Robust Pose Estimation for Multirotor UAVs Using Off-Board Monocular Vision

Qiang Fu, Quan Quan, and Kai-Yuan Cai

Abstract—**This paper deals with the problem of pose estimation (or motion estimation) for multirotor unmanned aerial vehicles (UAVs) by using only an off-board camera. An extended Kalman filter (EKF) is often adopted to solve this problem. However, the accuracy and robustness of an EKF are limited partly by the usage of an existing linear constant-velocity process model applicable to many rigid objects. For such a reason, a nonlinear constant-velocity process model featured with the characteristics of multirotor UAVs is proposed in this paper, the superiority of which is explained from the perspective of observability. With the new process model and a generic camera model, a practical EKF method suitable for conventional cameras and fish-eye cameras is then proposed. By taking EKF implementation into account, a general correspondence method that could handle any number of feature points is further designed. Simulation and real experiments show that the proposed EKF method is more robust against noise and occlusion than currently employed filtering methods.**

Index Terms—**Monocular vision, multirotor unmanned aerial vehicle (UAV), pose estimation, process model.**

I. Introduction

NOWADAYS multirotor unmanned aerial vehicles (UAVs) are being widely used in many applications. For example, Amazon has designed and tested a future delivery system—Prime Air to deliver goods using multirotor UAVs [1]. PRENAV uses a ground robot and a multirotor UAV in coordination (ground-air cooperation) to perform industrial inspection [2]. Accurate and reliable pose estimation is a fundamental issue for autonomous operation of these vehicles. Although global positioning system (GPS) is commonly-used, it is not suitable for GPS-denied situations, such as urban areas or inside buildings. In contrast, vision-based navigation methods do not depend on GPS and could provide high-precise pose within a close range. Considering that small multirotor UAVs often feature CPUs with limited capabilities [3], this paper focuses on the estimation of the pose of multirotor UAVs for visual servoing (vision-based control) by using only an off-board monocular camera. In existing monocular motion capture systems, such as Vicon [4] and OptiTrack [5] in single-camera mode, marker missing [e.g., caused by occlusion or camera field of view (FOV)] usually happens and accurate pose information cannot be obtained when the number of detected markers is less than three. For such a reason, this paper aims to solve this problem from the following two aspects: 1) use fish-eye cameras as they can provide images with a very large FOV (about 180°) without requiring external mirrors or rotating devices [6]; 2) propose a nonlinear constant-velocity process model with less unknowns.

In computer vision, estimating the pose of a calibrated camera from n 3D-2D point correspondences is known as Perspective-n-Point (PnP) problem [7]. A comprehensive overview of PnP algorithms could be found in [8] and references therein. It is easy to transform the problem of estimating rigid object pose into a PnP problem. Therefore, existing pose estimation methods for rigid objects (including multirotor UAVs) can be generally classified into three categories: linear methods [8]–[10], iterative methods [11]–[13], and recursive methods [14]–[18]. Linear methods are simple and intuitive, but are sensitive to noises. Iterative methods are more accurate and robust, but they are computationally more intensive than linear methods and prone to fall into local minima. Recursive methods rely on temporal filtering methods, especially Kalman filters. These methods are accurate, efficient, and suitable for image sequence processing. Since the measurement model is nonlinear in the system states (determined by the camera imaging model), an extended Kalman filter (EKF) is often adopted for visual servoing of rigid objects (including multirotor UAVs). However, the process model of EKF is a linear constant-velocity process model applicable to many rigid objects [14]–[16], which is not a very appropriate model for multirotor UAVs (more markers have to be detected so that the states can be observed). On the other hand, most of recursive methods [14]–[16] are based on conventional cameras, which obey the pinhole projection model and provide a limited FOV. But these methods are inapplicable to fish-eye camera pose estimation directly because fish-eye cameras can provide images with a very large FOV (about 180°) and the pinhole camera model is no longer valid.

In order to solve the process modeling problem, a nonlinear constant-velocity process model featured with the characteristics of multirotor UAVs is proposed in this paper. Multirotor

Manuscript received September 20, 2016; revised January 27, 2017; accepted March 29, 2017. Date of publication April 24, 2017; date of current version September 11, 2017. This work was supported by the National Key Project of Research and Development Plan under Grant 2016YFC1402500. *(Corresponding author: Qiang Fu.)*

Q. Fu was with the School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China. He is now with the School of Instrumentation Science and Opto-electronics Engineering, Beihang University, Beijing 100191, China (e-mail: fq@buaa.edu.cn).

Q. Quan and K.-Y. Cai are with the School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China (e-mail: qq_buaa@buaa.edu.cn; kycai@buaa.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIE.2017.2696482

0278-0046 © 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

UAVs are under-actuated systems with four independent inputs (a thrust force perpendicular to the propeller plane and three moments) and six coordinate outputs [19]. Thus, the number of unknowns in the proposed process model, namely a nonlinear constant-velocity model, are set to four instead of six as in linear constant-velocity process models [14]–[16]. Then, based on the new process model and a generic camera model [20], an EKF method is proposed to fuse the process model and off-board monocular vision information. Thanks to the proposed process model, the robustness of pose estimation against noise and occlusion is improved. The minimum number of feature points observed by the camera is reduced from three to two, which is analyzed from the perspective of observability in this paper. Also, thanks to the generic camera model, the proposed EKF method is suitable for conventional cameras as well as fish-eye cameras. The research results in this paper are helpful to improve the robustness of single-camera motion capture systems or ground-air cooperation systems. Additionally, the implementation of majority vision-based EKF methods requires 3D-2D point correspondences to be known. A correspondence algorithm for monocular pose estimation is proposed in [11], but it fails if less than four feature points are detected in the image (e.g., caused by occlusion or camera FOV). This means that EKF cannot work in these situations. To solve the correspondence problem, a general correspondence method combining EKF and the algorithm in [11] is also proposed in this paper, which could handle any number of image points.

The main contributions of this paper are as follows.

1) A nonlinear constant-velocity process model featured with the characteristics of multirotor UAVs is proposed.
2) Based on the new process model and a generic camera model, an EKF method for cameras equipped with generic lens is proposed.
3) To implement the EKF method in real systems, a general correspondence method that could handle any number of feature points is proposed.
4) The proposed EKF method is more robust against noise and occlusion than existing filtering methods, the superiority of which is explained from the perspective of observability.

This paper is organized as follows. Some preliminaries and problem formulation are introduced in Section II. In Section III, a correspondence-based EKF method and observability analysis are presented. Then, the experimental results are reported in Sections IV and V, followed by the conclusion in Section VI.

Following notations are adopted in this paper. $\mathbb{R}^n$ is Euclidean space of dimension n. $\|\cdot\|$ denotes the Euclidean vector norm or induced matrix norm. $\mathbf{I}_n$ is the identity matrix with dimension n. $\mathbf{0}_{m\times n}$ is a zero vector or a zero matrix with dimension $m \times n$. The gradient of the vector function $\mathbf{g} \in \mathbb{R}^m$ is given by $\partial\mathbf{g}(\mathbf{x})/\partial\mathbf{x} = [\partial\mathbf{g}(\mathbf{x})/\partial x_1 \ \cdots \ \partial\mathbf{g}(\mathbf{x})/\partial x_n] \in \mathbb{R}^{m\times n}$. diag$[a_1, a_2, \ldots, a_n]$ denotes a diagonal matrix with $a_1, a_2, \ldots, a_n$ as its diagonal elements. The rank of the matrix $\mathbf{A}$ is given by rank$(\mathbf{A})$.

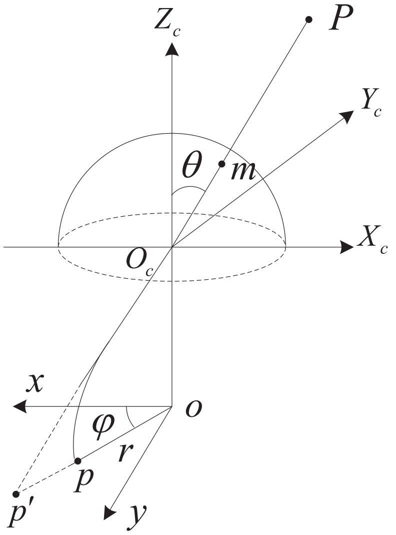


Fig. 1. Generic camera model [15]. The 3-D point P is imaged at p by a fish-eye camera, while it would be p' using a pinhole camera.

