2021 International Conference on Intelligent Computing, Automation and Systems (ICICAS) | 978-1-6654-2810-1/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICICAS53977.2021.00100

A Study of Autonomous Landing of UAV for Mobile Platform

Haojie Xiang[1], Jianjun Yi[1,*], Bo Zhou[1], Haili Wu[2,3], Jinzheng Mou[2,3]

[1] Department of Mechanical Engineering, East China University of Science and Technology, Shanghai 200237, China
[2] Shanghai Aerospace Control Technology Institute, Shanghai 201109, China
[3] Shanghai Key Laboratory of Aerospace Intelligent Control Technology, Shanghai 201109, China
[*]Corresponding author: jjyi@ecust.edu.cn

***Abstract*—Robot cooperation is the key to search and rescue (SAR) missions, which usually occur in complex situations affected by different types of disasters, and such situations usually require UAVs to cooperate with ground unmanned vehicles (UGVs) to provide valuable information, but achieving autonomous landing when return is urgently needed has been a research hotspot and difficulty in the field of mobile robotics. In this paper, an autonomous landing strategy for UAVs on mobile platforms is proposed to solve the above problems, and the specific research includes: (1) the construction of an autonomous landing platform and the design of cooperative target are completed considering the rapidity and stability of landing; (2) a method that can detect and track moving targets in real-time is proposed for the problem of moving target position movement; (3) the simulation scene and the actual scene of target identification is conducted accordingly. The experimental results show that the recognition of the target in both the simulation scene and the real scene is rapid and accurate.**

Keywords—Quadrotor, autonomous landing, mobile landing platform, target detection

I. Introduction

In recent years, with the continuous updating of technologies related to drones, more and more drones are being applied to various fields of life, including scientific exploration and data collection[1], commercial services, military reconnaissance and law enforcement [2], search and rescue [3], patrol [4], and entertainment [5]. At the same time, the cost of drones is gradually decreasing with the continuous development of related technologies such as microelectronics technology and digital communication technology, making more and more civilian drones appear. The increasingly widespread application of UAVs puts forward higher requirements for autonomous flight technology. After completing specific tasks, autonomous and safe landing and recovery are of great significance to the reproducible use of UAVs. The autonomous landing of a UAV refers to the process of the UAV using GPS, IMU, and other equipment for positioning and navigation and controlling the UAV to safely land at a designated location through the flight control system[6]. The most common landing methods are direct air recovery, parachute landing recovery, man-made remote gliding landing, blocking recovery, and autonomous landing recovery[7]. The first four methods still require human intervention and many restrictions. In addition, it is greatly affected by environmental conditions, and the landing sites are scattered and uncontrolled without autonomy, stability, and even less autonomy in landing. Therefore, autonomous landing and navigation have become a hot spot in the previous research on UAV intelligent landing. At the same time, in the field of computer vision, target tracking has always been an important topic. Obtain images from the camera, estimate the position of the target in a continuous image sequence, determine the motion information of the target's speed, direction, and trajectory, and realize the behavior of the moving target. Achieving behavioral analysis and understanding of moving targets has high research value and research significance in both military and security, so autonomous landing of UAVs on ground moving platforms based on visual guidance has received a lot of attention as a key technology[8]. For the recovery phase of SAR missions, a cooperative target landing system for mobile platforms is proposed in this paper. Through the identification and tracking of the cooperative target, combined with the movement prediction of the mobile platform, the target that the drone needs to track is obtained. Finally, it will land on the mobile platform and complete the final recovery of the drone.

The structure of this paper is as follows. In section II, we explain in detail the specific method of mobile platform landing recognition and tracking proposed in this paper. Section III will show the experimental results based on simulated scenarios and real-world data. Finally, the fourth part will provide some conclusions and future work.

II. Identification And Tracking Method

Landing guidance is the basis for the landing of the UAV mobile platform. During the recovery phase of SAR operations, the UAV needs to be guided to the area where the mobile platform is located via GPS before starting fine alignment and landing. For target recognition, autonomous tracking, and landing, this paper designs a set of autonomous landing systems for UAVs based on a visually guided mobile platform based on the requirements for autonomous landing of UAVs. The general process is shown in Fig.1.

