

A Fuzzy Controller: Using Monocular Computer Vision to See and Avoid Obstacle for Quadcopter

Geng Keke, Liu Wei, and Tan Liguo

Bauman Moscow State Technical University 1051005 Russian

Keywords: monocular computer vision, 3-D reconstructions, PID, fuzzy controller.

Abstract. In this paper use the monocular computer vision, which is in the direction forward of quadcopter, to calculate the spatial distribution of these feature points matching. So that we can find out the approximate contour of the obstacle. A fuzzy controller was designed to command quadcopter for avoiding collision, the sensors used to accomplish this task are a forward camera and inertial measurement unit (IMU). In order to evaluate the performance of this controller a big amount of tests was carried out with real web camera and simulation in MATLAB.

Introduction

Now UAV (unmanned aerial vehicle) especially quadcopter has attracted more interests for a wide range of applications, involving security, rescues, inspection, surveillance, and mapping, etc. A system of quadcopters is a kind of under-actuated systems with multiple and strong coupling. The control problem has been many solutions, some of which have been implemented successfully on hardware tests, such as PID and LQR [1-2]. Sliding mode controllers [3] and H_∞ controllers have been well applied to stabilize quadcopter [4].

In practice there is a specific problem, to avoid obstacle in the flight. The important content of the paper is to design a fuzzy controller for quadcopter, which is to control quadcopter for obstacle avoidance. In section 2 computer vision for navigation is presented, the mission has been accomplished by the three-dimensional reconstruction algorithm. In Section 3 introduce mathematical model of quadcopter, which is accord to Euler theorem and Newton's law, at the same time considering the effect of air resistance and rotating torque. Section 4 provides a fuzzy controller to help quadcopter avoid obstacle. Finally, give the simulation results in Section 5.

3D-reconstruction of keypoints obstacle

Computer vision is a field that includes methods for acquiring, processing, analyzing, and understanding images and, in general, high-dimensional data from the real world in order to produce numerical or symbolic information. At present, computer vision has been very widely used. We use monocular vision to rebuild the contours of obstacles.

At first, we need to detect the keypoints of obstacles from scene photos, making by computer vision. Present there are some classic keypoint detection: Moravec corner detection, Harris corner detection, SUSAN (Smallest Univalued Segment Assimilating Nucleus) corner detection and the others. To find more stable keypoints, considering the effect of the movement of the carrier-rotation, scale and illumination changes, we use SIFT method (Scale-invariant feature transform), which was proposed by David. G. Lowe in 2004 [5] [6]. The same time to obtain more reliable keypoints, we use the method combined with Hartley's Normalized Eight-Point Algorithm [7] [8] and RANSAC (Random sample consensus) algorithms to calculate the fundamental matrix and essentially matrix. By the singular value decomposition for the essentially matrix ($E = USV^T$), we can get the rotation matrix R and translation vector t [9].

Then we can get the projection matrix:

$$M = [K_{3 \times 3} | 0]_{3 \times 4} \cdot \begin{bmatrix} R_{3 \times 3} & t_{3 \times 1} \\ 0 & 1 \end{bmatrix} \quad (1)$$

Using Camera Calibration Toolbox for Matlab, written by the California Polytechnic University Dr. Bouquet, we can get the camera internal parameter matrix:

$$k = \begin{bmatrix} 3052.568 & 0 & 663.793 \\ 0 & 3277.952 & 504.364 \\ 0 & 0 & 0 \end{bmatrix} \quad (2)$$

The projection matrix formula indicates:

$$z_{ci} \cdot \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \begin{bmatrix} m_{11}^i & m_{12}^i & m_{13}^i & m_{14}^i \\ m_{21}^i & m_{22}^i & m_{23}^i & m_{24}^i \\ m_{31}^i & m_{32}^i & m_{33}^i & m_{34}^i \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \quad i = 1, 2. \quad (3)$$

Four linear equations can be obtained by the following:

$$\begin{cases} (u_1 m_{31}^1 - m_{11}^1)X + (u_1 m_{32}^1 - m_{12}^1)Y + (u_1 m_{33}^1 - m_{13}^1)Z = m_{14}^1 - u_1 m_{34}^1 \\ (v_1 m_{31}^1 - m_{21}^1)X + (v_1 m_{32}^1 - m_{22}^1)Y + (v_1 m_{33}^1 - m_{23}^1)Z = m_{24}^1 - v_1 m_{34}^1 \\ (u_2 m_{31}^2 - m_{11}^2)X + (u_2 m_{32}^2 - m_{12}^2)Y + (u_2 m_{33}^2 - m_{13}^2)Z = m_{14}^2 - u_2 m_{34}^2 \\ (v_2 m_{31}^2 - m_{21}^2)X + (v_2 m_{32}^2 - m_{22}^2)Y + (v_2 m_{33}^2 - m_{23}^2)Z = m_{24}^2 - v_2 m_{34}^2 \end{cases} \quad (4)$$

