Indoor Multi-Camera-Based Testbed for 3-D Tracking and Control of UAVs

Heng Deng, Qiang Fu, Quan Quan, Kun Yang, and Kai-Yuan Cai

***Abstract*— Flight testbeds with multiple unmanned aerial vehicles (UAVs) are especially important to support research on multi-vehicle-related algorithms. The existing platforms usually lack a generic and complete solution allowing for software and hardware design. For such a purpose, this paper presents the design and implementation of a comprehensive multi-camera-based testbed for 3-D tracking and control of UAVs. First, the testbed software consists of a multi-camera system and a ground control system, which performs image processing, camera calibration, 3-D reconstruction, pose estimation, and motion control. In the multi-camera system, the positions and orientations of UAVs are first reconstructed by using epipolar geometric constraints and triangulation methods and then filtered by an extended Kalman filter (EKF). In the ground control system, a classical proportional–derivative (PD) controller is designed to receive the navigation data from the multi-camera system and then generates control commands to the target vehicles. Then, the testbed hardware employs smart cameras with field-programmable gate array (FPGA) modules to allow for real-time computation at a frame rate of 100 Hz. Lightweight quadcopter Parrot Bebop drones are chosen as the target UAVs, which does not require any modification to the hardware. Artificial infrared reflective markers are asymmetrically mounted on target vehicles and observed by multiple infrared cameras located around the flight region. Finally, extensive experiments are performed to demonstrate that the proposed testbed is a comprehensive and complete platform with good scalability applicable for research on a variety of advanced guidance, navigation, and control algorithms.**

***Index Terms*— 3-D tracking, extended Kalman filter (EKF), indoor flight testbed, multi-camera system, unmanned aerial vehicles (UAVs), visual feedback.**

I. Introduction

In recent years, there has been growing attention in unmanned aerial vehicles (UAVs) in both academic research and industrial applications covering a range of mission scenarios, such as search and rescue [1], path planning [2], [3], performance assessment [4], [5], target tracking [6], [7], and formation flight [8], [9]. Besides, a great variety of guidance, navigation, and control algorithms are developed to implement autonomous and intelligent flight. However, many algorithms are only tested and validated with simple simulations or data sets, which are not as convincing as experimental verification. Therefore, the performance verification of algorithms through real flight experimentation plays an important role.

Most autonomous flight tests are performed in an outdoor environment with the aid of some reliable navigation systems, such as the global positioning systems (GPSs) or integrated navigation systems fusing GPS signal with inertial information [10], [11]. Nevertheless, these outdoor flight tests require wide flight area and reliable signal transmission and must be tolerant of the unpredictable weather condition. Besides, the safety of the UAV and people around is not guaranteed in outdoors since the GPS signals may be unavailable or sufficiently reliable in some confined areas, such as buildings and forests. Even if reliable GPS signals were available, it would be combined with a precise map of the surrounding areas to perform safe and stable trajectories. Accordingly, an indoor flight testbed using vision method is emerging as a promising solution recently since the vision-based method can gather rich information and span a wide field of view. Besides, the indoor testbed ensures the safety of flight test by adding protective measures, and the indoor environment can be easily controlled according to specific requirements [12].

For such a purpose, much progress has been made to verify advanced approaches and algorithms in the development of indoor flight testbeds for UAVs. A comprehensive survey on robotics testbeds is presented in [13]. To be specific, the MIT indoor multi-vehicle flight testbed [14] is developed to study long-duration UAV health management issues, such as fault detection, isolation, and recovery in a controlled environment. The testbed measures the position and orientation of markers installed on the UAV by a Vicon motion capture system, and it is verified that six-camera configuration can accurately track at least four UAVs and multiple ground vehicles in a $5 \times 5 \times 2$ m flight volume. Although it has a high estimation accuracy and can handle multiple vehicles, it requires expensive equipment, including a high-resolution motion capture system with multiple smart cameras. A localization system has been proposed for an indoor rotary-wing micro aerial vehicle (MAV) using three onboard blade LEDs and a base station-mounted active vision unit in [15]. The base station tracks the ellipse formed by a pair of cyan LEDs for five-degree of freedom (DoF)

Manuscript received November 9, 2018; revised June 26, 2019; accepted July 2, 2019. Date of publication July 26, 2019; date of current version May 12, 2020. This work was supported by the National Key Project of Research and Development Plan under Grant 2016YFC1402500. The Associate Editor coordinating the review process was Jochen Lang. *(Corresponding author: Quan Quan.)*

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

Q. Fu is with the School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing 100083, China (e-mail: fuqiang@ustb.edu.cn).

Q. Quan is with the School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China, and also with the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing 100191, China (e-mail: qq_buaa@buaa.edu.cn).

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

Digital Object Identifier 10.1109/TIM.2019.2928615

pose estimation with yaw estimation from the red LED by analyzing the captured images. The most significant advantage of the localization system is the unique LED configuration allowing six-DoF pose estimation by using only three LEDs and one camera. Michael *et al.* [16] have developed a General Robotics, Automation, Sensing, and Perception (GRASP) multiple-UAV testbed to support advanced research on coordinated, dynamic flight with broad applications. They aim at designing novel control methods and algorithms in the interactions among multiple UAVs. Oh *et al.* [17] have presented the control of an indoor UAV with four color markers using multi-camera visual feedback with multiple cameras. The visual feedback is employed by the development of an indoor flight testbed that uses only two low-cost cameras allowing the full six-DoF pose estimation and a classical proportional–integral–derivative (PID) controller. Tomer *et al.* [18] have developed a low-cost indoor multirotor adaptive navigation testbed with a suite of 12 small omnidrive robots tracked via a multi-camera OptiTrack system with its Motive optical motion capture software.

The testbeds mentioned earlier have some limitations that inhibit their utility. Most platforms rely heavily on the expensive commercial high-resolution motion capture systems, and it is difficult to modify the positioning systems to consider more specific applications. Moreover, most of the target vehicles are self-designed or modified, which may slow down the research process to adjust the internal structure and onboard sensing. Thus, such platforms usually lack a generic and complete solution allowing for software and hardware design. For such a purpose, this paper proposes a comprehensive indoor multi-camera-based testbed, including customized software and hardware to provide the pose estimation of the target UAVs and autonomously control the vehicles. The 3-D positions and orientations of the UAVs are estimated by using a multi-vision algorithm and an extended Kalman filter (EKF), and then, the visual feedback information is sent to the testbed to control the target vehicles. The proposed testbed is low cost and easy to use with high-precision estimation and control accuracy. Lightweight commercial quadcopter Parrot Bebop drones [19] are chosen as the target UAVs with available software development kit (SDK), which does not require special requirements or any modification to the hardware, and accurate state measurement and robust attitude control can be provided during flight operations. Therefore, extensive indoor flight tests can be repeatedly conducted in a short period. Different from the testbed of [17], the proposed testbed uses the infrared reflective markers instead of colored markers, which makes the marker detecting process more simple and robust to illumination. Besides, since the field-programmable gate array (FPGA) has become a widely used platform for smart camera implementations to perform a variety of embedded vision tasks [20]–[23] due to the superior performance in hardware acceleration, it is easier to scale up the system since the image processing is implemented on an FPGA inside individual camera to reduce the processing time and decrease the bandwidth greatly.

The contributions of this paper are as follows.