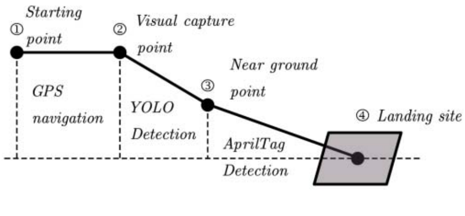


Fig. 1. Schematic diagram of the overall process

A. Overview

In this section, for the landing process of the mobile platform, a composite target design (in section 2.2.1) is adopted. When the altitude is high, YOLO is used to identify the target and land slowly; for the image processing module (section 2.3), when the altitude drops to after a certain threshold, AprilTag is used to identify a more accurate relative pose. Both results are passed to the prediction module (Section 2.4). Kalman filter is used to predict the position information of the car at the next moment. The predicted information at this point is used as the control input for the UAV, and the overall block diagram is shown in Fig.2.

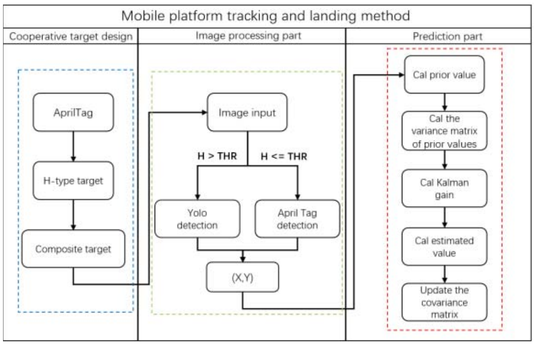


Fig. 2. Flow chart of target recognition and tracking method

B. Cooperative target design

For moving targets, it is necessary to design a target that is helpful for recognition and detection, so this paper is going to use the combination of the recognition subject and AprilTag as the recognition target, so that the recognition algorithm of Yolo and AprilTag can be used in stages during the recognition process, resulting in more accurate recognition.

1) Coordinate system transformation

The cooperation target of this paper is designed to be H-shaped as the main body combined with AprilTag, and at the same time add a red round frame to increase its characteristic information. The structure diagram is shown in Fig.3.

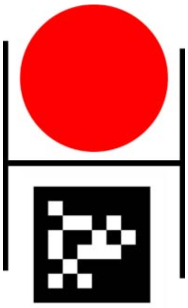


Fig. 3. Landing cooperative target

2) AprilTag library

Apriltag is a visual benchmark library. It has a wide range of applications in the fields of VR, robots, and camera calibration. With the specific two-dimensional code markers, the target location can be quickly detected and the relative position can be calculated in real-time. Its real-time is even possible on cell phone level processors. For cooperation targets with 3D information, AprilTag is the most widely used visual landing sign. One of the AprilTag series of signs is shown in Fig.4.

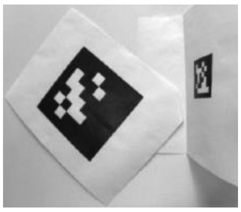


Fig. 4. AprilTag identification code

The AprilTag visual library contains many types of visual identifiers, such as Tag25h9, Tag36h11, etc. Each category has a different id. Take Tag36h11 as an example, '36' indicates that the number of squares that can be used for effective identification is 6*6, as shown in the white square in the above figure; h11 indicates that the Hamming distance is 11, which is used in data transmission error control coding; id=5 represents the serial number represented by this type of Tag36h11.

C. Image recognition algorithm

Considering that the scene will be affected by foggy weather, the defogging algorithm and recognition algorithm are combined. In foggy weather, defogging is considered first and then target recognition. For target recognition, this paper uses a combination of two recognition methods. When the distance to the target is greater than a threshold, the YOLO recognition algorithm is used. YOLO recognition is faster and more robust than traditional image recognition algorithms; AprilTag is used to calculate relative poses when the distance is close enough. The block diagram is shown in Fig.5.

