International Journal of Control and Automation
Vol.7, No.7 (2014), pp.65-84
http://dx.doi.org/10.14257/ijca.2014.7.7.06

Vision Based Robust Pose Estimation System in Different Observed Situations for an Outdoor Quadrotor

Wei Zheng[1], Fan Zhou[1] and Zengfu Wang[1,2*]

[1]*Dept. of Automation, University of Science and Technology of China, Hefei, China*
[2]*Institute of Intelligent Machines, Chinese Academy of Sciences, Hefei, China*
*zhengwei@mail.ustc.edu.cn, *zfwang@ustc.edu.cn*

Abstract

In this paper, we present a vision based robust pose estimation system in different observed situations for a quadrotor in outdoor environments. This system could provide us with approximate ground truth of pose estimation for an outdoor quadrotor, while most of existing vision based systems perform indoors. We only use the own features of the quadrotor, while most existing systems modify the architecture of the quadrotor or put additional components such as colored markers on it. We propose the novel robust pose estimation algorithms for different observed situations. With good observed results, we get all of the four rotors and calculate the pose. But when fewer than four rotors are observed, all of existing external vision based systems for the quadrotor do not mention this and could not get right results. By combining inertial measurement unit (IMU) data, our robust pose estimation system has solved these problems and obtained accurate results of pose estimation. We demonstrate in real experiments that our pose estimation system for the quadrotor could perform accurately and robustly in real time.

Keywords: *Pose estimation, Quadrotor, Vision, IMU data, Outdoor*

1. Introduction

Recent years, micro aerial vehicles (MAVs) have gained much attention, because of the quite wide range of applications like exploration, search-and-rescue. The quadrotor used in this paper is a kind of micro aerial vehicle with four rotary wings (see Figure 1). A quadrotor has distinct advantages in agility and maneuverability. The quadrotor has wonderful property of vertical take-off and landing, omni-directional flying and hovering. For quadrotor application, it is necessary to set up the external pose estimation system for the quadrotor. The pose estimation system is significant in evaluating experimental results and improving algorithms. It could also be employed for autonomous landing or take-off of the quadrotor.

Nowadays, most of external vision based systems worked in indoor environments. These systems were usually complex and expensive. They generally modified the architecture of the quadrotor or put some additional components such as colored blobs and LEDs on it. Colored blobs and LEDs are difficult to be observed in outdoor environments. Besides, these methods need to detect all the blobs or LEDs rightly. When one or more blobs are lost, these approaches do not mention these cases and are not able to get right results of pose estimation.

* Corresponding Author

ISSN: 2005-4297 IJCA

The motivation of our work is to present an external vision based robust pose estimation system which could deal with different observed situations for a quadrotor in outdoor environments (see Figure 1). This system could provide us with approximate ground-truth pose estimation of a quadrotor. Also we intend to detect and track a quadrotor well without modifying architecture or adding additional components such as colored blobs. When all of rotors or not all of rotors are observed, we give robust pose estimation algorithms to get accurate results in these different observed situations.

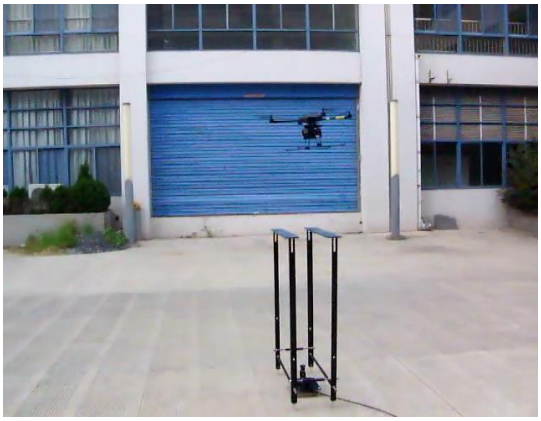


Figure 1. Our Robust Pose Estimation Relationship System for a Quadrotor

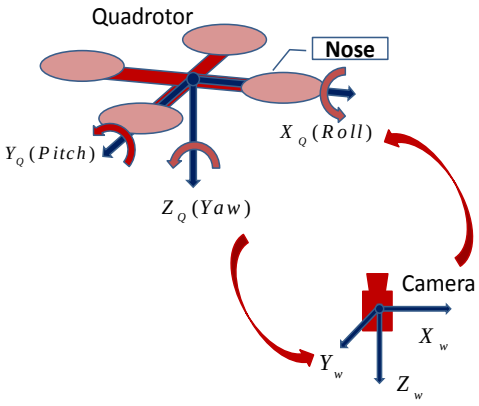


Figure 2. Relative Pose in our System

1.1. Related Work

Altug *et al.* [2] placed some colored blobs on the bottom of a quadrotor. Then they used a ground camera to estimate the pose of a quadrotor. Later, Altug *et al.*, [3] presented a novel two-camera method. One camera was located on-board the quadrotor, and the other camera was located on the ground.

Park *et al.*, [4] placed red and green LED markers at the bottom of a quadrotor to form a special landmark. By analyzing the color distribution of the markers, they got the position and orientation of a quadrotor. Achtelik *et al.*, [5] also made use of colored LEDs for pose estimation. The LEDs were inside some table-tennis balls and were observed by a stereo camera. Breitenmoser *et al.*, [10] presented a monocular vision-based system for 6D relative robot localization. It made a target module with four non-coplanar colored LED markers on a robot and placed one camera on the wall.

These systems were primarily designed for indoor environments. They usually made use of a few cameras which were located on the walls, ground or the ceilings. Besides, colored blobs and LEDs were difficult to be observed in outdoor environments.

How *et al.*, [6] presented a real-time indoor autonomous vehicle test environment which was named RAVEN. Ahrens *et al.*, [7] and Abeywardena *et al.*, [8] used the VICON(motion capture system) for observing the quadrotor indoors. These systems were mainly designed for multi-vehicle missions using both ground and air vehicles in indoor environments. Also they were complex and expensive.

Klose *et al.*, [9] put an additional rotor guard and casing around the quadrotor. Then they used CAD model of their quadrotor to match with the data calculated from current images. The computational cost of this pose estimation algorithm was high. And it might cause tracking loss under fast quadrotor movements.

Systems designed for outdoor environments were quite rare. Ha and Lee [1] used a team of a MAV and an unmanned ground vehicle (UGV). The UGV tracked the MAV and calculated its pose. Approaches [25, 26] employed results of offline SFM (Structure from Motion) to get pose estimation and could not perform in real-time. These approaches had quite large computation complexity.

In a word, most of external vision based pose estimation systems were designed for indoor environments and not suitable for outdoor environments. For pose estimation, Park *et al.*, [4] designed a special landmark both using points and regions information. Klose *et al.*, [9] compared the original CAD model of a quadrotor with the current data. Most systems used direct geometry algorithms. Approaches [2, 3, 5, 10] detected colored blobs or LEDs and analyzed their geometry relationship to get pose estimation. These approaches [2, 3, 5, 10] also need to observe all of the blobs or LEDs rightly. When one or more blobs were lost, these approaches did not mention these cases and were not able to get right results of pose estimation.

In this paper, we only use the quadrotor's own features to estimate the pose. The rotors of a quadrotor are seen as the reference points. We present the robust pose estimation algorithms for the different observed situations. Here our algorithms refer to the Perspective-n-Point (PnP) problem, which is an important problem in computer vision, robotics, and augmented reality.