II. Preliminaries and Problem Formulation

A. Generic Camera Model

As shown in Fig. 1, $O_c - X_cY_cZ_c$ denotes the camera coordinate system and $o - xy$ is the image coordinate system (unit mm). A three dimensional (3-D) point $\mathbf{P} = [X\ Y\ Z]^T \in \mathbb{R}^3$ is projected to $\mathbf{m} = [\sin\theta\cos\varphi\ \sin\theta\sin\varphi\ \cos\theta]^T$ on the unit hemisphere centered at O_c. Thus, it is easy to derive that

$$\mathbf{m} = \frac{\mathbf{P}}{\|\mathbf{P}\|}. \tag{1}$$

The 3-D point P is imaged at p by a fish-eye camera, while it would be p' by a pinhole camera.

A generic model suitable for both conventional and fish-eye cameras is proposed as follows [20]:

$$r(\theta) = k_1\theta + k_2\theta^3 + k_3\theta^5 + k_4\theta^7 + k_5\theta^9 + \cdots . \tag{2}$$

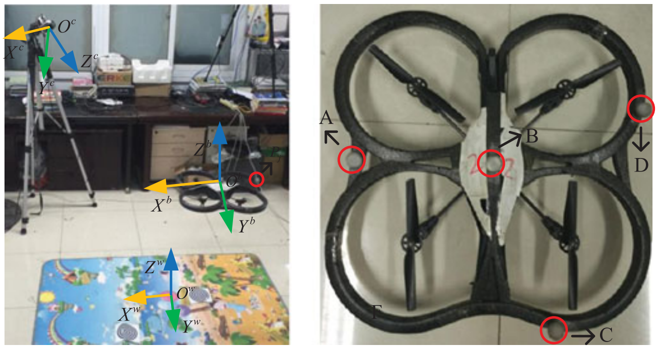
In this paper, we choose the model that contains only the five parameters k_1, k_2, k_3, k_4, k_5, as it is found that the first five terms can approximate different projection curves. Therefore, the image coordinates of p (or p') in $o - xy$ is obtained by

$$\begin{bmatrix} x \\ y \end{bmatrix} = r(\theta)\begin{bmatrix} \cos\varphi \\ \sin\varphi \end{bmatrix} \tag{3}$$

where $r(\theta)$ is defined in (2), and φ is the angle between the radial direction and the x-axis. Then, the pixel coordinates (u, v) can be derived from

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} m_u & 0 \\ 0 & m_v \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} \tag{4}$$

where (u_0, v_0) is the principal point, and m_u, m_v are the number of pixels per unit distance in horizontal and vertical directions, respectively. Thus, the intrinsic parameters of a camera



(a) (b)

Fig. 2. (a) Illustration of different coordinate systems. (b) Multirotor UAV (quadrotor) fixed with four reflective markers (reflective markers are used as feature points in this paper).

are $(k_1, k_2, m_u, m_v, u_0, v_0, k_3, k_4, k_5)$, which can be obtained through camera calibration [20], [21].

B. Feature Point Imaging

As shown in Fig. 2, $O^b - X^bY^bZ^b$ denotes the body coordinate system of a multirotor UAV, $O^c - X^cY^cZ^c$ is the camera coordinate system, and $O^w - X^wY^wZ^w$ is the world coordinate system. Let $(\mathbf{T}, \Theta)$ be the relative pose between the body coordinate system and the world coordinate system, where $\mathbf{T} = [X\ Y\ Z]^T$ is the relative position vector and $\mathbf{\Theta} = [\theta\ \varphi\ \phi]^T$ is the relative orientation defined by the *pitch*, *yaw*, and *roll* parameters. Then, the transformation of $\mathbf{P}_j$ (the jth feature point fixed rigidly with the multirotor UAV) from the body coordinate system to the world coordinate system is described as

$$\mathbf{P}_j^w = \mathbf{T} + \mathbf{R}(\Theta)\mathbf{P}_j^b \tag{5}$$

where $\mathbf{P}_j^w = \left[X_j^w\ Y_j^w\ Z_j^w\right]^T$ is the coordinate vector of the jth feature point in the world coordinate system, $\mathbf{P}_j^b = \left[X_j^b\ Y_j^b\ Z_j^b\right]^T$ is the coordinate vector of the jth feature point in the body coordinate system (known beforehand), and the rotation matrix $\mathbf{R}$ is given as follows:

$$\mathbf{R}(\Theta) = \begin{bmatrix} \mathrm{c}\theta \mathrm{c}\varphi & \mathrm{s}\phi\mathrm{s}\theta\mathrm{c}\varphi - \mathrm{c}\phi\mathrm{s}\varphi & \mathrm{c}\phi\mathrm{s}\theta\mathrm{c}\varphi + \mathrm{s}\phi\mathrm{s}\varphi \\ \mathrm{c}\theta\mathrm{s}\varphi & \mathrm{s}\phi\mathrm{s}\theta\mathrm{s}\varphi + \mathrm{c}\phi\mathrm{c}\varphi & \mathrm{c}\phi\mathrm{s}\theta\mathrm{s}\varphi - \mathrm{s}\phi\mathrm{c}\varphi \\ -\mathrm{s}\theta & \mathrm{s}\phi\mathrm{c}\theta & \mathrm{c}\phi\mathrm{c}\theta \end{bmatrix} \tag{6}$$

where c and s are abbreviations for cosine and sine, respectively. Suppose that the transformation of $\mathbf{P}_j$ from the world coordinate system to the camera coordinate system is described as

$$\mathbf{P}_j^c = \mathbf{T}_w^c + \mathbf{R}_w^c\mathbf{P}_j^w \tag{7}$$

where $\mathbf{P}_j^c = \left[X_j^c\ Y_j^c\ Z_j^c\right]^T$ is the coordinate vector of the jth feature point in the camera coordinate system and $\mathbf{R}_w^c, \mathbf{T}_w^c$ are known beforehand. Based on (5) and (7), it is derived that

$$\mathbf{P}_j^c = \mathbf{T}_b^c + \mathbf{R}_\mathbf{b}^\mathbf{c}\mathbf{P}_j^b \tag{8}$$

where $\mathbf{T}_b^c = \mathbf{T}_w^c + \mathbf{R}_w^c\mathbf{T}, \mathbf{R}_\mathbf{b}^\mathbf{c} = \mathbf{R}_\mathbf{w}^\mathbf{c}\mathbf{R}(\mathbf{\Theta})$. Then, the jth feature point $\mathbf{P}_j$ is imaged at $\mathbf{p}_j = [u_j^i\ v_j^i]^T$ and its coordinates are given by (1)–(4).

If the 3D-2D feature correspondences $\mathbf{P}_j \leftrightarrow \mathbf{p}_j$ are known; then, the measurement model that defines the relationships between the output measurements and the states could be expressed as follows:

$$\mathbf{z}_k = \mathbf{g}\left(\mathbf{x}_k\right) + \mathbf{v}_k \tag{9}$$

where $\mathbf{z}_k = \left[u_1^i\ v_1^i\ \ldots\ u_{n_F}^i\ v_{n_F}^i\right]_k^T \in \mathbb{R}^{2n_F}$ is the measurement vector for n_F feature points, and $\mathbf{g}\left(\mathbf{x}_k\right) \in \mathbb{R}^{2n_F}$ can be obtained from (8) and (1)–(4). Besides, $\mathbf{v}_k$ is the measurement noise vector and each element of $\mathbf{v}_k$ is considered as independent zero-mean Gaussian distribution with variance R_i. Therefore, the measurement-noise-variance matrix $\mathbf{R}_k \in \mathbb{R}^{2n_F \times 2n_F}$ is a diagonal matrix with diagonal elements being R_i.

Note that when $\mathbf{R}_w^c = \mathbf{I}_3, \mathbf{T}_w^c = \mathbf{0}_{3\times1}$, the problem discussed in this paper degenerates to that in [11]. Thus, the problem formulation of this paper is more general. Substituting (8) into (1)–(4) leads to two nonlinear equations with six unknown parameters. Therefore, at least three noncollinear points are required for pose estimation in theory. In fact, it is pointed out that the number of feature points n_F satisfies $4 \leq n_F \leq 6$ in many robotic visual servoing applications [15].