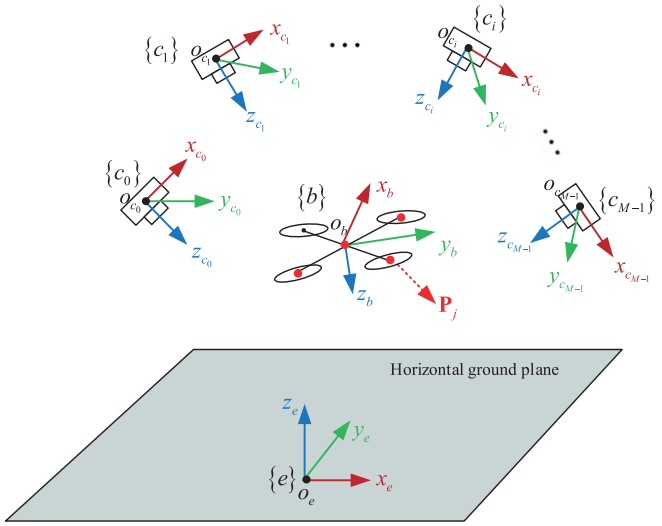


Fig. 1. Definitions of the coordinate frames in the testbed.

1) A comprehensive and generic multi-camera-based testbed for 3-D tracking and control of UAVs is proposed.
2) A complete description to the testbed software, including a multi-camera system and a ground control system, is provided, and the source code of the ground control system in a Simulink model is available at https://github.com/DengMark/BebopTest.
3) The testbed hardware solution is low cost and easy to implement.
4) The testbed provides a generic and complete solution for multi-camera based flight platforms with good scalability applicable for research on a great variety of advanced guidance, navigation, and control algorithms.

The remainder of this paper is organized as follows. Section II gives an overview of the proposed testbed architecture, components, and operation procedure, followed by the testbed software design III, including the multi-camera system design and the ground control system design. Then, the testbed hardware design is described in Section IV. Finally, Section V shows the experimental results of the proposed indoor testbed using multi-camera visual feedback, and Section VI gives the conclusions and future research work.

II. Problem Statement

A. Coordinate Frames

Note that the notations and definitions in this paper are consistent with that in [24]. For simplicity, consider a UAV attached with some infrared reflective markers flying in the view of a multi-camera system in Fig. 1, where there are three coordinate frames involved: earth-fixed coordinate frame (EFCF), aircraft-body coordinate frame (ABCF), and camera coordinate frame (CCF). The EFCF $\{e\} = \{o_e x_e y_e z_e\}$ denotes a right-hand frame with the coordinate origin o_e located on the horizontal ground plane, which is determined during the calibration process. The EFCF is employed to express UAV's states relative to the global frame. The ABCF

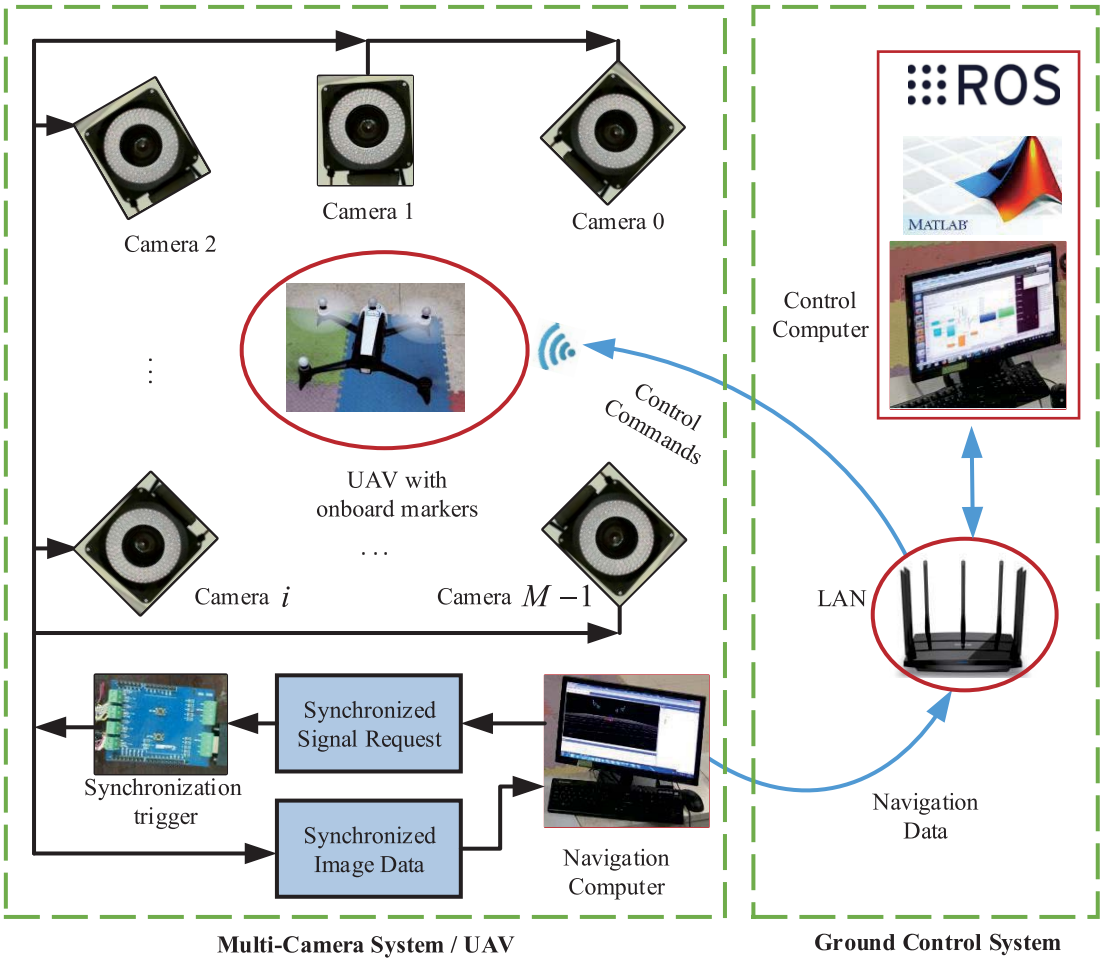


Fig. 2. Indoor multi-camera-based testbed configuration.

$\{b\} = \{o_b x_b y_b z_b\}$ is a right-hand frame fixed to the vehicle. The center of gravity (CoG) of the vehicle is chosen as the origin o_b of frame $\{b\}$. The onboard reflective markers are denoted by $\mathbf{P}_j (j = 1, \cdots, N_p)$. The CCF $\{c_i\} = \{o_{c_i} x_{c_i} y_{c_i} z_{c_i}\}, (i = 0, 1, \cdots, M-1)$ is attached to each external camera with its origin o_{c_i} located in the camera optical center and the $o_{c_i} z_{c_i}$-axis aligned with the optical axis. Within this research, the main job is first to calibrate the transformation between each CCF and EFCF, then estimate the pose between ABCF and EFCF accurately with the image information in each CCF, and finally control the UAV with ABCF fixed.