The PnP problem is to get the position and orientation of a camera given its intrinsic parameters and a set of n correspondences between 3D points and their 2D projections. The minimal number of correspondences to solve PnP problem is three. Some researches [11, 12] had found that P3P problem with non-collinear points would result in as many as four solutions. Fischler [13] found that P4P problem with non-coplanar points had several solutions. P4P problem with coplanar points had unique solution. Hu and Wu [14] gave a result that non-coplanar P4P problem could have at most four solutions. For P5P problem there were as many as two solutions. For more than 6 correspondences, it became classics Direct Linear Transformation (DLT) problem [23].

PnP approaches could be classified into two types: non-iterative and iterative solutions. Non-iterative approaches are generally linear algorithms or some ingenious transformations. Iterative approaches usually minimize an error function. They are generally slower but have higher accuracy than non-iterative ones. Iterative approaches may fall into local minimum and result in pose ambiguity.

Approaches [16, 17] presented linear non-iterative algorithms for PnP problems. But they had low accuracy. Lepetit *et al*. [20] proposed a non-iterative solution EPnP to the PnP problem for more than four 3D-to-2D point correspondences. It had better accuracy and lower computational complexity than other non-iterative approaches. Among iterative approaches, Dementhon et al. [15] presented POSIT algorithm to solve PnP problem for more than four non-coplanar correspondences. Lu *et al*. [18] introduced a very accurate iterative algorithm. It minimized a 3D space error and was faster than other iterative ones. But it often fell into local minimum and resulted in pose ambiguity. Schweighofer and Pinz [19] proposed a robust pose estimation algorithm from a planar target. It took full advantage of properties of four coplanar points and solved the pose ambiguity. Comparing with other iterative approaches, it could get right unique solution for four coplanar points.

Recently, some significant work has been done for PnP problem with some other equipment such as inertial measurement units (IMUs). IMU often consists of three-axis gyroscopes and three-axis acceleration sensors. It could provide accurate roll and pitch angle, *i.e.*, the vertical direction. The angular accuracy of roll and pitch angle in low

cost IMU is about 0.5 degree, and in high accuracy IMU is less than 0.02 degree. Fraundorfer et al. [21] presented a novel minimal case solution to the calibrated relative pose estimation problem using at least 3 point correspondences with known roll and pitch angles from IMU. It could cope with planes and even collinear points. Similar to the idea, Kukelova *et al.* [22] provided new closed-form solutions to the absolute pose estimation of a calibrated camera from two 2D-3D correspondences and a given vertical direction. They could also estimate the pose together with unknown focal length from three 2D-3D correspondences and a given vertical direction.

1.2. Overview of our Work

In this paper, the contributions of our work are mainly as follows. We propose an external vision based robust pose estimation system in different observed situations for a quadrotor in outdoor environments (see Figure 1). Our system could work well in outdoor environments, while most existing systems worked in indoor environments. Only the own features of the quadrotors are made use of, while most existing systems placed additional components such as LEDs or modified the architecture of a quadrotor. Also we present the robust pose estimation algorithms for different observed situations. When all of four rotors are observed, our pose estimation algorithm could obtain accurate results. When there are occlusions or poor tracking results, only two or three rotors may be observed. All of existing pose estimation systems do not mention these cases and are not able to get right results. By combining IMU data, our robust pose estimation system can solve these problems and get accurate results of pose estimation.

This paper is organized as follows: In Section I, we introduction our work and the related work. In Section II, the hardware structure and features selection of the quadrotor are shown. The preliminary position of the quadrotor is calculated in Section III. In Section IV, we describe our robust and accurate pose estimation algorithms in detail. The simulation experiments and real experiments were performed in Section V and Section VI. In the end, we give the conclusions.

2. Hardware and Features Selection of the Quadrotor

2.1. Hardware

The quadrotor used here is named X600D produced by XAircraft corporation. It has flight control system combining GPS and on-board inertial measurement unit (IMU) which consists of three-axis gyroscopes and three-axis acceleration sensors. We also equip our quadrotor with electronic compass, sonar or air pressure sensor. For transferring the selected sensor data to ground computer, we use ZigBee wireless transmission modules.

The camera fixed on the ground is an industrial camera (AVT Stingray F125B) made by allied vision technologies. The image data is transferred to our ground computer by 1394b connecting line.

2.2. Why Only Use the Quadrotor's Own Features

One main character of our work is that we only use the quadrotor's own features. We wish that this system is simple, effective and more general for quadrotor applications. For detection and tracking, we see a quadrotor as a dark blob and get its preliminary position. For getting accurate pose estimation, we only observe the rotors of a quadrotor. In outdoor environments, it is difficult to observe the colored markers or

appearance modification. The rotors of a quadrotors are good features. These rotors could also provide us with enough information for pose estimation and might be more reliable than colored markers.

3. Preliminary Position

3.1.Relative Pose Relationship Between the Quadrotor and the Camera

The industrial camera is fixed on the ground, and the quadrotor is flying in the sky. The quadrotor coordinate system is shown in Figure 2. Here Roll denotes the rotation about X_Q axis, Pitch denotes the rotation about Y_Q axis, and Yaw denotes the rotation about Z_Q axis. A simple illustration of the relative pose relationship between the camera and the quadrotor could be seen in Figure 2.

3.2.Detection and Tracking of a Quadrotor

The color of our quadrotor is black. The wingspan of the quadrotor is about 75 cm. In our work, we only use the quadrotor's own features. When our quadrotor is flying in the air, we can see it as a dark blob in the center and four wings around it. What we face is a detection and tracking problem under static background. This problem could be divided into two main sections: detection and tracking.

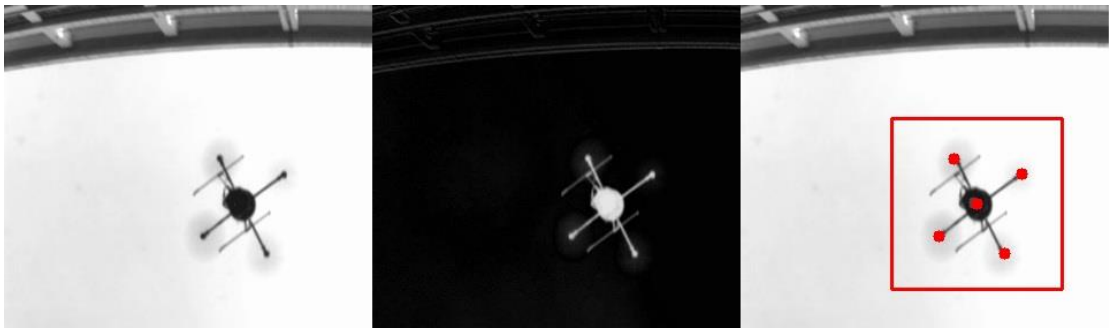
For detecting a quadrotor, we use the background difference method. In our work, we placed the ground camera upward with vertical direction. When a quadrotor is flying in the air, a moving dark blob is obtained from a sequence of images using background difference method. The four rotors of a quadrotor could also be detected. When a quadrotor is detected rightly, an accurate region of the quadrotor is also obtained.

For tracking a quadrotor, we employ the modified mean-shift tracking approach. Tracking windows with variable sizes would be employed. The original size of tracking window is decided by the detection results. When tracking steps going on, the size of the tracking window would be adjusted. The tracking of four rotors would also be performed to get positions of the rotors. A brief introduction of detection and tracking steps could be seen in Figure 3.

At first, the detection section is performed. It provides the tracking section with the initial results. And then the tracking section goes on. When tracking section fails, the detection section would carry out again. When a quadrotor flies for a long time, background updating is needed. In our work, background updating is performed according to different situations. When a quadrotor is observed rightly, the accurate window region which covers the quadrotor could be acquired. The background updating would update the region which is in image but outside the quadrotor region. This background updating performs every a few frames. When the quadrotor flies out the field of view, the background updating process would update the entire image region.