C. Process Model for Multirotor UAVs

The nominal model used in this paper is a constant-velocity motion model. The translational motion model for multirotor UAVs [19] is described as

$$\dot{\mathbf{T}} = \mathbf{V} \tag{10}$$

$$\dot{\mathbf{V}} = \mathbf{R}(\Theta)u\mathbf{a} - g\mathbf{a} \tag{11}$$

where $\mathbf{V} = [V_x\ V_y\ V_z]^T \in \mathbb{R}^3$ is the relative linear velocity vector, g is the gravity acceleration 9.81 m/s^2, $\mathbf{a} = [0\ 0\ 1]^T \in \mathbb{R}^3$, and $u \in \mathbb{R}$ is the control input. In [19], u could be known by using an inertial measurement unit (IMU). However, u is unknown in this paper since only off-board vision information is exploited (IMU information is not used in this paper). Here, it is assumed that $u \approx g$ in many situations such as hovering, so that we have $u = g + \varepsilon_1$, where ε_1 is modeled as a Gaussian noise.

Note that when multirotor UAVs are accelerating or decelerating, the roll and pitch could become significant, which will absolutely cause problem for the proposed translational motion model. However, the roll and pitch angles ϕ, θ are usually constrained by "saturation" in the implementation process of proportion–integration–differentiation controllers for multirotor UAVs. For example, there are $\|\phi\| \leq 0.2$ rad, $\|\theta\| \leq 0.2$ rad in the real-time experiments (see Section V) of this paper. It is known from (11) that $u = g/\left(\cos\theta\cos\phi\right) \leq 10.21$ m/s^2 in order to balance the gravity. As shown in Section V, the proposed translational motion model is still valid when ϕ, θ are small.

As only off-board vision information is used in this paper, the relative angular velocity $\mathbf{W} = [w_1\ w_2\ w_3]^T \in \mathbb{R}^3$ is also unknown. Here, it is assumed that $\mathbf{W}$ is constant during each sample period as in [14]–[16]. This assumption is reasonably valid for sufficiently small sample periods in robotic visual servoing systems [15]. Therefore, the rotational motion model for multirotor UAVs is described as

$$\dot{\boldsymbol{\Theta}} = \mathbf{W} \tag{12}$$

$$\dot{\mathbf{W}} = \xi \tag{13}$$

where $\xi = [\varepsilon_2\ \varepsilon_3\ \varepsilon_4]^T \in \mathbb{R}^3$ is modeled as Gaussian noises. Finally, the process model for multirotor UAVs is written as follows:

$$\dot{\mathbf{T}} = \mathbf{V} \tag{14}$$

$$\dot{\mathbf{V}} = \mathbf{R}(\Theta)(g + \varepsilon_1)\mathbf{a} - g\mathbf{a} \tag{15}$$

$$\dot{\boldsymbol{\Theta}} = \mathbf{W} \tag{16}$$

$$\dot{\mathbf{W}} = \xi. \tag{17}$$

Let T_s denote the sampling time. Using the first-order backward difference, the discrete form of the proposed process model for Kalman filtering is described as

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}) + \boldsymbol{\Gamma}(\mathbf{x_{k-1}})\,\varepsilon_{k-1} \tag{18}$$

where $\mathbf{x}_k = [X_k\ V_{x,k}\ Y_k\ V_{y,k}\ Z_k\ V_{z,k}\ \theta_k\ w_{1,k}\ \varphi_k\ w_{2,k}\ \phi_k\ w_{3,k}]^T \in \mathbb{R}^{12}$, $\varepsilon_{k-1} = [\varepsilon_{1,k-1}\ \varepsilon_{2,k-1}\ \varepsilon_{3,k-1}\ \varepsilon_{4,k-1}]^T \in \mathbb{R}^4$, and $\mathbf{f}(\mathbf{x}_{k-1}) \in \mathbb{R}^{12}$, $\boldsymbol{\Gamma}(\mathbf{x_{k-1}}) \in \mathbb{R}^{12\times 4}$ (see Appendix for details). The noises $\varepsilon_{i,k-1}$ $(i = 1, \ldots, 4)$ are considered as independent zero-mean Gaussian distribution with variance Q_i. Therefore, the process-noise-variance matrix $\mathbf{Q}_{k-1} \in \mathbb{R}^{4\times 4}$ is a diagonal matrix with diagonal elements being Q_i.

Note that (18) is a nonlinear constant-velocity process model. The commonly-used linear constant-velocity process model [14]–[16] is depicted as follows:

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \gamma_k \tag{19}$$

where $\mathbf{A} \in \mathbb{R}^{12\times 12}$ is a block diagonal matrix with 2×2 blocks of the form $\left[\begin{smallmatrix} 1 & T_s \\ 0 & 1 \end{smallmatrix}\right]$, and $\gamma_k = [0\ \gamma_k^1\ 0\ \gamma_k^2\ 0\ \gamma_k^3\ 0\ \gamma_k^4\ 0\ \gamma_k^5\ 0\ \gamma_k^6]^T \in \mathbb{R}^{12}$ models the motion uncertainties. Compared to (19), the dimension of the inputs in (18) reduces from six to four. Therefore, (18) may be more robust against occlusion (see Sections IV-C and V). In this paper, the linear constant-velocity process model (19) as used in [14]–[16] will be compared with the proposed nonlinear constant-velocity process model (18) by observability analysis and experimental results.

Note that in (19) both the relative linear velocity $\mathbf{V} \in \mathbb{R}^3$ and the relative angular velocity $\mathbf{W} \in \mathbb{R}^3$ are assumed to be constant during each sample period [14]–[16]. However, only the assumption for $\mathbf{W} \in \mathbb{R}^3$ is required in (18). Therefore, (18) is expected to be more robust against noise (see Section IV-B).

Suppose that EKF with the process model (19) and the measurement model (9) and EKF with the process model (18) and the measurement model (9) are denoted as *Traditional* EKF and *Proposed* EKF, respectively. Then, the objectives of this paper are as follows:

1) estimating the relative pose $(\mathbf{T}, \Theta)$ between the body coordinate system and the world coordinate system by using the *Proposed* EKF method;
2) performing observability analysis on both *Traditional* EKF and *Proposed* EKF;
3) comparing the performance of the *Traditional* EKF method with that of the *Proposed* EKF method.

Algorithm 1: Point Correspondence Determination and Pose Estimation.

```
Step 1: Use the brute-force matching method in [11] to find initial point correspondences and $\hat{\mathbf{x}}_{0,0}$, and let $k = 1$;
Step 2: Calculate $\hat{\mathbf{x}}_{k,k-1}$ using (20)–(22);
Step 3: With $\hat{\mathbf{x}}_{k,k-1}$, obtain the feature points $\mathbf{P}_j^c$ $(j = 1, \ldots, n_F)$ according to (8);
Step 4: Project $\mathbf{P}_j^c$ $(j = 1, \ldots, n_F)$ into the camera image and match each prediction with its closest detection if they are closer than a threshold $\lambda_r$ (e.g., 5 pixels);
Step 5: With the correspondences found in Step 4, calculate $\hat{\mathbf{x}}_{k,k}$ using (23)–(26). Then, go back to Step 2 with
    $k = k + 1$.
```

Note that this paper focuses on using (18) and (9) in the basic EKF framework. To further improve filter performances, several techniques could be adopted, such as the adaptive EKF (AEKF) [22] and iterative adaptive EKF [15].

III. EKF and Observability Analysis

Using the nonlinear constant-velocity process model (18) and measurement model (9) presented in Section II, a correspondence-based EKF method is proposed in this section. Since point correspondence is required in the implementation of the proposed EKF method, a point correspondence method combining EKF and the algorithm in [11] is also proposed. This point correspondence method still works when severe marker missing happens. Besides, observability analysis is performed on both *Traditional* EKF and *Proposed* EKF. This theoretically explains why the proposed nonlinear constant-velocity process model (18) is better than the existing linear constant-velocity process model (19).

