

Research Article

Quad-Rotor Helicopter Autonomous Navigation Based on Vanishing Point Algorithm

Jialiang Wang,¹ Hai Zhao,¹ Yuanguo Bi,¹ Xingchi Chen,¹ Ruofan Zeng,¹ and Shiliang Shao²

¹ College of Information Science & Engineering, Northeastern University, No. 11, Lane 3, Wenhua Road, Heping, Shenyang, Liaoning Province 110819, China

² Shenyang Institute of Automation Chinese Academy of Sciences, No. 114, Nanta Street, Shenhe District, Shenyang, Liaoning 110016, China

Correspondence should be addressed to Yuanguo Bi; biyuanguo@ise.neu.edu.cn

Received 14 April 2014; Accepted 22 June 2014; Published 22 July 2014

Academic Editor: Yuxin Zhao

Copyright © 2014 Jialiang Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Quad-rotor helicopter is becoming popular increasingly as they can well implement many flight missions in more challenging environments, with lower risk of damaging itself and its surroundings. They are employed in many applications, from military operations to civilian tasks. Quad-rotor helicopter autonomous navigation based on the vanishing point fast estimation (VPFE) algorithm using clustering principle is implemented in this paper. For images collected by the camera of quad-rotor helicopter, the system executes the process of preprocessing of image, deleting noise interference, edge extracting using Canny operator, and extracting straight lines by randomized hough transformation (RHT) method. Then system obtains the position of vanishing point and regards it as destination point and finally controls the autonomous navigation of the quad-rotor helicopter by continuous modification according to the calculated navigation error. The experimental results show that the quad-rotor helicopter can implement the destination navigation well in the indoor environment.

1. Introduction

As things get further away from us, they seem smaller. When they get far enough away, distances become ever tinier and so form a single point called vanishing point. In one-point perspective, all the horizontal lines go straight across, while the lines that move away from us—the sides of boxes, the road we are on, or the railway lines in front of us—all converge towards the center of the picture [1].

So the vanishing point contains much three-dimension information of the real environment, which can help to understand the real environment especially containing many parallel lines. The position estimation of vanishing point is widely used in applications such as navigation of robot, three-dimensional reconstruction, and camera calibration, and it has already become the hot research topic nowadays.

In the recent years, quad-rotor helicopter received an increasing attention from the research community. Many authors focused on the autonomous navigation control of

them. Hoffmann et al. [2] presented a model-based algorithm for autonomous flying with their STARMAC-quad rotor. Their system flew outdoors and utilized GPS and inertial measurement. Achtelika et al. [3] developed an indoor autonomous quad rotor equipped with a laser range scanner and cameras enabling autonomous hovering in a constraint indoor environment. Kendoul et al. [4] developed vision system for autonomous navigation 3D localization and control of small aerial vehicles based on optic flow.

Vanishing point is very important because it contains abundant three-dimension information for the indoor environment, and the innovation for this paper is that it focuses on the implementation of autonomous navigation using vanishing point algorithm and VPFE algorithm based on clustering principle in the indoor environment.

This paper is organized as follows: Section 2 presents system model of quad-rotor helicopter and its physical parameters. Section 3 describes the principle of vanishing point algorithm. Section 4 presents the implementation process

of proposed VPEE algorithm in detail. Section 5 describes the navigation based on improved PID control. Section 6 presents experimental results in the indoor environment in terms of execution time and flight deviation. Conclusions are given at the end of the paper.

2. System Model of Quad-Rotor Helicopter

The quad rotor has the flight motions such as vertical take-off and landing, pitching, rolling, and yawing, which are implemented by motor speed control so as to make system generate different force on each propeller.

$$\begin{aligned} \begin{bmatrix} Xg \\ Yg \\ Zg \end{bmatrix} &= \begin{pmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & \sin \varphi & \cos \varphi \\ 0 & -\cos \varphi & \sin \varphi \end{pmatrix} \cdot \begin{pmatrix} -\sin \theta & \cos \theta & 0 \\ -\cos \theta & -\sin \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \\ &= \begin{pmatrix} -\cos \psi \sin \theta - \sin \psi \sin \varphi \cos \theta & \cos \psi \cos \theta - \sin \psi \sin \theta \sin \varphi & \sin \psi \cos \varphi \\ \sin \psi \sin \theta - \cos \psi \sin \varphi \cos \theta & -\sin \psi \cos \theta - \cos \psi \sin \theta \sin \varphi & \cos \psi \cos \varphi \\ \cos \theta \cos \varphi & \cos \varphi \sin \theta & \sin \varphi \end{pmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}, \end{aligned} \quad (1)$$

Where straight lines $L_1, L_2,$ and L_3 represent the parallel lines of $OX, OY,$ and $OgXg$ separately. Lines $O'P_1$ and $O'P_2$ are the projection lines while lines OX and OY mapped into the ground coordinate. $\theta, \varphi,$ and ψ mean included angles as Figure 1 shows.

Besides, for the whole quad-rotor helicopter, the input variable is rotate speed of 4 electric motors [5], and they can be expressed as

$$\begin{aligned} u_1 &= F_1 + F_2 + F_3 + F_4, \\ u_2 &= F_4 - F_2, \\ u_3 &= F_3 - F_1, \\ u_4 &= F_1 - F_2 + F_3 - F_4, \end{aligned} \quad (2)$$

where $u_1, u_2, u_3,$ and u_4 represent the lifting force, rolling force, pitching force, and drifting force, respectively. In the meanwhile, the input variable contains 3 other pieces of poseinformation of $\theta, \varphi,$ and ψ for the whole system [6].

3. Principle of Vanishing Point Algorithm

3.1. Model of Vanishing Point. A group of parallel lines in three-dimension (3D) space can be mapped into some intersection lines in two-dimension (2D) image, and the intersection point formed by these intersection lines is called vanishing point [7].

Figure 2 shows the mapping relationship between point $A(x_0, y_0, z_0)$ in the three-dimension space and point $B(x'_0, y'_0, z'_0)$ in the two-dimension image [8], and we can obtain

$$\begin{pmatrix} x'_0 \\ y'_0 \\ z'_0 \end{pmatrix} = \begin{pmatrix} \frac{x_0 f}{y_0} \\ \frac{z_0 f}{y_0} \\ f \end{pmatrix}, \quad (3)$$

where f represents focal length of visual system, and point $p(x_0, y_0, z_0)$ can also be expressed as the sum of several

(Og, Xg, Yg, Zg) is the ground coordinate, and (O, X, Y, Z) is the quad-rotor helicopter coordinate as Figure 1 shows. Quad-rotor helicopter coordinate is used to control the state of motion, while ground coordinate is used to adjust the position and posture of the quad-rotor helicopter. From the analysis about transformation of above two coordinates, we can get the transformation equation as formula (1) shows:

vectors in the directions of $x, y,$ and z coordinate axes [9]. Using the attributes of $i, j,$ and k to express the base vector of the three axes, in the meantime vector \bar{p} can be expressed as $\bar{p} = x_0 \bar{i} + y_0 \bar{j} + z_0 \bar{k}$, so the vanishing point v can be calculated by

$$\bar{v} = x_v \bar{i} + y_v \bar{j} + f \bar{k}. \quad (4)$$

The indoor flight environment for quad-rotor helicopter navigation is shown as Figure 3(a), and the vanishing point can be obtained by the above calculation, and then it is set as the destination point navigation for quad-rotor helicopter in the indoor environment.