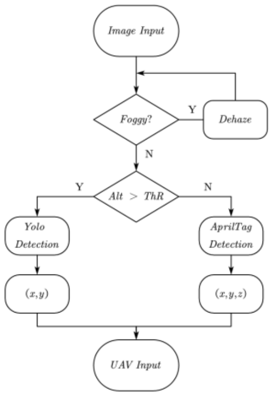


Fig. 5. Image processing flow

1) Dehazing algorithm

The defogging algorithm used in this paper is based on the dark channel prior proposed by He Kaiming to defog the image. The algorithm is mainly divided into five steps:

(1) The dark channel a priori, that is, in sunny and fog-free weather, there will always be at least one color channel with a lower value (or even close to zero) for certain elements;

(2) Estimate an approximate transmittance based on the a priori of the dark channel;

(3) Soft mapping, that is, an optimization of the

transmittance, so that the edge information is more obvious;

(4) Estimate the light intensity of the atmosphere. In the generated "dark channel" image, obtain the pixel position of 0.1% of the picture size. This position represents the light intensity of most atmospheric colonoscopy, and the average value is accumulated to obtain a relatively stable value;

(5) Restore the preliminary dehazing image through the general model of the foggy image.

This paper is based on the scene in the simulation environment and simulates the foggy effect in Photoshop to test the effect of the dehazing algorithm. The experimental results are shown in Fig.6.

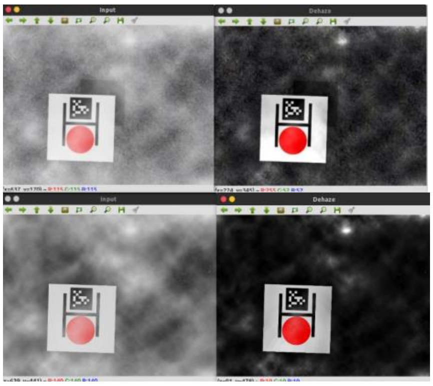


Fig. 6. Dehazing experiment

For different degrees of foggy weather, the dehazing effect is basically achieved, and the target can be clearly shown from the image, and the effect of the algorithm is basically achieved.

2) YOLO target detection

YOLO was proposed by Redmon et al[9]. The algorithm uses OverFeat to improve the R-CNN network structure and treats target detection as a linear regression problem. The performance of the algorithm makes YOLO the leader in the end-to-end algorithm. YOLO has a simple network structure, fast detection speed, good performance in real-time, and detection accuracy. In addition, the feature of YOLO's simultaneous detection of multiple targets makes it widely used.

We use the Tiny_YOLOv3 detection algorithm as the detection model. YOLOv2 [10] is the author of Redmon's algorithm improvement based on YOLOv1. Under the premise of ensuring the accuracy and speed of the v1 algorithm, the detection accuracy is further improved, and the recall rate and positioning accuracy errors are resolved. In 2018, author Redmon further optimized the YOLOv2 algorithm and proposed YOLOv3 based on previous research. The algorithm optimized the problem of insufficient detection accuracy of YOLOv2 in small targets. The algorithm won the best real-time target of the year at the CVPR International Conference Detection algorithm. The main difference between the YOLOv3-Tiny algorithm and YOLOv3 is that the backbone network is simplified from the Darknet network to a seven-layer CNN network, and the prediction layer is reduced from three to two.

The network model of Yolov3-Tiny is shown in Fig.7.

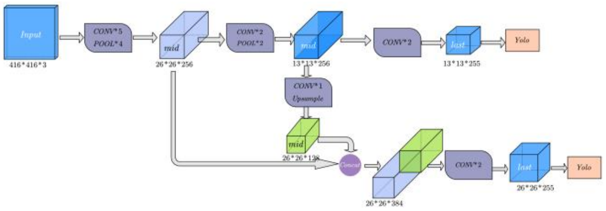


Fig. 7. YOLOv3-Tiny network structure

The input image undergoes seven layers of convolution to extract image features, and after 32 times downsampling, the feature map of the last layer is obtained, and after two convolutions, the output is obtained. After the feature map is also fully convolutional, the purpose is to process images of different sizes. Then the feature map of the seventh layer is up-sampled twice, merged with the feature of the sixth layer, and then subjected to a layer of convolution to obtain the output.