3.3. Preliminary Position of a Quadrotor

Here, the calculation of the pose of a quadrotor is divided into two steps. Firstly, the preliminary position of a quadrotor in the world coordinate system is obtained. Secondly, the accurate pose (position and orientation) of a quadrotor is calculated by our novel robust algorithms.



(a) (b) (c)

Figure 3. Detection and Tracking Steps for the Quadrotor

A quadrotor could be seen as a moving dark blob in images. The center of the dark blob denotes the position of the quadrotor (see Figure 3(c)). The center of an image is seen as the origin of the image coordinate system. The right direction of the origin is positive direction of x, and the upward direction of the origin is positive direction of y. Then the pixel positions of a quadrotor are:

$$x_q = \sum_{x_i, w_i \in C} x_i w_i \Big/ \sum_{w_i \in C} w_i \quad , \tag{1}$$

$$y_q = \sum_{y_i, w_i \in C} y_i w_i \Big/ \sum_{w_i \in C} w_i \quad , \tag{2}$$

$$w_i = \begin{cases} 1, & if\ w_i \in C_1 \\ 0, & if\ w_i \in C_2 \end{cases} \quad , \tag{3}$$

where (x_q, y_q) denotes the pixel coordinate position of a quadrotor. The C is the region of the quadrotor calculated from the detection step or tracking window. And (x_i, y_i) is the pixel position of a point which is considered as image of the quadrotor and in the region C. Here w_i is the weight of the point (x_i, y_i). w_i is given two different values which rely on the distance between the point and the center of region C. Here C_1 is the region near the center of C in region C. And C_2 is the region farther with the center of C in region C.

In this paper, the camera coordinate system is set to be equal to the world coordinate system. The center of camera's CCD sensor is regarded as the origin of the world coordinate system. The Z_w-axis points downward to the ground, the X_w-axis and Y_w-axis are the same with the image coordinate system (see Figure 2). Then the real position of a quadrotor could be obtained:

$$Z_q = \frac{D}{d} f \quad , \tag{4}$$

$$X_q = \frac{Z_q}{f} \tilde{x}_q \quad , \tag{5}$$

$$Y_q = \frac{Z_q}{f} \tilde{y}_q \quad , \tag{6}$$

where f is the focal length of the camera and Z_q, X_q, Y_q are real preliminary position of a quadrotor in world coordinate system. $\tilde{x}_q$ and $\tilde{y}_q$ are undistorted pixel coordinates. D is the real wingspan of the quadrotor, and d is undistorted pixel distance of the wingspan in image. From Equation 5 and Equation 6, we could get some results:

$$X_q = k \cdot Y_q \quad or \quad \tilde{x}_q = k \cdot \tilde{y}_q \quad , \tag{7}$$

where k is scaling factor of X_q and Y_q. And the scaling factor of X_q and Y_q is equal to the scaling factor of $\tilde{x}_q$ and $\tilde{y}_q$. When the intrinsic parameters of the camera are measured accurately, the scaling factor k could have a quite high accuracy.

4. Robust Pose Estimation Algorithms

4.1. Different Observed Situations of a Quadrotor

When a quadrotor flies at different positions, it could be seen as a fixed landmark. The pose estimation problems of a quadrotor could be divided into four different observed situations (see Figure 4).

(1): all of the four rotors are observed rightly (see Figure 4 a b c).
(2): three rotors of a quadrotor are observed (see Figure 4 d e f).
(3): only two rotors of a quadrotor are observed (see Figure 4 g h i).
(4): Fewer than two rotors are observed.

For these different observed situations, the last situation could not be solved up to now. In this paper, we mainly solve the former three situations. Most of current external vision based pose estimation systems of the quadrotor solve the first situation. For the second and third situations, all of other vision-based pose estimation systems don't mention these cases and could not get right results of pose estimation. In this paper, the second and third situations are also solved by making use of the IMU data.

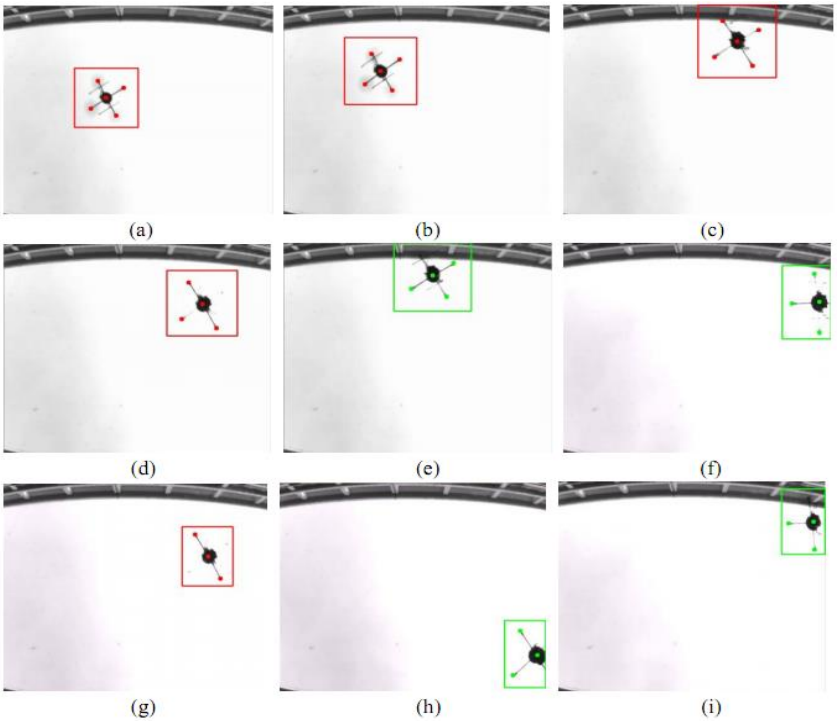


Figure 4. The Different Observed Results of a Quadrotor. Here we Show three Kinds of Results for Four, Three, or Two Rotors Observed.

4.2. Problem Formulation

The quadrotor is the object coordinate system. In this object coordinate system, the 3D coordinates of four rotors' centers are $M_i^o = (X_i^o, Y_i^o, Z_i^o)', i = 1,2,3,4$. The centers of four rotors could be seen as coplanar points, and then $M_i^o = (X_i^o, Y_i^o, 0)', i = 1,2,3,4$. The real values of X_i^o, Y_i^o are the half of wingspan and could be measured accurately in advance. Their corresponding coordinates in camera coordinate system are $M_i^c = (X_i^c, Y_i^c, Z_i^c)', i = 1,2,3,4$. They have the relationship:

$$M_i^c = R \cdot M_i^o + T \quad , \tag{8}$$

where $R = f(\alpha, \beta, \gamma)$ and $T = (t_x, t_y, t_z)'$ are the rotation matrix and the translation vector. The centers of rotors are expressed in the normalized image coordinate system as $m_i = (u_i, v_i, 1)', i = 1,2,3,4$, which are the projection of $M_i^c = (X_i^c, Y_i^c, Z_i^c)', i = 1,2,3,4$. Then we could have:

$$m_i \propto M_i^c = (R \cdot M_i^o + T) \quad , \tag{9}$$

In this paper, the intrinsic parameters of the camera are known and have been measured in advance.

4.3. EMRPP Algorithm for Four Rotors Observed

When all of four rotors of a quadrotor are observed, the quadrotor could be seen as a landmark. Different from common non-coplanar landmarks, this landmark is made up of four coplanar points. The coplanar P4P problem has unique solution.

