Team Control Number

201972054

Problem Chosen

B

Summary

With the continuous advancement of technology, the application of drones in social life has become more and more extensive. However, for the novice of the first drone, it is difficult to master the shooting of the drone. How to use the drone to take a satisfactory photo is one of the problems to be solved. In this paper, the UAV is used as the experimental platform to study the flight altitude, speed. The main research work and conclusions of this paper are as follows:

1) This paper uses the geometric relationship between the images to calculate the UAV position feature value, and uses the ground object resolution as a reference. In this paper, the camera's pose algorithm based on single image is used to derive the optimal flight range of the drone.

2) In this paper, the influence of shooting angle on the imaging of ground objects is studied under the condition that the flying height of the drone is fixed and the flying speed is constant. The optimization function of flight altitude, speed and shooting angle is established by using the essential matrix decomposition algorithm. Under the clarity requirements of the image of the ground object, the optimal range of the shooting angle of the drone is estimated.

3) The dynamic optimization process of the UAV shooting angle was studied. Firstly, the trajectory is represented by the trajectory sequence, the motion structure of the trajectory is described, the essential matrix function of flight height, speed and shooting angle is segmented to obtain the drone's shooting. Calculate the change law of the shooting angle of the drone.

4) For the situation that the UAV encounters obstacles during flight, the artificial potential field algorithm is used for path planning and obstacle avoidance. Combined with the linear flight characteristics of the UAV, the UAV shooting angle is determined. The effectiveness of the algorithm is verified by simulation experiments.

Key word: UAV, Pose estimation, Autonomous obstacle avoidance, Path planning

Content

1. Problem Retelling

Task1: Assume that the drone is not affected by external factors such as wind direction and humidity during flight, and the flight speed and shooting angle are constant, and the drone performs aerial photographing in a straight-line flight mode for an area with a certain width. What is the height range of the flying drone performing this aerial photographing and propose the best aerial photographing height at the specified shooting angle.

Task2: Assume that the UAV's flight altitude is fixed and the flight speed is constant and the UAV needs to perform more accurate image acquisition on the actual ground conditions and please calculate the optimal range of the UAV shooting angle.

Task3: Assume that the flying height of the drone is fixed and the flying speed is constant and the drone is collecting the trajectory of a certain moving object on the ground. Please calculate the changing pattern of the UAV shooting angle.

Task4: If the drone encounters obstacles such as signal towers, telephone poles, houses, etc. during its flight, please calculate the optimal flight speed, flight altitude and the shooting angle, and provide the relevant data simulation experiment.

2. Problem Analysis

2.1 Analysis of Question One

First of all, this topic mainly examines the research on the relationship between the construction of UAV flight height, shooting angle and ground object imaging. Here we can interpret the optimal flight height of UAV as the best height that can distinguish the ground object. Because what we take with the UAV is a picture, which is interpreted through the picture, and the picture is presented in the picture It is the size of the resolution. According to the composition relationship, we can know that the flight height should be designed according to the requirements of the ground resolution of the project. This situation is applicable to the case that the shooting angle is directly above. When the shooting angle changes, the composition relationship changes. The ground resolution changes at different positions from the projection point. The farther the distance is, the smaller the resolution is. We need to establish the included angle relationship between the target position and the flight height, the size of the objective lens, and then use the height design above to adjust.

2.2 Analysis of Question Two

We can express this question as an optimization problem. The optimization goal is the best shooting angle. The objective function is that the ground object imaging is the most clear at this shooting angle. That is to say, the high resolution constraint obtained is that the altitude and speed of flight are fixed. Therefore, first of all, they should be established under the influence of the altitude and speed of flight and the shooting angle. Goal of the resolution is that the optimization method can choose linear programming.

2.3 Analysis of Question Three

According to the description of task three, it can be concluded that this is a dynamic optimization problem. The so-called dynamic optimization refers to the dynamic cutting of the vehicle's motion track. The so-called dynamic cutting refers to that the vehicle's running track can be cut into several segments according to the different motion laws, and then each segment is established under the influence of the flight altitude and speed, as well as the shooting angle, and they work together The goal of the resolution used is to get the maximum optimization method.

2.4 Analysis of Question Four

This is also one of the optimization categories. Here we can choose the combination optimization idea of graph theory and genetic algorithm to construct. First, we use graph theory algorithm to calibrate each major obstacle point, and record the important observation and record objects in the aerial photography area. Our ultimate goal is to avoid the above obstacles and shoot the high-resolution object resolution at the same time. Then our optimization can According to this idea, first of all, the objective function is still to obtain high-resolution ground object image. Because of the flight altitude and speed, the other objective function is to take a shorter time, and the constraint condition is to cross the obstacles, that is, to avoid the obstacles in the graph theory, plus other constraints, such as area size, electric quantity, etc., the constraint method can choose the multi-objective planning model and select The genetic algorithm can be used to solve the multi-objective optimization model.

3. Symbol and Assumptions

3.1 Symbol Description

Symbol	Symbol Description
k_1, k_2	Distortion coefficient
p_1, p_2	Eccentric distortion coefficient
h	Camera focal length
E	Rigid body transformation matrix
R_0, t_0	Initial pose parameter

μ	Damping factor
p_i^w	Image point coordinates projected onto a normal plane
t	Translation vector
m_t	Track target number
F_i^b	Single rotor rotation lift
M_{sum}	Rotational motion
X_f	UAV end state
Xob_s	Obstacle position
α_{pitch}	Top view angle of lens

3.2 Fundamental Assumptions

Hypothesis1: Assume that the drone is not affected by external factors such as wind direction, humidity, and weather changes during the flight.

Hypothesis2: Assume that the drone camera has no quality problems. The shutter is turned on and off instantly, no delay, normal focus and proper exposure.

Hypothesis3: Assume that there are no abnormalities such as high frequency jitter, GPS positioning system interference and low battery power during the flight of the drone.

4. Task1: The Problem of Aerial Photographing Height of UAV

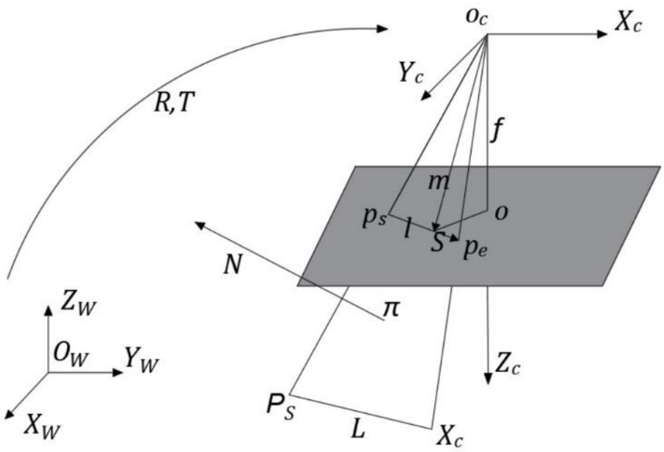


Fig. 4.1 Coordinate system selection reference map

As shown in Fig.4.1, the rectangular coordinate system O_w-$X_wY_wZ_w$ is the world coordinate system. L is a space line, $P_{s,}$ P_e is the two endpoints of L, and its parameters in the world coordinate system are known. The rectangular coordinate system O_c-$X_cY_cZ_c$ is the camera coordinate system. Its origin is based on the camera optical center. The Z_c axis points to the front of the camera along the optical axis. R_t is the rotation matrix and translation vector of the camera coordinate system relative to the world coordinate system. The projection image line of L in the image plane is l, and the line parameters of l in the camera coordinate system can be obtained by line extraction. P_s, P_e are the two endpoints of l.π represents the plane determined by the optical center O_C of the camera and the line l. The problem discussed in this section is how to estimate the altitude range f_1-f_2 and the optimal flight altitude f of UAV aerial photography from multiple known space lines and corresponding image projections when the UAV aerial photography is not affected by wind damping and humidity factors.

4.1 Calculation of the Optimal Height of the Image Taken by UAV

In traditional photogrammetry, "relative altitude=lens focal length / photographic scale", but this method of determining altitude is not suitable for digital camera, because the imaging principle of digital camera is different from that of film type aerial camera, because the digital camera acquires discrete pixels, because the scale of image is not everywhere due to the relief of terrain and tilt of image. Therefore, GSD[1] is often used to calculate the altitude when the UAV is used for tilt photogrammetry. In order to obtain the GSD that meets the user's requirements, it is necessary to control the camera's object distance and then control the altitude. The calculation formula of object distance D is:

$$D = \frac{GSD \cdot n_l}{2\tan\frac{FOV}{2}} \tag{4-1}$$

n_l- Number of pixels on long side; FOV- Lens field angle

The route can be divided into fixed altitude route, variable altitude route and surrounding route. The fixed altitude route refers to the flat terrain in the survey area, and the relative altitude of the aircraft is always maintained during flight; the variable altitude route refers to the relative altitude of the aircraft will rise with the rise of terrain when photographing the sloping terrain; the surrounding route refers to the relative altitude of the aircraft will rise with the rise of terrain when photographing a large terrain When the overhead angle is used for surround shooting, the calculation method of the altitude is different for different types of routes.