B. Testbed Architecture and Components

Since the focus of the proposed testbed is to provide an accurate and accessible platform to test a variety of vision-based algorithms in a real-time environment, the objective of this paper is to develop an accurate, efficient and robust pose estimation method for 3-D tracking of UAVs, serving as the visual feedback to enable autonomous flight. As shown in Fig. 2, the proposed testbed has four major components: a multi-camera system, a ground control system, onboard infrared reflective markers, and target UAVs. The multi-camera system providing accurate pose consists of multiple external cameras with infrared pass filters, an infrared light source to enhance light intensity, a synchronization trigger to synchronize image data from multiple cameras, and a navigation computer to run main visual algorithms. The ground control system with a control computer running robotics operation system (ROS) [25] is to receive the navigation data from the multi-camera system and then to upload the calculated control commands to the target UAV through wireless local area network (LAN). The reflective markers are asymmetrically mounted on the target UAVs and to be observed by infrared cameras with infrared-pass filters to provide reliable and easy-to-extract image features for the pose estimation. With at least four markers on the target vehicles and the corresponding observations in multiple cameras, the 3-D position and orientation can be obtained in real time.

Remark 1: The number and configuration of cameras remain as an important and complex factor in the testbed [26], [27]. In practice, a marker can be reconstructed when it is visible from at least two cameras. Besides the visibility problem, the reconstruction error depends on the convergence angle of two cameras [28]. In the proposed testbed, the cameras are just empirically placed to satisfy the visibility demands, while the optimal camera placement problem remains to be researched in the future.

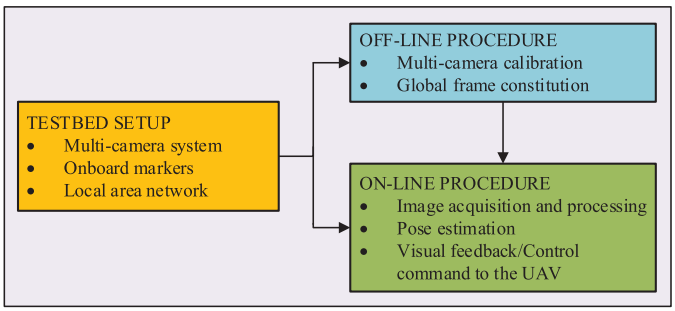


Fig. 3. Operation procedure of the indoor multi-camera testbed.

Remark 2: The configuration of the markers, namely, the placement of them on the target UAVs, is also an important factor in the testbed. The fundamental principle is that the arrangement is arbitrary, but it must be asymmetric. In addition, they should not lie in a plane to reduce the ambiguities of the pose estimation. To increase precision, the center of the markers should be identical to the CoG of the vehicles. As a result, the marker configuration in the initialization of estimation is more accurate. Besides, the markers should be visible from as more cameras as possible.

C. Operation Procedure

As shown in Fig. 3, the operation procedure of the proposed indoor testbed begins with the multi-camera setup, including the setup of all the cameras and computers. Then, the multi-camera calibration is executed to describe a mapping between the 3-D space feature and the corresponding 2-D image pixel, followed by the arrangement of the onboard markers to the target UAV. The three procedures are operated off-line before the UAV takes off, aiming to constitute a global frame coordinate, namely, the EFCF for the estimation algorithm in the following. When the flight test starts, the estimation algorithm starts working. First, all the images of the onboard markers are captured and sent to the navigation computer. Then, by processing the obtained images, the exact pixel positions of the detected markers with respect to the image frame are obtained. Since the marker configuration and initial correspondence can be estimated during the initialization process, the pose of the UAV, including 3-D position and orientation, can be estimated by using an EKF with the process model and visual measurement model. After that, the estimated pose of the UAV is broadcast and published in the wireless LAN, subscribed by the Simulink model[1] in the control computer via ROS. Finally, the control computer will deal with the visual feedback and return a control command to the UAV.

III. Testbed Software Design

This section presents the testbed software design, including the design of a multi-camera system and a ground control system. In addition, the software framework of the proposed indoor flight testbed is clarified.

A. Multi-Camera System Design

This section describes the multi-camera system design. First, the image processing is presented. Then, the main camera calibration based on a generic camera model suitable for conventional cameras and fish-eye lens cameras is explained. Finally, the pose estimation based on an EKF is proposed.

1) Image Processing: Image processing is the first and essential step in the vision algorithm. Only after image processing, the integrated vision data are employed by the camera calibration and pose estimation in the following. The goal of image processing is to deal with the input images to extract and locate the onboard markers. This process includes: 1) thresholding and smoothing the greyscale images; 2) segmenting the markers out of the background; 3) extracting the target features; and 4) matching these features. In order to simplify and speed up the image processing procedure, the design of the markers must be distinctive to make it easy to identify and segment from the background. Besides, the infrared light reflective markers are employed to reduce noise and disturbance.

Remark 3: There is an alternative to accelerate the image process in terms of hardware design. Smart cameras can be utilized instead of conventional cameras. The difference between them is that the image processing can be run on the computer unit of the smart cameras, and only the locations of feature points are sent to the navigation computer. Thus, it constitutes a decentralized multi-camera system. The significant advantage of decentralization is that less burden will be left to the navigation computer by reducing image processing time and the bandwidth of the network will decrease. In order to scale up the proposed system, i.e., increasing more external cameras and broadening the effective range, smart cameras are employed in the testbed. Thus, any number of cameras can be theoretically utilized on the condition that the total bandwidth is within the range of the LAN. Besides, in real experiments, the frame update rate is only up to 40 Hz with four conventional cameras, while for the smart cameras, the update rate can raise up to 100 Hz with eight cameras. Finally, smart cameras are utilized in the proposed testbed.

2) Camera Calibration:

a) Generic camera model: This paper considers a generic camera model suitable for conventional cameras and fish-eye lens cameras to describe a mapping between the 3-D space points and 2-D image pixels based on the work of [29]. The generic camera model is a more flexible radially symmetric projection model compared with the commonly used perspective projection model. The generic camera model is shown in Fig. 4(a). It mainly describes the nonlinear mapping from the incoming rays to the normalized image coordinates taking the radial distortion into account, which is defined as

$$\begin{bmatrix} x \\ y \end{bmatrix} = r(\theta) \begin{bmatrix} \cos\varphi \\ \sin\varphi \end{bmatrix} \tag{1}$$

where $\theta, \varphi \in \mathbb{R}$ is the direction of the incoming ray and θ is the angle between the principal axis and the incoming ray,

[1] Simulink is a block diagram environment for simulation and model-based design, which provides a graphical editor, customizable block libraries, and solvers for modeling. This paper employs the Robotics System Toolbox (RST) to establish communication between the Simulink model and the ROS-enabled UAV.

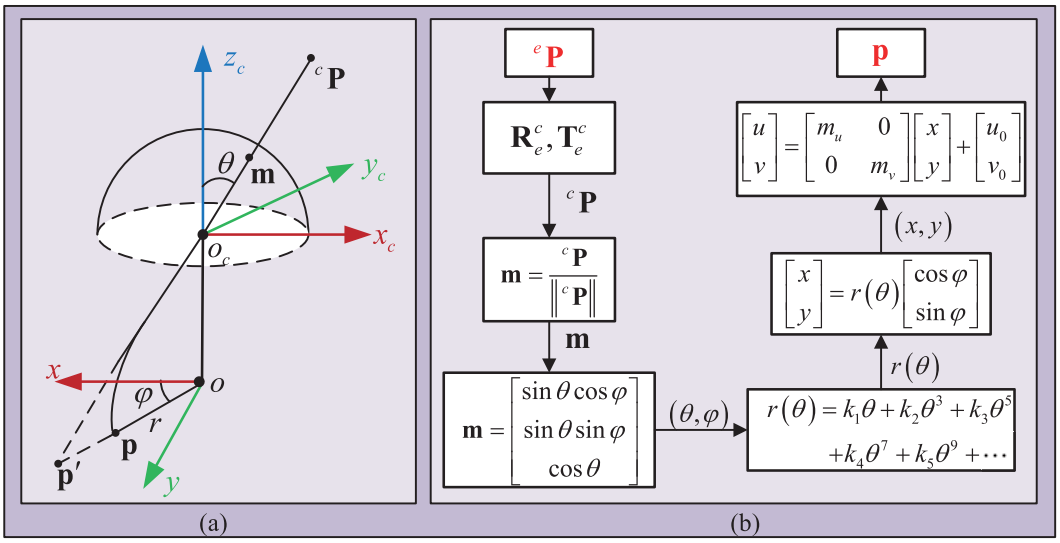


Fig. 4. (a) Generic camera model. The image of the point ${}^{c}\mathbf{P}$ is $\mathbf{p}$ by a fish-eye camera, whereas it would be $\mathbf{p}'$ by a pinhole camera. (b) Detailed projection procedure of the proposed generic camera model, taking the rotation and translation of the 3-D point ${}^{e}\mathbf{P}$ into account.