3.2. Straight Line Extraction of Destination Region. Detecting the parallel lines existing in the test environment is the premise for calculating the vanishing point, so this procedure is very important. It contains the below process: preprocessing of image, deleting noise interference, edge extracting using Canny operator, and extracting lines by RHT method.

3.2.1. Preprocessing of Image. While using histogram equalization method to improve contrast of overall situation, the luminance information can well be reflected in the histogram, and it is also effective for foreground and background no matter it is too bright or too dark. The most important thing is that histogram equalization is a reversible operation, which means only knowing the function expression of equalization, and then system can recover the original histogram with lower computational complexity.

3.2.2. Canny Operator. Canny operator is the optimization approximation operator obtained by measuring signal to noise ratio (SNR) and location product. Its principle in nature is to execute smooth image operation by using Gaussian

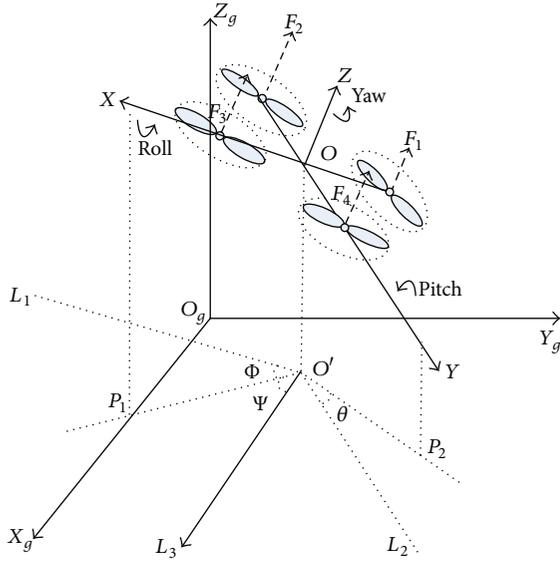


FIGURE 1: Transformation relationship between quad-rotor helicopter and ground coordinate.

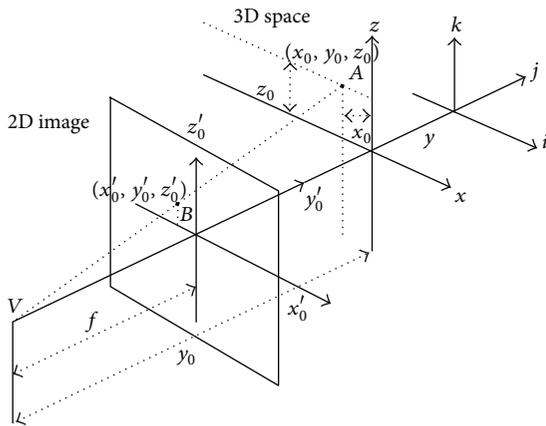


FIGURE 2: A point in 3D space mapped into 2D image.

function and calculating max derivative through first order differential operator. The process of edge detection is used to delete much useless information. In fact, it is used to detect points having obvious brightness variation, because these points include much information such as discontinuous depth and surface direction, change of material attribute, and scene illumination. Canny algorithm includes the below 4 procedures.

(1) *Gaussian Filter Denoising.* Gaussian blur is a filter for image penumbra, and it calculates the conversion for each pixel in the image by using below normal distribution:

$$G(r) = \frac{1}{\sqrt{2\pi\sigma^2}N} e^{-r^2/2\sigma^2}. \quad (5)$$

In the two-dimensional space, it can be expressed as

$$G(u, v) = \frac{1}{2\pi\sigma^2} e^{-(u^2+v^2)/2\sigma^2}, \quad (6)$$

where r is the blur radius which can be calculated by $r^2 = u^2 + v^2$; besides σ is the standard deviation of the above normal distribution, and it is used to control the smoothness of Gaussian filter parameters.

System executes convolution operation for original image and Gaussian smoothing template and then obtains an image having the slightest obscure. So image noise is reduced and the level of detail is also decreased after this process.

(2) *Gradient Operator Sobel.* The edges in image may point to different directions, so the Canny algorithm uses 4 masks to detect the margin of horizontal verticality and diagonals lines. The convolution value for original image and any mask is calculated and stored.

The first derivative of one two-dimension equation can be regarded as the gradient, and its vector can be expressed as

$$\bar{L}(x, y) = \begin{bmatrix} \bar{L}_x \\ \bar{L}_y \end{bmatrix} = \begin{bmatrix} \partial f / \partial x \\ \partial f / \partial y \end{bmatrix}. \quad (7)$$

The direction of vector $\bar{L}(x, y)$ stands for the direction having max changing rate, and the modulus of the gradient can be calculated by

$$|\bar{L}(x, y)| = \sqrt{\bar{L}_x^2 + \bar{L}_y^2}. \quad (8)$$

The direction of the gradient is

$$\delta(x, y) = \arctan\left(\frac{\bar{L}_y}{\bar{L}_x}\right), \quad (9)$$

where δ is the included angle formed by gradient and x -axis.

So the sample gradient model can be calculated by

$$\begin{aligned} \bar{L}_x &= f[i, j+1] - f[i, j], \\ \bar{L}_y &= f[i, j] - f[i+1, j], \end{aligned} \quad (10)$$

where i and j stand for the base vectors about y -axis negative direction and x -axis positive direction, respectively.

Canny edge detection function adopts a kind of gradient operator Sobel. Each pixel in the image uses this operator, and then the corresponding gradient vector or normal vector will be generated. The operator contains two three-dimensional matrices, and the modulus value of gradient can be expressed as

$$H = \sqrt{L_x^2 + L_y^2}. \quad (11)$$

Partial derivative can be calculated by

$$\begin{aligned} L_x &= (\beta \cdot n_3 + n_2 + n_4) - (\alpha \cdot n_7 + n_0 + n_6), \\ L_y &= (\beta \cdot n_1 + n_0 + n_2) - (\alpha \cdot n_5 + n_4 + n_6), \end{aligned} \quad (12)$$

where the constant values $\alpha = 2$ and $\beta = 2$ and then the above L_x and L_y can be expressed as the below convolution template:

$$L_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad L_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}. \quad (13)$$

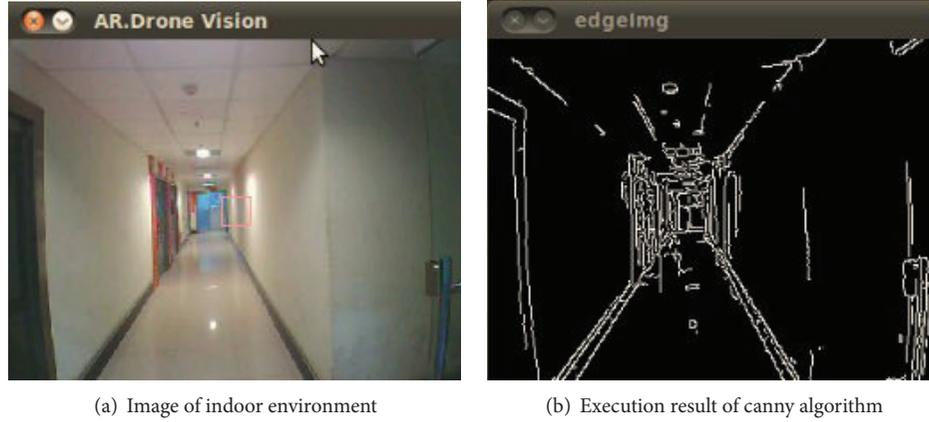


FIGURE 3: Edge detection of indoor flight environment.