For the fixed altitude route, the relative altitude of the aircraft is equal to the object distance, that is:

$$f_{\text{fixed height}}=D \tag{4-2}$$

For the variable altitude route, the relative altitude of the aircraft is related to the slope of the terrain fluctuation, and the calculation formula is as follows:

$$f_{\text{increase}}=L \cdot \tan\alpha + D \cdot \cos\alpha \tag{4-3}$$

L- Horizontal distance between UAV and takeoff; α- Slope angle.

For the circle route, the relative channel of the aircraft is related to the circle radius and pitch angle of the aircraft, and the calculation formula is as follows:

$$f_{\text{surround}}=D \cdot \sin\alpha_{pitch} \tag{4-4}$$

α_{pitch}- Top view angle of lens.

As shown in Fig. 1,the projection equation of any point P on the space straight line L can be obtained from the projection model of the camera, as shown in (4-5):

$$\begin{cases} x = \dfrac{f(r_{11}X + r_{12}Y + t_x)}{r_{31}X + r_{32}Y + t_z} \\ y = \dfrac{f(r_{21}X + r_{22}Y + t_y)}{r_{31}X + r_{32}Y + t_z} \end{cases} \tag{4-5}$$

Where $P=\left[X,Y,0\right]^T$ and $p=\left[x,y\right]^T$ are the projection points of space P in the image:

$$R=\begin{bmatrix} r_1 & r_2 & r_3 \end{bmatrix}=\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}, \quad t=\begin{bmatrix} t_x & t_y & t_z \end{bmatrix} \tag{4-6}$$

The projection line l of the space line L on the image plane can be expressed by equation (4-7):

$$x\cos\theta^c + y\sin\theta^c - \rho^c = 0 \tag{4-7}$$

According to the distance preserving property of Euclidean space, formula (4-13) can be obtained:

$$\left\| P_e^c - P_s^c \right\| = \left\| P_e - P_s \right\| \tag{4-8}$$

Where P_S, P_E are the two endpoints of line L, P_s^c and P_e^c are the coordinates of the two endpoints of line L in the camera coordinate system.

In the field work of aerial photography, the landform of the survey area is not very ideal. Due to the large fluctuation of the ground surface. In order to ensure that the fluctuation range of GSD is small, it is better to be near rather than far in the selection of aerial height and photographic object distance.

4.2 Height Range Calculation of UAV

The basic idea of orthogonal iterative algorithm based on line feature is consistent with that based on point feature. The orthogonal iterative algorithm based on line features takes the coplanar error of the line as the objective function, and obtains the optimal pose solution satisfying the objective function minimization through the orthogonal iterative process. The results of orthogonal iterative pose estimation ensure the orthogonality of rotation matrix, so it has high accuracy[2].

As shown in Fig.4.1, N is the unit normal vector of plane π under the camera coordinate system, and n is the unit direction vector of space straight line L. The vectors of O_{cP} and n in camera coordinate system can be expressed as $RP+t$ and R_n, where P is any point on line L. $RP+t$ is on the plane π, which is orthogonal to N, so formula (4-15) can be obtained:

$$N^T(RP+t)=0\,,N^T R\mathrm{n}=0 \tag{4-9}$$

Any vector in a space plane can be represented by a set of standard orthogonal bases of the plane. Let the vector ε be any vector on the plane π,then there is formula (4-10):

$$\varepsilon=(\varepsilon \mathrm{s})\mathrm{s}^T+(\varepsilon \mathrm{m})\mathrm{m}^T=(ss^T+\mathrm{mm}^T)\varepsilon \tag{4-10}$$

Let $C= s_{st} + m_{mt}$, (s,m) be a set of standard orthogonal basis of plane π.As shown in Fig.1,a set of expressions (4-11) of orthogonal basis can be obtained by using the image line l located on the plane π:

$$s=\begin{bmatrix}\sin\theta_c & -\cos\theta_c & 0\end{bmatrix},m=\frac{1}{\sqrt{f^2+\rho_c^2}}\begin{bmatrix}\rho_c\cos\theta_c & \rho_c\sin\theta_c & f\end{bmatrix}^T \tag{4-11}$$

Where [θ_c , ρ_c] is the coordinate of the image line l. Therefore, the vector $RP+ t$ on the plane π can be expressed by formula (4-12):

$$E(R,t)=\sum_{i=1}^{n}\|e_i\|^2=\sum_{i=1}^{n}\|(I-R)(RP_i+t)\|^2\ ,\quad E(R)=\sum_{i=1}^{n}\|e_i\|^2=\sum_{i=1}^{n}\|(I-C_i)R\mathrm{n}_i)\|^2 \tag{4-12}$$

When rank (B)=2,

$$S=\begin{cases} I, & \det(U)\det(V)=1 \\ diag(1,1,-1), & \det(U)\det(V)=-1 \end{cases} \tag{4-13}$$

The optimal solution of F can be obtained by the iterative process. Finally, h can be determined by equation (4-14):

$$t=(\sum_{i=1}^{n}(I-C_i))^{-1}\sum_{i=1}^{n}(I-C_i)RP_i \tag{4-14}$$

5. Task1: UAV Shooting Angle Estimation based on Multi-scale Information

In the process of UAV's flight, the image acquired by camera is mostly unknown scene information under actual ground conditions[3]. In order to ensure the accuracy of the captured image, enhance the robustness of the collected image, and reduce the influence of light, cloud, terrain and other factors on the camera imaging. In this chapter, based on the visual positioning in the location environment, the optimal camera angle of UAV is deduced through the geometric relationship between images[4].

5.1 Pose Estimation based on Essential Matrix SVD Decomposition

The essential matrix E is determined by the motion parameters R and α of the camera, independent of the internal parameters, E=[α]Tr, after singular value decomposition of the essential matrix, the motion parameter R and α with one scale factor difference can be obtained.

$$E = U\begin{bmatrix} \|t\| & 0 & 0 \\ 0 & \|t\| & 0 \\ 0 & 0 & \|t\| \end{bmatrix} V = U\begin{bmatrix} \|t\| & 0 & 0 \\ 0 & \|t\| & 0 \\ 0 & 0 & \|t\| \end{bmatrix}\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & \pm 1 \end{bmatrix} \tag{5-1}$$

As shown in Fig. 5.1, m, m' and m'' are the shooting points of space point m in positions 1, 2 and 3 of UAV respectively. Given three photographing position matrices corresponding to three frames of image, position 1 matrix $P=K[I|O]$, position 2 matrix $P'=K[R'|t']$ and position 3 matrix $P''=K[R''|t'']$, where I is the 3×3 unit matrix and K is the internal parameter matrix of camera[5].

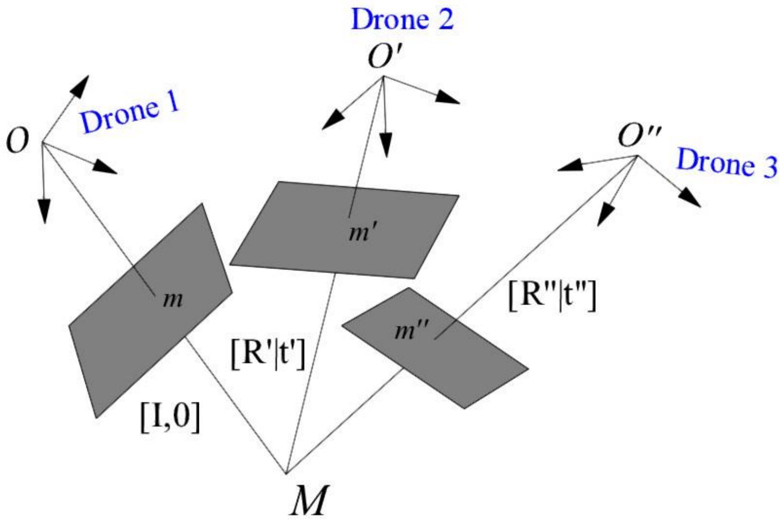


Fig. 5.1 Three-view geometry

5.2 Estimation of Relative Shooting Angle based on Optimal Decomposition

The relative shooting angle of no one else is determined by establishing the objective function and iterating the objective function for many times. This method avoids a large number of matrix operations in the existing algorithm, and obtains the unique solution of UAV's best shooting angle through iterative estimation. The formula (5-2) is obtained from the properties of the essential matrix:

$$EE^T = -[t]_x^2 \tag{5-2}$$

Expand both ends of formula (5-3):

$$EE^T = \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix},\ [t]_x^2 = \begin{bmatrix} t_y^2 + t_z^2 & t_x t_y & t_x t_z \\ t_x t_y & t_x^2 + t_y^2 & t_y t_z \\ t_x t_z & t_y t_z & t_x^2 + t_y^2 \end{bmatrix} \tag{5-3}$$