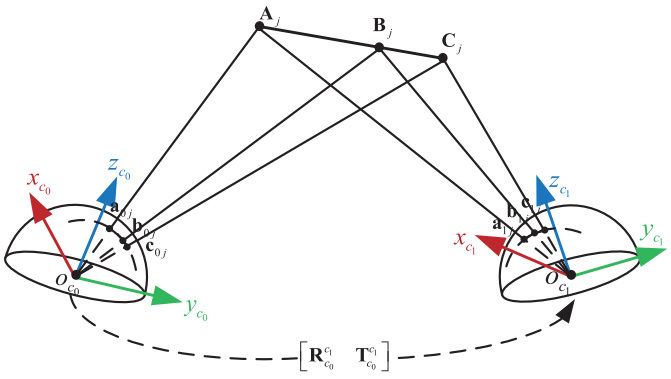


Fig. 5. Epipolar geometric constraints.

and $r \in \mathbb{R}_+$ is the distance between the image point and the principal point that has a general form as

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

where $k_1, k_2 \cdots \in \mathbb{R}$ are certain coefficients to be calibrated in which the first coefficient k_1 represents the focal length.

In summary, the basic form of the generic camera model can be written as

$$\mathbf{p} = \mathbf{G}\left(k_1, k_2, m_u, m_v, u_0, v_0, k_3, k_4, k_5, \mathbf{R}_e^c, \mathbf{T}_e^c, {}^{e}\mathbf{P}\right) \tag{3}$$

where the nonlinear function $\mathbf{G}(\cdot)$ projects the 3-D space point ${}^{e}\mathbf{P} \in \mathbb{R}^3$ with respect to the EFCF into the 2-D image point $\mathbf{p} \in \mathbb{R}^2$, the parameter $(m_u, m_v) \in \mathbb{Z}_+$ is the number of pixels per unit distance in horizontal and vertical directions, respectively, and $(u_0, v_0) \in \mathbb{R}_+$ is the principal point of the image. As for the parameters of the function $\mathbf{G}(\cdot)$, the first nine parameters $(k_1, k_2, m_u, m_v, u_0, v_0, k_3, k_4,$ and $k_5)$ are called the intrinsic parameters that describe the mapping from the point ${}^{c}\mathbf{P} \in \mathbb{R}^3$ with respect to the CCF to the image point $\mathbf{p}$, while the remaining parameters $(\mathbf{R}_e^c \in \mathbb{R}^{3\times3}, \mathbf{T}_e^c \in \mathbb{R}^3)$ are the corresponding extrinsic parameters that are the rotation and translation transformation matrices from the EFCF to the CCF. The specific detailed projection procedure of the generic camera model is shown in Fig. 4(b).

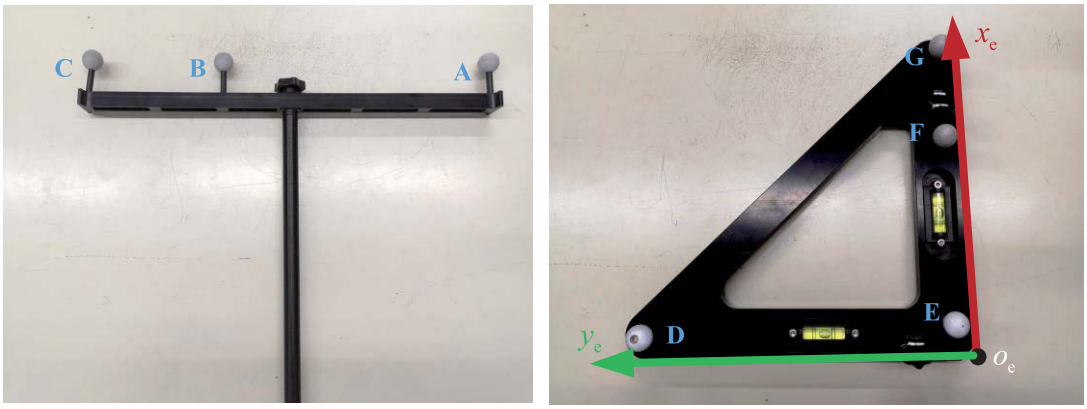
b) Epipolar geometric constraint: As shown in Fig. 5, the epipolar geometric constraint is the geometric constraint between two cameras. Suppose that 3-D features $\mathbf{A}_j, \mathbf{B}_j,$ and $\mathbf{C}_j$ are projected to the corresponding 2-D image point $\mathbf{a}_{0j}, \mathbf{b}_{0j},$ and $\mathbf{c}_{0j}$ on the unit hemisphere centered at o_{c_0} and $\mathbf{a}_{1j}, \mathbf{b}_{1j},$ and $\mathbf{c}_{1j}$ on the unit hemisphere centered at o_{c_1}. Without loss of generality, assume that a 3-D point $\mathbf{M}_j$ is projected to $\mathbf{m}_{0j} = [\sin\theta_{0j}\cos\varphi_{0j}\ \sin\theta_{0j}\sin\varphi_{0j}\ \cos\theta_{0j}]^T \in \mathbb{R}^3$, $\mathbf{m}_{1j} = [\sin\theta_{1j}\cos\varphi_{1j}\ \sin\theta_{1j}\sin\varphi_{1j}\ \cos\theta_{1j}]^T \in \mathbb{R}^3$ on the unit hemisphere centered at o_{c_0} and o_{c_1}, respectively. Thus, the epipolar geometric constraint is formulated as

$$\mathbf{m}_{1j}^T\left[\mathbf{T}_{c_0}^{c_1}\right]_\times \mathbf{R}_{c_0}^{c_1}\mathbf{m}_{0j} = 0 \tag{4}$$

where $\mathbf{R}_{c_0}^{c_1} \in \mathbb{R}^{3\times3}$ and $\mathbf{T}_{c_0}^{c_1} \in \mathbb{R}^3$ are the rotation and translation matrices from the left camera $\{c_0\}$ to the right camera $\{c_1\}$, respectively. The symbol $\times$ represents the skew-symmetry operation. For simplicity, define $\mathbf{E}_{c_0}^{c_1} = [\mathbf{T}_{c_0}^{c_1}]_\times \mathbf{R}_{c_0}^{c_1}$ as the essential matrix between the two cameras. Then, (4) can be simplified as

$$\mathbf{m}_{1j}^T\mathbf{E}_{c_0}^{c_1}\mathbf{m}_{0j} = 0. \tag{5}$$

The point correspondences ($\mathbf{M}_j, \mathbf{m}_{0j},$ and $\mathbf{m}_{1j}$) can be ($\mathbf{A}_j, \mathbf{a}_{0j},$ and $\mathbf{a}_{1j}$), ($\mathbf{B}_j, \mathbf{b}_{0j},$ and $\mathbf{b}_{1j}$) or ($\mathbf{C}_j, \mathbf{c}_{0j},$ and $\mathbf{c}_{1j}$) for instances, as shown in Fig. 5. According to (5), the epipolar geometric constraint depends only on the intrinsic parameters and relative pose of the cameras, while it does not rely at all on the scene structure. There are at least two potential applications. First, given an image point $\mathbf{p}$ in the left camera (or in the first view), the constraint can determine the position of the corresponding $\mathbf{p}'$ in the right camera (or in the second view) combined with the knowledge of the relative transformation matrix, which will give a prediction about the corresponding image point, making the tracking more accurate