In order to get accurate pose estimation results, iterative approaches are good choices. But many iterative approaches often fall into local minimums and result in pose ambiguity. Schweighofer and Pinz [19] proposed a robust pose estimation algorithm for four coplanar points. It solved the pose ambiguity problem and got unique solution. Their algorithm was faster and has higher accuracy than some other iterative approaches. We name this algorithm as RPP.

RPP made use of object-space error function. It considered that the orthogonal projection of M_i^c on m_i should be equal to M_i^c itself:

$$m_i \propto M_i^c = (R \cdot M_i^o + T) \quad , \tag{10}$$

$$F_i = \frac{m_i m_i'}{m_i' m_i} \quad , \tag{11}$$

where F_i is a projection operator, $F_i = F_i'$ and $F_i = F_i^2$. And the object-space error function [18] was:

$$E_{os} = \sum_{i=1}^{n} \| (I - F_i)(R \cdot M_i^o + T) \|^2 \quad , \tag{12}$$

By taking full advantage of the coplanar properties, RPP transformed the Equations 9 and 12. Then $\tilde{E}_{os}$ only depends on a rotation about the y-axis $\tilde{R}_y(\tilde{\beta})$ and $\tilde{T}$:

$$\tilde{m}_i \propto R_z(\tilde{\gamma})(\tilde{R}_y(\tilde{\beta}) \cdot \tilde{M}_i^o + \tilde{T}) \quad , \tag{13}$$

$$\tilde{E}_{os} = \sum_{i=1}^{n} \| (I - \tilde{F}_i) R_z(\tilde{\gamma})(\tilde{R}_y(\tilde{\beta}) \cdot \tilde{M}_i^o + \tilde{T}) \|^2 \quad , \tag{14}$$

$$\tilde{T} = (\tilde{t}_x, \tilde{t}_y, \tilde{t}_z)' \quad , \tag{15}$$

where symbol "~" above the variables denotes the transformations of these variables.

Here we give a modified RPP algorithm. In former section, we get the preliminary position of the quadrotor. From Equation 7, we could get extra constraint:

$$t_x = k \cdot t_y \quad , \tag{16}$$

When the intrinsic parameters of the camera are measured accurately and the pixel coordinates of the quadrotor are calculated rightly, the accuracy of scaling factor k is quite high. Then we put Equation 16 into Equations 9 and 12:

$$\tilde{m}_i \propto R_z(\tilde{\gamma})(\tilde{R}_y(\tilde{\beta}) \cdot \tilde{M}_i^o + \hat{T}) \quad , \tag{17}$$

$$\tilde{E}_{os} = \sum_{i=1}^{n} \| (I - \tilde{F}_i) R_z(\tilde{\gamma})(\tilde{R}_y(\tilde{\beta}) \cdot \tilde{M}_i^o + \hat{T}) \|^2 \quad , \tag{18}$$

$$\hat{T} = (\hat{k}\hat{t}_y, \hat{t}_y, \hat{t}_z)' \quad , \tag{19}$$

where $\hat{k}$ is the transformation of k and could be calculated from k. By using the scaling factor k, there are only two degrees of freedom in $\hat{T}$ here. Then $\tilde{E}_{os}$ only depends on a rotation about the y-axis $\tilde{R}_y(\tilde{\beta})$ and $\hat{t}_y$, $\hat{t}_z$. We refer to our modified RPP algorithm as MRPP algorithm.

As iterative algorithms, RPP and MRPP both need the initial pose guess which would affect the computation time and the accuracy of results. Lepetit et al. [20] proposed a non-iterative solution EPnP for more than four 3D-to-2D point correspondences. It had better accuracy and lower computational complexity than other non-iterative approaches. EPnP was faster than RPP, but had lower accuracy than RPP algorithm.

In order to get robust and accurate pose estimation results, we present a new algorithm EMRPP combined by EPnP and our MRPP algorithms. We first perform EPnP to get the initial pose estimation. Then we make use of this initial value as the input of MRPP. We name this algorithm as EMRPP. The first step of EMRPP could get quite accurate initial estimation of pose. A good initial estimation would speed up the following iterative section of EMRPP and get robust and accurate results of pose estimation. Because the EPnP is a fast non-iterative algorithm and the good initial estimation speeds up the MRPP algorithm. So the computation time of EMRPP is less than RPP, and the accuracy of EMRPP is higher than RPP.

4.4. Three/Two Rotors Observed

When only three or two rotors of a quadrotor are observed, we could not get unique pose estimation by pure vision approaches. Here we make use of the IMU data to get the accurate results. The IMU on the quadrotor could provide us with Roll and Pitch angles of the quadrotor. The angular accuracy of Roll and Pitch angles is about 0.3^{o}. This accuracy meets our requirements. We employ IMU data to get accurate results when no more than three rotors are observed.

In this paper, the image data and IMU data could not arrive in our computer at the same time. Besides, the current data of image and IMU in the computer may be observed at different time. Obviously, the image data and IMU data should be at the same time to get accurate results. The IMU data is sent by flight control system every 10ms. Then ZigBee modules take about 15ms to transfer it to ground computer. The total delay of IMU data is about 15~25ms and we assume that the average delay is 20ms. The industrial camera works at 30fps. Image data is transferred to ground computer by 1394b connector, which would take about 10ms. So the total delay of image data is about 10~40ms and we assume that the average delay is 25ms. In general, the image data we get currently is later than the IMU data we get currently. Strictly speaking, the image data is later from -5ms to 15ms than IMU data. The image data with the IMU data which arrived at the computer 10ms before would be used to get pose estimation.

Here our approach is improved from Kukelova et al. [22]. Kukelova et al. provided closed-form solutions to the absolute pose estimation of a calibrated camera from at least two 2D-3D correspondences and IMU data. They assumed that they got the 2D-3D correspondences and IMU data at the same time. They declared that they got Roll and Yaw angles from IMU data. In their paper, the rotation matrix R performed the Pitch first, then the Roll, and finally the Yaw. In fact, the angles we could get from IMU data are Roll and Pitch. For quadrotors, the rotation matrix R should be expressed by another order. In our work, the rotation matrix R is:

$$R = R_z R_y R_x \quad , \tag{20}$$

where R_z is the rotation matrix for the Yaw axis, R_y is the rotation matrix for the Pitch axis and R_x is the rotation matrix for the Roll axis. And they could be shown as follows:

$$R_z = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} , \quad (21)$$

$$R_y = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} , \quad (22)$$

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{bmatrix} , \quad (23)$$

Utilizing the data returned by the IMU, we can get the values of R_x and R_y . So the only one unknown parameter of the rotation matrix R is the rotation angle α around the Yaw axis. Then equation 9 is:

$$\lambda m_i = \left[R_z(\alpha) R_y R_x \mid T \right] M_i , \quad (24)$$

where λ is the scaling factor. m_i are normalized image coordinates and M_i are homogeneous coordinates. To simplify the former equation, we use the $q = \tan\alpha/2$. We could get:

$$(1+q^2) R_z(q) = \begin{bmatrix} 1-q^2 & -2q & 0 \\ 2q & 1-q^2 & 0 \\ 0 & 0 & 1+q^2 \end{bmatrix} , \quad (25)$$

So the equation 24 could be written as:

$$[m_i]_\times \left[R_z(q) R_y R_x \mid T \right] M_i = 0 , \quad (26)$$

where $[m_i]_\times$ is the skew symmetric matrix of m_i and the rank of $[m_i]_\times$ is two. The equation 26 produces three polynomial equations and only two are linearly independent. From Equation 7, we could have extra constraint of T:

$$t_x = k \cdot t_y , \quad (27)$$

Make use of this constraint in equation 26 and we get:

$$[m_i]_\times \left[R_z(q) R_y R_x \mid T(k t_y, t_y, t_z) \right] M_i = 0 , \quad (28)$$

In this case there are only three unknown variables t_y, t_z, q. From equation 28, we know that one 2D-3D point correspondence provides us with two independent polynomial equations. But there is variable q of degree two in independent polynomial equations. So the minimal number of point correspondences we need to get unique pose estimation is two. By eliminate the variable q of degree two in equations, we could get unique solution of q. And then we get the final results.