According to the translation vector of UAV, the angle matrix R is used as the unknown value. The matching feature points are established by the geometric relationship between the matching feature points of two views, and the solution of R satisfies the formula (5-4).

$$R = \arg\min_{R} \frac{1}{2}\sum_{i=1}^{n}\left\|\hat{m}_i^{'T}[t]_x R\hat{m}_i\right\|^2 \tag{5-4}$$

$R = R_0 R(\mu)$ and $d(\mu) = E - [t]_x R_0 R(\mu)$, R_0 be the initial value. The $d(\mu)$ is linearized: Let the Jacobian matrix be $J = \nabla d$, and derive the above formula to get the Jacobian matrix J_i.

$$J = \begin{bmatrix} Q1^1 & Q2^1 & Q3^1 \\ Q1^2 & Q2^2 & Q3^2 \\ Q1^3 & Q2^3 & Q3^3 \end{bmatrix} \tag{5-5}$$

Among them, the superscript i=1,2,3 represents the i column of the matrix. Equation (5-8) is transformed into the least squares problem, and the optimal solution of the angle R of the drone is obtained by iterative calculation.

5.3 Unmanned Shooting Angle Estimation Experiment based on Relative Geometric Relationship

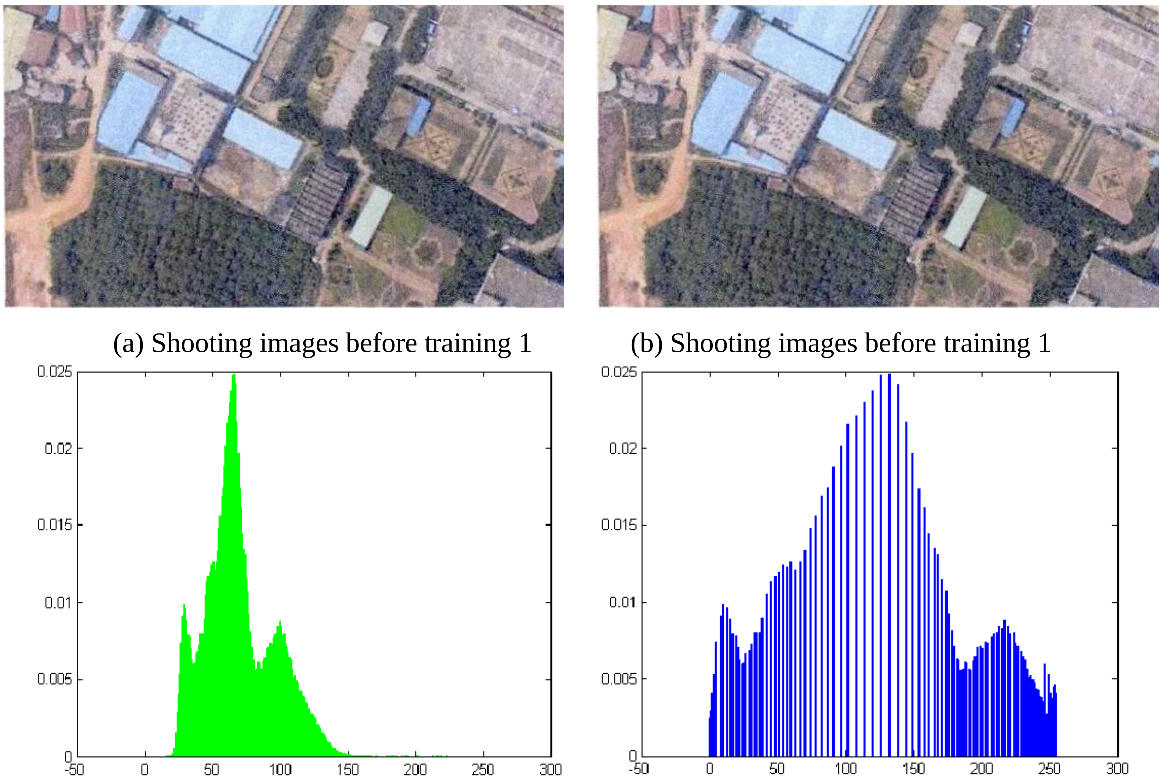
The internal parameters of the camera are limited: f=800, and the principal point coordinates under actual ground conditions are $u_0 = v_0 = 0$. Through the determination of the objective function and the derivation of the pose parameter Jacobian matrix during the iteration, the steps of the whole shooting angle experiment are shown in Tab. 5.1.

Table 5.1 Relative height direct estimation process based on the height Lie group

Relative height direct estimation algorithm based on homography mapping error minimization and high Lie group representation	
Step 1	Determine the initial value of the drone posture $E_0 = \begin{bmatrix} I & 0 \\ 0 & 1 \end{bmatrix}$
Step 2	Start iteration, the number of iterations is k， k=1
Step 3	Calculate the Jacobian matrix J and the objective function residual e, and then get the residual sum $g^k = sum(e)$
Step 4	Calculate the parameters to obtain the pose update matrix $M(\hat{u})$
Step 5	Number of iterations k=k+1, If k>1, judge whether g^k is greater than g^{k+1}, if g^k is less than g^{k+1} and the number of iterations is less than the set number of times, go to Step 3, otherwise go to Step 6
Step 6	End of iteration, output pose optimal solution E=E_0

In order to verify the validity of the algorithm, the effects of a set of 15 frames are processed. The image reference frame is shown in Fig. 5.2. The SIFT and RANSAC operators are used to extract and match the feature points for the remaining 14 frames and the reference frame respectively[6]. Then the essence is used. The matrix SVD decomposition algorithm performs pose calculation, and the shooting attitude angle sub-controller is:

$$\begin{bmatrix} p_d \\ q_d \\ r_d \end{bmatrix} = \begin{bmatrix} k_{P\phi}(\phi_d - \phi) + k_{1\phi}\int(\phi_d - \phi)dt + k_{D\phi}(\dot{\phi}_d - \dot{\phi}) \\ k_{P\theta}(\theta_d - \theta) + k_{1\theta}\int(\theta_d - \theta)dt + k_{D\theta}(\dot{\theta}_d - \dot{\theta}) \\ k_{P\psi}(\psi_d - \psi) + k_{1\psi}\int(\psi_d - \psi)dt + k_{D\psi}(\dot{\psi}_d - \dot{\psi}) \end{bmatrix} \tag{5-6}$$



(a) Shooting images before training 1 (b) Shooting images before training 1

(c) Shooting image histogram before training (d) Shooting image histogram after training

Fig. 5.2 Angle tracking error before and after training

6. Task 3: Dynamic Optimization Algorithm for Shooting Angle of Drone

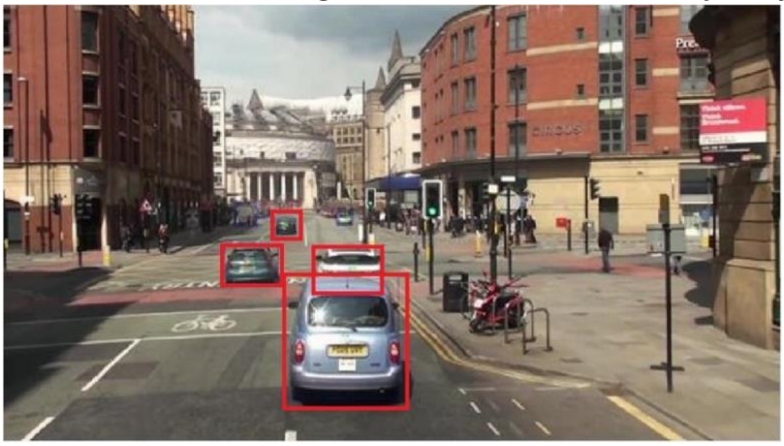
In the case that the flying height of the drone is fixed and the flight speed is constant, the trajectory of a moving car on the ground is detected by changing the shooting angle of the drone. In order to ensure the clarity of the captured image and the safety of the drone flight[7]. The dynamic optimization method of the drone shooting angle of deep learning: Firstly, the convolutional neural network detector is used to predict the key position of the collected image containing the target, and the key point based on the prediction result is fed back to the control system of the drone, and the control is performed. The system adjusts the angle of the drone to get the best shot quality.