(a) (b)

Fig. 6. Calibration kits. (a) Three-marker wand to calibrate the intrinsic and extrinsic parameters. (b) Four-marker triangle board to determine the EFCF.

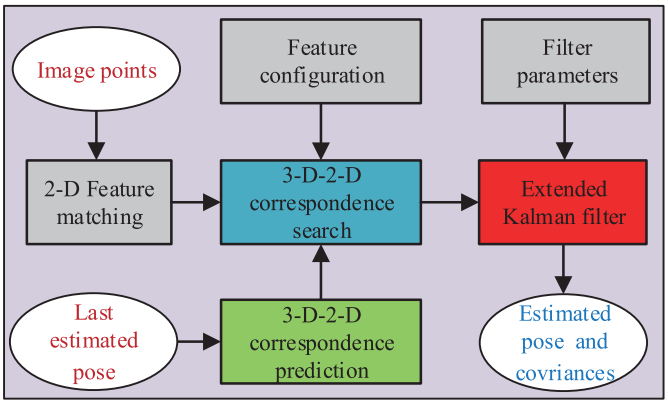


Fig. 7. Flow diagram of the pose estimation.

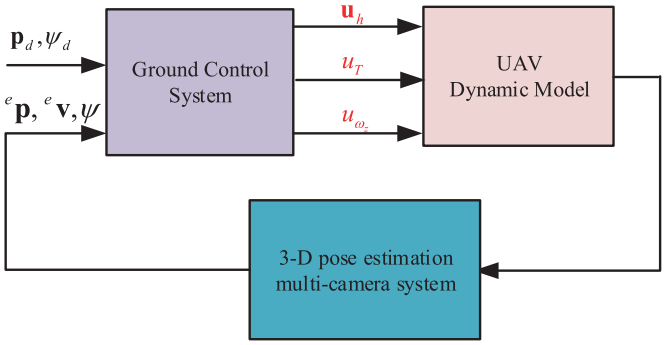


Fig. 8. Closed-loop control diagram in the proposed testbed.

and faster. Second, if the intrinsic camera parameters are known, given the point correspondences between two cameras (or two consecutive views by one camera), the transformation matrix between two cameras can be obtained up to a finite number of ambiguities. Then, the ambiguity can be solved by comparing each possible 3-D reconstruction with the true scene structure.

c) 3-D Reconstruction: Based on the epipolar geometric constraint mentioned earlier, 3-D reconstruction is to recover the location of a 3-D feature with respect to EFCF from a set of point correspondences of two cameras. More explicitly, supposing that a set of point correspondences $\mathbf{m}_{0j} \longleftrightarrow \mathbf{m}_{1j}$

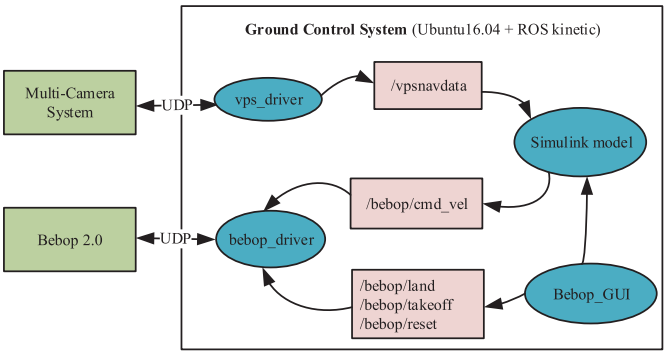


Fig. 9. Software framework of indoor flight control testbed.

with respect to the same 3-D space point $\mathbf{M}_j$ captured by two cameras, the reconstruction objective is to find the camera matrices $\mathbf{P}_0, \mathbf{P}_1$ as well as the 3-D point $\mathbf{M}_j$ such that

$$s_0\mathbf{m}_{0j} = \mathbf{P}_0\mathbf{M}_j, s_1\mathbf{m}_{1j} = \mathbf{P}_1\mathbf{M}_j \tag{6}$$

where s_0, s_1 are scaling factors and $\mathbf{P}_0 = [\mathbf{I}_3\mathbf{0}_{3\times1}] \in \mathbb{R}^{3\times4}$, $\mathbf{P}_1 = [\mathbf{R}_{c_0}^{c_1}\mathbf{T}_{c_0}^{c_1}] \in \mathbb{R}^{3\times4}$. The method for 3-D reconstruction from two views is summarized as follows.

1) Compute the essential matrix $\mathbf{E}_{c_0}^{c_1}$ from point correspondences based on the epipolar geometric constraint [see 5] using the normalized eight-point algorithm [30].
2) Recover the camera transformation matrices $\mathbf{R}_{c_0}^{c_1}$ and $\mathbf{T}_{c_0}^{c_1}$ from the essential matrix $\mathbf{E}_{c_0}^{c_1}$ based on singular value decomposition (SVD) [31] and marker configuration.
3) Based on each of point correspondences $\mathbf{m}_{0j} \longleftrightarrow \mathbf{m}_{1j}$ and camera matrices $\mathbf{P}_0, \mathbf{P}_1$, compute the position of the 3-D space point $\mathbf{M}_j$.

Remark 4: There are some inherent ambiguities involved in the 3-D reconstruction of a scene from point correspondences. The camera matrices may be retrieved by SVD up to a scale, but there are four possible solutions except for the overall scale. Thus, the 3-D point is further reconstructed based on each possible camera matrix, and the solution with the positive reconstruction coordinate is optimum. Besides, the absolute position of the space point cannot be determined without the knowledge of the scene structure. In practice, the scale can be