They are overdetermined equations, but the least squares method can be used for solving equations. In this way three-dimensional coordinates of the keypoints (X, Y, Z) can be obtained.

To verify the correctness of the three-dimensional reconstruction algorithm, experiment was indoors. Subject of this experiment was a corridor (fig.1).

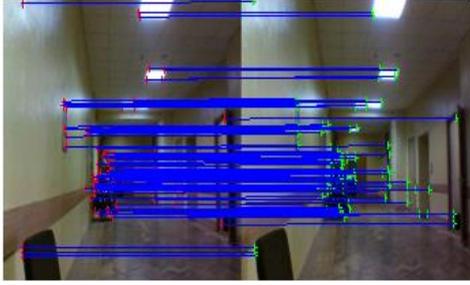


Fig.1 Match of keypoint

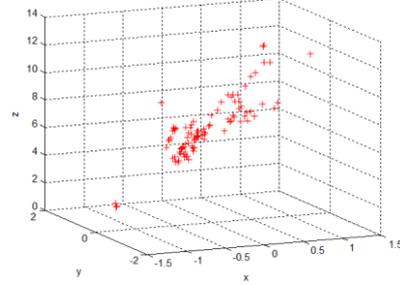


Fig.2 3D-reconstruction

The rotation matrix and translation vector are:

$$R = \begin{bmatrix} 0.9804 & -0.0036 & 0.1470 \\ 0.0011 & 0.9999 & 0.0131 \\ 0.1471 & 0.0126 & -0.9803 \end{bmatrix}, t = \begin{bmatrix} -0.1067 \\ -0.0080 \\ 0.9943 \end{bmatrix} \quad (5)$$

The coordinates of obstacle are in the world coordinate system, needs to be converting them to the base coordinate system.

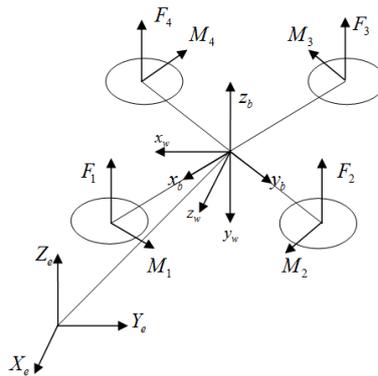


Fig.3 The quadcopter frame systems. $X_e Y_e Z_e$ - base coordinate system; $X_b Y_b Z_b$ -body coordinate system ; $X_w Y_w Z_w$ -world coordinate system; F_1, F_2, F_3, F_4 -draft; M_1, M_2, M_3, M_4 -Moment of resistance.

Set to the image plane of camera perpendicular to the plane of quadcopter, and the angle between the optical axis of camera and cross arms of quadcopter is 45° . Then, the based coordinates of obstacle:

$$[X, Y, Z]^T = R(\varphi, \theta, \psi) \cdot R\left(-\frac{\pi}{2}, 0, -\frac{\pi}{4}\right) \cdot [x_w, y_w, z_w]^T + [x_b, y_b, z_b]^T \quad (6)$$

Mathematical model of quadcopter and PID controller

In the simulation of flight quadcopter, with limited distance and duration, we can neglect the motion of the Earth, to consider the earth inertial coordinate system. In the initial stages of development, quadcopter can also be considered as the unit rigid body[10], and the wind takes into account only as an external perturbation. Establishment of movement and kinetic as follows:

1). the relationship between Euler angular velocity and the angular velocity of body

$$\begin{cases} \dot{\varphi} = p + \sin \varphi \cdot \tan \theta \cdot q + \cos \varphi \cdot \tan \theta \cdot r \\ \dot{\theta} = \cos \varphi \cdot q - \sin \varphi \cdot r \\ \dot{\psi} = \frac{\sin \varphi}{\cos \theta} \cdot q + \frac{\cos \varphi}{\cos \theta} \cdot r \end{cases} \quad (7)$$