Before performing algorithm detection, target training weight files are required. Many of the YOLO-related data sets open-sourced on the Internet are mainly PASCAL VOC data sets, ImageNet data sets, and COCO data sets. The PASCAL VOC data set has about 10,000 images, each of which is for transportation (Airplanes, cars, trains, etc.), indoor items, and animals. However, to achieve a specific detection purpose, a new data set needs to be constructed based on a specific scene, and the model needs to be retrained.

For the target designed in this paper, it is necessary to re-acquire images and train the model. The simulated training images are collected by the drone's lower camera, taking into account the position change, scale change, rotation change, and height change of the mobile platform during the movement, and similar signs interference and other factors, but also the influence of the camera's field of view, there will be the problem of incomplete field of view of the mobile platform, so in the process of collecting the image of the mobile platform, different heights, distances, and angles are selected. Video capture on mobile platforms. Considering the issues such as camera focal length and moving platform, samples were collected for all of the above issues, and samples were collected as shown in Fig.8.

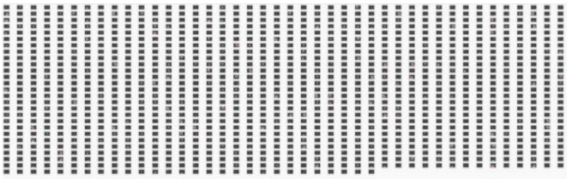


Fig. 8. Samples

In this paper, target images of different heights and angles have been collected in response to the above situation, and 1120 samples have been collected and made into a data set.

The Tiny-YOLOv3 model used in this paper uses the darknet deep learning training framework under the Ubuntu operating system and is integrated into the ROS robot operating system. It mainly loads the trained weight file into the YOLO node, creates its own *.name file, and add the target category, which needs to be the same as the name when creating the training set; modify the *.data file, set the classes category to 1, and the corresponding file path, and finally modify the *.cfg file, set the number of batches, which is the total image input into the training grid at one time Number; set the subdivision size to 4, divide the 64 images into 4 forward calculations, set the learning rate to 0.001, and other parameters by default. The basic parameters of the network model are shown in TABLE I:

TABLE I. YOLO NETWORK PARAMETERS

Main parameters	**Parameter information**	
	Parameter settings	***Parameter Description***
Batch	32	Number of samples in one training
Subdivision	8	Divided into subdivision sub-convolution forward calculation
Width	416	Training image width
Height	416	Training image height
Channels	3	Number of channels for training images
Learning_rate	0.001	Learning rate
Classes	1	The network needs to detect the number of categories
Angle	0	Image angle change
Hue	0.1	Tonal variation range
Saturation	1.5	Saturation size
Exposure	1.5	Exposure value
Steps	400000,450000	Number of iterations when learning rate changes
Filters	16	Number of output feature maps

Finally, for the weight files of different training times, 100 test samples of 640 x 480 are selected for performance testing. Since the initial training set samples are small, after adding some samples, it is found that when the weight of the target is trained to 26000 times, the detected mAP can reach 90%; from the point of view of the recognition accuracy rate, as the model training time increases, the parameters of the model have optimized accordingly, and the detection effect is better. The maximum number of training times in this paper reached 76,000 times. The target recognition rate can reach 90%, which meets the needs of target detection in this paper.

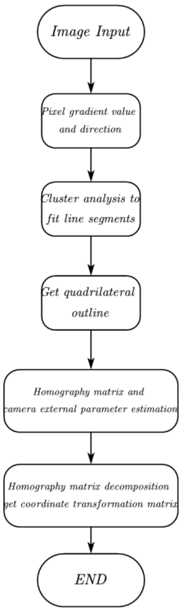


Image Input
Pixel gradient value and direction
Cluster analysis to fit line segments
Get quadrilateral outline
Homography matrix and camera external parameter estimation
Homography matrix decomposition get coordinate transformation matrix
END

Fig. 9. AprilTag identification process

3) AprilTag recognition algorithm

AprilTag[11,12] is a visual reference system designed by the University of Michigan's APRIL Laboratory. Through a specially designed QR code label of known size and the internal reference information of the camera, the coordinates of the QR code label relative to the camera can be estimated. The specific algorithm flow is shown in Fig.9.