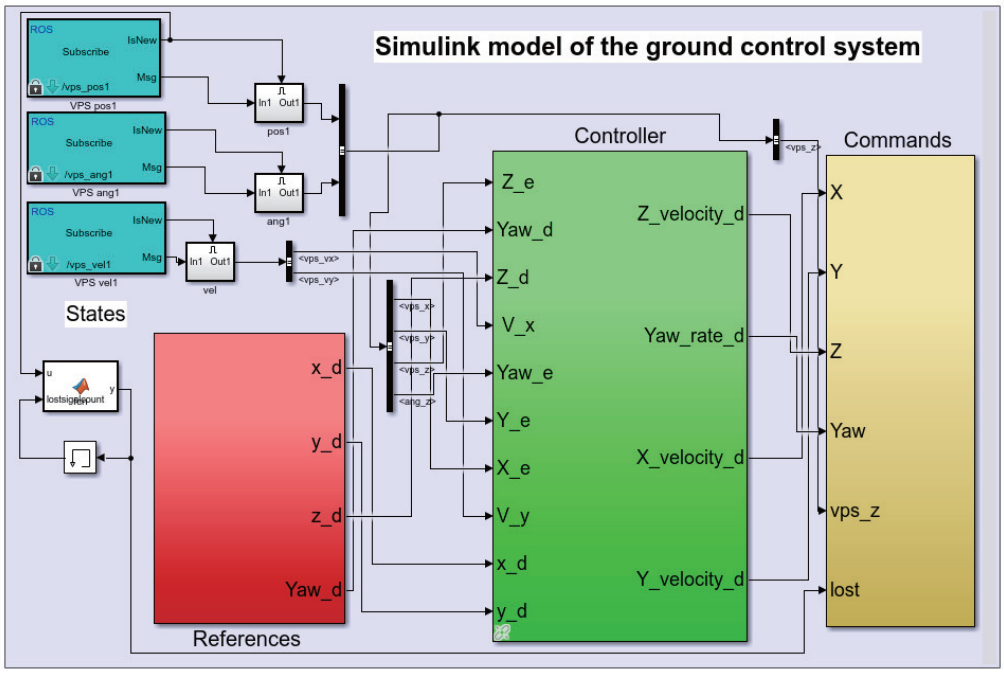


Fig. 10. Simulink model of the ground control system.

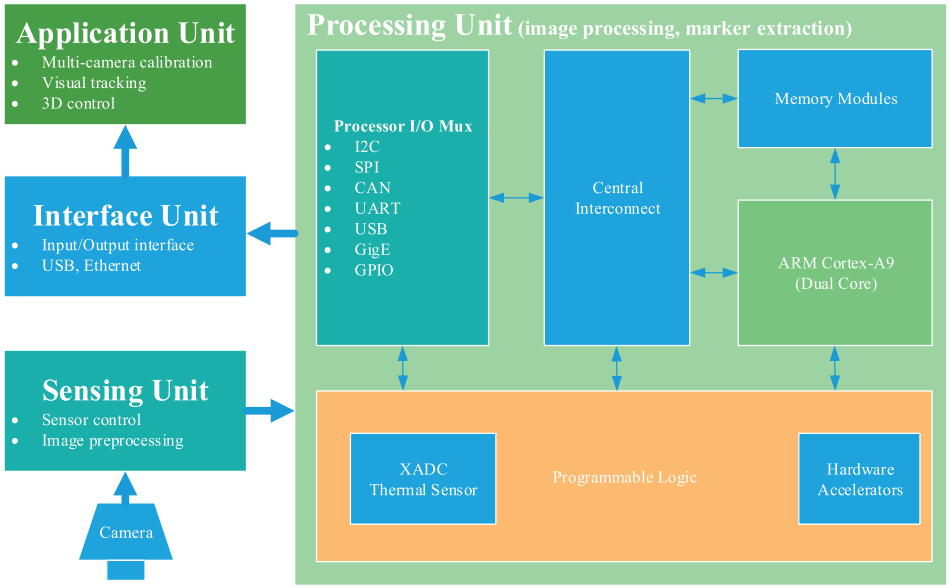
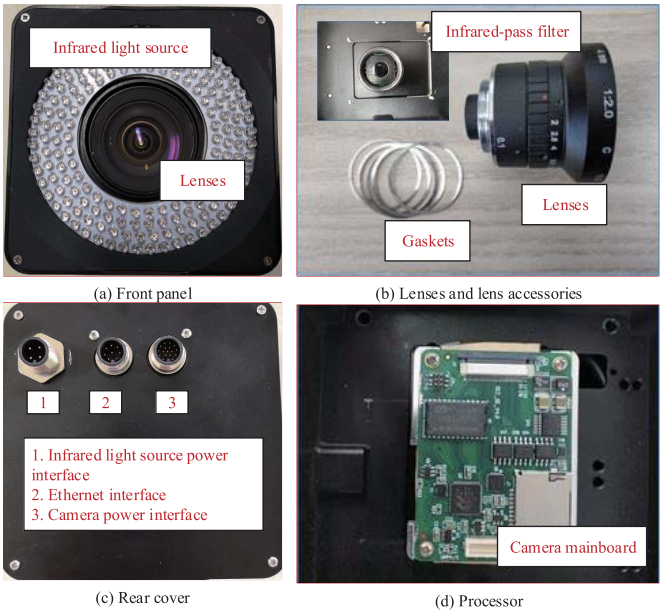


Fig. 11. Hardware architecture of the proposed testbed.

computed based on the actual length and the reconstruction length of **AC** on the 1-D calibration wand.

d) Calibration algorithm: As for a monocular camera, the calibration is to determine the nine intrinsic parameters $(k_1, k_2, m_u, m_v, u_0, v_0, k_3, k_4,$ and $k_5)$, and the calibration procedure for estimating those parameters is detailed in [29] based on viewing a planar object with several control points at known positions. In addition to the intrinsic parameters of a camera, however, the multiple cameras are placed at unchanged locations during the tracking process in our testbed. Thus, their relative poses need to be calibrated, so are their corresponding poses with respect to the global frame. These are referred to as the calibration of the extrinsic parameters of each camera $(\mathbf{R}_e^c, \mathbf{T}_e^c)$, and our previous work [32] proposed a method to calibrate the intrinsic and extrinsic parameters of multiple fish-eye cameras using a 1-D freely moving wand based on the generic camera model. In our testbed, the calibration method in [32] is employed, and then, another calibration kit is used to uniform the pose of cameras with respect to the global frame. As shown in Fig. 6(a), a 1-D wand with three identical reflective markers with length configuration that $L_{\mathbf{AC}} = 390$ mm and $L_{\mathbf{AB}} = 260$ mm are used to calibrate the intrinsic parameters and extrinsic parameters of each camera with respect to the reference camera. Similarly,



(a) Front panel (b) Lenses and lens accessories

(c) Rear cover (d) Processor

Fig. 12. Camera structure of the proposed testbed. The camera sensor is protected by a square stainless steel housing with a size of 105 × 105 × 90 mm. (a) Front panel. (b) Lenses and lens accessories. (c) Rear cover. (d) Processor.

Fig. 6(b) shows a 2-D triangle board with four markers with length configuration that $L_{\mathbf{DE}} = 240$ mm, $L_{\mathbf{EG}} = 230$ mm, and $L_{\mathbf{EF}} = 150$ mm. The 2-D calibration kit is utilized to set the origin and determine the EFCF with the three axes; besides the calibration results are the extrinsic parameters of each camera with respect to the EFCF.