A. Extended Kalman Filter

Using the nonlinear constant-velocity process model (18) and measurement model (9), the well-known EKF consisting of prediction and estimation parts is given as follows:

$$\mathbf{F}_{k-1} = \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\hat{\mathbf{x}}_{k-1,k-1}} \tag{20}$$

$$\hat{\mathbf{x}}_{k,k-1} = \mathbf{f}(\hat{\mathbf{x}}_{k-1,k-1}) \tag{21}$$

$$\mathbf{P}_{k,k-1} = \mathbf{F}_{k-1}\mathbf{P}_{k-1,k-1}\mathbf{F}_{k-1}^T + \boldsymbol{\Gamma}_{k-1}\mathbf{Q}_{k-1}\boldsymbol{\Gamma}_{k-1}^T \tag{22}$$

$$\mathbf{H}_k = \frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\hat{\mathbf{x}}_{k,k-1}} \tag{23}$$

$$\mathbf{K}_k = \mathbf{P}_{k,k-1}\mathbf{H}_k^T(\mathbf{R}_k + \mathbf{H}_k\mathbf{P}_{k,k-1}\mathbf{H}_k^T)^{-1} \tag{24}$$

$$\hat{\mathbf{x}}_{k,k} = \hat{\mathbf{x}}_{k,k-1} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{g}(\hat{\mathbf{x}}_{k,k-1})) \tag{25}$$

$$\mathbf{P}_{k,k} = \mathbf{P}_{k,k-1} - \mathbf{K}_k\mathbf{H}_k\mathbf{P}_{k,k-1}. \tag{26}$$

Here, $\mathbf{P}_{k,k-1}$ is a *priori* covariance of the estimation error, $\mathbf{P}_{k,k}$ is a *posterior* covariance of the estimation error, and $\mathbf{K}_k$ is the Kalman gain matrix at step k.

Note that the implementation of the proposed EKF method requires point correspondences to be known. The steps used to find point correspondences and estimate pose using the proposed EKF method are summarized in Algorithm 1.

Note that the proposed point correspondence method could handle any number of feature points, whereas the correspondence method in [11] fails when there are less than four feature points detected in the image. Besides, the proposed point correspondence method is suitable for conventional cameras as well as fish-eye cameras, whereas the correspondence method in [11] is only suitable for conventional cameras.

B. Observability Analysis

Consider a general nonlinear system of the form

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) \\ \mathbf{y} = \mathbf{h}(\mathbf{x}) \end{cases} \tag{27}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the state vector, and $\mathbf{y} = [y_1 \dots y_m]^T \in \mathbb{R}^m$ is the measurement vector with $y_k = h_k(\mathbf{x})$ $(k = 1, \dots, m)$. The zeroth-order Lie derivative of any scalar function is the function itself, namely, $L^0 h_k(\mathbf{x}) = h_k(\mathbf{x})$. The first-order Lie derivative of $h_k(\mathbf{x})$ with respect to $\mathbf{f}(\mathbf{x})$ is defined as $L_f^1 h_k(\mathbf{x}) = \nabla h_k(\mathbf{x}) \cdot \mathbf{f}(\mathbf{x}) = \nabla L^0 h_k(\mathbf{x}) \cdot \mathbf{f}(\mathbf{x})$, where "$\nabla$" denotes the gradient operator, and "$\cdot$" means the inner product. The second-order Lie derivative of $h_k(\mathbf{x})$ with respect to $\mathbf{f}(\mathbf{x})$ is $L_f^2 h_k(\mathbf{x}) = L_f^1(\mathbf{L}_f^1 h_k(\mathbf{x})) = \nabla L_f^1 h_k(\mathbf{x}) \cdot \mathbf{f}(\mathbf{x})$. Higher order Lie derivatives of $h_k(\mathbf{x})$ can be computed similarly. Based on the Lie derivatives, the observability matrix is defined as the matrix with rows

$$\boldsymbol{\Omega} = \left[\nabla L_f^l h_k(\mathbf{x}) | k = 1, \dots, m; l = 0, \dots, n-1\right]. \tag{28}$$

It is proved in [23] that if $\boldsymbol{\Omega}$ is full column rank, then the nonlinear system (27) is locally observable.

Next, observability analysis is performed on *Traditional* EKF and *Proposed* EKF, according to the criteria mentioned before. In practice, there may be only two feature points observed by the camera because of occlusion or camera FOV. In this situation, for the nonlinear system consisting of (19) and (9), $\mathbf{f}(\mathbf{x}) = [V_x \;\; 0 \;\; V_y \;\; 0 \;\; V_z \;\; 0 \;\; w_1 \;\; 0 \;\; w_2 \;\; 0 \;\; w_3 \;\; 0]^T$

$$\nabla \mathbf{h}(\mathbf{x}) = \begin{bmatrix} \times & 0 & \times & 0 & \times & 0 & \times & 0 & \times & 0 & \times & 0 \\ \times & 0 & \times & 0 & \times & 0 & \times & 0 & \times & 0 & \times & 0 \\ \times & 0 & \times & 0 & \times & 0 & \times & 0 & \times & 0 & \times & 0 \\ \times & 0 & \times & 0 & \times & 0 & \times & 0 & \times & 0 & \times & 0 \end{bmatrix} \tag{29}$$

where "×" generally means a nonzero element. For *Traditional* EKF, the observability matrix $\boldsymbol{\Omega}_1$ is

$$\boldsymbol{\Omega}_1 = \begin{bmatrix} \nabla L_f^0 \mathbf{g}(\mathbf{x}) \\ \nabla L_f^1 \mathbf{g}(\mathbf{x}) \\ \mathbf{0}_{4\times12} \\ \vdots \\ \mathbf{0}_{4\times12} \end{bmatrix} \in \mathbb{R}^{48\times12} \tag{30}$$

where $\nabla L_f^i \mathbf{g}(\mathbf{x}) \in \mathbb{R}^{4\times12}$ $(i = 0, 1, \dots)$ and $\nabla L_f^j \mathbf{g}(\mathbf{x}) = \mathbf{0}_{4\times12} (j \geqslant 2)$. It is easy to find that $\text{rank}(\boldsymbol{\Omega}_1) < 12$ at each state point, so the pose parameters are unobservable using the linear constant-velocity model (19) and the measurement model (9). This is consistent with the conclusions in [14]–[16], where at least three noncollinear features are required for pose estimation. Similarly, for the nonlinear system consisting of (18) and (9), $\mathbf{f}(\mathbf{x}) = [V_x \;\; a_1 \;\; V_y \;\; a_2 \;\; V_z \;\; a_3 \;\; w_1 \;\; 0 \;\; w_2 \;\; 0 \;\; w_3 \;\; 0]^T$, where $a_1 = g\cos\phi\sin\theta\cos\varphi + g\sin\phi\sin\varphi, a_2 = g\cos\phi\sin\theta\sin\varphi - g\sin\phi\cos\varphi$, $a_3 = g\cos\phi\cos\theta - g$, $\nabla\mathbf{h}(\mathbf{x})$ is the same as in (29). For *Proposed* EKF, the observability matrix

$$\boldsymbol{\Omega}_2 = \begin{bmatrix} \nabla L_f^0 \mathbf{g}(\mathbf{x}) \\ \nabla L_f^1 \mathbf{g}(\mathbf{x}) \\ \nabla L_f^2 \mathbf{g}(\mathbf{x}) \\ \vdots \\ \nabla L_f^{11} \mathbf{g}(\mathbf{x}) \end{bmatrix} \in \mathbb{R}^{48\times12} \tag{31}$$

is formulated. It is found that $\text{rank}(\boldsymbol{\Omega}_2) = 12$ through MATLAB Symbolic Toolbox,[1] so the pose parameters are observable using the nonlinear constant-velocity model (18) and the measurement model (9). This is consistent with the experimental results in Sections IV and V. It can be concluded from above discussions that the proposed nonlinear constant-velocity process model (18) outperforms the existing linear constant-velocity process model (19).