Neighbourhood pixel point can be expressed as

$$L_{nb} = \begin{bmatrix} m_0 & m_1 & m_2 \\ m_7 & [i, j] & m_3 \\ m_6 & m_5 & m_4 \end{bmatrix}. \quad (14)$$

So the emphasis on calculating Sobel operator is to obtain the central pixel in the area whose size is similar to the template.

(3) *Control of Gradient Value.* The system compares the gray level and direction of current pixel with the pixel gradient value whose position is two pixels prior to current pixel, if it is smaller than the pixel gradient value of prior pixel, and then it would be set as 0, which means it is not an edge.

(4) *Partition of Lag Threshold.* The system could obtain the point whose gradient is changing by the above principle, but for those having high luminance gradient, there would exist some interference factors to judge whether they are edges, so a threshold is introduced to solve this problem. To be exact, Canny uses two lag thresholds of high and low threshold. If the important edge is continuous curve in the image, then the system can track the obscure part of the given curve and avoid to regard these noise pixels which actually do not form curve as edges.

The system begins this process with a large threshold, firstly marks these luminance gradients having high edge probability, and then tracks them in the whole image by the calculated direction information. For the process of edge tracking, the system uses the low threshold to control tracking path and makes the tracking path finally return the beginning position along with the obscure part.

Figure 3(a) shows the indoor environment captured by the front camera of quad-rotor helicopter. System stores the detecting results in the form of binary image, and Figure 3(b) is the edge effect picture using the above edge detecting process. Experimental results show the basic edge information can be successfully extracted.

3.2.3. Randomized Hough Transformation (RHT) Method Used to Extract Straight Lines. The main idea used to detect

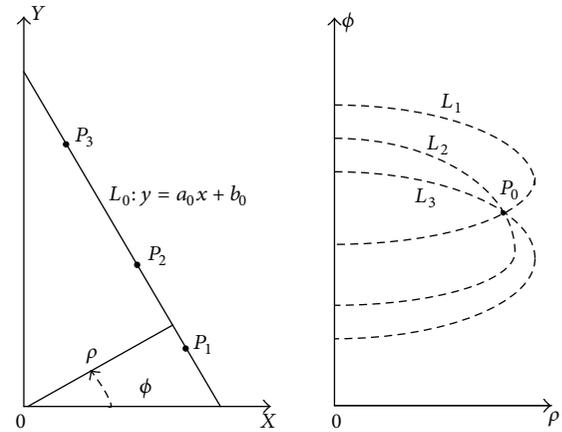


FIGURE 4: The mapping relation for a line in the image space and parameter space.

straight lines is to establish a mapping between image coordinate space and parameter coordinate space. Many pixels in the image space are mapped into the parameter space, if these pixels are mapped into the same point in the parameter coordinate, and then the counter will be increased by 1. The point's coordinate having largest counter value is corresponding to a straight line in the image space.

As Figure 4 shows, for any three points P_1 , P_2 , and P_3 in the straight line $Y = a_0x + b_0$ in the image space, they will form three sinusoidal curves L_1 , L_2 , and L_3 while they are mapped into the parameter space, and the intersection point P_0 of the three curve is corresponding to the straight line L_0 in the image space.

This paper uses improved randomized Hough transformation for the quad-rotor helicopter by using randomized sampling pixels in the image space and mapping them into parameter space. System firstly randomly picks two or n pixels and maps them into one point in the parameter space, then uses a set P with each element containing both a real-valued vector and an integer score to represent the parameter space, and updates the set P at each step by the point mapped from

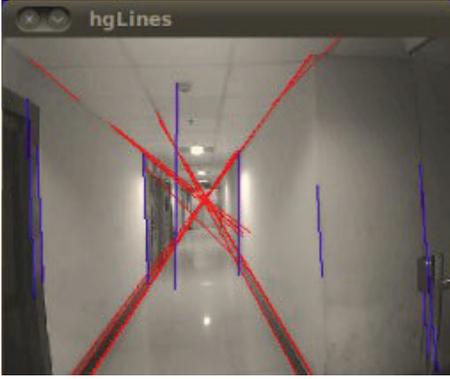


FIGURE 5: Straight lines extraction.

the randomly picked pixels. Finally, system finds accumulated cells of the detected curves [10].

RHT method uses many-to-one mapping so as to avoid huge amount of calculations in the traditional Hough transformation (HT) method using one-to-many mapping.

The system uses dynamic list structure and allocates corresponding memory space to these parameters only accumulated by the many-to-one mapping. The experimental result about using RHT method to extract straight lines is shown as Figure 5.

The interfering accumulations caused by disturbance pixels can be significantly improved in RHT method. For Figure 5, HT method uses a $320 * 240$ array while RHT method uses the storage of only around 36 cells. Besides, the computing time of HT method is about 2.4 times larger than that of RHT method.

Consequently, RHT method has the advantages of infinite parameter space, small storage, and high computation speed, which can better satisfy the real-time control of the quadrotor helicopter.

4. Implementation of Vanishing Point Algorithm

Vanishing point is obtained by evaluating the straight lines extracted from collected image. As in the indoor environment, there inevitably existed some disturbing lines, so the emphasis on this vanishing point algorithm is to reduce the effect of error component and estimate the exact position of vanishing point.

4.1. The Vanishing Point Algorithm.

(1) *Deleting Unreasonable Straight Line.* There is a straight line set as $C\{l_0, l_1, \dots, l_k \mid k = 0, \dots, M\}$, where M is the number of straight lines system that has been detected and k is integer. Each line is expressed by two endpoints $\{(x_{s0}, y_{s0}), (x_{e0}, y_{e0})\}, \{(x_{s1}, y_{s1}), (x_{e1}, y_{e1})\}, \dots, \{(x_{sM}, y_{sM}), (x_{eM}, y_{eM})\}$.

For endpoints set as C_p , the angle between each line and horizontal coordinate axis can be calculated by

$$\text{deg} = \arctan \frac{|y_1 - y_0|}{|x_1 - x_0|}. \quad (15)$$

There are some straight lines in the indoor scenario such as the edges of ceilings, the edges of floors, etc., and these lines are disturbing lines in the calculation of the vanishing point. Therefore, the removal of these lines is essential to accurately calculate the vanishing point and alleviate the error of the algorithm.