6.1·Dynamic Running Track Segmentation

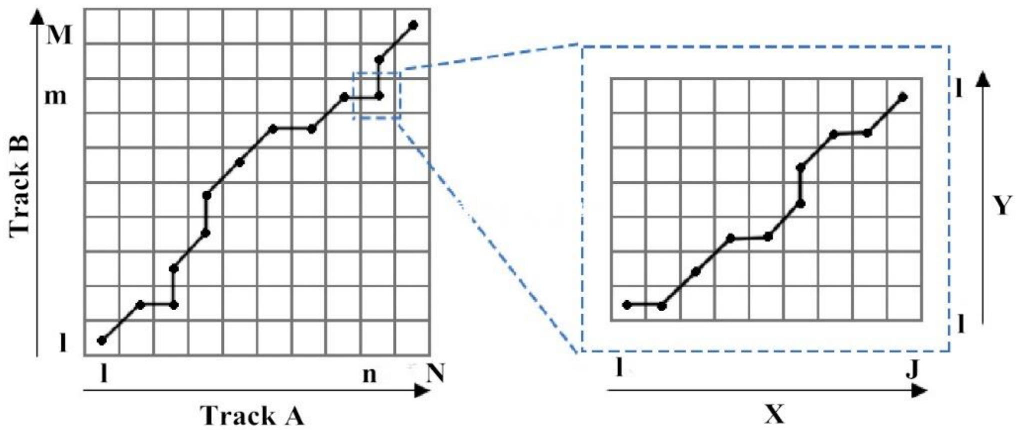
The motion trajectory refers to the moving path of the object, so the motion process can be described by the moving direction of the object. For a trajectory segment of a straight type, since the trajectory segment does not turn at any time, the direction of motion at each moment of the trajectory segment does not change. The angle of change $p(t)$ of the trajectory at the point of motion of point $p(t)$ can be calculated by equation (6-1).

$$\alpha(t) = a\cos\left(\frac{V_f(t)\cdot V_b(t)}{\|V_f(t)\|\|V_b(t)\|}\right) \tag{6-1}$$

Aiming at the problem of unknown vehicle trajectory, the target detection-based matching algorithm is used to dynamically segment the running trajectory of the car[8]. The metric combines the detection target candidate frame and the target center coordinates, and combines the affine transformation to segment the vehicle behavior trajectory.



(a) Automobile motion track image acquisition



(b) Dynamic path segmentation

Fig. 6.1 Car dynamic track segmentation

6.2 Calculation Method of Flight Angle of Drone based on Key Point Coordinates

The key points of the detection process are used to locate the four key points t_1, t_2, t_3, and t_4 of the drone using the deep convolutional neural network shown in Fig. 7. The network has 8 convolutional layers, and 2 fully connected layers have a total of 1200 neuron nodes. The size of the convolutional layer is 5×5[9]. After the convolution calculation, the corrected linear unit is input, and the output layer outputs the eight-dimensional data to jointly predict the position coordinates of the four points (t_1, t_2, t_3, t_4), thereby obtaining the axis of the drone accurate positioning.

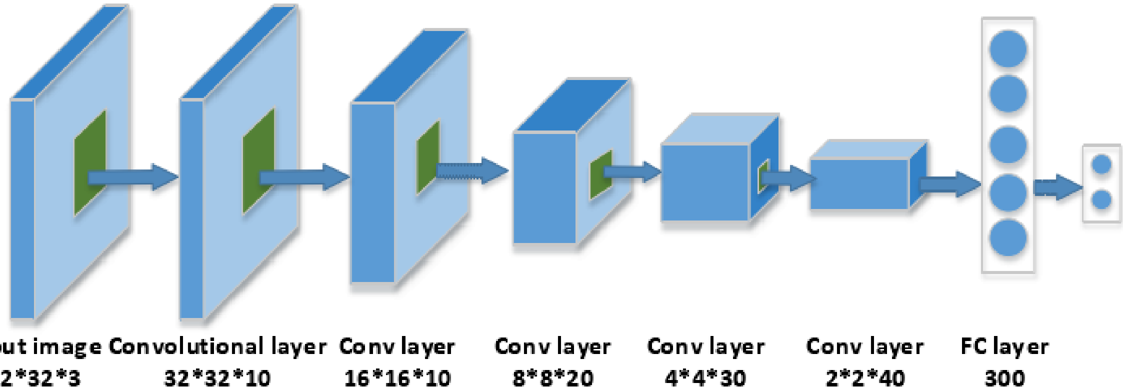


Fig. 6.2 Key point convolutional neural network structure

For a point in space, the coordinate values in the camera coordinate system are known (X_c, Y_c, Z_c). According to the similar triangle theorem, the relationship with the coordinates of the object can be expressed as:

$$\begin{cases} \dfrac{x-C_x}{F_x} = \dfrac{X_c}{Z_c} \\ \dfrac{y-C_y}{F_y} = \dfrac{Y_c}{Z_c} \end{cases} \tag{6-2}$$

Among: i—Actually photographing the sequence of key points 1-4 of the drone on the ground; x_i, y_i—Corresponding to the coordinates of the key point on the image plane; X_{ci}, Y_{ci}, Z_{ci}—Camera angle corresponding to the key point； Similarly, the solution formula of the roll angle AY and the pitch angle AX can be known.

6.3·Unmanned Aerial Vision Binocular Visual Position Measurement and Calculation Method

The target detection problem in the UAV[10] pose detection process is a complex and challenging task. This task needs to combine image classification and positioning. Accurate target recognition and positioning can be the key to determining the position coordinates of the UAV. The role of the UAV's flight angle is accurately estimated, thus solving the two problems of "what is the image taken" and "where is the shooting clearer". This paper uses the depth learning-based Faster-RCNN algorithm for the airborne drone.

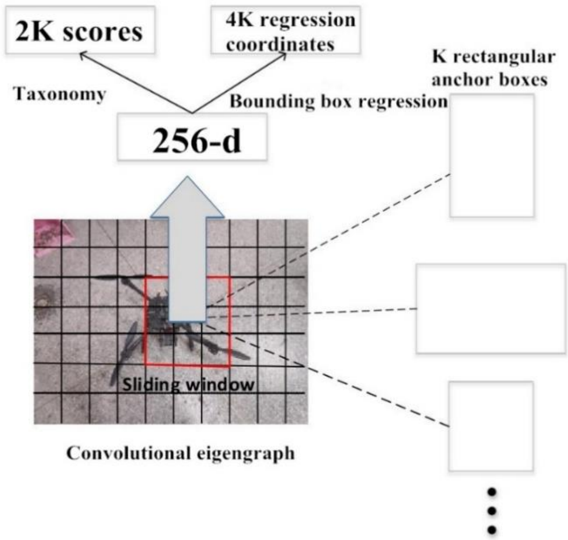


Fig.6.3 RPN network structure

Through the three-dimensional coordinates of the spatial point, the cosine theorem (6-2) can be used to find the angle between the left camera and the drone connection V_1, the left camera and the right camera connection V_2, and the target and the double The calculation formula for the regular change between the angles of the camera shooting is shown in (6-3).

$$\cos\theta = \frac{v_1 v_2}{\|v_1\|\|v_2\|} \tag{6-3}$$

The calibrated camera parameters are substituted into the P4P algorithm to solve the target pose[11]. Each shot can be used to solve a set of flight attitude angles based on the automatically detected key points into the calibrated P4P model. The rotation angle around the Z axis is called the heading angle. The rotation angle around the X axis is called the elevation angle. The rotation angle around the Y axis is called the horizontal angle. The flight attitude measurement results are shown in Table 2.