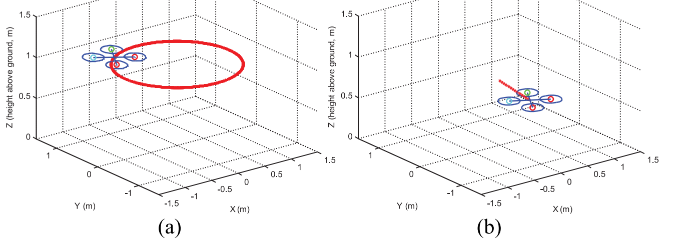
IV. Numerical Simulation Results

A. Simulation Setting

In the simulation experiments, the camera has an image resolution of 752 pixels × 480 pixels and its framerate is 40 Hz (namely, the sampling time $T_s = 0.025$ s). The focal length of the camera is 1.8 mm with a FOV of 185° and the principle point is assumed to lie at the image center. Suppose that the translation from the world coordinate system to the camera coordinate system $\mathbf{T}_w^c = [\,-0.35\ 0.65\ 3.3]^T$ m, and the rotation $\mathbf{R}_w^c$ is described in the form of Euler angles $[\,-0.175\ -0.436\ 2.182]^T$ rad $(z - y - x)$. There are four feature points fixed rigidly with a quadrotor and the 3-D coordinates of each feature point in the body coordinate system are $[0.14 {-0.215}\ -0.013]^T$ m, $[-0.21\ -0.075\ -0.004]^T$ m, $[0.04\ 0.033\ 0.033]^T$ m, and $[0.032\ 0.256\ -0.015]^T$ m. Two trajectories of the quadrotor are generated: 1) a 2-D circular trajectory with the fix height of 1 m and the diameter of 2 m (see Fig. 3(a), the period for one rotation is 12 s); 2) a 3-D curve trajectory from $(-0.455, -0.570, 1.008)^T$ m to $(0.745, 0.230, 0.368)^T$ m [see Fig. 3(b)]. The world coordinate system locates at the origin. The simulations have been performed using the Robotics and Machine Vision Toolbox for MATLAB [24]. The initial values of the pose is obtained using the P4P method [25]. The simulation time is set to 80 s.

Three filter methods for pose estimation are evaluated: 1) *Proposed* EKF [with the nonlinear constant-velocity process

[1] The MATLAB code is available at http://rfly.buaa.edu.cn/resources



(a) (b)

Fig. 3. Quadrotor trajectories generated in the simulation experiments: (a) 2-D circular trajectory; (b) 3-D curve trajectory.

model (18)]; (2) *Traditional* EKF [with the linear constant-velocity process model (19)]; (3) a loosely-coupled filtering method P4P + KF, which means that the pose estimation problem is solved using the P4P method, and the pose is refined using a Kalman filter [with the linear constant-velocity process model (19)] at the same time. The filter parameters are as follows: for *Proposed* EKF, $\mathbf{P}_{0,0} = 0.01\mathbf{I}_{12}$, $\mathbf{Q}_0 = 0.04\mathbf{I}_4$, $\mathbf{R}_0 = \sigma^2\mathbf{I}_8$, where σ is the standard deviation of Gaussian noises added to image points; for *Traditional* EKF, $\mathbf{P}_{0,0} = 0.01\mathbf{I}_{12}$, $\mathbf{Q}_0 = \text{diag}\ [0, p, 0, p, 0, p, 0, p, 0, p, 0, p]$ with $p = 0.04$, $\mathbf{R}_0 = \sigma^2\mathbf{I}_8$; for P4P + KF, $\mathbf{P}_{0,0} = 0.01\mathbf{I}_{12}$, $\mathbf{Q}_0 = \text{diag}\ [0, q, 0, q, 0, q, 0, q, 0, q, 0, q]$ with $q = 0.04$, $\mathbf{R}_0 = 0.0004\mathbf{I}_6$.

If there are m frames in each experiment, the accuracy of each pose parameter η is evaluated by the root-mean-squared (rms) error

$$D^{\eta}_{rms} = \sqrt{\frac{1}{m}\sum_{j=1}^{m}(\eta_j - \hat{\eta}_j)^2} \qquad (32)$$

where η_j denotes the ground-truth data and $\hat{\eta}_j$ is the corresponding data obtained by using the above different filtering methods. Another criterion is to use the rms reprojection error

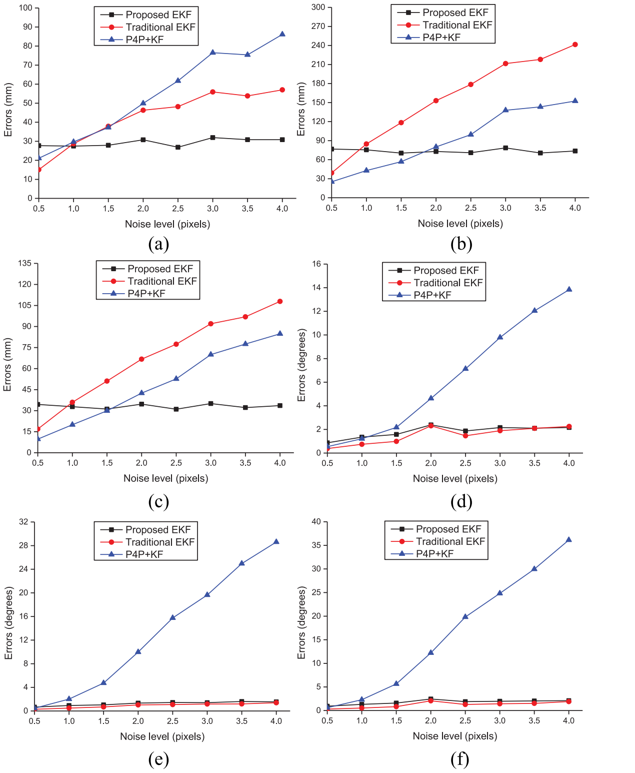
$$E_{rms} = \sqrt{\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\|u_{ij} - \hat{u}_{ij}\|^2} \qquad (33)$$

where u_{ij} denotes the image point of the jth 3-D point in the ith frame and $\hat{u}_{ij}$ is the corresponding reprojection point obtained by using the pose estimation results.

B. Noise Simulations

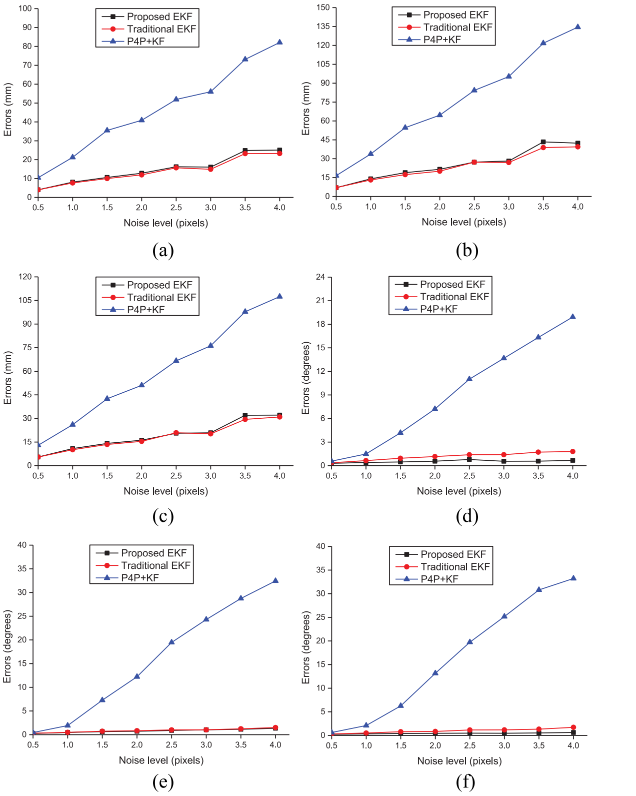
Gaussian noises with the mean value $\mu = 0$ and the standard deviation σ varying from 0.5 to 4 pixels are added to the image points. Fig. 4 shows the estimation errors of the pose parameters for the 2-D circular trajectory, whereas Fig. 5 shows the estimation errors of the pose parameters for the 3-D curve trajectory. The rms reprojection errors for the 2-D circular trajectory and the 3-D curve trajectory are shown in Fig. 6.

As shown in Figs. 4–6, all the evaluated methods offer good accuracy under different levels of noises except the P4P + KF method. One possible reason is that the loosely coupling structure of the P4P + KF method would result in error propagation. The *Proposed* EKF method gives better results than the *Traditional* EKF method in Fig. 4(a)–(c), while they give quite



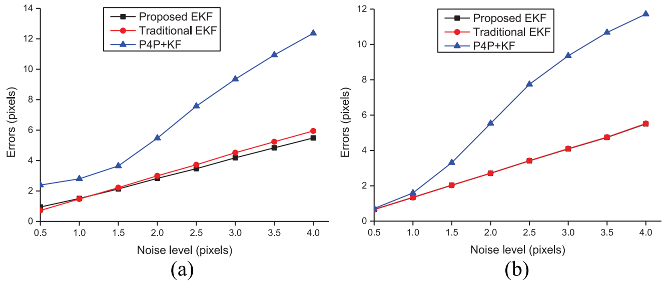
(a) (b) (c) (d) (e) (f)

Fig. 4. Pose estimation errors for the 2-D circular trajectory: (a) D^X_{rms} (mm); (b) D^Y_{rms} (mm); (c) D^Z_{rms} (mm); (d) D^{θ}_{rms} (°); (e) D^{φ}_{rms} (°); (f) D^{ϕ}_{rms} (°).