Unreasonable lines in the image are deleted by means of judging the deg angle. As OpenCV library function is used by this system, so the value ranges from 0 to 360 degrees after transformation. Judging about whether a line is close to horizontal or vertical straight lines depends on

$$\begin{aligned} f(\text{deg}) &= 1, \\ \text{VT} &> |90 - \text{deg}|, \\ \text{VT} &> |270 - \text{deg}|, \\ \text{HT} &> |\text{deg}|, \\ \text{HT} &> |180 - \text{deg}|, \end{aligned} \quad (16)$$

where VT (vertical line threshold) and HT (horizontal line threshold), respectively, represent the vertical and horizontal angle thresholds used to delete undesired lines.

In order to obtain the optimized value of VT and HT, we do the below test experiments so as to make the coverage rate to achieve the max value. Supposing the error radius is r for the vanishing point, so these straight lines will generate some points both inside and outside the circular error of vanishing point. The coverage rate is defined as $I_i/(I_i + O_k)$ used to mean the ratio of points inside the circular error of vanishing point accounts for the percentage of total points, where i and k are both positive integers.

As Figure 6(a) shows, there exist many intersections caused by straight lines, and these intersections are enlarged in order to be seen clearly; for example I_1 and I_2 (inside the circular error of vanishing point) and O_1, O_2, O_3 , and O_4 (outside the circular error of vanishing point) consist in straight line L_1 . If L_1 is deleted, then these six points are all deleted, which will affect the coverage rate. So using the threshold of horizontal line and vertical line is necessary so as to reduce the calculation work and improve coverage rate.

The values of VT and HT are both integers so as to lower the calculation. While VT and HT are both 0, which means no straight lines are deleted in this case, $\text{VT} + \text{VT} > 90$, which means all straight lines will be deleted. So choosing a suitable value to make the algorithm have the best accuracy is the key point. Figure 6(b) shows the test results about the relationship of VT and HT affecting the coverage rate. The corresponding value of VT and HT is 10 and 15, respectively, while the coverage rate reaches to its peak value.

Figure 6(b) shows the coverage rate affected by the thresholds of VT and HT. Because VT and HT both vary from 0 to 90 degrees, so there are total 810 image frames (while accuracy is 1 degree) in Figure 6(b). As the coverage rate will affect the outcome accuracy of vanishing point, which directly affects the flight deviation, the experiment of these 810 image frames well reflects the flight deviation according to

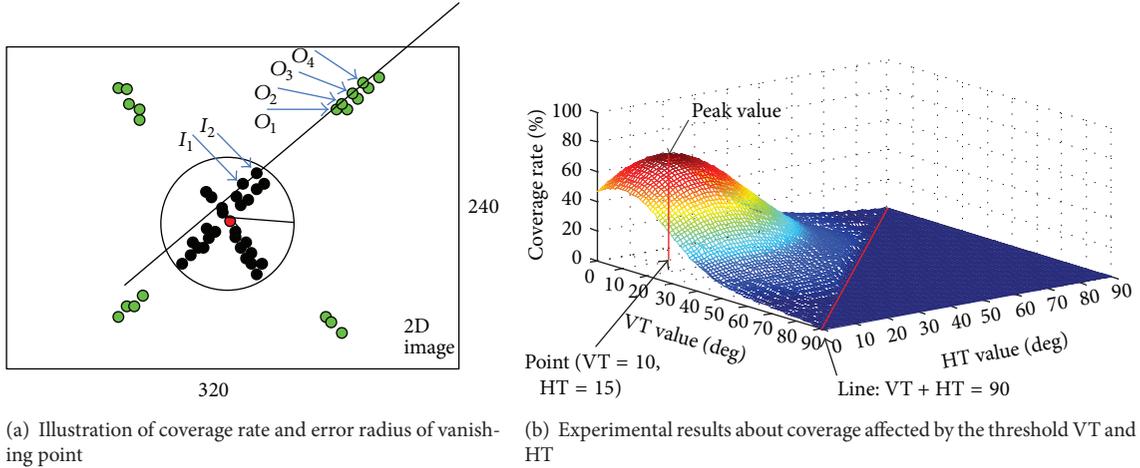


FIGURE 6: Error radius of vanishing point and coverage rate affected by the threshold VT and HT.

the thresholds VT and HT while the number of images frames increase.

For those lines making $f(\text{deg}) = 1$ is undesirable, so those lines should be deleted, and they are marked as blue in Figure 5.

(2) *Calculating the Intersection Point.* After deleting undesirable lines, the number of lines in the line set is M , so the number of their intersections is $N = M \cdot (M - 1)/2$, and it is stored as the initial value in the intersection array.

Any two straight lines in image can be expressed as

$$\begin{aligned} A_1x + B_1y + C_1 &= 0, \\ A_2x + B_2y + C_2 &= 0. \end{aligned} \quad (17)$$

Points $A(x_1, y_1)$ and $B(x_2, y_2)$ are on a straight line while points $C(x_3, y_3)$ and $D(x_4, y_4)$ are on another straight line. Then distance d between the two lines can be calculated by

$$d = (x_4 - x_3) \cdot (y_2 - y_1) - (x_2 - x_1) \cdot (y_4 - y_3). \quad (18)$$

While d is 0, it means straight lines AB and CD form parallel line; otherwise, they are intersection lines, and then the coordinate of their intersection point can be calculated by

$$\begin{aligned} x_0 &= x_1 \cdot (x_4 - x_3) \cdot (y_2 - y_1) \\ &+ (x_2 - x_1) \cdot (x_4 - x_3) \cdot (y_3 - y_1) \\ &- \frac{x_3}{d} \cdot (x_2 - x_1) \cdot (y_4 - y_3), \\ y_0 &= y_1 \cdot (x_2 - x_1) \cdot (y_4 - y_3) \\ &+ (x_3 - x_1) \cdot (y_2 - y_1) \cdot (y_4 - y_3) \\ &+ \frac{y_3}{d} \cdot (x_4 - x_3) \cdot (y_2 - y_1). \end{aligned} \quad (19)$$

Calculate all the intersections by the above equations, where n is the number of all intersection points, and their coordinates are expressed as $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$,

respectively. Those intersection points whose coordinate does not exist in the range of $(0, 0)$ to (h, w) are undesirable ones, and they should be deleted, where h and w are the height and weight of the image.

The system uses Canny algorithm to obtain the edge of the collected image and then adopts RHT method to extract straight lines. From the analysis of abundant practical flight experimental results, we can know that the successful rate about collecting elements of intersection point set can reach to above 92.5%.

4.2. Proposed VPFE Algorithm Based on Clustering Principle.

As the bottleneck for quad-rotor helicopter navigation is the speed of image processing, it includes the calculation of every pixel in the image. In order to improve the real-time ability of visual navigation, it is important to eliminate some unnecessary calculations.

