A Pose Estimation Method of a Moving Target Based on Off-board Monocular Vision*

Fu Jia-Ning, Fu Qiang, Usman Arif and Quan Quan

***Abstract*—Pose estimation plays a vital role in landing guidance of quadcopter unmanned aerial vehicles. In this paper, a continuous-time optimization method is proposed to estimate the pose of a moving object only using an off-board camera. The optimization time that can vary depending upon the desired accuracy and computational speed, and it is finally transferred into solving an ordinary differential equation. Based on the proposed method, a pose estimation system with a camera and a pan-tilt unit is established. The system accuracy and algorithm correctness are tested in an experiment with a moving board as the target. Experimental results indicate that the proposed method can track a moving target and estimate its position and attitude in real time. Therefore, it will be promising to use this method in landing guidance of quadcopter unmanned aerial vehicles.**

I. INTRODUCTION

In the landing guidance process of quadcopter Unmanned Aerial Vehicles (UAVs), one important issue is how to acquire position and attitude parameters, which are used as inputs to the flight control system. Moreover, these parameters are required to be determined precisely, even at low altitude and for small translation motions. This makes pose estimation of the quadcopter UAV different from that used for flying guidance or the fixed-wing UAV landing guidance.

During the past decades, researchers have been putting a significant effort and some pose estimation methods have been developed. Global Position System (GPS), Differential Global Position System (DGPS), Global Navigation Satellite System (GNSS) and some other electromagnetic navigation technologies are the most commonly used methods [1-3]. There are also some methods for GPS-denied environment which involve the use of the Inertial Measurement Unit (IMU) or Ultra-Wideband (UWB) as means for navigation [4,5]. Another GPS-independent method is based on computer vision. For example, a UAV can be guided to land on a pad [6] or map the terrain [7] through vision. While most vision based methods are on-board [6-9], there are also off-board methods like infrared stereo vision systems [10]. However, the methods mentioned have some disadvantages: During low altitude flight and in urban areas, GPS systems may lose satellite signals. If a quadcopter UAV uses electromagnetic navigation, it will need to have more sensors which increases its payload. For on-board vision navigation methods, not only the equipment will increase UAV's payload, but it also requires a prefabricated landing pad or a specific landing target position.

Therefore, it is natural and convenient to develop a new off-board monocular vision method for the landing guidance of quadcopter UAVs, which should provide real-time pose data. For a practical system, the vision sensor should track the target as well, therefore the vision system includes an active target tracking device. Specifically, a Pan-Tilt Unit (PTU) is used in experiment to rotate the camera.

In computer vision, determining the relative pose between an object and a calibrated camera is often known as the Perspective-n-Point (PnP) problem [11]. They can be classified into two categories [11]: iterative methods [12] and non-iterative methods [13-15]. Our method of pose estimation is a P4P problem solution. Specifically, it is a continuous time optimization method and is solved by computer iteratively. Besides, our method has some resemblance with [16] but more concise.

The contribution of this paper includes an optimization method that estimates a moving target pose in real time. This method is a continuous-time optimization method which estimates relative pose between image pixel and camera system. Meanwhile, an experiment is also presented, which calculates the real-time attitude and position of the moving target.

This paper is organized as follows. Section II presents preliminaries and problem formulation; Section III describes the main algorithm of pose estimation; In Section IV, experiment details and results are provided, and accuracy is evaluated; In Section V, this paper is concluded with final discussions and some future work.

We use the following notation. $\mathbb{R}^n$ denotes Euclidean space of dimension n. $\boldsymbol{R}_A^B$ and $\boldsymbol{T}_A^B$ denote rotation matrix and translation vector from coordinate system $\{A\}$ to coordinate system $\{B\}$, respectively.

II. PRELIMINARIES AND PROBLEM FORMULATION

A. Coordinate system transformation

Let $\boldsymbol{P} \in \mathbb{R}^3$, vectors $\boldsymbol{P_A} = [X_A \;\; Y_A \;\; Z_A]^{\mathrm{T}}$ and $\boldsymbol{P_B} = [X_B \; Y_B \; Z_B]^{\mathrm{T}}$ be the coordinates of P in two different coordinate system $\{A\}$ and $\{B\}$ respectively. In Euclidean geometry, they satisfy [17]:

*Research supported by National Natural Science Foundation of China under grant #61473012 and #51375462.