From the relationship between the world coordinate system and the image coordinate system, a homography matrix can be used to calculate, the formula is as follows, the theoretical point coordinates are equal to the transformation matrix multiplied by the actual image point coordinates:

$$s\begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix} = H_{3\times 3}\begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{21} & h_{31} \\ h_{12} & h_{22} & h_{32} \\ h_{13} & h_{23} & h_{33} \end{bmatrix}\begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} \tag{1}$$

Expand the formula to get:

$$\begin{cases} su_2 = u_1 h_{11} + v_1 h_{21} + h_{31} \\ sv_2 = u_1 h_{12} + v_1 h_{22} + h_{32} \\ s = u_1 h_{13} + v_1 h_{23} + h_{33} \end{cases} \tag{2}$$

Further transformation, eliminating the scale factor s, and treating the elements of the H matrix as a column vector, the formula can be obtained:

$$\begin{bmatrix} u_1 & v_1 & 1 & 0 & 0 & 0 & -u_1 & -u_2 & -v_1 & u_2 & -u_2 \\ 0 & 0 & 0 & u_1 & v_1 & 1 & -u_1 & v_2 & -v_1 & v_2 & -v_2 \end{bmatrix}\begin{bmatrix} h_{11} \\ h_{21} \\ h_{31} \\ h_{12} \\ h_{22} \\ h_{32} \\ h_{13} \\ h_{23} \\ h_{33} \end{bmatrix} = 0 \tag{3}$$

Because of the influence of the scale factor, the last element of the H matrix can be normalized to 1, so the degree of freedom of the H matrix is 8. To obtain the solution of the formula, at least 4 point pairs are required. The above equation can be obtained by direct linearization, and the homography matrix H can be obtained. At this point, the rotation matrix and translation matrix converted from the world coordinate system to the camera coordinate system are reached, that is, the external parameter matrix of the camera.

Then establish two threshold models (beyond the threshold is considered "1", less than the threshold is considered "0"), one is based on the spatial change of the pixel intensity of "black", and the other is based on the spatial variation of the pixel intensity of "white" Change model:

$$I(x,y) = Ax + Bxy + Cy + D \tag{4}$$

The model has four parameters, which can be solved by least squares regression. The final threshold size is obtained from the average value of the two models' predictions. According to different black and white blocks, the labels are decoded to obtain different code patterns, and the decoded code patterns are compared with the code pattern library stored locally to eliminate the interference of other code patterns and judge whether it is the correct label.

4) AprilTag recognition algorithm Coordinate system transformation

Now that we know the homography matrix H and the camera internal parameters fx, fy, they can be decomposed into a rotation matrix R and a translation matrix T. Combined with the camera external parameter model formula:

$$\begin{bmatrix} X_C \\ Y_C \\ Z_C \\ 1 \end{bmatrix} = \begin{bmatrix} R & T \\ 0_3^T & 1 \end{bmatrix} \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix} \quad (5)$$

The transformation from the camera coordinate system to the world coordinate system can be obtained. Combining the transformation between the camera coordinate system and the drone coordinate system, the AprilTag's pose relative to the drone can be obtained.

In addition to the camera coordinate system and the world coordinate system, there are the AprilTag coordinate system and the robot coordinate system. The coordinate system of AprilTag can be set in the tag.YAML file. Since the bottom camera of the drone is used to identify the target, I hope Yes, the x-axis is the main direction of its movement, so it is necessary to transform the coordinate system of AprilTag, and set the posture of AprilTag as: x:0.0000, y:0.0000, z:0.0, qw:0.707, qx:0.0, qy:0.0, qz:-0.707, the coordinate relationship is shown in Fig.10.

