. During the descent process, the position deviation in the x and y directions will continue to be adjusted to meet the threshold conditions. If the ground cargo disappears in the image in the camera during the descent, the drone will fly higher to allow the camera to recapture the object. When the vertical distance of the drone from the cargo is less than the threshold, the drone will land vertically above the cargo and use a mechanical claw to grab the cargo. Finally, after grabbing the cargo, the drone takes off again. [7]

4.2. Position control of the drone when grabbing objects

The PID controller [8] is used to calculate the control quantity, so that the deviation value of the UAV from the desired position is as small as possible. The mathematical expression for the PID controller is:

DOI: 10.54254/2755-2721/98/20241092

$$\mu_x(t) = K_p e_x(t) + K_I \int_0^t e_x(t)\,dt + K_D \frac{de_x(t)}{dt} \tag{5}$$

$$\mu_y(t) = K_p e_y(t) + K_I \int_0^t e_y(t)\,dt + K_D \frac{de_y(t)}{dt} \tag{6}$$

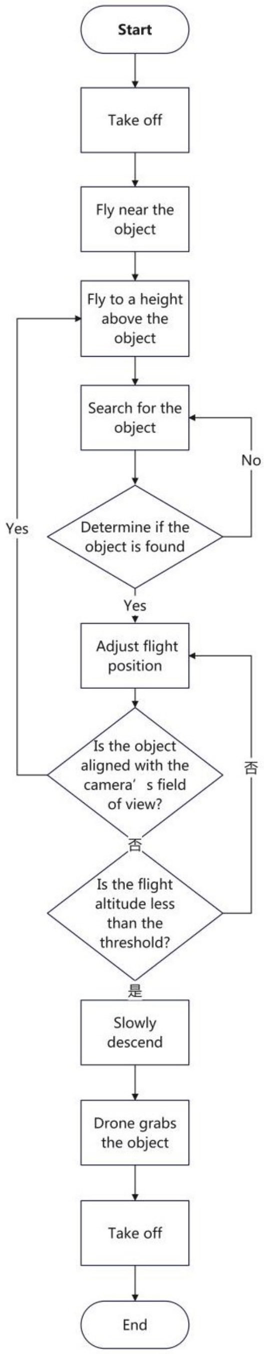

Figure 5. Integrated System for UAV Cargo Recognition and Acquisition

In this formula, $e_x(t)$ and $e_y(t)$ represent the deviation values of the x-axis and y-axis components of the aircraft's current position from the desired position, respectively. Due to the change in altitude during the landing process of the UAV (Unmanned Aerial Vehicle), the speed of the UAV's position adjustment needs to be altered. Therefore, the calculation of the PID control parameters in this paper will be based on different heights $K_p = f(z_r)$, $K_i = g(z_r)$, $K_d = h(z_r)$. As a result, the control value will only be related to the height and relative position, which means that other disturbances such as wind will not affect the control value.

5. Analysis of Experimental Results

5.1. Mechanical claw grasping cargo experiment

In order to evaluate the gripping performance of the grippers, we carried out a series of tests in the laboratory. We mainly tested the success rate of the robotic arm in grasping objects of various shapes. For this purpose, we have prepared three geometries in three shapes, namely triangular prisms, cylinders, and cuboids, the dimensions of which are shown in table 1.

Table 1. A table with headings spanning two columns and containing notes.

shape	Dimensions (mm)		
Triangular prism	Edge length		high
	60.62		70
cylinder	diameter		high
	50		70
cuboid	long	wide	high
	80	50	70

We used a 3D printer to manufacture these objects for testing. The theoretical gripping area of the mechanical claw is shown in figure 6, which is a circular area with a diameter of 183.77mm. As long as the object is within this area, it can theoretically be grabbed by the mechanical claw.

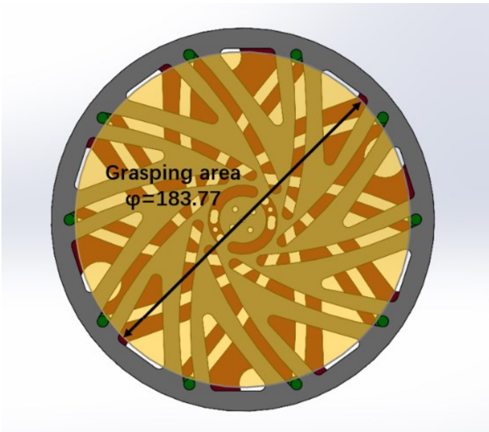

Figure 6. Gripping Surface Area of the Mechanical Claw

We take the center of the grasping area as the origin of the coordinate system, with the right side designated as the positive x-axis direction and the upper side as the positive y-axis direction, thereby establishing the coordinate system.