Fu Jia-Ning is with School of Automation Science and Electrical Engineering, Beihang University, Beijing, 100191 China (phone: 86-15210966735; e-mail: fjnty@buaa.edu.cn).

Fu Qiang is with School of Automation Science and Electrical Engineering, Beihang University, Beijing, 100191 China. (e-mail: fq@buaa.edu.cn).

Usman Arif is with School of Automation Science and Electrical Engineering, Beihang University, Beijing, 100191 China. (e-mail: usman.arif3711@yahoo.com).

Quan Quan is associate professor of School of Automation Science and Electrical Engineering, Beihang University, Beijing, 100191 China. (e-mail: qq_buaa@buaa.edu.cn).

978-1-4673-8318-9/16/$31.00©2016 IEEE

$$P_B = R_A^B P_A + T_A^B, \qquad (1)$$

where $R_A^B \in \mathbb{R}^{3\times 3}$ is the rotation matrix, and $T_A^B \in \mathbb{R}^3$ is the translation vector.

In this paper, the coordinate systems are defined as follows. The moving target's coordinate system $\{T\}$, is attached with it, and its origin and orientation can be defined arbitrarily. The origin of the camera coordinate system $\{C\}$ is the optical center of the camera, with X_C axis pointing right, Y_C axis pointing downward, and Z_C axis pointing forward. Besides, the origin of ground coordinate system $\{G\}$ is on the ground surface, and its orientation can be defined arbitrarily.

Figure 1 shows the experimental setup along with the coordinate system. O_G-$X_GY_GZ_G$ denotes the ground coordinate system; O_C-$X_CY_CZ_C$ denotes the camera coordinate system; O_T-$X_TY_TZ_T$ denotes the target coordinate system.

The standard definition of the rotations about the three principle axes is presented [18]. A rotation of ψ radians about the x-axis is defined as

$$R_x(\psi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & \sin\psi \\ 0 & -\sin\psi & \cos\psi \end{bmatrix}.$$

Similarly, a rotation of θ radians about the y-axis is defined as

$$R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix}.$$

Finally, a rotation of φ radians about the z-axis is defined as

$$R_z(\phi) = \begin{bmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

The angles ψ, θ, and ϕ are the Euler angles.

In this paper, the rotation matrix from the target coordinate system to the camera coordinate system R_T^C satisfies:

$$R_T^C = R_z(\phi) R_y(\theta) R_x(\psi), \qquad (2)$$

Moreover, the rotation matrix from the camera coordinate system to the ground coordinate system R_C^G satisfies:

$$R_C^G = R_x\left(\frac{\pi}{2}\right) R_z(\beta) R_x(\alpha),$$

where α and β are the rotation angles about the PTU Horizontal axis and vertical axis separately, and they can be provided by PTU. Due to coordinate system definitions, the original camera coordinate system rotates $\pi/2$ rad about X_G axis relative to the ground coordinate system. Based on the descriptions above, we get:

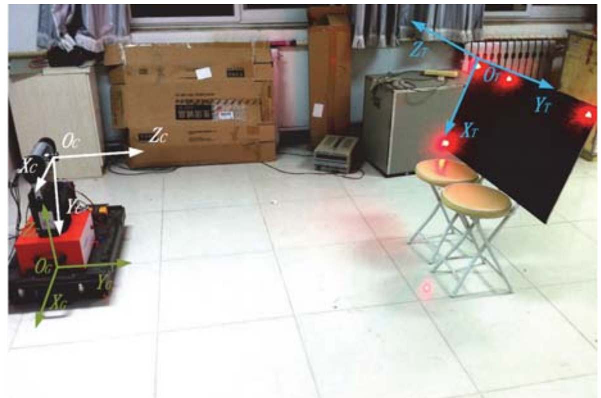


Figure 1. Illustration of various coordinate systems

$$\begin{aligned} P_C &= R_T^C P_T + T_T^C, \\ P_G &= R_C^G P_C + T_C^G, \\ P_G &= R_C^G\left(R_T^C P_T + T_T^C\right) + T_C^G \\ &= R_T^G P_T + T_T^G, \end{aligned} \qquad (3)$$

where $P_C = [X_C \quad Y_C \quad Z_C]^T$, $P_G = [X_G \quad Y_G \quad Z_G]^T$, $P_T = [X_T \quad Y_T \quad Z_T]^T$. In (3), subscript C denotes camera coordinate system; G denotes ground coordinate system; T denotes target coordinate system. Based on (3), the transformation from the ground coordinate system to the target coordinate system is:

$$P_T = \left(R_T^G\right)^{-1}\left(P_G - T_T^G\right) = \left(R_T^G\right)^{-1} P_G - \left(R_T^G\right)^{-1} T_T^G. \qquad (4)$$

Using (4), equations of position parameters can be established and solved.