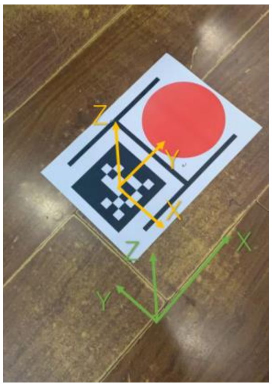


Fig. 10. Coordinate system

The homography matrix and camera parameters have been obtained from the above and can be decomposed to get the rotation matrix R_W^C and the translation matrix t_W^C of the world coordinate system (w) and the camera coordinate system (c) defined by yourself, and the last required It is the transformation from the drone coordinate system(u) to the world coordinate system (w), so the camera coordinate system and the drone coordinate system are transformed again, as shown in the following formula:

$$T_u^w = T_c^w T_u^c = \begin{bmatrix} R_c^w & t_c^w \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_u^c & t_u^w \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} [R_w^c] & -[R_w^c] t_w^c \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_u^c & 0 \\ 0 & 1 \end{bmatrix} (6)$$

$$R_u^w = [R_w^c]^T R_u^c, \ t_u^w = - [R_w^c]^T t_w^c \quad (7)$$

D. Prediction algorithm

Since the cooperative target is on a moving car, if only the detected target position is used as input to the drone, when the drone reaches the target position detected at the previous moment, the target has reached the position of the next moment during this time. To meet the real-time requirements, it is necessary to predict the relative pose at the next moment based on the motion trajectory of the mobile platform. Since the recognition and tracking in this paper are processed separately, the recognition relies on the YOLO box to select the target position and only needs to predict the next moment of movement based on the movement of the mobile platform. This paper selects the Kalman filter algorithm as the prediction algorithm and predicts the relative pose information is used as the control information between the two frames to control the quadcopter.

The state equation of the system is as follows:

$$\begin{cases} X(k) = AX(k-1) + BU(k) + GW(k) \\ Z(k) = HX(k) + V(k) \end{cases} \quad (8)$$

Where X(k) and X(k-1) are the system state vectors at t_k and t_{k-1} respectively, A is the state transition matrix of the system, B is the control matrix of the system, and U(k) is the system input Matrix, G is the interference transition matrix, W(k) is the process noise, its mean is zero, and the covariance matrix is Q; Z(k) is the observation vector, H is the observation matrix, V(k) is the observation noise, and its mean is zero, the covariance matrix is R, and the transmission time interval of two consecutive frames is Δt.

The Kalman filtering process is divided into the following five steps:

(1)Based on the state equation, predict the state of the next moment through the previous moment;

$$X(k|k-1) = Ax(k-1|k-1) + BU(k) \quad (9)$$

(2) After the system state vector is updated, update its corresponding covariance matrix;

$$P(k|k-1) = AP(k-1|k-1)A^T + GQ(k-1)G^T \quad (10)$$

(3) Calculate the Kalman gain $K_g(k)$;

$$K_g(k) = P(k|k-1)H^T/(HP(k|k-1)H^T + R) (11)$$

(4) From the updated covariance matrix, and merge the observation vector;

$$X(k|k) = X(k-1|k-1) + K_g(k)(Z(k) - HX(k|k-1)) \quad (12)$$

(5) Update the covariance matrix of the fused state vector;

$$P(k|k) = (I - K_g(k)H)P(k|k-1) \tag{13}$$

Within the camera's field of view, it is considered that the target is moving in a two-dimensional plane, and Δt is considered to be small. Therefore, it is considered that the system moves linearly and linearly within Δt. Let x(k), y(k) represent the position of the target at t_k, $\dot{x}(k), \dot{y}(k)$ represent the target speed at t_k, according to the kinematics equation:

$$\begin{cases} x(k+1) = x(k) + \dot{x}(k) + \ddot{x}(k)\dfrac{\Delta t^2}{2} \\ y(k+1) = y(k) + \dot{y}(k) + \ddot{y}(k)\dfrac{\Delta t^2}{2} \\ \dot{x}(k+1) = \dot{x}(k) + \ddot{x}(k)\Delta t^2 \\ \dot{y}(k+1) = \dot{y}(k) + \ddot{y}(k)\Delta t^2 \end{cases} \tag{14}$$