When three rotors are observed, we would have six independent polynomial equations. By using least squares method, we get the optimal results of pose estimation. We name this algorithm as IMU+3P. If only two rotors are observed, there would be just four independent polynomial equations. So we get unique solution of pose estimation in the end. This algorithm is named as IMU+2P.

5. Simulation Experiments

We have performed our algorithms in simulation and real experiments. In simulation and real experiments, we compared our algorithms with some state-of-the-art and classics algorithms, such as GAO algorithm [12], LHM approach [18], RPP algorithm [19], and EPnP algorithm [20].

Here we use a virtual calibrated camera to generate a set of 3D-to-2D point correspondences. We use the intrinsic parameters of real camera as the intrinsic parameters of the virtual camera. We generate different Gaussian random noise to simulate different image noise and IMU noise. For each experiment, we set the ground-truth translation and rotation values of the quadrotor in advance. The translation error is the angle between the estimated translation direction and the ground-truth direction. The rotation error is the smallest angle of rotation to bring the estimated rotation to the ground-truth rotation.

5.1.Four Rotors Observed

5.1.1. Pose Error with Different Image Noise

The pose error involves the translation error and the rotation error. Here we set the tilt angle as 30 degrees and change the image noise. Figure 5a and Figure 5b show the translation error and rotation error of these algorithms. When image noise increases, the translation error and the rotation error would also increase. Here the results of LHM, RPP and EMRPP are better than GAO, EPnP. When image noise is larger than four pixels, the rotation accuracy of EPnP falls quickly. The accuracy of EMRPP is slightly higher than RPP, LHM.

5.1.2. Pose Error with Different Tilt Angles

For different tilt angles, we set the image noise as 1.0 pixel and test these algorithms. Figure 5c and 5d show the translation error and rotation error separately with different tilt angles. GAO algorithm has the worst results. When tilt angle is smaller than 10 degrees, EPnP is better than LHM, RPP and EMRPP. When tilt angle of the planar points is larger than 10 degrees, EMRPP, LHM and RPP have better results than EPnP. Meanwhile, EMRPP has slightly better results than RPP and LHM. When tilt angle is more than 40 degrees, the rotation error of EPnP increases rapidly. In order to show the details of rotation error better, the rotation errors are shown when tilt angles are smaller than 40 degrees.

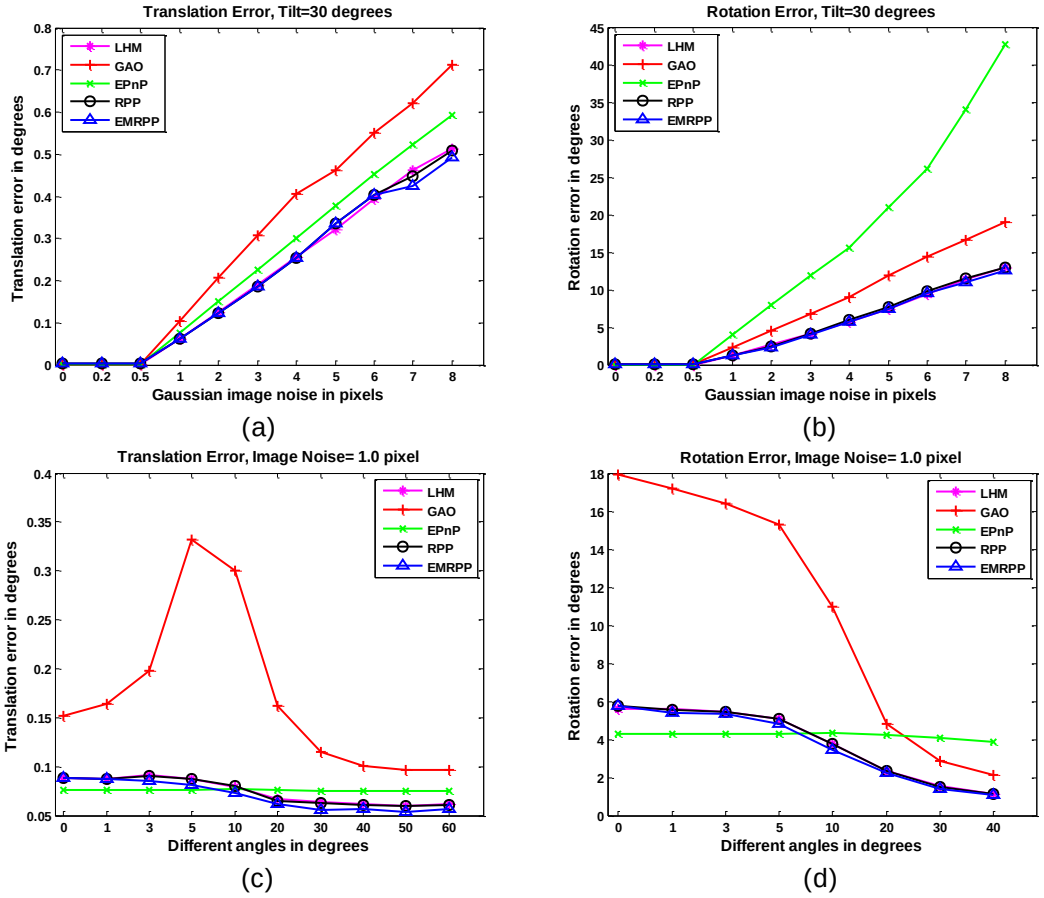


Figure 5. Translation Error and Rotation Error with Different Image Noise are Shown in (a) and (b). Translation Error and Rotation Error with Different Tilt Angles are Shown in (c) and (d)

5.2.Discussion

The computation time is also important. In this paper, algorithms were written in C/C++ and tested in a notebook which had two i3 CPUs and 4G memory. The computation time of these algorithms could be seen in Table 1. GAO and EPnP algorithms perform fast, but they have low accuracy. LHM has higher accuracy and less computation time, but it suffers from pose ambiguity. EMRPP and RPP have quite good accuracy of pose estimation. Both computation time of EMRPP and RPP are below 2ms and certainly meet the requirements of real-time applications. In fact, our EMRPP usually has higher accuracy than RPP and performs 10% faster than RPP.

Here LHM, RPP and EMRPP are iterative algorithms, so they have higher accuracy than non-iterative ones. RPP is in fact the modified algorithm of LHM in plane situation. So RPP and LHM have similar accuracy when they both obtain the correct results. Our EMRPP is combined by EPnP and MRPP. EPnP is quite fast and could provide good initial pose estimation. The good initial estimation would speed up the following iterative section of EMRPP and get more robust and accurate pose results. So EMRPP is faster and has higher accuracy than RPP.