B. Camera pinhole model

The camera pinhole model (Figure 2) is used to transform P_C in $\{C\}$ and P_T in $\{T\}$ to p which is in image coordinate system (O_I-X_IY_I) as follows,

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha_x & 0 & u_0 & 0 \\ 0 & \alpha_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} X_C \\ Y_C \\ Z_C \\ 1 \end{bmatrix} = M\begin{bmatrix} R_T^C & T_T^C \\ 0^T & 1 \end{bmatrix}\begin{bmatrix} X_T \\ Y_T \\ Z_T \\ 1 \end{bmatrix}, \qquad (5)$$

and

$$M = \begin{bmatrix} \alpha_x & 0 & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{bmatrix},$$

where s in (5) is the scaling factor; M is the camera intrinsic matrix. And α_x, α_y, u_0, v_0 in M are determined by camera calibration [19].

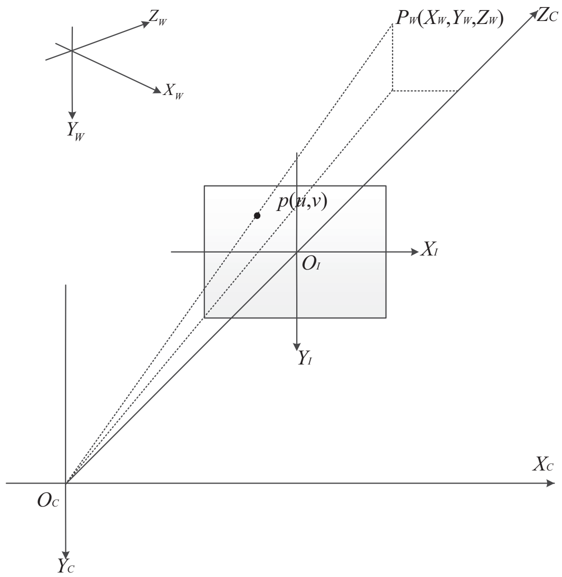


Figure 2. Camera pinhole model

C. Problem formulation

From (5), we get a description of coordinate transformation including pose information, which should be solved using real-time information.

Specifically, in (5), suppose that:

$$\boldsymbol{R}_T^C = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix},$$

$$\boldsymbol{T}_T^C = \begin{bmatrix} t_1 & t_2 & t_3 \end{bmatrix}^T,$$

where the elements of $\boldsymbol{R}_T^C$ are the simplified algebra expressions of (2), and t_1, t_2, t_3 are scalar translation amount of X, Y, Z respectively. By expanding (5), for every point, we can get:

$$u = \frac{(\alpha_x r_{11} + u_0 r_{31}) X_T + (\alpha_x r_{12} + u_0 r_{32}) Y_T + (\alpha_x r_{13} + u_0 r_{33}) Z_T + \alpha_x t_1 + u_0 t_3}{r_{31} X_T + r_{32} Y_T + r_{33} Z_T + t_3},$$

$$v = \frac{(\alpha_y r_{21} + v_0 r_{31}) X_T + (\alpha_y r_{22} + v_0 r_{32}) Y_T + (\alpha_y r_{23} + v_0 r_{33}) Z_T + \alpha_y t_2 + v_0 t_3}{r_{31} X_T + r_{32} Y_T + r_{33} Z_T + t_3}.$$

The above equations can be used directly to solve our problem. Image pixel coordinates u and v, camera intrinsic parameters α_x, α_y, u_0, v_0 and target system coordinates X_T, Y_T, Z_T are known. The three rotation angles ϕ, θ, ψ, which compose the $\boldsymbol{R}_T^C$, and the three elements of the $\boldsymbol{T}_T^C$, namely t_1, t_2, t_3, are to be calculated.

Define $\boldsymbol{X} = [t_1 \ t_2 \ t_3 \ \phi \ \theta \ \psi]^{\mathrm{T}} \in \mathbb{R}^6$. In order to solve $\boldsymbol{X}$, three feature points in a same image are needed at least. If more than three points' pixel coordinates are provided, non-linear equations will be needed to solve.