Table 2 Attitude Measurement Results

Actual value	*Relative attitude measurement in yaw direction*		*Relative attitude measurement in pitch direction*		*Relative attitude measurement in roll direction*	
l(°)	*Measured value l(°)*	*Error l(°)*	*Measured value l(°)*	*Error l(°)*	*Measured value l(°)*	*Error l(°)*
0	0.69	0.69	1.47	1.47	1.32	1.32
10	11.44	1.44	9.1	-0.9	9.13	-0.87
20	20.16	0.16	19.6	-0.4	20.32	0.32
30	30.57	0.57	31.21	1.21	30.39	0.39
40	39.59	-0.41	39.06	-0.94	40.02	0.02

7. Task4: Algorithm and Simulation Experiment of Aerial Photography Obstacle Avoidance Path for UAV

7.1 Obstacle Avoidance Path Planning Algorithm based on Artificial potential field method

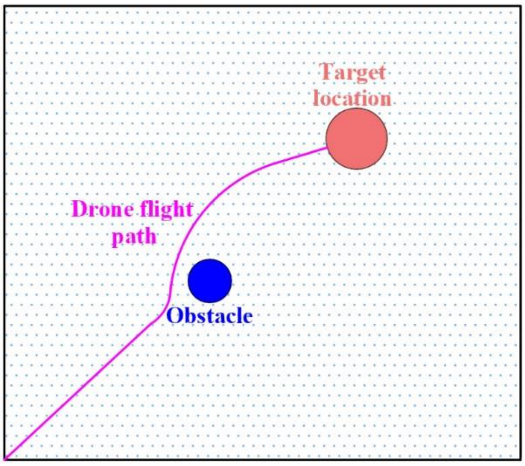


Figure 7.1 Schematic diagram of the aerial obstacle avoidance path of the drone

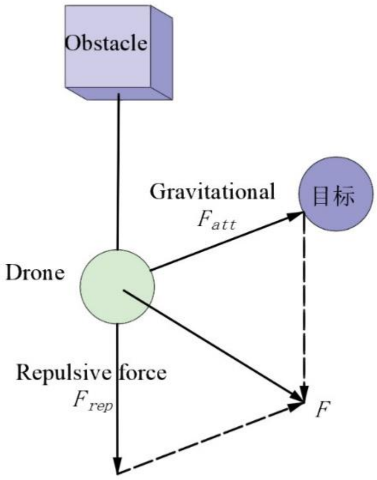


Figure 7.2 Force direction of the UAV algorithm in the artificial potential field

The artificial potential field algorithm establishes the virtual repulsion and gravity respectively by the obstacle and the target point. In the process of motion, the target point always attracts the moving body. The obstacle in the path generates the repulsive force to the moving body, and the artificial potential field is utilized. The path of the law planning is usually smoother[12]. The algorithm only needs to transform the distance information into the corresponding potential field function, but in the case of more obstacles, the planned path is easy to fall into the local optimum. If the drone encounters obstacles such as signal towers, utility poles, houses, etc. during the flight, as shown in Fig. 7.1. The virtual force of the UAV in the artificial potential field algorithm is shown in Figure 7.2. The potential field function is designed as follows:

(1) Gravitational potential field function

The UAV is regarded as a mass point, and the position is defined. The force field can be differentiated. The target point generates a gravitational potential field for the UAV. The magnitude of the potential field energy is positively related to the distance between the UAV and the target point. The potential energy is proportional to the square of the distance. The design potential energy function equation is:

$$U_{att}(q,a,v)X = \frac{1}{2}k_{att}d^2(X,X_g) \quad (4\text{-}1)$$

The gravitational force in the direction of the UAV movement is the negative gradient of the gravitational potential field, and the corresponding gravitational function is:

$$F_{att}(q,a,v) = -\nabla U_{att}(q,a,v) \quad (4\text{-}2)$$

Substituting equation (4-2) into the obtainable gravitational function is:

$$F_{att}(q,a,v) = -k\left|X - X_g\right| \quad (4\text{-}3)$$

In the formula, k_{att} represents the positive proportional gain coefficient of the gravitational field, X represents the position of the drone, X_g represents the position of the flying target point, d represents the distance from X to the position of the target point X_g, and represents the Euclidean distance of X to the target point X_g. Analysis of the gravitational function can be seen that the gravitational force is oriented by the drone to the target point, and its size is related to its distance from the target point. As the distance increases, the attraction of the target point to the drone is greater. The k_{att} gain coefficient is related to the obstacle avoidance effect. When the coefficient k_{att} is large, the obstacle avoidance effect is obvious.

(2) Repulsive potential field function

In the motion space, the obstacle generates a repulsion potential field function for the drone. Within a certain range, the potential energy is inversely related to the distance between the drone and the obstacle. Therefore, the magnitude and distance of the repulsion potential energy can be set. In inverse proportion, the design potential energy function equation is:

$$U_{rep}(q,a,v) = \begin{cases} \frac{1}{2}k_{ep}\left(\frac{1}{X - X_{obs}} - \frac{1}{\rho_0}\right), X - X_{obs} \le \rho_0 \\ 0 \qquad\qquad , X - X_{obs} > \rho_0 \end{cases} \quad (4\text{-}4)$$

Within the range of the obstacle's influence, the repulsion is along the negative gradient direction of the repulsion potential field function:

$$F_{rep}(q,a,v) = -\nabla U_{rep}(q,a,v) \quad (4\text{-}5)$$

Substituting equation (3.34) into the available repulsion function is:

$$F_{rep}(q,a,v) = \begin{cases} k_{ep}\left(\frac{1}{X - X_{obs}} - \frac{1}{\rho_0}\right)\frac{1}{\left(X - X_{obs}\right)^2}\frac{\partial\left(X - X_{obs}\right)}{\partial X}, X - X_{obs} \le \rho_0 \\ 0 \qquad\qquad , X - X_{obs} > \rho_0 \end{cases} \quad (4\text{-}6)$$

Therefore, the total field of the drone in the environment is the superposition field of the gravitational field and the repulsive field to obtain the total potential:

$$U(q,a,v) = U_{att}(q,a,v) + U_{rep}(q,a,v) \quad (4\text{-}7)$$

The combined force of the drone is:

$$F(q,a,v)=F_{att}(q,a,v)+F_{rep}(q,a,v) \quad (4\text{-}8)$$

(3) Description of obstacle avoidance problems

In order to facilitate the calculation of the obstacle avoidance algorithm, the algorithm regards the drone as a rigid body structure, and uses a single integral dynamic system to represent the dynamic system of the rotor less drone. The state space expression can be expressed as:

$$\dot{X}=AX+BU \quad (4\text{-}9)$$

In the formula: A=0, B=1, X=x, U=u, X represents the position coordinates of the drone, and U represents the control input of the drone.

In the formula: $X_f \in R$ is indicates the end state of the drone. Since the target point is known, the end state is constant; U_f is the end control input. When the drone reaches the target point, it will not continue to move, so the end control input U_f is zero. The dynamic error equation can be further written as:

$$\dot{\tilde{X}}=A\tilde{X}+BU \quad (4\text{-}11)$$

(4) UAV artificial potential field algorithm flow

According to the above ideas, the specific implementation process of the algorithm program is as follows:

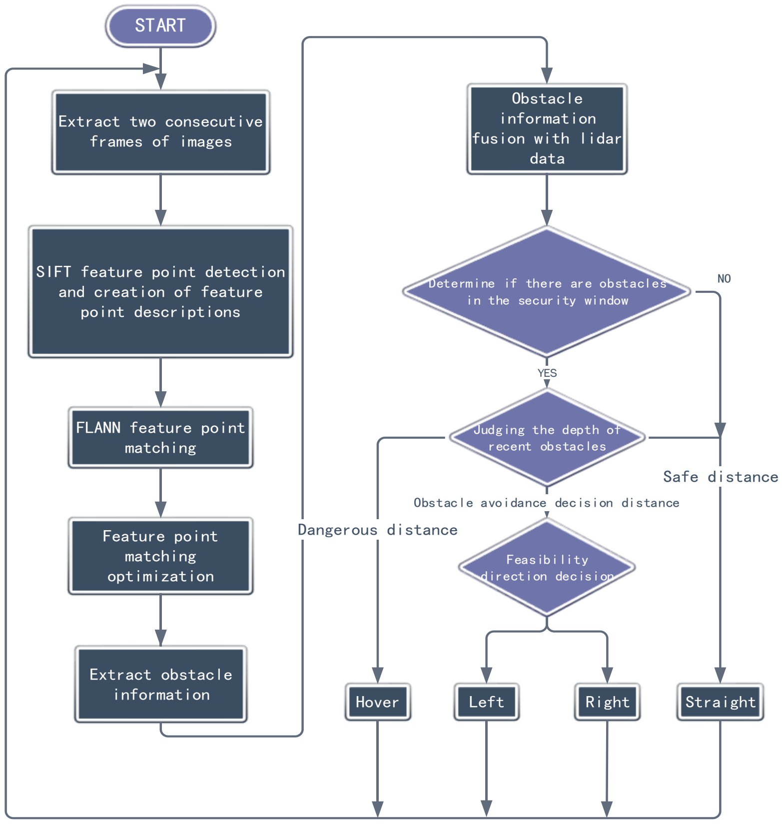


Fig. 7.3 Flow chart of artificial potential field method

According to the modified gravitational field function, the modified repulsion field function and the method of solving the local minimum value proposed in this section, the above-mentioned obstacle avoidance strategy is simulated in the MATLABR 2016a environment. Whether the drone can successfully fly to the target location if a certain number of obstacles are verified separately. The starting position q= (0,0) of the drone starts to start moving toward the target point (10,10) with acceleration and zero speed of motion. Set the artificial potential field function to have a repulsion gain coefficient η of 4, a gravitational field gain coefficient k of 12, a velocity gain coefficient of 2, and an acceleration gain factor of 1. The simulation results are shown in Figure 7.4.

