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 \( n \), \( R^i \) and \( T^i \) 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 \( P \in \mathbb{R}^3 \), vectors \( P_a = [X_a \ Y_a \ Z_a]^T \) and \( P_b = [X_b \ Y_b \ Z_b]^T \) be the coordinates of \( P \) in two different coordinate system \( \{A\} \) and \( \{B\} \) respectively. In Euclidean geometry, they satisfy [17]:

\( P_a = R_a P + T_a \)

\( P_b = R_b P + T_b \)
\[
P_T = R_T^c P_s + T_s^c, \tag{1}
\]

where \( R_T^c \in \mathbb{R}^{3 \times 3} \) is the rotation matrix, and \( T_s^c \in \mathbb{R}^3 \) 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. \( O_GX_GY_GZ_G \) denotes the ground coordinate system; \( O_CX_CY_CZ_C \) denotes the camera coordinate system; \( O_TX_TY_TZ_T \) denotes the target coordinate system.

The standard definition of the rotations about the three principle axes is presented [18]. A rotation of \( \psi \) 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 \( \theta \) 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 \( \phi \) 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 \( \psi, \theta, \) and \( \phi \) are the Euler angles.

In this paper, the rotation matrix from the target coordinate system to the camera coordinate system \( R_C^T \) satisfies:

\[
R_C^T = R_z(\phi)R_y(\theta)R_x(\psi), \tag{2}
\]

Moreover, the rotation matrix from the camera coordinate system to the ground coordinate system \( R_G^C \) satisfies:

\[
R_G^C = R_z(\beta)R_y(\alpha)R_x(\alpha),
\]

where \( \alpha \) and \( \beta \) 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:

\[
P_T = R_T^c P_s + T_s^c, \tag{3}
\]

where \( P_C = [X_C \ Y_C \ Z_C]^T \), \( P_G = [X_G \ Y_G \ Z_G]^T \), \( P_T = [X_T \ Y_T \ Z_T]^T \). 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:

\[
P_T = \left( R_T^G \right)^{-1}(P_G - T_G^C) = \left( R_T^G \right)^{-1} P_C - \left( R_T^G \right)^{-1} T_G^C. \tag{4}
\]

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 \( (O_XY) \) as follows,

\[
s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{u_0} \begin{bmatrix} \alpha_x & 0 & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_T \\ Y_T \\ Z_T \end{bmatrix} = M \begin{bmatrix} X_T \\ Y_T \\ Z_T \end{bmatrix}, \tag{5}
\]

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 \( \alpha_x, \alpha_y, u_0, v_0 \) in \( M \) are determined by camera calibration [19].
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:

\[
R^c = \begin{bmatrix}
    r_{11} & r_{12} & r_{13} \\
    r_{21} & r_{22} & r_{23} \\
    r_{31} & r_{32} & r_{33}
\end{bmatrix},
\]

\[
T^c = \begin{bmatrix} t_1 & t_2 & t_3 \end{bmatrix}^T,
\]

where the elements of \( R^c \) 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:

\[
u = \frac{(\alpha r_{11} + u_0 r_{31}) X_i + (\alpha r_{12} + u_0 r_{32}) Y_i + (\alpha r_{13} + u_0 r_{33}) Z_i + \alpha t_1 + u t_3}{r_{11} X_i + r_{12} Y_i + r_{13} Z_i + t_1},
\]

\[
v = \frac{(\alpha r_{21} + v_0 r_{31}) X_i + (\alpha r_{22} + v_0 r_{32}) Y_i + (\alpha r_{23} + v_0 r_{33}) Z_i + \alpha t_2 + v t_3}{r_{21} X_i + r_{22} Y_i + r_{23} Z_i + t_2}.
\]

The above equations can be used directly to solve our problem. Image pixel coordinates \( u, v \), camera intrinsic parameters \( \alpha, \alpha_0, u_0, v_0 \) and target system coordinates \( X_t, Y_t, Z_t \) are known. The three rotation angles \( \phi, \theta, \psi \), which compose the \( R^c \), and the three elements of the \( T^c \), namely \( t_1, t_2, t_3 \), are to be calculated.