From (5), for every feature point, subscript i means the i-th point. we have:

$$\begin{aligned} f_{2i-1}(\boldsymbol{X}) &= (\alpha_x r_{11} + u_0 r_{31}) X_{Ti} + (\alpha_x r_{12} + u_0 r_{32}) Y_{Ti} \\ &\quad + (\alpha_x r_{13} + u_0 r_{33}) Z_{Ti} + \alpha_x t_1 + u_0 t_3 \\ &\quad - u_i (r_{31} X_{Ti} + r_{32} Y_{Ti} + r_{33} Z_{Ti} + t_3), \\ f_{2i}(\boldsymbol{X}) &= (\alpha_y r_{21} + v_0 r_{31}) X_{Ti} + (\alpha_y r_{22} + v_0 r_{32}) Y_{Ti} \\ &\quad + (\alpha_y r_{23} + v_0 r_{33}) Z_{Ti} + \alpha_y t_2 + v_0 t_3 \\ &\quad - v_i (r_{31} X_{Ti} + r_{32} Y_{Ti} + r_{33} Z_{Ti} + t_3). \end{aligned} \quad (i=1,2,3,4). \tag{6}$$

Theoretically, $f_j(\boldsymbol{X})$ (j=1, 2, …, 7, 8) should equal to zero. However, due to the noise, we have to solve using an optimization method. Therefore we have:

$$\min_{\boldsymbol{X} \in \mathbb{R}^6} F(\boldsymbol{X}) = \frac{1}{2} \boldsymbol{f}(\boldsymbol{X})^T \boldsymbol{f}(\boldsymbol{X}), \tag{7}$$

Where $\boldsymbol{f}(\boldsymbol{X}) = [f_1(\boldsymbol{X}) \ f_2(\boldsymbol{X}) \ \cdots \ f_7(\boldsymbol{X}) \ f_8(\boldsymbol{X})]^{\mathrm{T}} \in \mathbb{R}^8$ is a column vector.

Next, the proposed method will be introduced in Section III in detail. This method will focus on (7) to obtain six pose parameters.

III. Main Algorithm of Pose Estimation

In order to optimize (7) to its minimum value, we take the time derivate of the objective function $F(X)$, and get

$$\frac{dF(\boldsymbol{X})}{dt} = \frac{dF(\boldsymbol{X})}{d\boldsymbol{f}(\boldsymbol{X})} \frac{d\boldsymbol{f}(\boldsymbol{X})}{d\boldsymbol{X}} \frac{d\boldsymbol{X}}{dt} = \boldsymbol{f}(\boldsymbol{X})^T \boldsymbol{D} \frac{d\boldsymbol{X}}{dt},$$

where

$$\boldsymbol{D} = \begin{bmatrix} \frac{\partial f_1(\boldsymbol{X})}{\partial t_1} & \frac{\partial f_1(\boldsymbol{X})}{\partial t_2} & \frac{\partial f_1(\boldsymbol{X})}{\partial t_3} & \frac{\partial f_1(\boldsymbol{X})}{\partial \phi} & \frac{\partial f_1(\boldsymbol{X})}{\partial \theta} & \frac{\partial f_1(\boldsymbol{X})}{\partial \psi} \\ \frac{\partial f_2(\boldsymbol{X})}{\partial t_1} & \frac{\partial f_2(\boldsymbol{X})}{\partial t_2} & \frac{\partial f_2(\boldsymbol{X})}{\partial t_3} & \frac{\partial f_2(\boldsymbol{X})}{\partial \phi} & \frac{\partial f_2(\boldsymbol{X})}{\partial \theta} & \frac{\partial f_2(\boldsymbol{X})}{\partial \psi} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial f_8(\boldsymbol{X})}{\partial t_1} & \frac{\partial f_8(\boldsymbol{X})}{\partial t_2} & \frac{\partial f_8(\boldsymbol{X})}{\partial t_3} & \frac{\partial f_8(\boldsymbol{X})}{\partial \phi} & \frac{\partial f_8(\boldsymbol{X})}{\partial \theta} & \frac{\partial f_8(\boldsymbol{X})}{\partial \psi} \end{bmatrix}.$$