A new vanishing point fast estimation (VPFE) algorithm is proposed to evaluate the vanishing points and obtain a navigation destination point finally. In fact, it is 1-mean cluster process. The system deals with all the points in the intersection set by the method of convergence. Specifically, the system firstly deletes those points whose distances to mass center are larger than cluster radius, and then adjusts value of the cluster radius by the method of convergence. The above procedure is executed repeatedly, and finally the mass center is obtained as the vanishing point. The total number of iterative procedure can be calculated by

$$R \cdot p^n < R_e. \quad (20)$$

As n is the number of iterative process, the value is

$$n = \left\lceil \log \frac{R_e}{R} \right\rceil + 1, \quad (21)$$

where $p (< 1)$ is the convergence coefficient, R represents radius of convergence, and R_e is a dynamical value that varies from each cluster process, which stands for the deviation of mass center about each cluster process; finally, it is the



FIGURE 7: Clustering processing of VPFE algorithm.

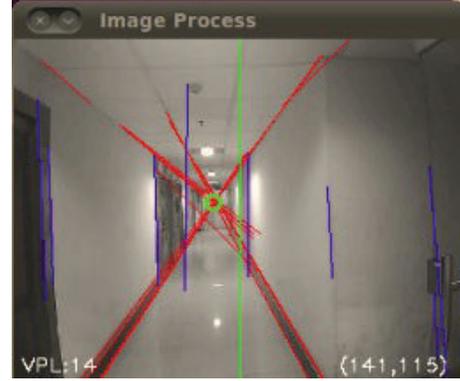


FIGURE 8: Collected image of the indoor environment.

radius of convergence used to describe the region how much vanishing point will be located. In order to facilitate the calculation, an element amount threshold NT is preset.

The procedure of algorithm is described as follows.

Step 1. Set the initialization values of R and R_e and calculate the n according to R and R_e .

Step 2. The new radius of convergence is $p \cdot R$.

Step 3. Calculate the mass center of the intersection points set as C and regard it as the vanishing point.

Step 4. For all the elements in the intersection point set as C , calculate the distance between intersection point and vanishing point.

Step 5. If the number of elements in the intersection points is smaller than NT or the amount of iterative process is larger than n , then go to Step 6; otherwise, return to Step 2.

Step 6. Output value of the mass center in the intersection point set is regarded as the position of the vanishing point.

The verification result of the above algorithm is shown as Figure 7. The green points are the undesirable ones who do not meet the requirement of threshold in the convergence process, and these points should be deleted. The red marker is the vanishing point obtained from the above VPFE algorithm, and the corresponding image in the indoor environment collected by the front camera of the quad-rotor helicopter is shown as Figure 8.

5. Navigation Based on PID Control

The implementation of quad-rotor helicopter navigation is composed of image data collection and receiving control command, which forms a closed-loop control system [11, 12]. Besides, the control command calculated by PID control forms another closed-loop control processing as Figure 9 shows.

In this experiment, the destination point is the vanishing point for the quad-rotor helicopter, and the image central

position collected by the quad rotor is its current position. The quad-rotor helicopter modifies the image central position according to the vanishing position. Suppose the image central position is $O(x_o, y_o)$ and the vanishing point coordinate is $VP(x_{vp}, y_{vp})$, so the error for the navigation can be expressed as

$$\begin{aligned} dx &= x_{vp} - x_o, \\ dy &= y_{vp} - y_o. \end{aligned} \quad (22)$$

The system adapts to PID to implement the process of navigation by close-loop control. In fact, the effect of PID control is to decrease the difference of current position and the vanishing point position till the difference reaches to a certain threshold [13]:

$$V = K_p \cdot e + K_I \cdot \sum e + K_D \cdot de, \quad (23)$$

where e_1 is the difference between current value and real value, e_2 represents accumulated error sum for the period ranges from beginning of current close-loop control to current control, e_3 stands for difference between the current control difference and last control difference in the process of the same close-loop control, and V is the real parameter to control the quad rotor. This navigation process is implemented by adjusting the three attitude angles of yaw, roll, and pitch.

The adjustment includes direction of three-dimension axis, dy is the position difference of y -axis used to control the vertical flight, which is corresponding the height control of quad-rotor helicopter, and dx is the position difference of x -axis used to control the horizontal flight, which is corresponding the yaw angle and roll angle control of quad-rotor helicopter [14]:

$$e_i = dx_i = x_{vi} - x_o \quad (24)$$

$$\text{out} = K_p \cdot e_i + \alpha \cdot K_I \cdot \sum e_i + K_D \cdot (e_i - e_{i-1}),$$

where α is defined by

$$\alpha = \begin{cases} 0, & |e_i| > \varepsilon, \\ 1, & |e_i| < \varepsilon, \end{cases} \quad (25)$$

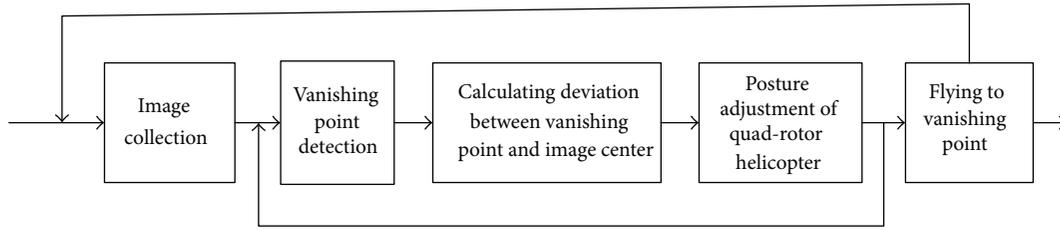


FIGURE 9: The workflow of quad-rotor helicopter navigation control.

where ε is the error threshold of integral separation while, under the condition of y -axis and z -axis, the e_i is $dy_i = y_{vi} - y_o$ and $dz_i = z_{vi} - z_o$, respectively.

Due to the fact that the control of quad-rotor helicopter belongs to complex nonlinear control, so in the process of PID control, while biased error became larger, the current integral separation parameter should be set as 0 to close integral action in order to eliminate shake, and while biased error became smaller, the current integral separation parameter should be set as 1 so as to open integral action.

In order to test the performance of improved PID control, indoor flight experiments about testing flight angle (roll, pitch, and yaw) and flight deviation (horizontal and vertical) are executed.

The quad-rotor helicopter is AR.Drone which is developed by Parrot Corporation in France, and it is designed based on four-axis intelligent controlling by WIFI, and it has the sensors such as three-axis accelerometer, two-axis gyrometer, one-axis yaw gyrometer, and ultrasonic wave height meter. Besides, it has a front camera (320 * 240 pix, 15 fps) and bottom camera (176 * 144 pix, 60 fps). AR.Drone has an ARM9 microcontroller which is based on Linux OS development platform.

Quad-rotor helicopter collects the sensors data such as gyroscope, accelerometer, and altimeter and system further calculates the navigation data such as posture information of pitch, roll, yaw, and height. The quad rotor sends these navigation data and video streaming to PC through WIFI [15]. Then PC refreshes the current navigation data and handles the video streaming including decoding, video navigation, and image handling algorithms. After PC calculating the navigation error according to the output of visual navigation, then system sends the navigation error to the PID controller and finally obtains the output of flight control command. The output is sent to quad rotor by WIFI to control its flight effectively [16].