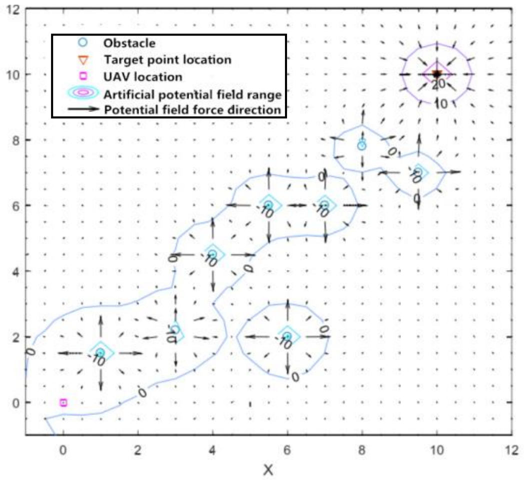


Figure 7.4 Improved artificial potential field method Gravitational potential field and repulsion potential field vector

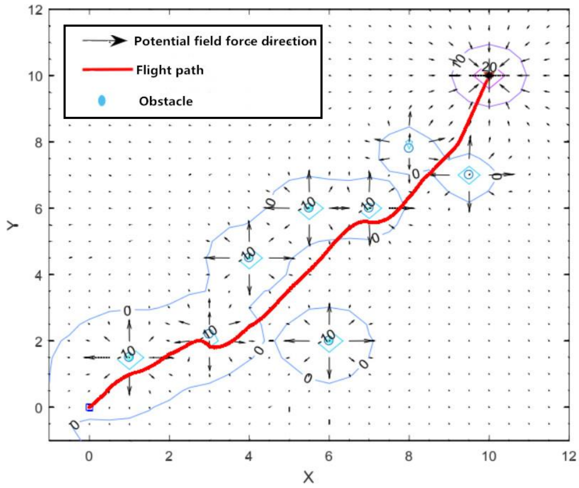


Figure 7.5 Artificial potential field method path planning and obstacle avoidance algorithm simulation

In order to simulate the path planning and obstacle avoidance process of the UAV in real environment, a three-dimensional environment map was established, and the improved artificial potential field path planning method was verified by simulation experiments. The drone flies from the starting point coordinates (2,1,0.5) to the target point (19,19,0.5).The algorithm planning results are shown in Figure 7.5.

By establishing a three-dimensional simulated terrain map, the simulated drone is flying in a space with dense obstacles, and the red solid line is the planned path trajectory. From the simulation diagram 7.6, the map plane coordinates (5,4), (11,9), (17,16) are the obstacles between the starting point and the target point in the map, and the improved artificial potential field method passes the obstacles. A repulsion field function is established, and the obstacle forms a repulsion F_{rep}. During the flight path, the algorithm calculates three repulsive fields and finally plans the flight path, as shown in Fig. 7.6. From the path planning result, the yaw angle is less than 45°, and the planned path is relatively smooth. From the distance of the path, the distance from the starting point to the target point does not show a significant deviation from the straight line.

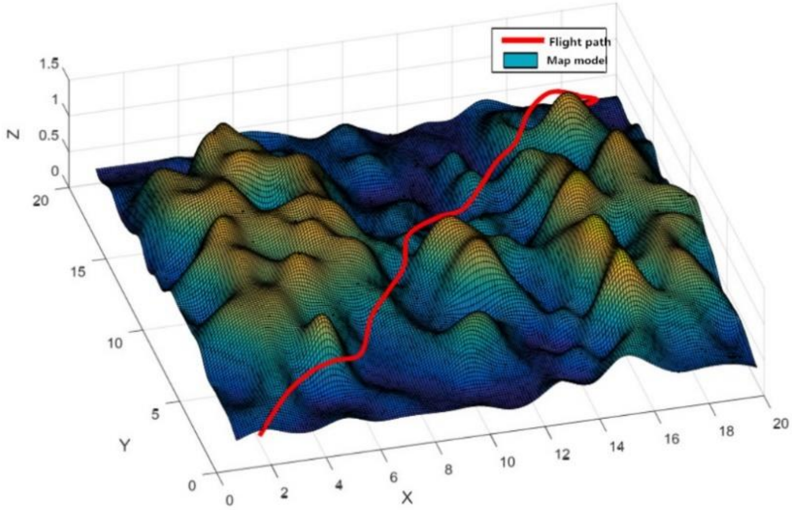


Fig. 7.6 Improved artificial potential field method path planning simulation in 3D map

7.2 Results and Analysis of UAV Aerial Photography Obstacle Avoidance Simulation Experiment

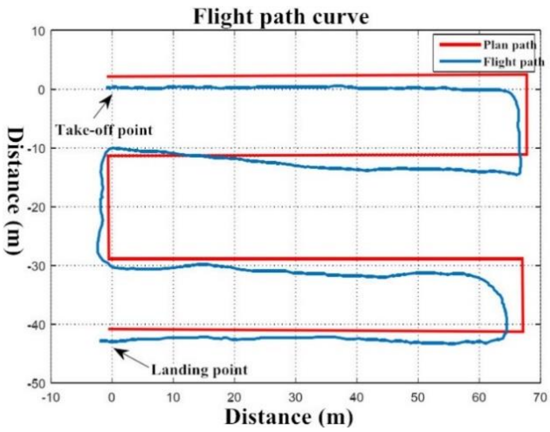


Fig. 7.7 Path task planning diagram in the actual test environment of the playground

Fig. 7.8 Comparison of UAV planning path and actual flight path

The experimental results show that the autonomous obstacle avoidance module can realize the real-time acquisition of the forward environment information of the multi-UAV. The multi-UAV can realize the real-time pair in the flight process by integrating the obstacle information detected by the millimeter wave radar and the binocular vision sensor. The tower performs autonomous obstacle avoidance, and finally the obstacle around the tower reaches the target waypoint. In actual flight, the coordinate point of the drone, the given coordinate point, and the path planning task through the ground station host computer are shown in Fig. 7.7. Fig. 7.8 shows the path trajectory of the actual flight. The system has given 8 flight target points by GPS positioning, and a tortuous path is planned.

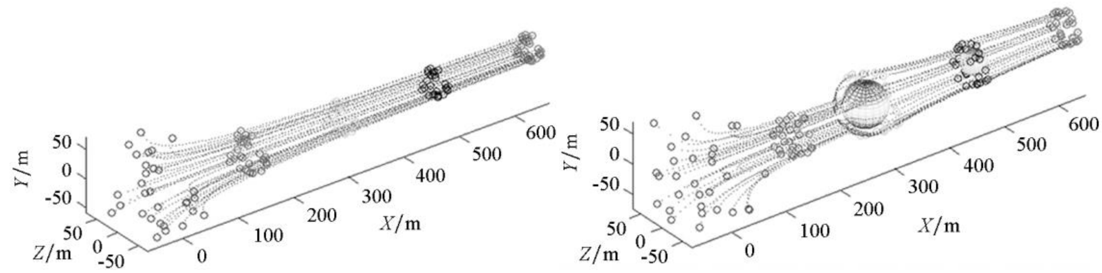


Fig. 7.9 Flight path of drones in barrier-free and obstructed spaces

Looking at the actual path planning, the movement distance in the X-axis direction is at most 6.5 meters, and the movement distance in the Y-axis direction is 40 meters. Figure 4.16 shows the flying height in the three-dimensional space of the drone. The flying height during the experimental flight is about 2 meters. The experiment was carried out on a relatively empty playground in the campus. From the actual path planning, the maximum distance of movement in the X-axis direction was 65 meters, and the distance in the Y-axis direction was 40 meters. Figure 4.16 shows the flying height in the three-dimensional space of the drone. The flying height during the experimental flight is about 2 meters. Flight path of drones in barrier-free and obstructed spaces is shown as Fig. 4.10