As for the calibration algorithm, the main point is to transform the calibration into an optimization problem as

$$\begin{aligned}\mathbf{y}^* = \arg\min_{\mathbf{y}} \sum_{i=0}^{M}\sum_{j=1}^{N_i} &\left(L_{\boldsymbol{AB}} - \left\|\mathbf{A}_j^r - \mathbf{B}_j^r\right\|\right)^2 \\ &+\left(L_{\mathbf{AC}} - \left\|\mathbf{A}_j^r - \mathbf{C}_j^r\right\|\right)^2 + \left(L_{\boldsymbol{BC}} - \left\|\mathbf{B}_j^r - \mathbf{C}_j^r\right\|\right)^2 \\ &+\left\|\mathbf{a}_{ij} - \mathbf{G}\left(\mathbf{y}, \mathbf{A}_j^r\right)\right\|^2 + \left\|\mathbf{b}_{ij} - \mathbf{G}\left(\mathbf{y}, \mathbf{B}_j^r\right)\right\|^2 \\ &+\left\|\mathbf{c}_{ij} - \mathbf{G}\left(\mathbf{y}, \mathbf{C}_j^r\right)\right\|^2 \end{aligned} \tag{7}$$

where the optimization variable is defined as $\mathbf{y} = [k_1^i\ k_2^i\ m_u^i\ m_v^i\ u_0^i\ v_0^i\ k_3^i\ k_4^i\ k_5^i\ \mathbf{R}_{c0}^{ci}\ \mathbf{T}_{c0}^{ci}]^T, i = 1, \cdots M$, and $\mathbf{a}_{ij}, \mathbf{b}_{ij}$, and $\mathbf{c}_{ij}$ are the image points of 3-D reconstruction features $\mathbf{A}_j^r, \mathbf{B}_j^r$, and $\mathbf{C}_j^r$ in the ith camera and N_i is the total number of times that the features are viewed in the ith camera. The former three errors are the reconstruction length errors, while the last three terms are the reprojection errors of each feature point. The sparse Levenberg–Marquardt algorithm [33] is employed to minimize the errors.

Remark 5: It is noted that the intrinsic parameters remain unchanged wherever the cameras are placed. Thus, in practice, the intrinsic parameters are first calibrated alone, followed by the calibration of the remaining extrinsic parameters of (7). Thus, the computation of the optimization process is reduced.

3) Pose Estimation: Accurate pose estimation is a fundamental issue for 3-D tracking and control of UAVs. In this section, the pose is estimated by an EKF under the assumptions that the system is corrupted by zero-mean Gaussian white noises and the covariance of measurement models is known. A generic linear constant velocity process model with 12 states is employed to describe the motion characteristics of the vehicles, and the feature imaging of point correspondences constitutes the visual measurements model with noises. The general flow diagram of the pose estimation is shown in Fig. 7. In the diagram, the 3-D–2-D correspondence search is to associate the reflective markers with the image points at each camera independently, which is extremely important for the visual measurement of the filter. Thus, it is important for the initialization process to obtain a nice initial value of the 2-D feature matching and feature configuration. The 2-D feature matching is to find the point correspondences in different camera images of the same space point based on the epipolar geometric constraint, and thus, 3-D reconstruction of each feature can be executed, so is the feature configuration. The feature configuration is to calibrate the position of the markers with respect to the ABCF, which is essential for visual measurement. The filter needs to be reinitialized whenever the tracking loss of features happens. Based on the last estimated pose, the 3-D–2-D correspondences are predicted based on the process model and then search for the correspondences, constituting the visual measurement. Finally, an EKF is utilized to estimate the pose and corresponding covariance.

a) Process model: For pose estimation of rigid objects, the system state vector of process model comprises of the position ${}^e\mathbf{p} = [p_{x_e}\ p_{y_e}\ p_{z_e}]^T \in \mathbb{R}^3$, velocity ${}^e\mathbf{v} = [v_{x_e}\ v_{y_e}\ v_{z_e}]^T \in \mathbb{R}^3$, Euler angle $\boldsymbol{\Theta} = [\phi\ \theta\ \psi]^T \in \mathbb{R}^3$, and angular velocity ${}^b\boldsymbol{\omega} = [\omega_{x_b}\ \omega_{y_b}\ \omega_{z_b}]^T \in \mathbb{R}^3$ and thus is defined as

$$\mathbf{x} = [p_{x_e}\ v_{x_e}\ p_{y_e}\ v_{y_e}\ p_{z_e}\ v_{z_e}\ \theta\ \omega_{y_b}\ \psi\ \omega_{z_b}\ \phi\ \omega_{x_b}]^T \in \mathbb{R}^{12}. \tag{8}$$

Then, a generic linear constant velocity process model is chosen in which the relative linear velocity and angular velocity are assumed to be constant during each sample period. This assumption is reasonable for sufficiently small sample periods in real-time tracking systems. Thus, the process model can be written as [34], [35]

$$\begin{aligned} {}^e\dot{\mathbf{p}} &= {}^e\mathbf{v} \\ {}^e\dot{\mathbf{v}} &= \boldsymbol{\gamma}_p \\ \dot{\boldsymbol{\Theta}} &= {}^b\boldsymbol{\omega} \\ {}^b\dot{\boldsymbol{\omega}} &= \boldsymbol{\gamma}_a \end{aligned} \tag{9}$$

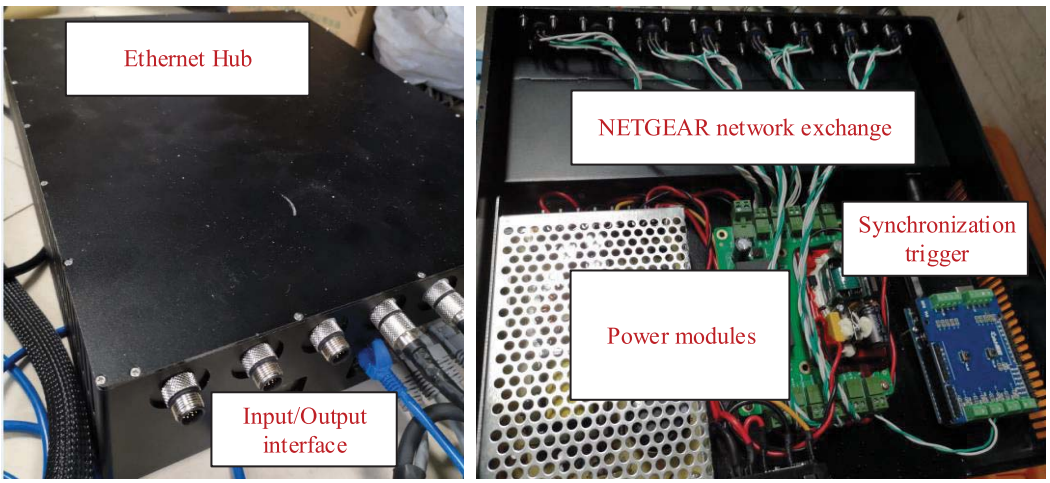
where $\boldsymbol{\gamma}_p, \boldsymbol{\gamma}_a \in \mathbb{R}^3$ are modeled as zero-mean Gaussian white noises. Suppose that T_s is the sampling time, and applying the first-order backward difference, the abstract discrete-time form of the process model (9) is described as

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \boldsymbol{\gamma}_k \tag{10}$$

where $\mathbf{F} \in \mathbb{R}^{12\times 12}$ is a block diagonal matrix with 2×2 blocks of the form $\begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix}$, and

$$\boldsymbol{\gamma}_k = [0\ \gamma_{1,k}\ 0\ \gamma_{2,k}\ 0\ \gamma_{3,k}\ 0\ \gamma_{4,k}\ 0\ \gamma_{5,k}\ 0\ \gamma_{6,k}]^T \in \mathbb{R}^{12}$$

models the motion uncertainties described by a zero-mean Gaussian distribution with covariance $\mathbf{Q}_k \in \mathbb{R}^{12\times 12}$.



(a) External structure (b) Internal structure

Fig. 13. Ethernet hub of the proposed testbed. The hub is used to send the synchronous trigger signal to cameras and receive image data from all cameras. (a) External structure. (b) Internal structure.