Experiment 1. While the quad-rotor helicopter hovers in the indoor environment, disturbance is imposed to the three angles of roll angle, pitch angle, and yaw angle at the time 22 s. The response process was reflected as Figures 10(a), 10(b), and 10(c) show. The experimental results show that the improved PID control is better than original PID control in terms of overshoot and adjustment time.

Experiment 2. While the quad-rotor helicopter hovers in the indoor environment, disturbance is imposed in the direction of horizontal and vertical at the time 22 s. The response

process was reflected as Figures 10(d) and 10(e) show. The experimental results show that the improved PID control is also better than original PID control as to overshoot and adjustment time.

Experiment 3. The quad-rotor helicopter was flied to the coordinate of destination point (3,3,3) from the coordinate of original point (1,1,1) in order to show the performance test of flight stability for the improved PID. The experimental results about x , y , and z coordinate axes are shown in Figures 11(a), 11(b), and 11(c). The experimental results show that the improved PID control is also better than original PID control in terms of overshoot and adjustment time.

From the analysis of the above experimental results, no matter whether disturbance is imposed in the quad-rotor helicopter, we can know that the improved PID can well conduct the stability of flight for the quad-rotor helicopter in the indoor environment compared with the original PID.

6. Flight Experimental Results in the Indoor Environment

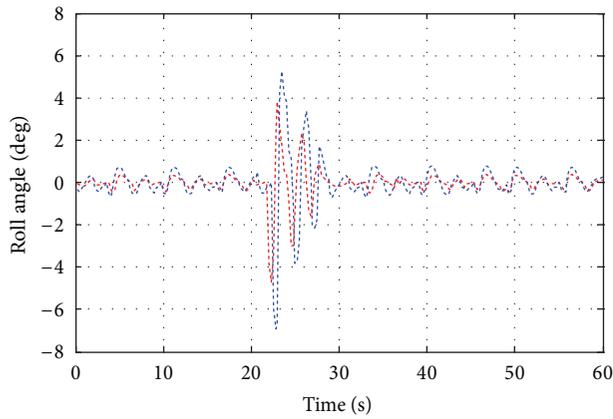
In order to test the performance of the VPFE algorithm, we test the flight of the quad-rotor helicopter in the indoor environment as Figure 3(a) shows. Figure 12 shows two frames of the software display image in PC. Each image describes the video image collected by the quad rotor after the corresponding handling by the above algorithm. A smaller picture in the right bottom of the image is used to show the flight scene of the quad-rotor helicopter at the corresponding time point.

In the below experiments, we measure the system efficiency in terms of algorithm execution time, and the experimental results about calculating vanishing point for these continuous 150 frames are shown in Figure 13.

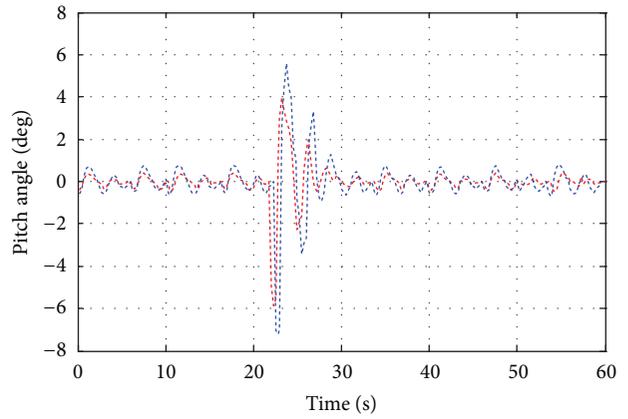
As we can see from the results, the average execution time for the whole algorithm is 26.46 ms, which can meet the indoor flight environment for AR.Drone whose video flowing returning rate is 15 fps [17, 18].

There are two other kinds of algorithm used to detect the vanishing point.

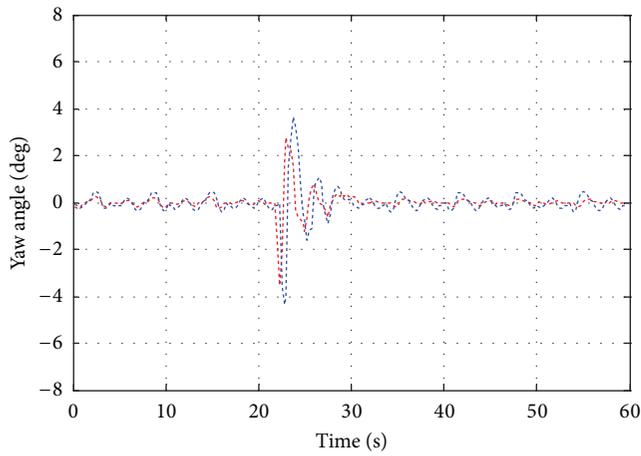
(1) *Spatial Alternation Technique.* Information of image coordinate space is mapped into another coordinate space, and then operations of deleting straight lines are executed. Gaussian ball transformation method is proposed by Tayebi and McGilvray [18]; after the process of mapping, the straight



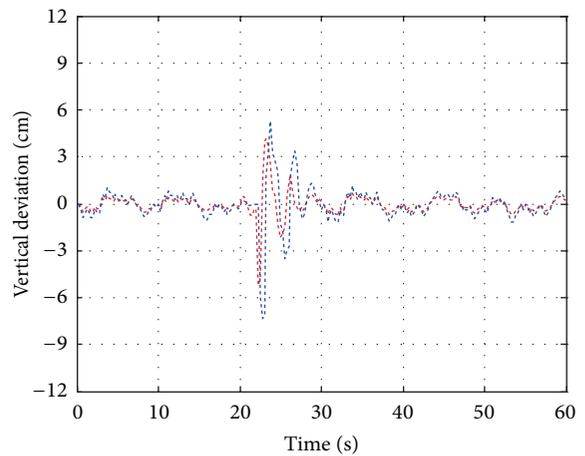
(a) Flight test of roll angel



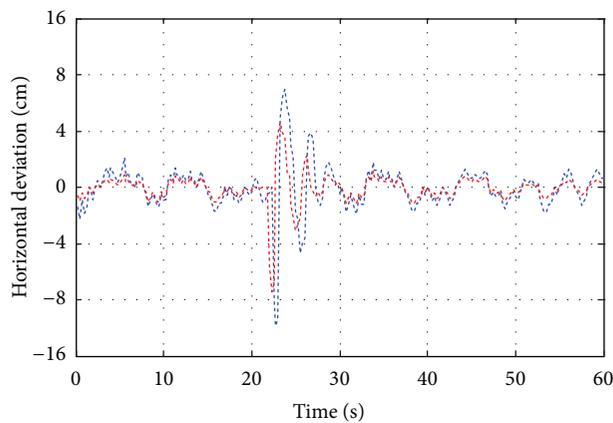
(b) Flight test of pitch angel



(c) Flight test of yaw angle



(d) Flight test of vertical deviation



(e) Flight test of horizontal deviation

FIGURE 10: Performance comparison about flight angle and flight deviation.