Table 1. Computation Time in Microsecond

	LHM	GAO	EPnP	RPP	EMRPP
Mean (μs)	544	30	109	1739	1592
Median (μs)	508	28	99	1613	1484

5.3.Three/Two rotors Observed

When there are only two or three rotors observed, pose could not be calculated by pure vision algorithm. IMU data would be used to calculate the pose. We test these algorithms with different image noise and IMU noise separately. We compare IMU+2P and IMU+3P with GAO, LHM and EPnP. Here GAO, LHM and EPnP are performed with all of the four rotors. Especially, they are all pure vision algorithms and don't use the IMU data. They are shown here only for comparision.

5.3.1. Pose Error with Different Image Noise

Figure 6a shows the translation error with different image noise. IMU+2P algorithm is figured in cyan square and IMU+3P is figured in blue triangles. Here 0.5 degree IMU noise is only for IMU+2P and IMU+3P. It could be seen in Figure 6a that IMU+2P has the largest translation error in these algorithms. The translation error of IMU+3P is between EPnP and GAO. LHM and RPP have good translation accuracy. Figure 6b shows the rotation error with different image noise. Here IMU+2P and IMU+3P have higher rotation accuracy than GAO, LHM and EPnP algorithms. IMU+2P have slightly smaller rotation error than IMU+3P.

5.3.2. Pose Error with Different IMU Noise

Figure 6c shows the translation error with different IMU noise. The image noise is set as 1.0 pixel here. IMU+2P has worst translation accuracy than other algorithms. When IMU noise is small, IMU+3P has good accuracy. But when IMU noise is more than 1.0 degree, translation error of IMU+3P increases to a high level. Figure 6d shows the rotation error with different IMU noise. When IMU noise is not large, IMU+2P and IMU+3P have quite good rotation accuracy. Besides, IMU+2P has higher rotation accuracy than IMU+3P.

5.4 Discussion

From Figure 6, we could see that IMU+2P has the lowest translation accuracy. The translation accuracy of IMU+3P is higher than IMU+2P and similar to GAO, EPnP. But they are all lower than translation accuracy of LHM and RPP. However, IMU+2P and IMU+3P have higher rotation accuracy than GAO, LHM, EPnP and RPP. Usually IMU+2P has higher rotation accuracy than IMU+3P. This could be explained. The number of point correspondences is related to translation estimation. When number of points is in a certain range, the more points we use, the higher translation accuracy we will have. Here the quadrotor has only four rotors, and each observed rotor is important. When only two rotors are observed, the translation accuracy would be lowest. When three rotors are observed, IMU+3P has higher accuracy. But the translation error of IMU+2P and IMU+3P is usually higher than EPnP, LHM and RPP. Because EPnP, LHM and RPP usually use four point correspondences to get translation results. As for the rotation error, roll and pitch degrees could be had by IMU. Usually IMU has a quite high accuracy, so IMU+2P and IMU+3P have higher rotation accuracy than GAO,

LHM, EPnP and RPP. When the number of point correspondences is small and not large, the more point correspondences may disturb the accuracy of rotation calculating especially that there are high accuracy IMU data. From Figure 6b and Figure 6d we could see IMU+3P have lower rotation accuracy than IMU+2P, when IMU+3P uses three point correspondences. LHM, GAO, EPnP and RPP all have lower accuracy than IMU+2P and IMU+3P, even though they have used all of the four point correspondences.

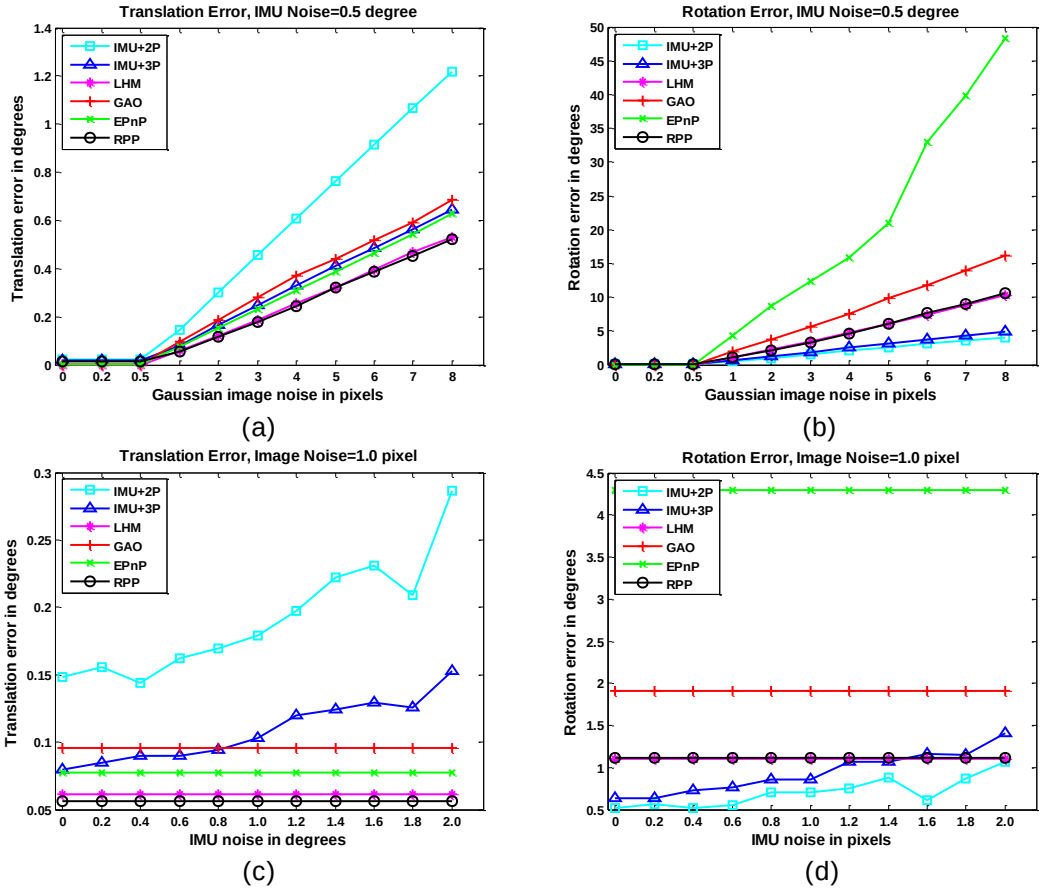


Figure 6. Translation Error and Rotation Error with Different Image Noise are Shown in (a) and (b). Translation Error and Rotation Error with Different IMU Noise are Shown in (c) and (d).

6. Real Experiments

We have performed our algorithms in real outdoor environments. An industrial camera is fixed on the ground and connected to a desktop computer by 1394b connecter. In order to compare the accuracy, we need to get the ground-truth here. Here we select the translation results of RPP as true height Z_q. Then this value Z_q is used in equation 5 and equation 6 to get the true X_q and Y_q. The real rotation angles are obtained from IMU data and electronic compass. From IMU data, we have the true roll angle and pitch angle. By electronic compass, the true yaw angle is obtained. The translation error is the angle between the estimated translation direction and the ground-

truth direction. The rotation error is the smallest angle of rotation to bring the estimated rotation to the ground-truth rotation.

6.1. Results of Real Experiments

The results of real experiments are shown in Figure 7. When all of the four rotors are observed, EMRPP is performed. IMU+3P and IMU+2P are also performed by using three or two point correspondences randomly to compare with other algorithms. When only two or three rotors are observed, only IMU+2P or IMU+3P is performed. Here we only figure the results of position, the results of orientation are not figured. The ground-truth is figured in yellow. Figure 7a shows the entire experiment results. In real experiment, GAO and EPnP have higher pose error than other algorithms. EMRPP, RPP and IMU+3P have good results of pose estimation in our real experiments. The results of EMRPP are slightly better than RPP. And the results of IMU+2P are slightly worse than IMU+3P. In order to show the details of results clearly, results of 20 sequential frames are shown in Figure 7b.