To ensure that the $F(X)$ converges to a local minimum value, the time derivative of (7) should not be larer than 0, and:

$$\frac{dF(\boldsymbol{X})}{dt} = \frac{dF(\boldsymbol{X})}{d\boldsymbol{f}(\boldsymbol{X})} \frac{d\boldsymbol{f}(\boldsymbol{X})}{d\boldsymbol{X}} \frac{d\boldsymbol{X}}{dt} = \boldsymbol{f}(\boldsymbol{X})^T \boldsymbol{D} \frac{d\boldsymbol{X}}{dt} \le 0.$$

A direct way of designing $\frac{d\boldsymbol{X}}{dt}$ is [16] :

$$\frac{d\boldsymbol{X}}{dt} = -\boldsymbol{D}^T \boldsymbol{f}(\boldsymbol{X}). \tag{8}$$

The differential equation (8) can be solved by using many iterative methods, such as Runge-Kutta method. Another advantage is that this differential equation is continuous to time, so it is convenient to balance between accuracy and computing speed to satisfy the program's operation need.

Based on this algorithm, it is easy to obtain the relative pose between the camera and the moving target in real time.

IV. Experiment and Results

The proposed method was tested on a real system, which includes several functional components (Figure 3).

A. Hardware of System

The experimental setup includes a PTU and a camera is mounted on it. This PTU is REVS-50M-G from ShenZhen Reinovo Technology Co., Ltd. The PTU can pan with a range of –135~135 degrees, and tilt with a range of –63~63 degrees. Its resolution is 0.09 degree. The camera is acA640-100gc from Basler AG corporation. It has Sony ICX618 CCD sensor and the sensor patch is 5.6 µm × 5.6 µm. The pixel resolution of the camera is 658 pixel × 492 pixel. And its max frame rate is 100 frames per second. The lens installed in the system is Pentax C60402KP. Its focal length is 4.2 millimeter. And its maximum angle of view with 86.77 degrees. The target is designed to be easily identifiable: it is a black plastic flat board with four red LED lights mounted. The lights are distributed in L shape (Figure 4). The size of the board is 100cm×50cm. The ground station of the experiment with dual-core processor of 2.4GHz connects the PTU and the camera with Ethernet cable.

B. LED Identification and Correspondence

The LEDs appear to be bright in each image frame that system acquired. Thus, image gray processing (9) and thresholding function (10) are sufficient to detect LEDs. Then erode the bright pixel blob to eliminate noise [20]. Finally, we obtain the center coordinates of the pixel blob (11), that is, LED pixel coordinates.

$$\begin{aligned} Gray(u,v) &= 0.114B(u,v) + 0.587G(u,v) \\ &\quad + 0.299R(u,v) \end{aligned} . \tag{9}$$

$$Gray(u,v) = \begin{cases} 255, & if\ Gray(u,v) \geq Threshold \\ 0, & otherwise \end{cases} . \tag{10}$$

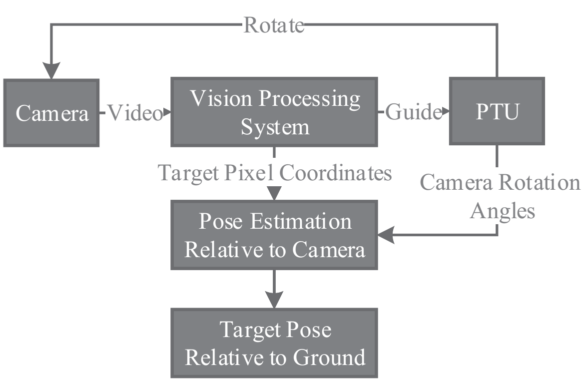


Figure 3. Experimental system

Figure 4. Experimental target

$$u_p = \frac{\sum u}{pixelnum}, \quad v_p = \frac{\sum v}{pixelnum} . \tag{11}$$

In (9), B, G and R denote the blue, green and red pixel component value separately. *Gray* in (9) and (10) denotes the pixel gray value. u and v in (11) denotes pixel coordinates in the blob; *pixelnum* denotes the number of pixel in blob; u_p and v_p are the average values of the sum of the pixel coordinates, i.e., the coordinates of the center of the blob.