Using MATLAB, we randomly generated a number of positions, placed our test gripping objects on them, and then activated the mechanical claw to test whether we could grasp the objects. The randomly generated positions in MATLAB are shown in the grabbed area as shown in figure 7. The outer red circle is the grasping area, and the graphics of various colors in the grasping area represent the position of the randomly generated grasped object.

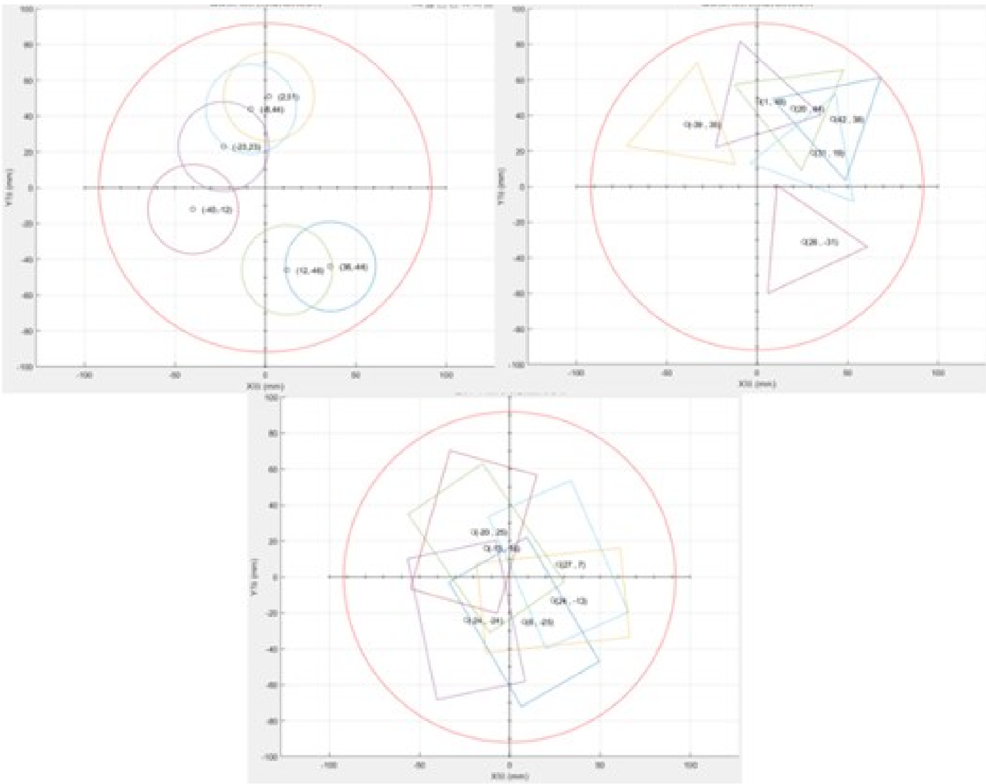

Figure 7. MATLAB-Generated Gripping Positions for Mechanical Claw Evaluation

After the test in reality, the robotic arm grabbed the object 100%, and we filmed the process of the robotic arm grabbing the object, it is shown in the figure below

In general, the robotic arm mounted on the aircraft has good grasping performance and is capable of grasping objects of various shapes.

5.2. The camera recognizes the object and returns the object's 3D coordinates

In order to achieve accurate grasping of objects by drones, we need highly accurate information about the relative position of objects and aircraft. And this information is obtained by the Intel® RealSense™ depth camera D435i. This section tests the accuracy of the information acquired by the camera.

The team first designed a set of experimental equipment, as shown in figure 10, which consists of a camera mount, a camera and a yellow object located on the ground, which can adjust the distance of the camera relative to the object on the ground on the z-axis by adjusting the height of the camera mount, and can change the distance of the object relative to the camera on the y-axis by moving the object on the ground.

We used MATLAB to randomly generate a set of X and Y data at different altitudes, and placed objects in these positions, and by comparing the data measured by the camera with the real data, we can get whether the relative position information of the object obtained by the camera is accurate enough.

The analysis of the data shows that the real coordinates and measurement coordinates increase with the increase of altitude, but the overall level is small, so the coordinates of the aircraft obtained by the camera relative to the object are relatively accurate.