The state vector is:

$$X(k) = [\, x(k) \quad y(k) \quad \dot{x}(k) \quad \dot{y}(k)]^T \tag{15}$$

Then there are:

$$X(k-1) = [\, x(k-1) \quad y(k-1) \quad \dot{x}(k-1) \quad \dot{y}(k-1)]^T \tag{16}$$

$$A = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{17}$$

$$G = \begin{bmatrix} \dfrac{\Delta t^2}{2} & 0 \\ 0 & \dfrac{\Delta t^2}{2} \\ \Delta t & 0 \\ 0 & \Delta t \end{bmatrix} \tag{18}$$

Since we want to estimate the position and velocity of the target, the observation vector is set as:

$$Z(k) = [\, x(k) \quad y(k) \quad \dot{x}(k) \quad \dot{y}(k)]^T \tag{19}$$

The observation matrix is:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{20}$$

Obtain the target state equation and observation equation as formula:

$$\begin{cases} X(k) = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} X(k-1) + \begin{bmatrix} \dfrac{\Delta t^2}{2} & 0 \\ 0 & \dfrac{\Delta t^2}{2} \\ \Delta t & 0 \\ 0 & \Delta t \end{bmatrix} W(k-1) \\ Z(k) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} X(k) + V(k) \end{cases} \tag{21}$$

III. Experimental Result

The experimental results of this paper are divided into simulation experiments and real experiments. The processor of the simulation experiment platform is AMD 3900X and the graphics card is NVIDIA GeForce RTX 2060. Simulation scenarios are built in the Gazebo environment under ROS operating system for recognition tests, while real experiments are conducted using a USB camera as the image input and a paper-printed target for recognition tests.

A. Simulation Scene

The simulation experiment in this paper is set up as shown in the Fig11. It mainly uses the car model and the Iris UAV model with a camera at the bottom. The images collected by the lower camera are transmitted to the YOLO detection node through the ROS node.

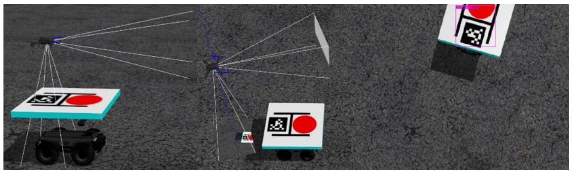
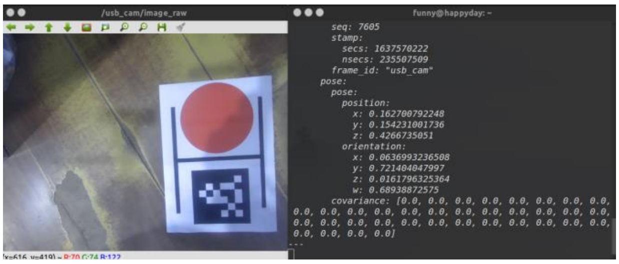


Fig. 11. Experimental result in simulation scene

As can be seen from Fig.11, the model trained in this paper can effectively identify the target. After testing, it can effectively identify the target at different heights, directions and postures, and the detection speed is faster to meet the real-time requirements. According to the YOLO detection BoundingBox information, the position of the target in the image can be obtained, which can be used as the input of the prediction algorithm.

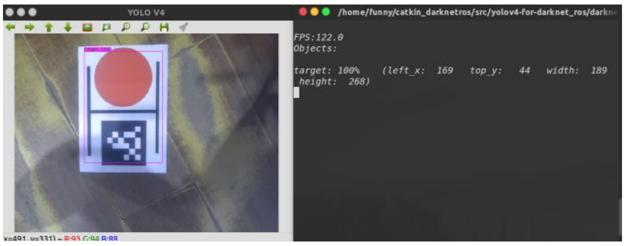
B. Real world Scene

The real experiment in this paper is to print the cooperative target on A4 paper and use the USB camera to collect the image to the detection node. YOLO detection can ignore the impact of light, color difference, etc., and quickly and accurately identify the target. The test results are shown in Fig12 and Fig13.



```
        seq: 7605
        stamp:
          secs: 1637570222
          nsecs: 235507509
        frame_id: "usb_cam"
      pose:
        pose:
          position:
            x: 0.162700792248
            y: 0.154231001736
            z: 0.4266735051
          orientation:
            x: 0.0636993236508
            y: 0.721404047997
            z: 0.0161796325364
            w: 0.68938872575
        covariance: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0]
---
```