Since all LEDs look similar in the image, it is necessary to identify LEDs using pixel coordinates. Fortunately, according to geometric projection invariance theory, LEDs can be distinguished by the geometric feature. From Figure 4, we can see that A, B, C are in a line, which means

$$S_{AC} = S_{AB} = S_{BC} ,$$

where S is the slope of line, and the subscript defines the line. For example, S_{AC} is the slope of line AC. Using the slope information, we can identify LED D in the image, and:

$$L_{AB} > L_{AC} ,$$

where L is the pixel distance. Similarly, LED A, B and C can also be identified.

C. Target Tracking

This paper focuses on the moving target pose estimation. Thus, the target tracking method is proposed here. The control inputs are calculated using the acquired information from camera images, therefore this information must be extracted before start of control loop.

Specifically, in order to make LEDs in the field of view, the control objective of PTU is to make mid-point of A and D in image coincide with the center-point of image. Based on this, we have:

$$\min DIS = \sqrt{\left(\frac{u_A + u_D}{2} - u_0\right)^2 + \left(\frac{v_A + v_D}{2} - v_0\right)^2} . \tag{12}$$

In (12), *DIS* is the pixel distance between the center of the board and the center of image; u_A, v_A and u_D, v_D are image coordinates of A and D respectively; u_0, v_0 are camera intrinsic parameters, and represent the image coordinates of the camera optical center.

TABLE I. PIXEL COOORDINATES OF FEATURE POINTS MEASURED AND REPROJECTED

		Pixel Coordinates of Feature Points	Pixel Coordinates of Reprojection	Corresponding Point Absolute Distance Error (Pixel)
First Frame	***D***	(138,406)	(137.0,405.7)	1.044
	C	(141,236)	(140.3,235.0)	1.221
	B	(263,225)	(262.2,224.4)	1.000
	A	(506,211)	(505.9,212.3)	1.304
Second Frame	***D***	(134,326)	(133.7,326.6)	0.671
	C	(141,164)	(140.9,164.4)	0.412
	B	(267,160)	(266.9,160.7)	0.707
	A	(498,157)	(497.8,157.6)	0.632
Third Frame	***D***	(427,301)	(426.9,300.0)	1.005
	C	(211,246)	(211.5,245.8)	0.539
	B	(308,192)	(306.8,190.5)	1.921
	A	(432,121)	(432.1,120.4)	0.608
Fourth Frame	***D***	(411,364)	(410.7,363.6)	0.500
	C	(133,312)	(132.9,312.7)	0.707
	B	(233,225)	(233.1,223.4)	1.603
	A	(354,120)	(353.8,120.0)	0.200
Fifth Frame	***D***	(114,366)	(114.2,365.9)	0.224
	C	(118,167)	(116.8,167.1)	1.204
	B	(262,157)	(261.9,159.3)	2.302
	A	(532,149)	(532.1,149.6)	0.608

D. Experiment Result

To evaluate the pose estimation algorithm and the whole experimental system, we have conducted a real-time experiment. First, our main algorithm is to be verified by the reprojection method [21].

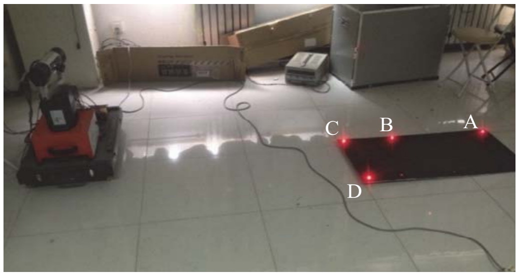
In our research, the reprojection method is to obtain the point image coordinates with estimated rotation angles and translation vector by using (2) and (5), and evaluate algorithm by reprojection error – the distance between estimated coordinates and real coordinates, which is presented in Table 1. We can see that the absolute distance error is less than 3 pixels, which is relatively small. The experiment is performed for many times, but the distance error remains small which verifies our main algorithm.

Next step is a set of experiment (Figure 5). The pose result of target relative to ground coordinate system is shown as follows: x = 0.0541m, y = 1.4447m, z = 0.7101m, ϕ = 0.4194deg, θ = 2.0609deg, ψ = 3.1214deg. It should be noted that, for the convenience of measurement, coordinate origin is the camera optical center, instead of on the ground plane, which influences the Z coordinate exclusively. In order to testify this result, a measuring tape was used to get the coordinates: $x \approx$ 0.05m, $y \approx$ 1.45m, $z \approx$ 0.70m. Despite the measuring tape's accuracy is not good enough, our result is still satisfying. Later the distance from camera optical center to target system origin had also been measured, which is 1.61m, and the distance calculated by result is 1.6107m, which suggests a good accuracy of this experiment. The rotation angles are all close to zero degree, because the target rotates little from the ground coordinate system.