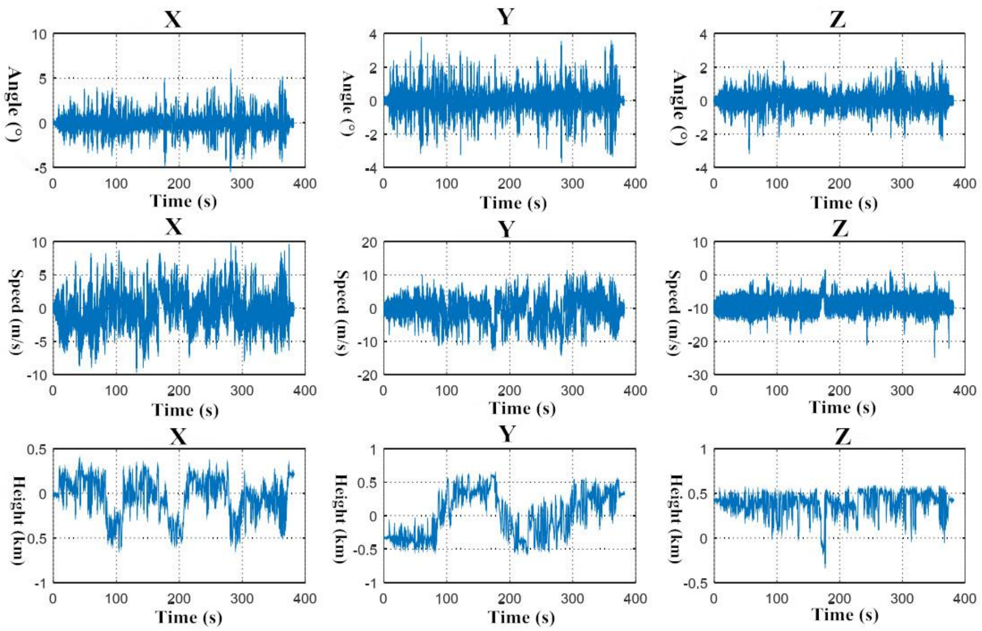


Figure 7.10 Obstacle simulation raw data curve under height, speed and angle

From the flight error results, the system deviation is small at the path point 2, and the path following error reaches the maximum at the path point 4, with an error of about 5 meters. The experimental results show that the designed path planning system error is within a reasonable range and achieves the expected design effect. Fig. 7.10 shows the sensor raw data curve data during the UAV path planning flight.

8.Conclusion

With the maturity of aerial survey technology for drones, it has become one of the important technological innovations. For newcomers who use drones for the first time, it is difficult to master the flying height, flight speed and shooting angle of the drone. How to use the drone to take satisfactory photos is an urgent problem to be solved. According to the requirements of the title, this model is based on the rotor less drone as the main experimental platform. The research focuses on the visual pose height. The binocular vision angle, the flying height of the drone and the ground object imaging relationship. The following conclusions are drawn:

1）This paper analyzes the factors affecting the imaging of UAV, discusses the imaging characteristics of the UAV vision system, and uses the geometric constraint relationship between the images to calculate the Pose feature values of the UAV's linear flight mode. Then the ground object resolution is used as the reference quantity. The optimal flying height of the drone is derived by a camera pose algorithm based on a single image. Experiments show that the proposed algorithm can make full use of the target characteristics of ground objects to derive the optimal flight altitude range of the drone.

2）In order to ensure more accurate image acquisition under actual ground conditions, for the static shooting targets such as flat land and hills, the essential matrix decomposition algorithm is used to establish the optimization function of flight altitude, flight speed and shooting angle to solve the optimal shooting angle of the drone.

3）For the shooting target in motion, this paper takes the moving car as the shooting object, and divides the car's motion trajectory according to the motion law. Based on the essential matrix decomposition algorithm, the essential matrix of flight height, flight speed and shooting angle is segmented. The function, the application of the single-relationship relationship to obtain the law of the angular motion of the drone.

4）In order to avoid obstacles such as signal towers, telephone poles and houses during the flight of the drone, the artificial potential field algorithm is used to plan the UAV path. Determine the shooting distance according to the environmental constraints around the camera internal reference and the target, calibrate the obstacles encountered in the flight, obtain the high-resolution ground image as the objective function. Determine the shooting angle of the drone; and on this basis, the drone. Smooth flight speed and uniform flight altitude are used as the objective functions. Combined with the linear flight characteristics of the drone. The attitude estimation function of the single image is used to calculate the flight speed and flight altitude. Through the simulation test comparison, the proposed algorithm can realize the obstacle avoidance shooting task of the UAV quickly and efficiently.

Although the algorithm proposed in this paper has achieved good results, the established model is not perfect. We need to continuously improve and optimize the model in the next step, and there is still much content worth studying, such as how to take better. The method of extracting image targets, how to find the adaptive adjustment method of parameters during flight, and improving the intelligence level of obstacle avoidance system are all worthy of further study. By deepening, can the shooting technology of drones be more mature and the application range is more extensive.

References

[1] Tao L, Aigong X U, Xin S, et al. Application of UWB/INS combination in indoor nabigation and positioning[J]. Science of surveying&Mapping,2016.
[2] Altug E, Ostrowski J P, Taylor C J. Control of a quadrotor helicopter using dual camera visual feedback[J]. Int. j. robots Res,2005,24(50):329-341.
[3] Breitenmoser, Andreasl; Kneip, Laurent 1; Siegwart,Roland1.A Monocular Vision-Based System For[C]. IEEE/RSJ International Conference on Intelligent Roborts&Systems.IEEE,2011.
[4] Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition[J]. Computer Science,2014.
[5] Miao C, Li J. Autonomous landing of small unmanned aerial rotorcraft based on monocular vision in GPS-denied area[J]. IEEE/CAA Journal of Automatica sinica, 2015,2(1):109-114.
[6] Huh S, Shim D H. A vision-based landing system for small unmanned aerial vehicles using an airbag[J].Control Engineering Practice,2010,18(7):812-823.
[7] Thrun S, Liu Y, Koller D, et al. Simultaneous localization and mapping with sparse extended information filters[J]. The International Journal of Robotics Research,2004,23(7-8):693-716
[8] Javier Ortiz, Alberto Gomez. License Plate Recognition for Parking Control System by Mathematical Morphology[J]. Journal of Electronic Science and Technology, 201,12(01):81-84.
[9] Lou J, Gwun O.A Comparison of SIFT,PCA-SIFT and SURF[J].International Journal of Image Processing,2013,3(4):143-152.
[10] S.Ren, K. He, R. Girshick, et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks[J]. IEEE Transaction on Pattern Analysis & Machine Intelligence,2017,39(6):1137.
[11] J. Redmon, S.Divvala, R. Girshick, et al. You Only Look Once: Real-Time Object Detection[C]. Computer Vision and Pattern Recgnition. IEEE, 106, 779- 788.
[12] J. F. Henrigues, R. Caseiro, P. Martins, et al. High-Speed Tracking with Kernelized Correlation Filters. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2015, 37,3,583-596.