Here φ, θ, ψ - The angles of yaw, pitch, roll; p, q, r - The angular velocities in the body frame B;

2).On the base of Euler theorem and Newton's law with the effect of air resistance and rotating torque.

$$\begin{cases} \dot{\zeta} = v \\ m\dot{v} = F_f + F_d + F_g \\ J\dot{\Omega} + \Omega \times J\Omega = M_f - M_d + M_c \end{cases} \quad (8)$$

F_f - Translational force; F_d - Air resistance; F_g - gravitational force; $\zeta = [x, y, z]$ -position in the system E; $\Omega \times J\Omega$ - gyroscopic moments; M_c - Coriolis torque.

3). Combined with the above equations, obtain:

$$\begin{cases} \dot{x} = (C_\varphi C_\psi S_\theta + S_\varphi S_\psi) \frac{b(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2)}{m} - \frac{f_{dx}}{m} \dot{x} \\ \dot{y} = (C_\varphi S_\psi S_\theta - S_\varphi C_\psi) \frac{b(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2)}{m} - \frac{f_{dy}}{m} \dot{y} \\ \dot{z} = C_\varphi C_\theta \frac{b(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2)}{m} - \frac{f_{dz}}{m} \dot{z} - g \\ \dot{p} = qr \left(\frac{I_y - I_z}{I_x} \right) + \frac{l}{I_x} b(\omega_4^2 - \omega_2^2) - \frac{f_{dmx}}{I_x} p + \frac{J_r}{I_x} q(\omega_1 + \omega_3 - \omega_2 - \omega_4) \\ \dot{q} = pr \left(\frac{I_z - I_x}{I_y} \right) + \frac{l}{I_y} b(\omega_3^2 - \omega_1^2) - \frac{f_{dmy}}{I_y} q + \frac{J_r}{I_y} p(\omega_2 + \omega_4 - \omega_1 - \omega_3) \\ \dot{r} = pq \left(\frac{I_x - I_y}{I_z} \right) + \frac{l}{I_z} d(\omega_1^2 + \omega_3^2 - \omega_2^2 - \omega_4^2) - \frac{f_{dmz}}{I_z} r \\ p = \dot{\varphi} - s_\theta \dot{\psi} \\ q = C_\varphi \dot{\theta} + S_\varphi C_\theta \dot{\psi} \\ r = -S_\varphi \dot{\theta} + C_\varphi C_\theta \dot{\psi} \end{cases} \quad (9)$$

The other parameters are listed in Table.2.

PID [11] controllers of angular position and height have the following forms:

$$\begin{cases} U_2 = K_{p\varphi}(\varphi_d - \varphi) + K_{i\varphi} \int (\varphi_d - \varphi) dt + K_{d\varphi}(\dot{\varphi}_d - \dot{\varphi}) \\ U_3 = K_{p\theta}(\theta_d - \theta) + K_{i\theta} \int (\theta_d - \theta) dt + K_{d\theta}(\dot{\theta}_d - \dot{\theta}) \\ U_4 = K_{p\psi}(\psi_d - \psi) + K_{i\psi} \int (\psi_d - \psi) dt + K_{d\psi}(\dot{\psi}_d - \dot{\psi}) \\ U_1 = \frac{m}{C_\theta C_\varphi} (K_{pz}(z_d - z) + K_{iz} \int (z_d - z) dt + K_{dz}(z_d - \dot{z}) - g) \end{cases} \quad (10)$$

Values of the coefficients regulators are showed in the follown table:

Table 1: Values of the coefficients regulators

	K_p	K_i	K_d
z	25	10	30
φ	8	0.02	5
θ	10	0.1	8

ψ	20	0.1	15
--------	----	-----	----

Fuzzy Controller

The controller was designed using Fuzzy Logic Techniques to avoid obstacle. The aim of the controller is to generate desired yaw commands for the aerial vehicle based on the location of the target in the image plane. With the commands generated by the controller, the aircraft must avoid the obstacle, which is lie in its trajectory. The Fuzzy PID controller was implemented using Fuzzy Logic Toolbox MATLAB. This toolbox has a hierarchical class definition for each part of the fuzzy-logic environment (variables, rules, membership functions, and defuzzification modes).