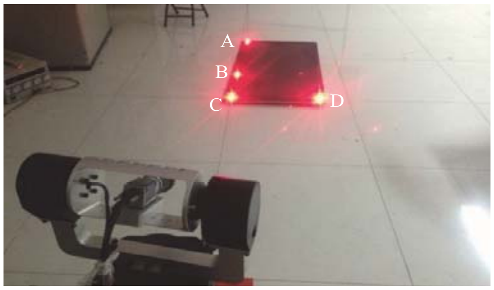
Finally, we conducted a continuous real-time experiment. The experiment scenario along with corresponding output at software interface are shown in Figure 6.

Although the algorithm and experiment are accurate, position and attitude information contains small errors. These errors may be generated due to the following reasons:

1) The optimization time is limited. The optimal solution can not be reached, so an error has occurred and accumulated.
2) The image distorts, which will lead to image barrel distortion. In such a case, the reprojection error increases as the point gets closer to the edge.
3) The experimental board can bend, and will bring error into calculation.



(a) Experiment result scene description picture 1



(b) Experiment result scene description picture 2

Figure 5. Pictures of the experiment scene

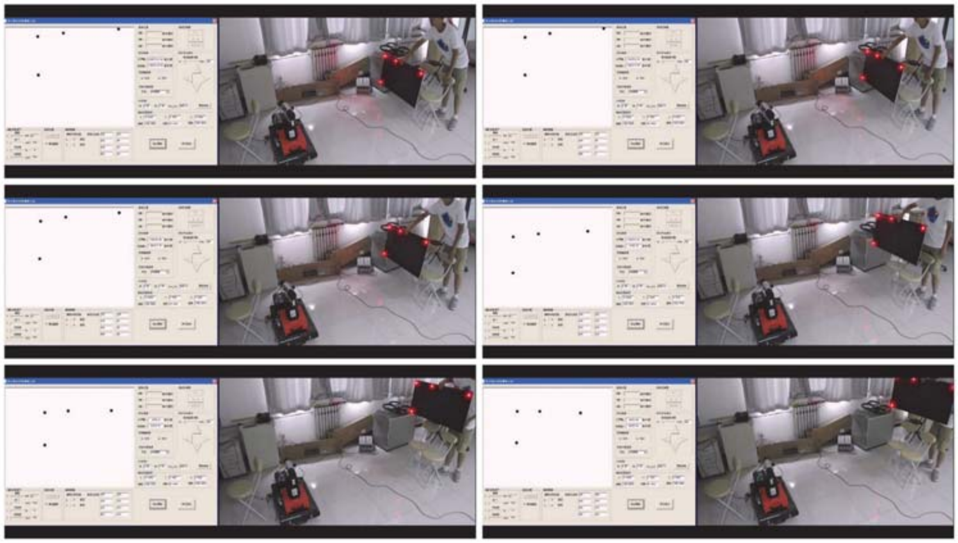


Figure 6. Simultaneous experiment scenario and software interface in a continuous real-time experiment

V. CONCLUSION

In this research, an off-board vision pose estimation method for a moving target was presented. This continuous-time optimization method has a unique advantage, that is, optimization time is adjustable. Meanwhile, the pose estimation can be achieved in real time. This method has been verified by the reprojection method, and the target pose of the experiment is accurate. Therefore, our off-board monocular vision pose estimation method is promising to guide quadcopter UAVs landing by providing its real-time pose, especially in GPS-denied environment.

The future work is to test the complete method in a real quadcopter UAV landing environment. Besides, some extreme circumstance such as losing the target should also be considered.

REFERENCES

[1] A. R. Lopez, "GPS landing system reference antenna," *IEEE Trans. Antennas and Propagation*, vol. 52, no. 1, pp. 104-113. Feb 2010.

[2] T. Walter, P. Enge, J. Blanch, and B. Pervan, "Worldwide vertical guidance of aircraft based on modernized GPS and new integrity augmentations," *Proceedings of the IEEE*, vol. 96, no. 12, pp. 1918-1935, Dec 2008.