Define \( X = [t_1, t_2, t_3, \phi, \theta, \psi]^T \in \mathbb{R}^6 \). In order to solve \( X \), 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:

\[
f_{2i-1}(X) = (\alpha(r_{11} + u_0 r_{31}) X_i + (\alpha(r_{12} + u_0 r_{32}) Y_i + (\alpha(r_{13} + u_0 r_{33}) Z_i + \alpha t_1 + u t_3 \notag
\]

\[+ (\alpha r_{21} + v_0 r_{31}) X_i + (\alpha r_{22} + v_0 r_{32}) Y_i + (\alpha r_{23} + v_0 r_{33}) Z_i + \alpha t_2 + v t_3 \notag
\]

\[+ \alpha_0 r_{31} X_i + r_{32} Y_i + r_{33} Z_i + t_3),
\]

\[
f_{2i}(X) = (\alpha(r_{11} + v_0 r_{31}) X_i + (\alpha r_{12} + v_0 r_{32}) Y_i + (\alpha r_{13} + v_0 r_{33}) Z_i + \alpha t_1 + v t_3 \notag
\]

\[+ (\alpha r_{21} + u_0 r_{31}) X_i + (\alpha r_{22} + u_0 r_{32}) Y_i + (\alpha r_{23} + u_0 r_{33}) Z_i + \alpha t_2 + u t_3 \notag
\]

\[+ \alpha_0 r_{32} X_i + r_{31} Y_i + r_{33} Z_i + t_3). \]

Theoretically, \( f_i(X) \) should equal to zero. However, due to the noise, we have to solve using an optimization method. Therefore we have:

\[
\min_{X \in \mathbb{R}^6} F(X) = \frac{1}{2} \left( f(X)^T f(X) \right), \tag{7}
\]

where \( f(X) = [f_1(X) f_2(X) \cdots f_7(X) f_8(X)] \in \mathbb{R}^8 \) 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 derive of the objective function \( F(X) \), and get:

\[
\frac{dF(X)}{dt} = \frac{dF(X)}{dX} \frac{dX}{dt} = f(X)^T D \frac{dX}{dt},
\]

where

\[
D = \begin{bmatrix}
\frac{\partial f_1}{\partial t_1} & \frac{\partial f_1}{\partial t_2} & \frac{\partial f_1}{\partial t_3} & \frac{\partial f_1}{\partial \phi} & \frac{\partial f_1}{\partial \theta} & \frac{\partial f_1}{\partial \psi} \\
\frac{\partial f_2}{\partial t_1} & \frac{\partial f_2}{\partial t_2} & \frac{\partial f_2}{\partial t_3} & \frac{\partial f_2}{\partial \phi} & \frac{\partial f_2}{\partial \theta} & \frac{\partial f_2}{\partial \psi} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots
\end{bmatrix}.
\]

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

\[
\frac{dF(X)}{dt} = \frac{dF(X)}{dX} \frac{dX}{dt} = f(X)^T D \frac{dX}{dt} \leq 0.
\]

A direct way of designing \( \frac{dX}{dt} \) is [16]:

\[
\frac{dX}{dt} = -D^T f(X). \tag{8}
\]

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 REV50M-G from Shenzhen Reinovo Technology Co., Ltd. The PTU can pan with a range of −135 to 135 degrees, and tilt with a range of −63 to 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 100 cm × 50 cm. The ground station of the experiment with dual-core processor of 2.4 GHz 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.

\[
\text{Gray}(u,v) = 0.114B(u,v) + 0.587G(u,v) + 0.299R(u,v) \quad (9)
\]

\[
\text{Gray}(u,v) = \begin{cases} 255, & \text{if } \text{Gray}(u,v) \geq \text{Threshold} \\ 0, & \text{otherwise} \end{cases} \quad (10)
\]

In (9), \(B\), \(G\) and \(R\) denote the blue, green and red pixel component value separately. \(\text{Gray}\) in (9) and (10) denotes the pixel gray value. \(u\) and \(v\) in (11) denotes pixel coordinates in the blob; \(\text{pixelnum}\) denotes the number of pixel in blob; \(u_0\) and \(v_0\) 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:

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

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 Coordinates of Feature Points Measured and Reprojected

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

**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: \( x = 0.0541m, y = 1.4447m, z = 0.7101m, \phi = 0.4194\text{deg}, \theta = 2.0609\text{deg}, \psi = 3.1214\text{deg} \). 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 cannot 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.

![Figure 5. Pictures of the experiment scene](image)

(a) Experiment result scene description picture 1

(b) Experiment result scene description picture 2

---

2086
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