(a) (b) (c) (d) (e) (f)

Fig. 5. Pose estimation errors for the 3-D curve trajectory: (a) D^X_{rms} (mm); (b) D^Y_{rms} (mm); (c) D^Z_{rms} (mm); (d) D^{θ}_{rms} (°); (e) D^{φ}_{rms} (°); (f) D^{ϕ}_{rms} (°).



(a) (b)

Fig. 6. RMS reprojection errors: (a) E_{rms} for the 2-D circular trajectory (pixels); (b) E_{rms} for the 3-D curve trajectory (pixels).

TABLE I
NUMBER OF FEATURE POINTS OBSERVED BY THE CAMERA (T_1 = 25, T_2 = 50)

Method	Time		
	$0\,(\mathrm{s}){-}T_1\,(\mathrm{s})$	$T_1\,(\mathrm{s}){-}T_2\,(\mathrm{s})$	$T_2\,(\mathrm{s}){-}80\,(\mathrm{s})$
Proposed EKF	4	3	2
Traditional EKF	4	3	2
P4P + KF	4	4	4

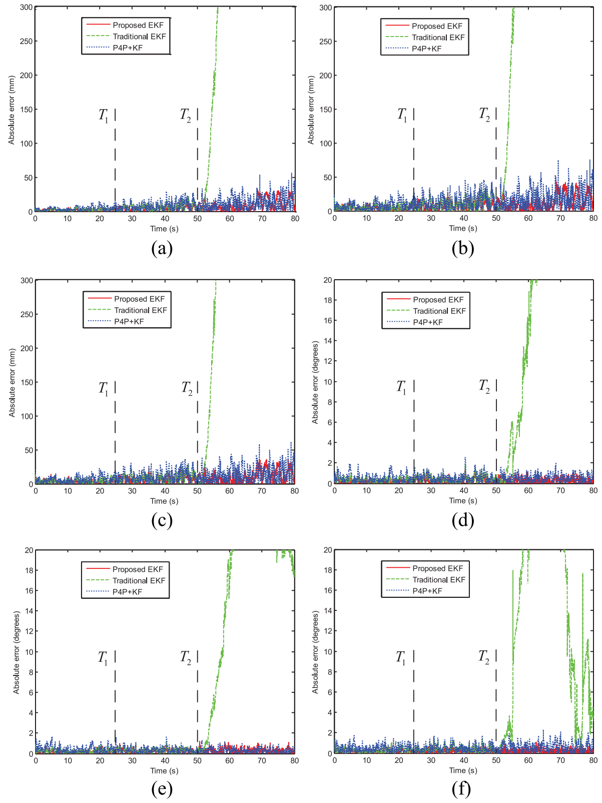
similar results in the rest of the comparisons. This is probably because the assumption that the relative linear velocity is constant during each sample period is not valid for the 2-D circular trajectory (the direction of the velocity is constantly changing). Thus, the linear constant-velocity process model (19) is inferior to the nonlinear constant-velocity process model (18) for the 2-D circular trajectory. However, the two models have almost the same performance for the 3-D curve trajectory.

C. Occlusion Simulations

The performance of the three filtering methods will be compared when the number of feature points observed by the camera n is varying (see Table I). Gaussian noise with the mean value $\mu = 0$ and the standard deviation $\sigma = 0.5$ pixel is added to the image points. Taking the 3-D curve trajectory for example, the absolute errors of pose estimation are shown in Fig. 7. It is known from Fig. 7 that the *Proposed* EKF method offers comparable accuracy with the *Traditional* EKF method when $3 \le n \le 4$ (the duration is 50 s). When $n = 2$ (the duration is 30 s), the *Traditional* EKF method fails to produce good results, whereas the *Proposed* EKF method still works. The accuracy of the *Proposed* EKF method when n is varying is also investigated. As shown in Tables II and III (std refers to standard deviation), the errors of position and orientation estimation increase as the number of feature points n decreases. The P4P + KF method requires that $n \ge 4$.

V. Real-Time Experimental Results

To show the robustness and effectiveness of our method, closed-loop control experiments of the quadrotor [AR.Drone 2.0, see Fig. 2(b)] with pose estimates using the *Proposed* EKF method at 40 Hz are performed. A video of this work is available at https://www.youtube.com/watch?v=wqCWSlqlATE or http://rfly.buaa.edu.cn/.



(a) (b) (c) (d) (e) (f)

Fig. 7. Absolute errors of pose estimation for the 3-D curve trajectory: (a) X (mm); (b) Y (mm); (c) Z (mm); (d) θ (°); (e) φ (°); and (f) ϕ (°).

TABLE II
POSITION ERRORS FOR THE PROPOSED METHOD WHEN THE NUMBER OF FEATURE POINTS IS VARYING

n	X (mm)		Y (mm)		Z (mm)	
	mean	std	mean	std	mean	std
4	1.99	1.36	4.36	3.06	3.38	2.47
3	4.42	3.15	7.90	5.41	6.34	4.42
2	9.57	7.93	13.08	10.52	10.78	8.58

TABLE III
ORIENTATION ERRORS FOR THE PROPOSED METHOD WHEN THE NUMBER OF FEATURE POINTS IS VARYING

n	$\theta(^\circ)$		$\varphi(^\circ)$		$\phi(^\circ)$	
	mean	std	mean	std	mean	std
4	0.20	0.15	0.18	0.14	0.19	0.16
3	0.29	0.23	0.24	0.17	0.20	0.15
2	0.30	0.22	0.31	0.23	0.22	0.17

In the first experiment, two MUC36M (MGYYO) cameras[2] are used (each camera has an infrared-pass filter and an image resolution of 752 pixels × 480 pixels). One is equipped with

[2] http://en.catchbest.com/index.asp

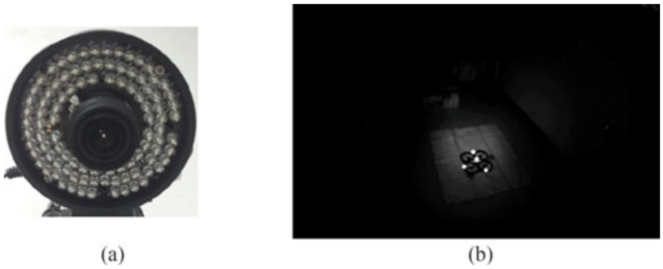

(a) (b)

Fig. 8. (a) Infrared conventional camera and active LEDs; (b) sample image of the infrared conventional camera (exposure time is 34 ms).

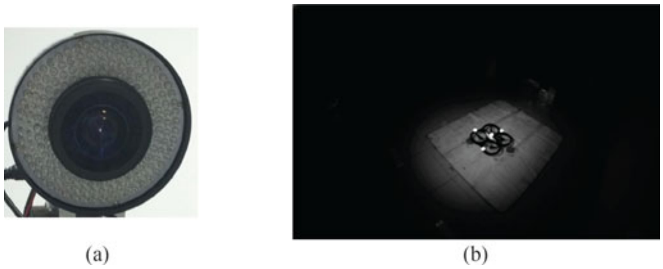
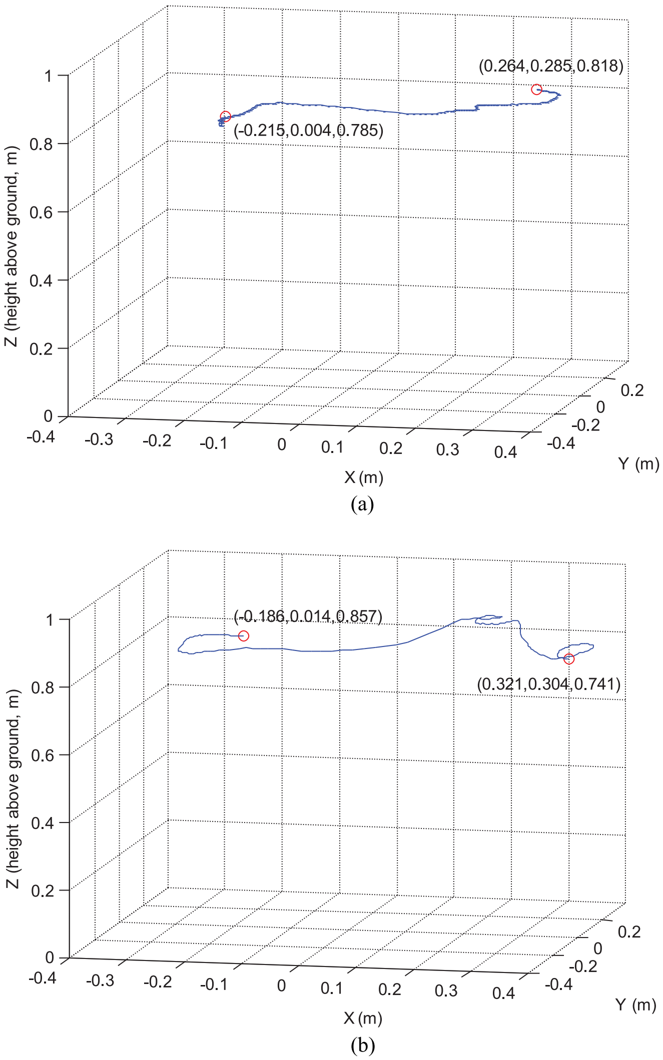