Appendix

Core program

```
function [Rp, tp] = CaculPose ( Camerb, Object)
        XXc = Camerb;
        XXw = Object;
        n = size(XXw,2);
        Q = zeros(4,1);
            for i = 1:n
            if i == 1 || i == 2
                continue
                    end
         a1 = XXw(:,1)-XXw(:,i);
         a2 = XXw(:,2)-XXw(:,i);
         a3 = xcross(a1,a2);
```

```
            a4 = xcross(a2,a3);

            a2 = a2/norm(a2);
            a3 = a3/norm(a3);
            a4 = a4/norm(a4);
            A = [a4 a2 a3];

            b1 = XXc(:,1)-XXc(:,i);
            b2 = XXc(:,2)-XXc(:,i);
            b3 = xcross(b1,b2);
            b4 = xcross(b2,b3);

            b2 = b2/norm(b2);
            b3 = b3/norm(b3);
            b4 = b4/norm(b4);
            B = [b4 b2 b3];

            Ri = B * A';
            Qi = quater(Ri);
            Q = Q + Qi;
        end
        Qp = Q /(n-2);
        Rp = Eler(Qp);

        t = zeros(3,1);
        for j = 1:n
            t = t + XXc(:,j);
        end
        tp = t / n;
return
function c = xcross(a,b)
        c = [a(2)*b(3)-a(3)*b(2);
            a(3)*b(1)-a(1)*b(3);
            a(1)*b(2)-a(2)*b(1)];
 return
 function Q = quater(R)
        r11 = R(1); r12 = R(4);r13 = R(7);
        r21 = R(2); r22 = R(5);r23 = R(8);
        r31 = R(3); r32 = R(6);r33 = R(9);
        q1 = 0.5*sqrt(1+r11+r22+r33);
        q2 = 0.25*(r32-r23)/q1;
        q3 = 0.25*(r13-r31)/q1;
        q4 = 0.25*(r21-r12)/q1;
        Q = [q1 q2 q3 q4]';
 return
 function    R = Eler(Q)
         q0 = Q(1);q1 = Q(2);q2 = Q(3);q3 = Q(4);
         r11 = (q0)^2+(q1)^2-(q2)^2-(q3)^2;
         r12 = 2*(q1*q2-q0*q3);
         r13 = 2*(q1*q3+q0*q2);
         r21 = 2*(q1*q2+q0*q3);
         r22 = (q0)^2-(q1)^2+(q2)^2-(q3)^2;
         r23 = 2*(q2*q3-q0*q1);
         r31 = 2*(q1*q3-q0*q2);
         r32 = 2*(q0*q1+q2*q3);
         r33 = (q0)^2-(q1)^2-(q2)^2+(q3)^2;
         R = [r11 r12 r13;
             r21 r22 r23;
```

```
                r31 r32 r33];
  return
function main()
MM=size(G,1); % G Topographic map is 01 matrix, if 1 is an obstacle
Tau=ones(MM*MM,MM*MM); % Tau initial pheromone matrix
Tau=8.*Tau;
K=100; % Number of iterations
M=50;
S=1 ; % The starting point of the shortest path
E=MM*MM; % destination of the shortest path
Alpha=1; % Alpha A parameter that characterizes the importance of the pheromone
Beta=7; % Beta A parameter that characterizes the importance of a heuristic factor
Rho=0.3 ; % Rho pheromone evaporation coefficient
Q=1; % Q pheromone increases the intensity factor
Minkl=inf;
Mink=0;
Minl=0;
D=G2D(G);
N=size(D,1); %N indicates the size of the problem (number of pixels)
   a=1; the side length of the % small square pixel
   Ex=a*(mod(E,MM)-0.5); %End point abscissa
   if Ex==-0.5
      Ex=MM-0.5;
end
Ey=a*(MM+0.5-ceil(E/MM));%Terminal point ordinate
  Eta=zeros(N); % heuristic information, taken as the reciprocal of the straight line distance to the
target point
% following heuristic information matrix
For i=1:N
Ix=a*(mod(i,MM)-0.5);     if ix==-0.5
     ix=MM-0.5;
     end
iy=a*(MM+0.5-ceil(i/MM));
     if i~=E
     Eta(i)=1/((ix-Ex)^2+(iy-Ey)^2)^0.5;
     else
   Eta(i)=100;
   end
end
ROUTES=cell(K,M); % uses the cell structure to store the crawling route of each ant of each
generation.
PL=zeros(K,M);
%Use matrix to store the length of crawling route of each ant of each generation
% Start K round ant foraging activities, sending M ants per round
for k=1:K
     for m=1:M
          W=S; % current node is initialized to the starting point
          Path=S; % crawl route initialization
          PLkm=0; % crawl route length initialization
          TABUkm=ones(N); % taboo table initialization
          TABUkm(S)=0; % is already at the initial point, so to exclude
          DD=D; % adjacency matrix initialization
                    % next step can go to the node
          DW=DD(W,:);
          DW1=find(DW);
          for   j=1:length(DW1)
               if TABUkm(DW1(j))==0
                              DW(DW1(j))=0;
```

```
                end
          end
end
LJD=find(DW);
Len_LJD=length(LJD);
while W~=E&&Len_LJD>=1
% reel gambling method choose how to go nextPP=zeros(Len_LJD);
          for i=1:Len_LJD
                          PP(i)=(Tau(W,LJD(i))^Alpha)*((Eta(LJD(i)))^Beta);
          end
end
sumpp=sum(PP);
PP=PP/sumpp;
Pcum(1)=PP(1);
for i=2:Len_LJD
             Pcum(i)=Pcum(i-1)+PP(i);
end
Select=find(Pcum>=rand);
to_visit=LJD(Select(1)); % establish probability distribution
Path=[Path,to_visit]; % path increase
PLkm=PLkm+DD(W,to_visit); % path length increased
w=to_visit;
      for kk=1:N
            if TABUkm(kk)==0
                DD(W,kk)=0;
                DD(kk,W)=0;
              end
      end
TABUkm(W)=0; % visited nodes are removed from the taboo table
  DW=DD(W,:);
DW1=find(DW);
for j=1:length(DW1)
        if TABUkm(DW1(j))==0
            DW(j)=0;
          end
end
LJD=find(DW);
Len_LJD=length(LJD); % number of optional nodes
  end
% note the foraging route and route length of each ant for each generation ROUTES{k,m}=Path;
      if Path(end)==E
            PL(k,m)=PLkm;
            if PLkm<minkl
                  mink=k;minl=m;minkl=PLkm;
            end
      else
            PL(k,m)=0;
      end
end
Delta_Tau=zeros(N,N); % update amount initialization
      for m=1:M
          if PL(k,m)
              ROUT=ROUTES{k,m};
              TS=length(ROUT)-1; % hops
                PL_km=PL(k,m);
              for s=1:TS
                  x=ROUT(s);
                  y=ROUT(s+1);
```

```
                Delta_Tau(x,y)=Delta_Tau(x,y)+Q/PL_km;
                Delta_Tau(y,x)=Delta_Tau(y,x)+Q/PL_km;
            end
        end
    end
Tau=(1-Rho).*Tau+Delta_Tau;
 end
% drawing
plotif=1;
    if plotif==1 % draw convergence curve
            minPL=zeros(K);
            for i=1:K
                    PLK=PL(i,:);
                    Nonzero=find(PLK);
                    PLKPLK=PLK(Nonzero);
                    minPL(i)=min(PLKPLK);
            end
figure(1)
plot(minPL);
hold on
grid on
title(' Convergence curve change trend ');
xlabel(' Number of iterations ');
ylabel(' Minimum path length ');
figure(2)
axis([0,MM,0,MM])
for i=1:MM
     for j=1:MM
          if G(i,j)==1
            x1=j-1;y1=MM-i;
            x2=j;y2=MM-i;
            x3=j;y3=MM-i+1;
            x4=j-1;y4=MM-i+1;
           fill([x1,x2,x3,x4],[y1,y2,y3,y4],[0.2,0.2,0.2]);
hold on
     else
           x1=j-1;y1=MM-i;
           x2=j;y2=MM-i;
           x3=j;y3=MM-i+1;
           x4=j-1;y4=MM-i+1;
           fill([x1,x2,x3,x4],[y1,y2,y3,y4],[1,1,1]);
hold on
           end
     end
end
hold on
title('Drone movement track ');
xlabel('coordinatex');
ylabel('coordinatey');
ROUT=ROUTES{mink,minl};
LENROUT=length(ROUT);
Rx=ROUT;
Ry=ROUT;
for
       ii=1:LENROUT
       Rx(ii)=a*(mod(ROUT(ii),MM)-0.5);
       if
       Rx(ii)==-0.5
```

```
        Rx(ii)=MM-0.5;
end
        Ry(ii)=a*(MM+0.5-ceil(ROUT(ii)/MM));
end
        plot(Rx,Ry)
end
plotif2=0;
if  plotif2==1
    figure(3)
    axis([0,MM,0,MM])
    for  i=1:MM
        for  j=1:MM
              if G(i,j)==1
                x1=j-1;y1=MM-i;
                x2=j;y2=MM-i;
                x3=j;y3=MM-i+1;
                x4=j-1;y4=MM-i+1;
                fill([x1,x2,x3,x4],[y1,y2,y3,y4],[0.2,0.2,0.2]);
hold on
        else
                x1=j-1;y1=MM-i;
                x2=j;y2=MM-i;
                x3=j;y3=MM-i+1;
                x4=j-1;y4=MM-i+1;
fill([x1,x2,x3,x4],[y1,y2,y3,y4],[1,1,1]);
hold on
              end
      end
end
for k=1:K
      PLK=PL(k,:);
      minPLK=min(PLK);
      pos=find(PLK==minPLK);
      m=pos(1);
      ROUT=ROUTES{k,m};
      LENROUT=length(ROUT);
      Rx=ROUT;
      Ry=ROUT;
for ii=1:LENROUT
      Rx(ii)=a*(mod(ROUT(ii),MM)-0.5);
if Rx(ii)==-0.5
    Rx(ii)=MM-0.5;
    end
    Ry(ii)=a*(MM+0.5-ceil(ROUT(ii)/MM));
end
plot(Rx,Ry)
hold on
    end
end
function D=G2D(G)
l=size(G,1);
D=zeros(l*l,l*l);
for i=1:l
        for j=1:l
                if G(i,j)==0
                        for m=1:l
                                for n=1:l
                                        if G(m,n)==0
```

```
                        im=abs(i-m);jn=abs(j-n);
                        if im+jn==1||(im==1&&jn==1)
                        D((i-1)*l+j,(m-1)*l+n)=(im+jn)^0.5;
                        end
                    end
                end
            end
        end
    end
end
```