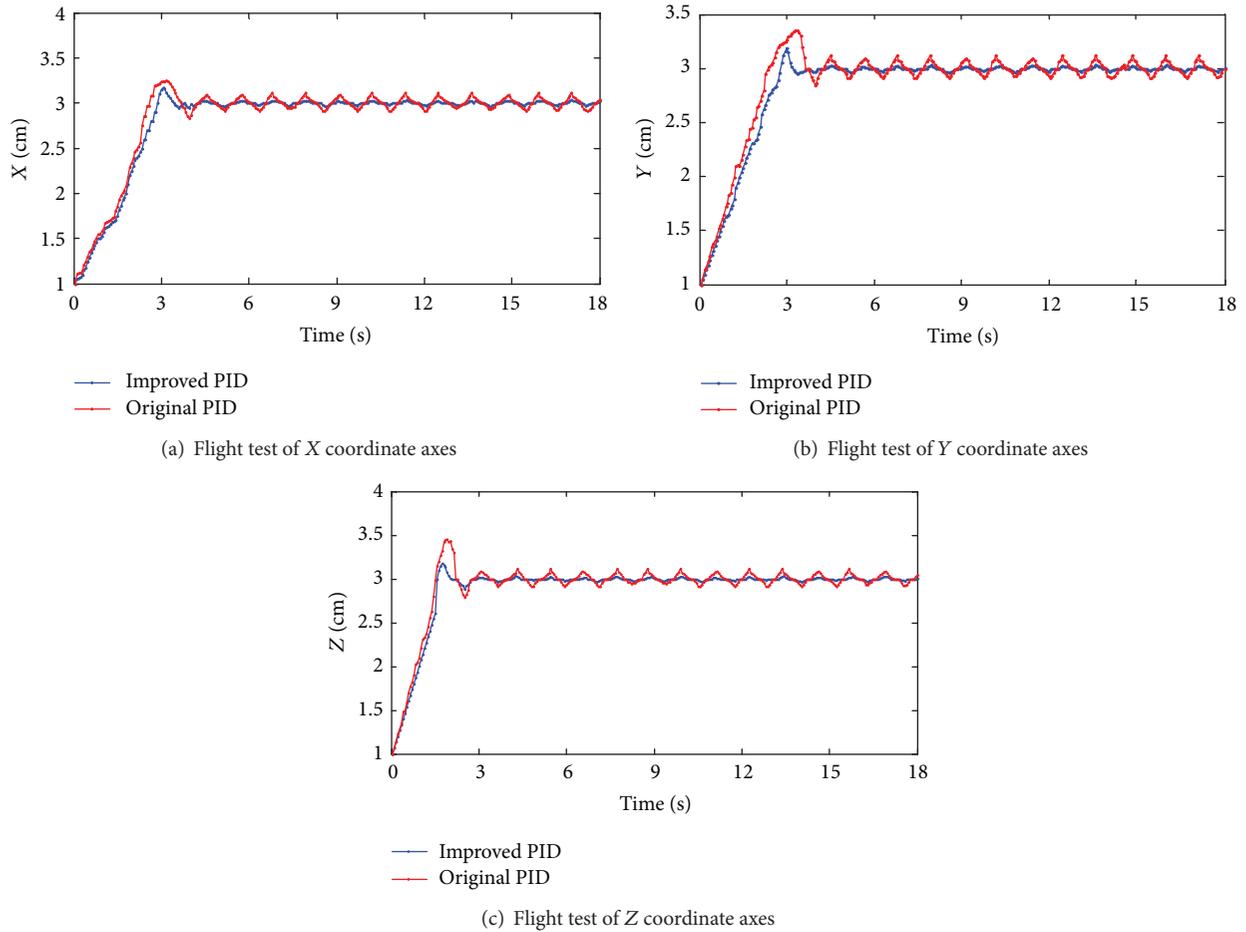


FIGURE 11: Performance test of flight stability.

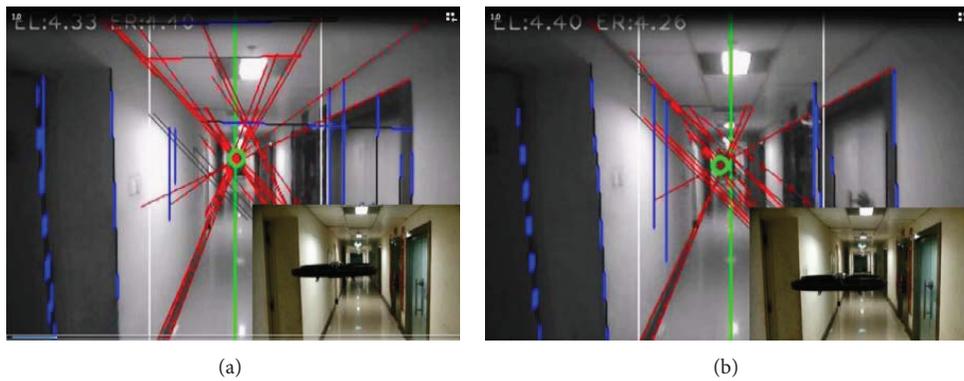


FIGURE 12: Flight test in the indoor environment.

lines and spatial location information of vanishing point is decreased at the cost of missing the information of straight line length and the distance between straight line and vanishing point. In the meantime, the calculation accuracy is affected by the accuracy of accumulate units.

(2) *Statistical Estimation Method.* Parameter of lines is estimated by the feature points in the edge of image, and these

parameters are used to calculate the vanishing point, or cost function is constructed using vanishing point and feature points of edge [19], but in the practical application, the algorithm is having higher computation complexity and lower computational efficiency.

This paper proposes algorithm mainly using straight line information. Straight line information is directly used to obtain the vanishing point in the image, and then the

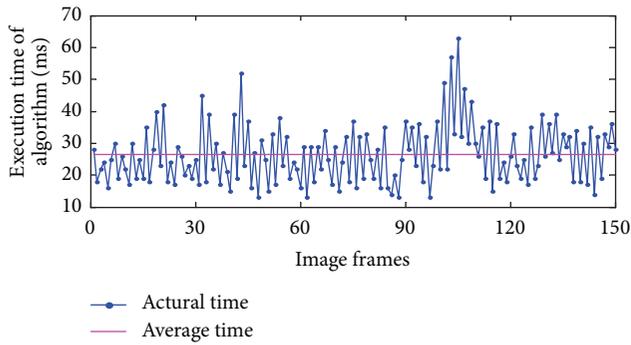


FIGURE 13: The efficiency of VPFE algorithm.