(a) (b)

Fig. 9. (a) Infrared fish-eye camera and active LEDs; (b) sample image of the infrared fish-eye camera (exposure time is 51 ms).

a conventional lense (13FM22IR) having a FOV of 118.6° and the other is equipped with a fish-eye lense (Fujinon FE185C057HA-1) having a FOV of 185°. The camera hardwares and sample images of each camera are shown in Figs. 8 and 9. The markers fixed with the quadrotor could be easily detected in the camera image because: (a) the light reflected by the markers lies in the infrared spectrum; (b) the outside light could be minimized by setting the exposure time of the camera to a small value. After camera calibration [20], [21], the positions of the markers in the reference frame of the quadrotor can be estimated as in [11]. Furthermore, the corresponding relations between a marker and its image point are obtained by using the proposed correspondence method. The quadrotor is expected to hover at the waypoint $(0.3, 0.3, 1)$ m from the starting point $(-0.2, 0, 1)$ m all along with only two markers detected. The 3-D trajectories of the quadrotor recorded by the *Proposed* EKF method are shown in Fig. 10 and the corresponding distance between the camera and the quadrotor is given in Fig. 11. It can be concluded from Fig. 10 that the quadrotor can fly to and hover at the desired waypoint all along with only two markers detected by the conventional camera or the fish-eye camera. As shown in Fig. 11, the durations of having only two observable markers are about 7 s and 8.5 s for the conventional camera and the fish-eye camera, respectively. Note that the distance for the fish-eye camera is smaller than that for the conventional camera because the active LEDs are not customized for the fish-eye camera.

In the second experiment, a conventional camera (i.e., the fish-eye lense in Fig. 9 is replaced by a conventional lense (AZURE-0420MM) having a FOV of 77.32°) is used. The quadrotor is expected to track a line from the point $(-0.55, -0.27, 1)$ m to the point $(0.32, 0.64, 1)$ m all along with only two markers detected. As the height is fixed to be 1 m, only the $X - Y$ trajectory



(a)

(b)

Fig. 10. Quadrotor is controlled to hover at the waypoint $(0.3, 0.3, 1)$ m from the starting point $(-0.2, 0, 1)$ m all along with only two markers detected: (a) 3-D trajectory of the quadrotor when a conventional camera is used; (b) 3-D trajectory of the quadrotor when a fish-eye camera is used.

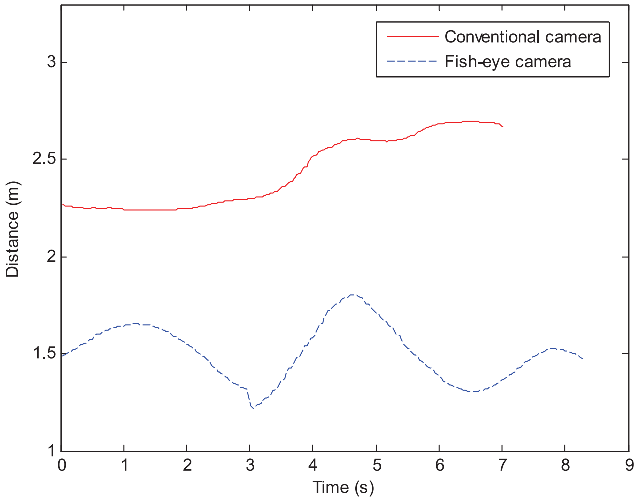


Fig. 11. Distance between the camera and the quadrotor in the hovering experiment (all along with only two markers detected).

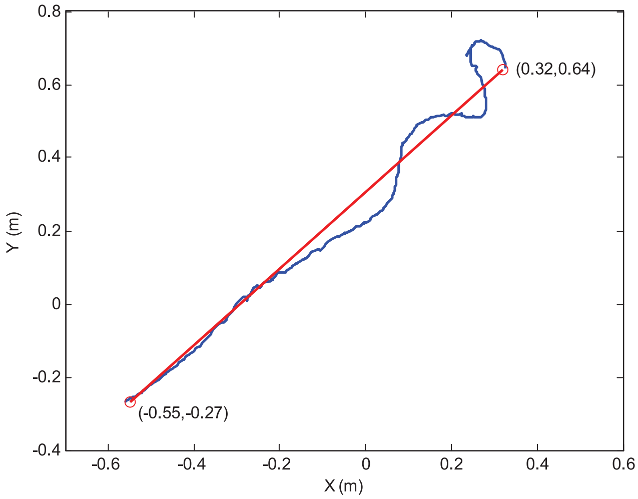


Fig. 12. *X–Y* trajectory (fixed height is 1 m) of the quadrotor when a conventional camera is used. The quadrotor is controlled to track a line from the point $(-0.55, -0.27, 1)$ m to the point $(0.32, 0.64, 1)$ m all along with only two markers detected.

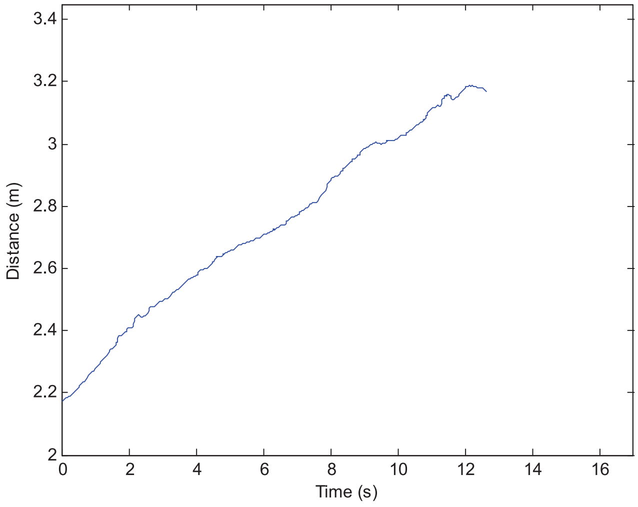


Fig. 13. Distance between the camera and the quadrotor in the line-tracking experiment (all along with only two markers detected).

of the quadrotor recorded by the *Proposed* EKF method is shown in Fig. 12 and the corresponding distance between the camera and the quadrotor is given in Fig. 13. It is known from Figs. 12 and 13 that the quadrotor can achieve line-tracking all along with only two markers detected by the conventional camera and the duration is about 12.5 s. Furthermore, the tracking result deteriorates when the quadrotor approaches the end-point. This is probably because the accuracy of pose estimation decreases as the distance between the camera and the quadrotor increases [11]. In summary, the experiments above demonstrate the practicability of the *Proposed* EKF method when the number of feature points detected is two.

VI. Conclusion

A more appropriate process model featured with the characteristics of multirotor UAVs was proposed in this paper to achieve accurate and robust pose estimation for multirotor UAVs based on off-board monocular vision. Based on the proposed nonlinear constant-velocity process model, a correspondence-based EKF method together with a general point correspondence technique handling any number of feature points (see Algorithm 1) was proposed. Observability analysis shows that the *Proposed* EKF method outperforms the *Traditional* EKF method. Simulations and real experiments have demonstrated that the *Proposed* EKF method is more robust against noise and occlusion than the *Traditional* EKF method. The *Proposed* EKF method is suitable for conventional cameras as well as fish-eye cameras. It will be promising to use this method in the single-camera motion capture system or ground-air cooperation system.