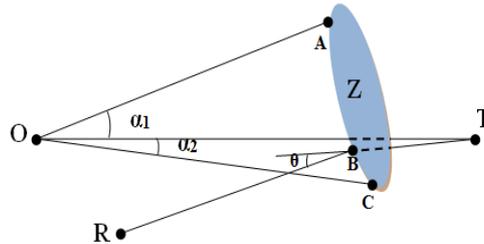


Fig. 4: Schematic diagram of obstacle avoidance. O-Original point, T-Target point, R-Real-time point, Z-Zone obstacle, A – The leftmost keypoints on obstacle, C-The right keypoint, B-The nearest keypoint, RB=d - distance, θ - supplementary angle of the angle between quadcopter, keypoint obstacle and the target point.

Some rules: 1).if $\alpha_1 > \alpha_2$, then quadcopter turn right; if $\alpha_1 < \alpha_2$, then quadcopter turn left; 2).if slope of the line BT larger than slope of the line RB, then $\theta > 0$, otherwise $\theta < 0$.

Input and output signals can be represented with triangular membership function:

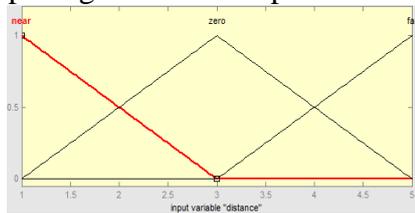


Fig.5 Membership function of the first input, the distance from quadcopter to the nearest keypoint of the obstacle.

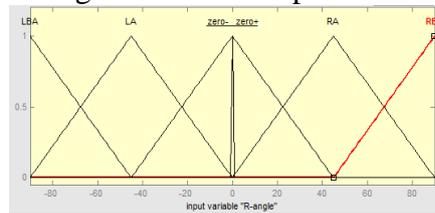


Fig.6 Membership function of the second input, the supplementary angle of the angle between quadcopter, keypoint obstacle and the target point when quadcopter turn right.

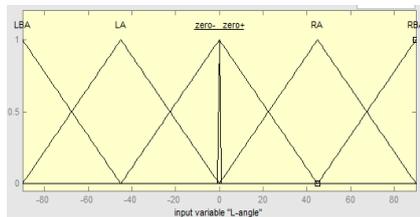


Fig.7 Membership function of the third input, the supplementary angle of the angle between quadcopter, keypoint obstacle and the target point when quadcopter turn left.

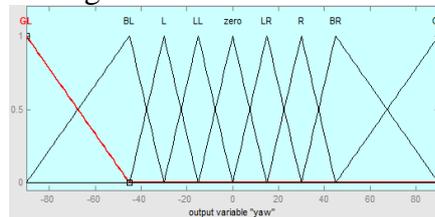


Fig.8 Membership function of the output, heading degrees to turn.

When quadcopter turn right the rules of fuzzy logic are shown in the following table:

Table 2: The rules of fuzzy logic

yaw	distance
-----	----------

		near	zero	far
R-angle	LBA	GR	GR	GR
	LA	BR	R	LR
	Zero-	GR	BR	R
	Zero+	GR	BR	R
	RA	BR	R	LR
	RBA	zero	zero	zero

When quadcopter turn left the rules will be the analogous way.

Simulation and Results

The entire structure for obstacle avoidance method of quadcopter in Simulation presents in the Fig.9. The blocks are:1- target point ; 2,4-PID controller; 3-initial attitude angles; 5-signal splitter; 6-voltage limiter;7- motor and propeller model; 8- mathematical model of quadcopter; C- camera; S - 3D reconstruction of obstacles; O –calculation of the distance and angle; F- fuzzy controller. Due to limited conditions, we only verify the feasibility of obstacle avoidance algorithm by simulation in MATLAB.