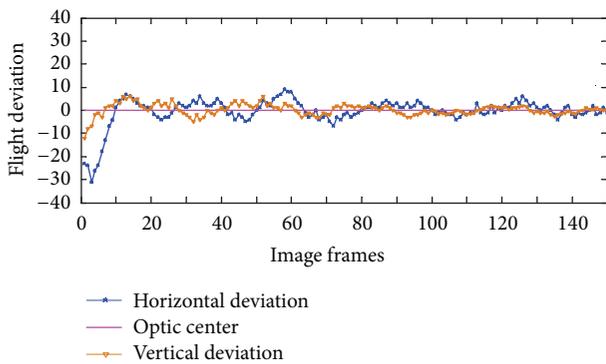


FIGURE 14: The horizontal deviation of autonomous navigation.

intersection of these straight lines is calculated, and this process is implemented by calling the matrix operations function in the OpenCV. Finally clustering principle is used to obtain the vanishing point so as to reduce the computation complexity of the algorithm.

The average execution time of proposed algorithm is about 26.46 ms. Compared with the other two vanishing point detection algorithms, the experimental results show that the execution time is separately about 13.7% and 21.3% shorter than Gaussian ball transformation method and statistical estimation method [20]. Consequently, the proposed algorithm has better real-time ability than the other two vanishing point detection algorithms.

Figure 14 shows the horizontal and vertical deviations between vanishing point and optic center for these continuous 150 image frames. In this experiment, the flight speed for the quad-rotor helicopter is constant value (0.4 m/s). As we can see from the experimental results, the quad-rotor helicopter can be well controlled during the whole process of autonomous navigation flight in the indoor environment [21, 22].

7. Conclusions

After the implementation of preprocessing of image, deleting noise interference, edge extracting using Canny operator, and extracting straight lines by RHT, the system finally obtains the position of vanishing point through the proposed VPFE

algorithm and effectively operates the autonomous navigation of the quad-rotor helicopter by continuous calculation about the navigation error. The experiment shows that it can implement the destination navigation toward the calculated vanishing point well in the indoor environment, and the experimental results show that its average execution time for the whole algorithm is 26.46 ms per image frame, besides the flight deviation error can well be controlled into the acceptable range, so it can meet the requirements of real-time and good stability in the indoor environment.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is partly supported by the National Natural Science Foundation of China (no. 61101121), the National High Technology Research and Development Program (863 Program) (no. 2013AA102505), the Key Laboratory Project Funds of Shenyang ligong University under Grant no. 4771004kfs03, and the Educational Committee of Liaoning Province Science and Technology Research Projects under Grant no. L2013096. The authors thank Jan Vitek, Ales Plsek, and Lei Zhao for their help while the first author studied at the Purdue University as a Visiting Scholar from September, 2010, to September, 2012. The authors also thank the anonymous reviewers for their valuable comments.

References

- [1] <http://drawsketch.about.com/od/drawingglossary/g/vanishingpoint.htm>.
- [2] G. Hoffmann, D. G. Rajnarayan, S. L. Waslander, D. Dostal, J. S. Jang, and C. J. Tomlin, "The Stanford testbed of autonomous rotorcraft for multiagent control," in *Proceedings of the 23rd Digital Avionics Systems Conference*, vol. 2, October 2004.
- [3] M. Achtelika, A. Bachrach, R. He, S. Prentice, and N. Roy, "Autonomous navigation and exploration of a quadrotor helicopter in GPS-denied indoor environments," in *Proceedings of the Robotics: Science and Systems*, 2008.
- [4] F. Kendoul, I. Fantoni, and K. Nonami, "Optic flow-based vision system for autonomous 3D localization and control of small aerial vehicles," *Robotics and Autonomous Systems*, vol. 57, no. 6-7, pp. 591–602, 2009.
- [5] S. Grzonka, G. Grisetti, and W. Burgard, "A fully autonomous indoor quadrotor," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 90–100, 2012.
- [6] P. Pounds, R. Mahony, and P. Corke, "Modelling and control of a large quadrotor robot," *Control Engineering Practice*, vol. 18, no. 7, pp. 691–699, 2010.
- [7] F. Schaffalitzky and A. Zisserman, "Planar grouping for automatic detection of vanishing lines and points," *Image and Vision Computing*, vol. 18, no. 9, pp. 647–658, 2000.
- [8] J. A. Shufelt, "Performance evaluation and analysis of vanishing point detection techniques," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 3, pp. 282–288, 1999.

- [9] K. Seo, J. Lee, and H. Choi, "An efficient detection of vanishing points using inverted coordinates image space," *Pattern Recognition Letters*, vol. 27, no. 2, pp. 102–108, 2006.
- [10] L. Xu, E. Oja, and P. Kultanen, "A new curve detection method: randomized Hough transform (RHT)," *Pattern Recognition Letters*, vol. 11, no. 5, pp. 331–338, 1990.
- [11] G. M. Hoffmann, H. Huang, S. L. Waslander, and C. J. Tomlin, "Precision flight control for a multi-vehicle quadrotor helicopter testbed," *Control Engineering Practice*, vol. 19, no. 9, pp. 1023–1036, 2011.
- [12] C. Nicol, C. J. B. MacNab, and A. Ramirez-Serrano, "Robust adaptive control of a quadrotor helicopter," *Mechatronics*, vol. 21, no. 6, pp. 927–938, 2011.
- [13] T. Dierks and S. Jagannathan, "Output feedback control of a quadrotor UAV using neural networks," *IEEE Transactions on Neural Networks*, vol. 21, no. 1, pp. 50–66, 2010.
- [14] G. V. Raffo, M. G. Ortega, and F. R. Rubio, "An integral predictive/nonlinear H_∞ control structure for a quadrotor helicopter," *Automatica*, vol. 46, no. 1, pp. 29–39, 2010.
- [15] G. R. Rodríguez-Canosa, S. Thomas, J. del Cerro, A. Barrientos, and B. MacDonald, "A real-time method to detect and track moving objects (DATMO) from unmanned aerial vehicles (UAVs) using a single camera," *Remote Sensing*, vol. 4, no. 4, pp. 1090–1111, 2012.
- [16] M. Bonak, D. Matko, and S. Blai, "Quadrocopter control using an on-board video system with off-board processing," *Robotics and Autonomous Systems*, vol. 60, no. 4, pp. 657–667, 2012.
- [17] J. F. Guerrero-Castellanos, N. Marchand, A. Hably, S. Lesecq, and J. Delamare, "Bounded attitude control of rigid bodies: real-time experimentation to a quadrotor mini-helicopter," *Control Engineering Practice*, vol. 19, no. 8, pp. 790–797, 2011.
- [18] A. Tayebi and S. McGilvray, "Attitude stabilization of a VTOL quadrotor aircraft," *IEEE Transactions on Control Systems Technology*, vol. 14, no. 3, pp. 562–571, 2006.
- [19] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, Cambridge, UK, 2nd edition, 2003.
- [20] A. Minagawa and N. Tagawa, "Vanishing point and vanishing line estimation with line clustering," *IEICE Transactions on Information & Systems*, vol. 7, pp. 1574–1582, 2000.
- [21] W. Sun, H. Gao, and B. Yao, "Adaptive robust vibration control of full-car active suspensions with electrohydraulic actuators," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 6, pp. 2417–2422, 2013.
- [22] W. Sun, Y. Zhao, J. Li, L. Zhang, and H. Gao, "Active suspension control with frequency band constraints and actuator input delay," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 1, pp. 530–537, 2012.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