Appendix

The concrete form of (18) is described as follows:

$$\begin{aligned}
X_k &= X_{k-1} + T_s V_{x,k-1} \\
V_{x,k} &= V_{x,k-1} + \left(T_s \mathrm{c}\phi_{k-1} \mathrm{s}\theta_{k-1} \mathrm{c}\varphi_{k-1} + T_s \mathrm{s}\phi_{k-1} \mathrm{s}\varphi_{k-1}\right) \bar{g} \\
Y_k &= Y_{k-1} + T_s V_{y,k-1} \\
V_{y,k} &= V_{y,k-1} + \left(T_s \mathrm{c}\phi_{k-1} \mathrm{s}\theta_{k-1} \mathrm{s}\varphi_{k-1} - T_s \mathrm{s}\phi_{k-1} \mathrm{c}\varphi_{k-1}\right) \bar{g} \\
Z_k &= Z_{k-1} + T_s V_{z,k-1} \\
V_{z,k} &= V_{z,k-1} - T_s g + T_s \mathrm{c}\phi_{k-1} \mathrm{c}\theta_{k-1} \bar{g} \\
\theta_k &= \theta_{k-1} + T_s w_{1,k-1} \\
w_{1,k} &= w_{1,k-1} + T_s \varepsilon_{2,k-1} \\
\varphi_k &= \varphi_{k-1} + T_s w_{2,k-1} \\
w_{2,k} &= w_{2,k-1} + T_s \varepsilon_{3,k-1} \\
\phi_k &= \phi_{k-1} + T_s w_{3,k-1} \\
w_{3,k} &= w_{3,k-1} + T_s \varepsilon_{4,k-1}
\end{aligned}$$

where $\bar{g} = g + \varepsilon_{1,k-1}$, and c,s are abbreviations for cosine and sine, respectively. Thus, we have

$$\boldsymbol{\Gamma}\left(\mathbf{x_{k-1}}\right) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \Gamma_{21} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \Gamma_{41} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \Gamma_{61} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & T_s & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & T_s & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & T_s \end{bmatrix}$$

where $\Gamma_{21} = T_s \mathrm{c}\phi_{k-1} \mathrm{s}\theta_{k-1} \mathrm{c}\varphi_{k-1} + T_s \mathrm{s}\phi_{k-1} \mathrm{s}\varphi_{k-1}$, $\Gamma_{41} = T_s \mathrm{c}\phi_{k-1} \mathrm{s}\theta_{k-1} \mathrm{s}\varphi_{k-1} - T_s \mathrm{s}\phi_{k-1} \mathrm{c}\varphi_{k-1}$, and $\Gamma_{61} = T_s \mathrm{c}\phi_{k-1} \mathrm{c}\theta_{k-1}$.

References

[1] Amazon Prime Air, 2016. [Online]. Available: http://www.amazon.com/b?node=8037720011

[2] PRENAV, 2016. [Online]. Available: http://www.prenav.com/#intro2

[3] V. Cichella *et al.*, "Cooperative path following of multiple multirotors over time-varying networks," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 3, pp. 945–957, Jul. 2015.

[4] Vicon Motion Capture Systems, 2016. [Online]. Available: https://www.vicon.com/

[5] OptiTrack Motion Capture Systems, 2016. [Online]. Available: http://optitrack.com/

[6] S. Abraham and W. Főrstner, "Fish-eye-stereo calibration and epipolar rectification," *ISPRS J. Photogramm. Remote Sens.*, vol. 59, no. 5, pp. 278–288, Aug. 2005.

[7] M. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.

[8] S. Li, C. Xu, and M. Xie, "A robust $O(n)$ solution to the perspective-n-point problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1444–1450, Jul. 2012.

[9] A. Ansar and K. Daniilidis, "Linear pose estimation from points or lines," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 5, pp. 578–589, May 2003.

[10] C. Martínez, P. Campoy, I. Mondragón, and M. A. Olivares-Méndez, "Trinocular ground system to control UAVs," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2009, pp. 3361–3367.

[11] M. Faessler, E. Mueggler, K. Schwabe, and D. Scaramuzza, "A monocular pose estimation system based on infrared LEDs," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2014, pp. 907–913.

[12] C. P. Lu, G. D. Hager, and E. Mjolsness, "Fast and globally convergent pose estimation from video images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 6, pp. 610–622, Jun. 2000.

[13] A. Assa and F. Janabi-Sharifi, "Virtual visual servoing for multicamera pose estimation," *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 2, pp. 789–798, Apr. 2015.

[14] W. J. Wilson, C. W. Hulls, and G. S. Bell, "Relative end-effector control using cartesian position based visual servoing," *IEEE Trans. Robot. Autom.*, vol. 12, no. 5, pp. 684–696, Oct. 1996.

[15] F. Janabi-Sharifi and M. Marey, "A Kalman-filter-based method for pose estimation in visual servoing," *IEEE Trans. Robot.*, vol. 26, no. 5, pp. 939–947, Oct. 2010.

[16] A. Assa and F. Janabi-Sharifi, "A robust vision-based sensor fusion approach for real-time pose estimation," *IEEE Trans. Cybern.*, vol. 44, no. 2, pp. 217–227, Feb. 2014.

[17] H. D. Taghirad, S. F. Atashzar, and M. Shahbazi, "Robust solution to three-dimensional pose estimation using composite extended Kalman observer and Kalman filter," *IET Comput. Vis.*, vol. 6, no. 2, pp. 140–152, Mar. 2012.

[18] S. Y. Chen, "Kalman filter for robot vision: A survey," *IEEE Trans. Ind. Electron.*, vol. 59, no. 11, pp. 4409–4420, Nov. 2012.

[19] R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor," *IEEE Robot. Autom. Mag.*, vol. 19, no. 3, pp. 20–32, Sep. 2012.

[20] J. Kannala and S. S. Brandt, "A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 8, pp. 1335–1340, Aug. 2006.

[21] Q. Fu, Q. Quan, and K.-Y. Cai, "Calibration of multiple fish-eye cameras using a wand," *IET Comput. Vis.*, vol. 9, no. 3, pp. 378–389, Jun. 2015.

[22] M. Ficocelli and F. Janabi-Sharifi, "Adaptive filtering for pose estimation in visual servoing," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2001, pp. 19–24.

[23] M. Anguelova, "Observability and identifiability of nonlinear systems with applications in biology," Ph.D. dissertation, Dept. Math. Sci., Chalmers Univ. Technol., Göteborg, Sweden, 2007.

[24] P. Corke, *Robotics, Vision & Control: Fundamental Algorithms in MATLAB*. Berlin, Germany: Springer-Verlag, 2011.

[25] L. Kneip, D. Scaramuzza, and R. Siegwart, "A novel parameterization of the perspective-three-point problem for a direct computation of absolute camera position and orientation," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 2969–2976.

Qiang Fu received the B.S. degree in thermal energy and power engineering from Beijing Jiaotong University, Beijing, China, in 2009, and the Ph.D. degree in control science and engineering from Beihang University (formerly Beijing University of Aeronautics and Astronautics), Beijing, China, in 2016.

He is currently a Postdoctoral Fellow in the School of Instrumentation Science and Optoelectronics Engineering, Beihang University. His main research interests include vision-based navigation and 3-D vision.

Quan Quan received the B.S. and Ph.D. degrees in control science and engineering from Beihang University (formerly Beijing University of Aeronautics and Astronautics), Beijing, China, in 2004 and 2010, respectively.

He has been an Associate Professor at Beihang University since 2013. His main research interests include vision-based navigation and reliable flight control.

Kai-Yuan Cai received the B.S., M.S., and Ph.D. degrees in control science and engineering from Beihang University (formerly Beijing University of Aeronautics and Astronautics), Beijing, China, in 1984, 1987, and 1991, respectively.

He has been a Full Professor at Beihang University since 1995. He is currently a Cheung Kong Scholar (Chair Professor), jointly appointed by the Ministry of Education of China and the Li Ka Shing Foundation of Hong Kong in 1999. His main research interests include software testing, software reliability, reliable flight control, and software cybernetics.