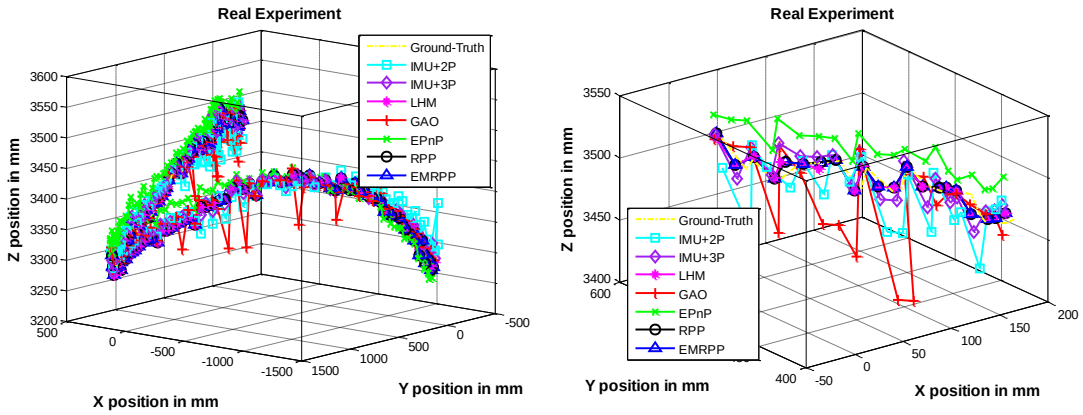


Figure. 7. The Results of Real Experiments. The Entire Result of the Experiment is Shown in Figure 7a. In Order to show the Details Clearly, a Small Part of the Experiment Results is shown in Figure 7b.

6.2. Pose Error in Real Experiments

Figure 8a shows the translation error in real experiments. The X axis indicates the sequential frames. Here the total quantity of sequential is 270. It should be enough to show the error of algorithms. From Figure 8a we could see that EMRPP, RPP, IMU+3P and LHM have low translation error. The results of EPnP are a little unstable. EMRPP and RPP have better translation results. Besides, the translation errors of EMRPP are slightly lower than those of RPP. Figure 8b shows the rotation error of these algorithms in real experiments. The results of EPnP and GAO are bad. EMRPP, LHM, IMU+2P and IMU+3P have good rotation accuracy. The rotation results of EMRPP and IMU+2P are better than other algorithms.

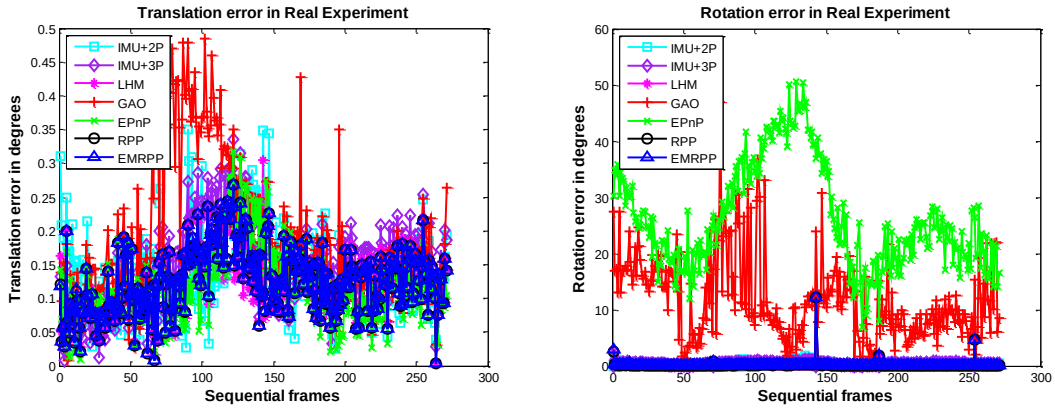
6.3. Discussion

For real applications of quadrotors in outdoor environments, we need the combination of EMRPP, IMU+3P and IMU+2P algorithms. When we have observed all of the four rotors, we employ the EMRPP algorithm to get the robust pose estimation.

EMRPP is a pure vision algorithm which has quite good accuracy and don't need extra IMU data. When there is no IMU device or IMU data has error, EMRPP could have good results independently.

Sometimes, there may be only three or two rotors observed. In our real experiments, this situation seldom happens. But we also need pay much attention to it. Some person may say that they could track or forecast the pose. Attitude tracking algorithm was proposed for a quadrotor when faults occurred in [24]. But for quadrotors, some sudden situations may occur at this moment. So calculated results which using current real data are necessary, especially when three or two rotors are observed.

When three rotors are observed, we could use IMU+3P algorithm. IMU+3P has both quite good accuracy of translation and rotation. It has the balance between translation and rotation, its accuracy is close to RPP and EMRPP. So IMU+3P can meet the requirement of our real applications. When two rotors are observed, we can perform the IMU+2P algorithm. Because the rotation accuracy of IMU+2P is quite high, we could trust the rotation results of IMU+2P. Usually, the translation error of IMU+2P is a little big. So we can combine the results of IMU+2P and former other results to have the better translation estimation.



(a) (b)

Figure 8. The Translation Error and the Rotation Error in Real Experiments

7. Conclusion

In this paper, we present an external vision based robust pose estimation system in different observed situations for a quadrotor in outdoor environments. This system could provide us with approximate ground truth of a flying quadrotor.

We don't need to modify the architecture of a quadrotor or put some additional components such as colored markers on it. The flying quadrotor is seen as a fixed landmark. Only using the quadrotor's own features, we propose robust and accurate algorithms to deal with the different observed situations. Before pose estimation algorithms perform, a pure vision-based approach is implemented to get preliminary position results. When four rotors are observed, we present the EMRPP algorithm which has higher accuracy and less computation time than RPP algorithm. When dealing with occlusion or poor tracking results, there may be only three or two rotors observed. This moment, most of other vision-based pose estimation systems for the quadrotor don't mention these cases and could not get right pose results. Here we

present IMU+3P and IMU+2P algorithms combining IMU data to have right results of pose estimation for these cases.

We have performed our system in real experiments. The results demonstrate that our real-time pose estimation system for the quadrotor could perform accurately and robustly in different observed situations.

Acknowledgements

This research was supported by the National Science and Technology Major Project of the Ministry of Science and Technology of China: ITER(No.2012GB102007).

The authors would like to thank the editors and the anonymous reviewers for their thoughtful reviews and suggestions.

References

[1] C. S. Ha and D. Lee, "Vision-Based Teleoperation of Unmanned Aerial and Ground Vehicles", Proceedings of the IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, doi: 10.1109/ICRA.2013.6630764, **(2013)**, pp. 1465-1470.

[2] E. Altug, J. P. Ostrowski and R. Mahony, "Control of A Quadrotor Helicopter Using Visual Feedback", Proceedings of the IEEE International Conference on Robotics and Automation, Washington, USA, doi: 10.1109/ROBOT.2002.1013341, **(2002)**, pp. 72-77.

[3] E. Altug, J. P. Ostrowski and C. J. Taylor, "Quadrotor Control Using Dual Camera Visual Feedback", Proceedings of the IEEE International Conference on Robotics and Automation, doi: 10.1109/ROBOT.2003.1242264, **(2003)**, pp. 4294-4299.