Fig. 12. AprilTag recognition result


YOLO V4
/home/funny/catkin_darknetros/src/yolov4-for-darknet_ros/darkn
x=491, y=331) ~ R:93 G:94 B:88

```
FPS:122.0
Objects:

target: 100%    (left_x:  169   top_y:   44   width:  189
 height:  268)
```

Fig. 13. Target experiment results

As can be seen from Fig13, for the actual target recognition, the FPS is high (indicating that the real-time performance is good), and the target of the recognition frame is more accurate, and it has a better effect on the target tracking module. When the height is lower than the set threshold, the recognition effect of AprilTag has also achieved the expected effect. It can be seen by taking the center of the AprilTag target as the coordinate system. At this time, the x and y coordinate values output by the camera are all positive, indicating that it is at the upper left of the target. It conforms to the drone coordinate system we set.

IV. Conclusion And Future Work

From the third part of the experimental results, the recognition effect is good. The targets in the simulation scene and the actual scene can be effectively identified, and the detected coordinates are converted from the AprilTag coordinate system to the drone coordinate system. The relative pose of the UAV relative to the target can be directly obtained. The future work is to put the above-mentioned work on the real UAV to execute and use the output of the coordinates as the control input of the UAV to realize the control process of the entire UAV landing on the mobile platform.

Acknowledgment

This paper was supported by the Major Program of National Natural Science Foundation of China under Grant No. 61690214, Shanghai Science and Technology Action Plan under Grant No.18DZ1204000, 18510745500, 18510750100, 18510730600, Shanghai Aerospace Science and Technology Innovation Fund (SAST) under Grant No. 2019-080, 2019-116 and Shanghai Sailing Program under Grant No. 19YF1420200, and the Natural Science Fund of China (NSFC) under Grant No.51575186

References

[1] Neumann P P, Bennetts V H, Lilienthal A J. Gas source localization with amirco-drone using bio-inspired and ppaper filter-based algorithms. Adv. Robot.2013,27,725-738.

[2] Colorado J, Perez M, Mondragon I. An integrated aerial system for landmine detection: SDR-based Ground Penetrating Radar onboard an autonomous drone.Adv. Robot. 2017,31,791-808.

[3] Tomic T, Schmid K, Lutz P. Toward a fully autonomous UAV: Research platform for indoor and outdoor urban search and rescue. IEEE Robot. Autom. Mag.2012,19,46-56.

[4] Kruijff G, Keshavdas S, Larochelle B, et al. Designing, developing, and depolying systems to support human-robot teams in disaster response. Adv. Robot. 2014,28,1547-1570.

[5] Minaeian S, Liu J, Son Y J. Vision-based target detection and localization via a team of cooperative UAV and UGVs. IEEE Trans, Syst. Man Cybern. Syst.2016,46,1005-1016.

[6] Falanga D, Zanchettin A , Simovic A , et al. Vision-based autonomous quadrotor landing on a moving platform[C]// 2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR). IEEE, 2017.

[7] P. Shao, C. Wu and S. Ma, "Research on key problems in assigned-point recovery of UAV using parachute," 2013 IEEE International Conference of IEEE Region 10, 2013, pp. 1-4.

[8] Shakernia O, Ma Y , Koo T J , et al. Landing an Unmanned Air Vehicle: Vision Based Motion[J]. Asian Journal of Control, 2010, 1(3):128-145.

[9] Exner D, Bruns E, Kurz D, et al. Fast and robust CAMShift tracking[C]. 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, 2010: 9-16.

[10] Redmon J, Farhadi A. YOLO9000: Better, Faster, Stronger[C]. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017: 6517-6525.

[11] Olson E. AprilTag: A robust and flexible visual fiducial system[C]//IEEE International Conference on Robotics and Automation. 2011: 3400-3407.

[12] Krogius M, Haggenmiller A, Olson E. Flexible Layouts for Fiducial Tags[C]//IEEE International Conference on Intelligent Robots and Systems. 2019: 1898-1903.