[3] J. F. Vasconcelos, C. Silvestre, P. Oliveira, and B. Guerreiro, "Embedded UAV model and LASER aiding techniques for inertial navigation systems," *Control Engineering Practice*, vol. 18, no. 3, pp. 262-278, Mar 2010.

[4] J. Meyer-Hilberg, T. Jacob, "High accuracy navigation and landing system using GPS/IMU system integration," *IEEE Trans. Aerospace and Electronic Systems*, vol. 9, no. 7, pp. 11-17, July 1994.

[5] A. Benini, A. Mancini, and S. longhi. "An IMU/UWB/vision-based extended Kalman filter for mini-UAV localization in indoor environment using 802.15.4a wireless sensor network," *Journal of Intelligent and Robotic Systems*, vol. 70, no. 1, pp. 461-476, Aug 2012.

[6] S. Saripalli, J. F. Montgomery and G. S. Sukhatme, "Visually guided landing of an unmanned aerial vehicle," *IEEE Trans. Robotics and Automation*, vol. 19, no. 3, pp. 371-380, June 2003.

[7] T. Templeton, D. H. Shim, C. Geyer and S. S. Sastry, "Autonomous vision-based landing and terrain mapping using an MPC-controlled unmanned rotorcraft," *Robotics and Automation*, *2007 IEEE Int. Conf.*, Roma, 2007, pp. 1349-1356.

[8] S. Weiss, D. Scaramuzza, and R. Siegwart, "Monocular-SLAM-based navigation for autonomous micro helicopters in GPS-denied environments," *Journal of Field Robotics*, vol. 28, no. 6, pp. 854-874, Oct 2011.

[9] A. D. Wu, E. N. Johnson, K. Michael, F. Dellaert, and G. Chowdhary. "Autonomous flight in GPS-denied environments using monocular vision and inertial sensors," *Journal of Aerospace Information Systems*, vol. 10, no. 4, pp. 172-186, Apr 2013.

[10] W. Kong, D. Zhang, X. Wang, Z. Xian and J. Zhang, "Autonomous landing of an UAV with a ground-based actuated infrared stereo vision system," *Intelligent Robots and Systems, 2013 IEEE/RSJ Int. Conf.*, Tokyo, 2013, pp. 2963-2970.

[11] S. J. Zhang, X. B. Cao, F. Zhan, and L. He, "Monocular vision-based iterative pose estimation algorithm from corresponding feature points," *Science China Information Sciences*, vol. 53, no. 8, pp. 1682-1696, Aug 2010.

[12] P. Y. Lee, J. B. Moore, "Gauss-Newton-on-manifold for pose estimation," *Journal of Industrial and Management Optimization*, vol. 1, no. 4, pp. 565-587, Nov 2005.

[13] D. DeMenthon, L. S. Davis, "Exact and approximate solutions of the perspective-three-point problem," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, no. 11, pp. 1100-1105, Nov 1992.

[14] R. M. Haralick, C. N. Lee, K. Ottenberg, and M. Nölle, "Review and analysis of solutions of the three point perspective pose estimation problem," *International Journal of Computer Vision*, vol. 13, no. 3, pp. 331-356, Dec 1994.

[15] Y. Q. Zheng, Y. B. Kuang, S. Sugimoto, K. Astrom and M. Okutomi, "Revisiting the PnP problem: a fast, general and optimal solution," *Computer Vision, 2013 IEEE Int. Conf.*, Sydney, NSW, 2013, pp. 2344-2351.

[16] Q. Quan, K. Y. Cai, "A new continuous-time equality-constrained optimization to avoid singularity," *IEEE Trans. Neural Networks and Learning Systems*, vol. 27, no. 2, pp. 262-272, Feb 2016.

[17] G. J. Zhang, *Machine Vision*. Beijing: Science Press, 2005, ch. 2, (in Chinese).

[18] G. G. Slabaugh, "Computing euler angles from a rotation matrix," [Online], available at the website: http://www.staff.city.ac.uk/~sbbh653/publications/euler.pdf, May 24, 2016.

[19] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330-1334, Nov 2000.

[20] R. Z. Liu, S. Q. Yu, *OpenCV Tutorial-Basics.* Beijing: Beihang University Press, 2009, ch. 5, (in Chinese).

[21] R. Hartley, A. Zisserman, *Multiple View Geometry in Computer Vision Second Edition.* Cambridge: Cambridge University Press, 2004, ch. 4.