[4] S. Park, D. H. Won, M. S. Kang, T. J. Kim and H. G. Lee, "RIC(Robust Internal-loop Compensator) Based Flight Control of a Quad-Rotor Type UAV", Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton, Canada, doi: 10.1109/IROS.2005.1545113, **(2005)**, pp. 3542-3547.

[5] M. Achtelik, T. Zhang, K. Kuhnlenz and M. Buss, "Viusal Tracking and Control of a Quadcopter Using a Stereo Camera System and Inertial Sensors", Proceedings of the International Conference on Mechatronics and Automation, Changchun, China, doi: 10.1109/ICMA.2009.5246421, **(2009)**, pp. 2863-2869.

[6] J. P. How, B. Bethke, A. Frank, D. Dale and J. Vian, "Real-Time Indoor Autonomous Vehicle Test Environment", IEEE Control Systems Magazine, doi: 10.1109/MCS.2007.914691, vol. 28, no. 2, **(2008)**, pp. 51-64.

[7] S. Ahrens, D. Levine, G. Andrews and J. P. How, "Vision-Based Guidance and Control of a Hovering Vehicle in Unknown, GPS-denied Environments", Proceedings of the IEEE International Conference on Robotics and Automation, Kobe, Japan, doi: 10.1109/ROBOT.2009.5152680, **(2009)**, pp. 2643-2648.

[8] D. Abeywardena, Z. Wang, S. Kodagoda and G. Dissanayake, "Visual-Inertial Fusion for Quadrotor Micro Air Vehicles with Improved Scale Observability", Proceedings of the IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, doi: 10.1109/ICRA.2013.6631015, **(2013)**, pp. 3148-3153.

[9] S. Klose, J. Wang, M. Achtelik, G. Panin, F. Holzapfel and A. Knoll, "Markerless, Vision-Assisted Flight Control of a Quadrocopter", Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, doi: 10.1109/IROS.2010.5649019, **(2010)**, pp. 5712-5717.

[10] A. Breitenmoser, L. Kneip and R. Siegwart, "A Monocular Vision-based System for 6D Relative Robot Localization", Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, USA, doi: 10.1109/IROS.2011.6094851, **(2011)**, pp. 79-85.

[11] W. J. Wolfe, D. Mathis, C. W. Sklair and M. Magee, "The Perspective View of Three Points", IEEE Transactions on Pattern Analysis and Machine Intelligence, doi: 10.1109/34.67632, vol. 13, no. 1, **(1991)**, pp. 66-73.

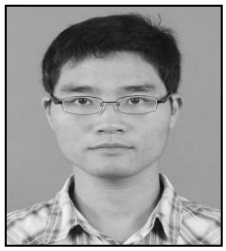
[12] X.-S. Gao, X.-R. Hou, J. Tang and H.-F. Cheng, "Complete Solution Classification for the Perspective-Three-Point Problem", IEEE Transactions on Pattern Analysis and Machine Intelligence, doi: 10.1109/TPAMI.2003.1217599, vol. 25, no. 8, **(2003)**, pp. 930-943.

[13] M. Fischler and R. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography", Communications of the ACM, vol. 24, no. 6, **(1981)**, pp. 381-395.

[14] Z.Y. Hu and F.C. Wu, "A Note on the Number of Solutions of the Noncoplanar P4P Problem", IEEE Transactions on Pattern Analysis and Machine Intelligence, doi: 10.1109/34.993561, vol. 24, no. 4, **(2002)**, pp. 550-555.
[15] D. F. DeMenthon and L. S. Davis, "Model-Based Object Pose in 25 Lines of Code", International Journal of Computer Vision, vol. 15, no. 1-2, **(1995)**, pp. 123-141.
[16] L. Quan and Z. Lan, "Linear N-point Camera Pose Determination", IEEE Transactions on Pattern Analysis and Machine Intelligence, doi: 10.1109/34.784291, vol. 21, no. 7, **(1999)**, pp. 774-780.
[17] A. Ansar and K. Daniilidis, "Linear Pose Estimation from Points or Lines", IEEE Transactions on Pattern Analysis and Machine Intelligence, doi: 10.1109/TPAMI.2003.1195992, vol. 25, no. 5, **(2003)**, pp. 578-589.
[18] C. P. Lu, G. D. Hager and E. Mjolsness, "Fast and Globally Convergent Pose Estimation from Video Images", IEEE Transactions on Pattern Analysis and Machine Intelligence, doi: 10.1109/34.862199, vol. 22, no. 6, **(2000)**, pp. 610-622.
[19] G. Schweighofer and A. Pinz, "Robust Pose Estimation from a Planar Target", IEEE Transactions on Pattern Analysis and Machine Intelligence, doi: 10.1109/TPAMI.2006.252, vol. 28, no. 12, **(2006)**, pp. 2024-2030.
[20] V. Lepetit, F. Moreno-Noguer and P. Fua, "EPnP: Accurate Non-Iterative O(n) Solution to the PnP Problem", International Journal of Computer Vision (IJCV), doi: 10.1007/s11263-008-0152-6, vol. 81, no. 2, **(2008)**, pp. 151-166.
[21] F. Fraundorfer, P. Transkanen and M. Pollefeys, "A Minimal Case Solution to the Calibrated Relative Pose Problem for the Case of Two Known Orientation Angles", Proceedings of the European Conference on Computer Vision, Crete, Greece, doi: 10.1007/978-3-642-15561-1_20, **(2010)**, pp. 269-282.
[22] Z. Kukelova, M. Bujnak and T. Pajdla, "Closed-form Solutions to the Minimal Absolute Pose Problems with Known Vertical Direction", Proceedings of the Asian Conference on Computer Vision, **(2010)**, Queenstown, New Zealand, doi: 10.1007/978-3-642-19309-5_17, pp. 216-229.
[23] R. Hartley and A. Zisserman, "Multiple View Geometry in Computer Vision (Second Edition)", Cambridge University Press, **(2004)**.
[24] H. Khebbache, "Robust Control Algorithm Considering the Actuator Faults for Attitude Tracking of an UAV Quadrotor Aircraft", International Journal of Control and Automation, vol. 5, no. 4, **(2012)**, pp. 55-66.
[25] A. Wendel, A. Irschara and H. Bischof, "Natural Landmark-based Monocular Localization for MAVs", Proceedings of the IEEE International Conference on Robotics and Automation, Shanghai, China, doi: 10.1109/ICRA.2011.5980317, **(2011)**, pp. 5792-5799.
[26] H. Lim, S. N. Sinha, M. F. Cohen and M. Uyttendaele, "Real-time Image-based 6-DOF Localization in Large-Scale Environments", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Providence, Rhode Island, doi: 10.1109/CVPR.2012.6247782, **(2012)**, pp. 1043-1050.

Authors

Wei Zheng, he is currently a Ph.D. candidate in Pattern Recognition and Intelligent System at Department of Automation, University of Science and Technology of China (USTC). He received his Bachelor's degree of Engineering from USTC in 2009. His research interests include unmanned aerial vehicle, robot localization and navigation, visual SLAM and multi-sensor fusion.



Fan Zhou, he received the B.Eng. degree from University of Science and Technology of China (USTC) in 2009. He is now a Ph.D. candidate in pattern recognition and intelligent system, USTC. His research interests include unmanned aerial robot, integrated navigation, vision SLAM and adaptive signal processing.

Zengfu Wang, he is a Professor at Department of Automation, University of Science and Technology of China (USTC). He is also a Professor at Institute of Intelligent Machines, Chinese Academy of Sciences. His research interests include: computer vision, human computer interaction and intelligent robot.