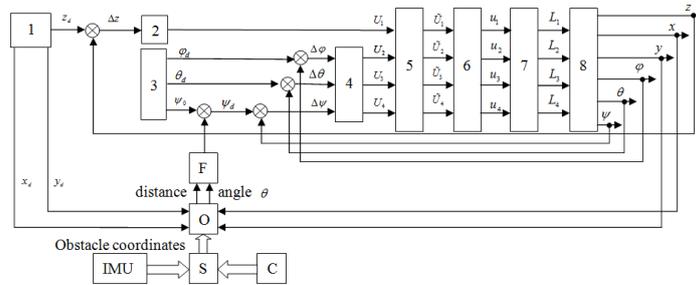


Fig.9 The structure of whole control system

Table 3: The simulation parameters

Total weight m (kg)	1.017	Thrust coefficient b	$2.984 \cdot 10^{-5}$
Gravitational acceleration g (m/s ²)	9.810	Drag coefficient d	$3.232 \cdot 10^{-7}$
Moment of inertia of the rotor about its axis of roataion J_r (kg*m ²)	$2.838 \cdot 10^{-5}$	Quadcopter's arm length l (m)	0.182
Moment of inertia of the quadcopter $J = [I_x \ I_y \ I_z]$ (kg*m ²)	diag(0.0217,0.022,0.0397)	Translational drag coefficient f_{dm}	diag(5.537 5.567 6.354)* 10^{-4}
		Rotational drag torque coefficient f_d	diag(0.032 0.032 0.048)

The simulation results are shown in the Fig.10, on the left the 3d trajectory for obstacle avoidance, on the right 2d trajectory. Quadcopter flies from A (0,0,3) to D (20,20,3). Red dots represents the keypoints of the obstacle; blue area represents the vertical region formed by the outermost feature points of the obstacle. The entire flight process can be divided into several stages: OA- departure; AB- flight to the target point; BC- avoiding obstacle; CD- flight to the target point.

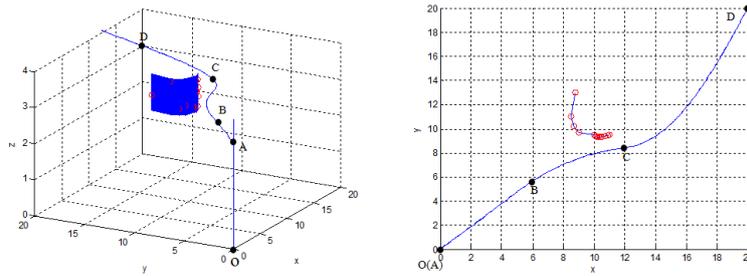


Fig.10 Trajectory of avoidance obstacle

Conclusions

In this article is proposed an obstacle avoidance method for quadcopter. In order to achieve obstacle avoidance, mainly control the yaw angle by using monocular vision and PID-fuzzy controller. To get the spatial location of the obstacle, the method in this article is getting key points of two consecutive image by algorithm SIFT, then form three-dimensional reconstruction for the keypoints, the final converts the coordinates of the key points into the base coordinate. We can see by simulation test, that avoidance strategy is feasible.

References

- [1] Hoffmann G M, Huang H, Waslander S L, et al. Quadrotor helicopter flight dynamics and control: Theory and experiment[C]//Proc. of the AIAA Guidance, Navigation, and Control Conference. 2007, 2.
- [2] Bouabdallah S, Noth A, Siegwart R. PID vs LQ control techniques applied to an indoor micro quadrotor[C]//Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on. IEEE, 2004, 3: 2451-2456.
- [3] Bouabdallah S, Siegwart R. Backstepping and sliding-mode techniques applied to an indoor micro quadrotor[C]//Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on. IEEE, 2005: 2247-2252.
- [4] Raffo G V, Ortega M G, Rubio F R. MPC with nonlinear H_{∞} control for path tracking of a quad-rotor helicopter[C]//Proc. of the IFAC World Congress. 2008, 2008.
- [5] D.G. Lowe. Object recognition from local scale-invariant features. In International Conference On Computer Vision, Corfu, Greece, September 1999: 1150-1157.
- [6] D.G. Lowe. Distinctive image features from scale-invariant key points. International Journal of Computer Vision, 2004, 2(60):91-110.
- [7] Wojciech Chojnacki , Michael J. Brooks , Anton Van Den Hengel , Darren Gawley. Revisiting Hartley's normalized eight-point algorithm // IEEE Transactions on Pattern Analysis and Machine Intelligence. - 2003.
- [8] Benjamin Cohen, Jeffrey Byrne. Inertial Aided SIFT for Time to Collision Estimation.[C] 2009 IEEE International Conference on Robotics and Automation, Kobe International Conference Center. Kobe, Japan, May, 2009.
- [9] Nister. An efficient solution to the five-point relative pose problem[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2004, 26 (6); 756-770.
- [10] Rodic A, Mester G. Modeling and simulation of quad-rotor dynamics and spatial navigation[C]//Intelligent Systems and Informatics (SISY), 2011 IEEE 9th International Symposium on. IEEE, 2011: 23-28.

[11] K.J. Astrom, TO. Hagglund Revisiting the Ziegler–Nichols step response method for PID control // science @direct. 2003.