5.3. Drone grabbing cargo test

In this section, we design an experiment in which a drone grabs an object and drops it. First, we designed a map as shown in figure 11.

DOI: 10.54254/2755-2721/98/20241092

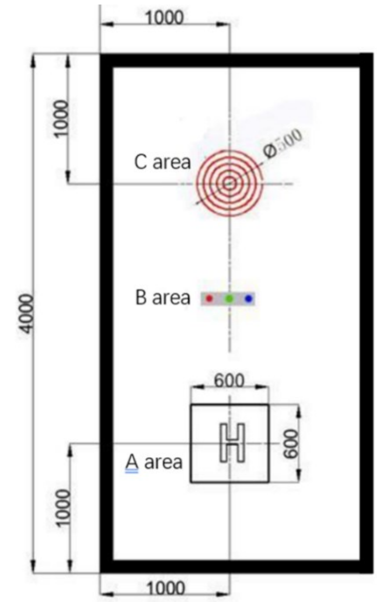

Figure 8. Map of drone picking and dropping goods

In this diagram, Area A indicates the take-off and landing areas of the drone; Area B is the cargo storage area, with a size of 450mm×100mm, located at the midpoint of area A and area C; Area C is a circular target with an outer diameter of 500 mm and a line width of 5 mm.

The aircraft first takes off from Area A, flies to Area B to land and grabs the cargo. After the cargo is successfully captured, it takes off again and flies to Area C for cargo drop-off. After landing in Area C and successfully dropping the cargo, the aircraft took off again and returned to Area A. At this point, the whole process is complete.

In order to objectively evaluate the success of the aircraft's grasping and delivery mission, we have formulated several criteria, which are whether the aircraft successfully took off, whether it grabbed the cargo, whether the dropped object was accurate (the object was dropped in the area of the first ring), and whether the aircraft successfully returned to the landing point (whether the projection of the aircraft on the ground in the area other than the propeller was included in the take-off and landing area).

We conducted ten experiments and compiled the results presented in table 2. Among them, success is represented by 1, failure is represented by 0, the number of rings is best 10 rings, the lowest is 6 rings, and if it is not cast in the target, it is 0 rings.

Table 2. Performance Evaluation of Aircraft in Grasping and Delivery Tasks

Number of experiments	takeoff	Grab the goods	Number of delivery rings	landing
1	1	1	8	1
2	1	1	8	1
3	1	0	0	1
4	1	1	7	1
5	1	1	10	1
6	1	0	0	1
7	1	1	6	1
8	1	1	8	1
9	1	1	7	0
10	1	1	9	1

6. Conclusion

This paper presents a four-rotor UAV cargo grabbing system. With the assistance of various useful ROS software packages, the paper has accomplished tasks such as camera calibration, cargo identification, and drone control, etc. The paper employs a PID controller to manage the speed of the drone, ensuring the drone's stable landing. After the drone lands on top of the cargo, it activates the mechanical claw to grasp the cargo before taking off again. In future research, the paper will continue to refine the control strategies and pose estimation algorithms.

References

[1] H. D. Cheng, X. H. Jiang, Y. Sun, and J. Wang, “Color image segmentation: Advances and prospects,” Pattern Recognition, vol. 34, no. 12, pp. 2259-2281, 2001.
[2] Ma, Y., Soatto, S., Kosecka, J., & Sastry, S. S. (2004). An Invitation to 3-D Vision: From Images to Geometric Models. Springer Science & Business Media.
[3] Lee, J. J., and K. F. Huang. "Geometry analysis and optimal design of Geneva mechanisms with curved slots." Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 218.4 (2004): 449-459.
[4] Lvzhao Bai, Jianshu Liang, LinLin Zhang Trajectory tracking control of quadrotor UAV based on T265 positioning
[5] A.S. Silva; F.M.Q. Severgnini; M.L. Oliveira; V.M.S. Mendes; Z.M.A. Peixoto Object Tracking by Color and Active Contour Models Segmentation
[6] Wilczkowiak M, Boyer E, Sturm P. Camera calibration and 3D reconstruction from single images using parallelepipeds[C]//Proceedings eighth IEEE international conference on computer vision. ICCV 2001. IEEE, 2001, 1: 142-148.
[7] Ming-You Ma;Shang-En Shen;Shang-En Shen; Enhancing UAV Visual Landing Recognition with YOLO’s Object Detection by Onboard Edge Computing
[8] Kiam Heong Ang; G. Chong; Yun Li PID control system analysis, design, and technology