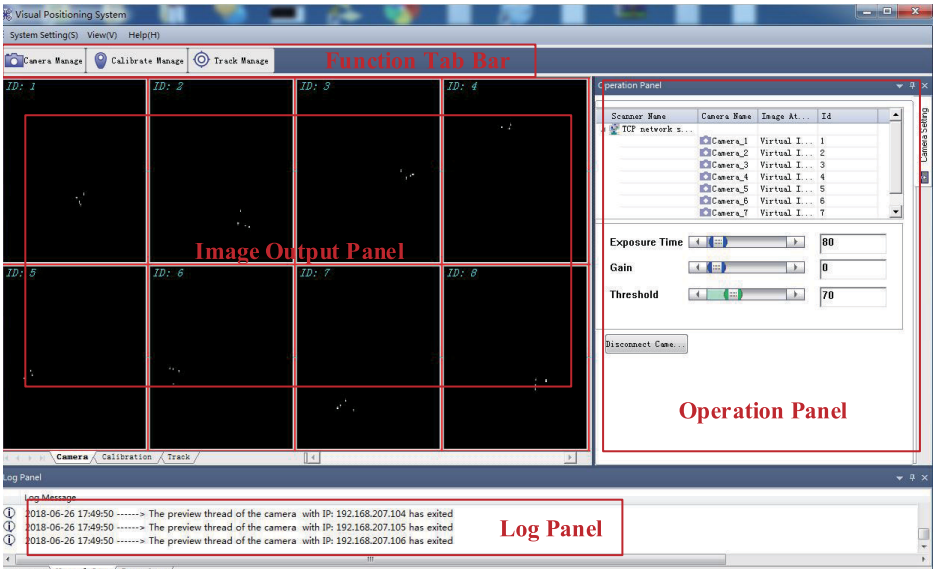


Fig. 14. 3-D pose estimation multi-vision ground station with eight cameras in the navigation computer.

b) Measurement model: Before describing the measurement model of the filter, the feature imaging principle needs to be derived, which explains the relationship between the 3-D space point on the UAV and the corresponding 2-D image point. Fig. 1 shows the situation of the multi-camera system, including the UAVs with onboard markers attached. Suppose that $\mathbf{T} \in \mathbb{R}^3, \mathbf{R}(\mathbf{\Theta}) \in \mathbb{R}^3$ are the relative position and orientation of the body frame ABCF with respect to the global frame EFCF, and they are also the pose to be estimated and tracked. The feature imaging principle is summarized as

$$\mathbf{p}_j^{c_i} = \mathbf{G}\left(k_1^i, k_2^i, m_u^i, m_v^i, u_0^i, v_0^i, k_3^i, k_4^i, k_5^i, \mathbf{R}_b^{c_i}, \mathbf{T}_b^{c_i}, {}^b\mathbf{P}_j\right), \quad i = 0, \cdots M-1;\ j = 1, \cdots N_p \tag{11}$$

where $\mathbf{p}_j^{c_i} = [u_j^{c_i}\ v_j^{c_i}]^T$ is the image point of the jth marker captured by the ith camera, the feature configuration ${}^b\mathbf{P}_j$ is the coordinate of the jth marker with respect to the ABCF, which needs to be accurately calibrated before filtering. Besides, we have the relationship that $\mathbf{R}_b^{c_i} = \mathbf{R}_e^{c_i}\mathbf{R}(\mathbf{\Theta})$ and $\mathbf{T}_b^{c_i} = \mathbf{R}_e^{c_i}\mathbf{T} + \mathbf{T}_e^{c_i}$, where the parameters $\mathbf{R}_e^{c_i}$ and $\mathbf{T}_e^{c_i}$ are obtained through the calibration process, so are the intrinsic parameters $(k_1^i, k_2^i, m_u^i, m_v^i, u_0^i, v_0^i, k_3^i, k_4^i$, and $k_5^i)$. Getting back to (11), given the 3-D point correspondence $\mathbf{p}_j^{c_i} \longleftrightarrow {}^b\mathbf{P}_j$ of each marker, the relationship with the estimated pose $(\mathbf{R}(\mathbf{\Theta}), \mathbf{T})$ can be obtained, which constitutes the visual measurement. What is essential is to determine the feature configuration and figure out the correspondences.

The measurement outputs for the filter are the 2-D image plane coordinates of the 3-D space points directly, and the measurement model defines the feature correspondences between the 2-D feature points and 3-D markers according to feature point imaging principle. Based on (11), the measurement model is expressed as

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k \tag{12}$$

where $\mathbf{z}_k = [\mathbf{z}_k^{c_0} \cdots \mathbf{z}_k^{c_{M-1}}]^T \in \mathbb{R}^{2N_pM}, \mathbf{z}_k^{c_i} = [u_1^{c_i}\ v_1^{c_i} \cdots u_{N_p}^{c_i}\ v_{N_p}^{c_i}]^T \in \mathbb{R}^{2N_p}$, and the symbol $\mathbf{z}_k^{c_i}$ is defined

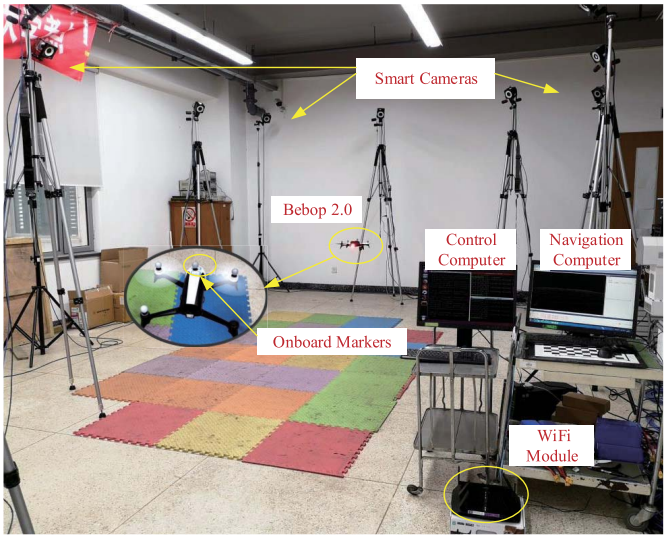


Fig. 15. Indoor flight control testbed.

as the measurement outputs of N_p feature points captured by the ith camera. The corresponding measurements degenerate to zero when there is no point captured by the camera. The function $\mathbf{h}(\mathbf{x}_k) = [\mathbf{h}_{c_0}(\mathbf{x}_k) \ \cdots \ \mathbf{h}_{c_{M-1}}(\mathbf{x}_k)]^T \in \mathbb{R}^{2N_pM}$, and the expression of $\mathbf{h}_{c_i}(\mathbf{x}_k) \in \mathbb{R}^{2N_p}$ can be obtained using (11). Besides, $\mathbf{v}_k = [\mathbf{v}_k^{c_0} \ \cdots \ \mathbf{v}_k^{c_{N-1}}]^T \in \mathbb{R}^{2N_pM}$, $\mathbf{v}_k^{ci} \in \mathbb{R}^{2N_p}$ is the measurement noise vector assumed as a zero-mean Gaussian white noise with covariance $\mathbf{R}_k \in \mathbb{R}^{2N_pM \times 2N_pM}$.

c) Extended Kalman Filter: Based on the linear constant velocity process model (10) and the nonlinear measurement model (12), a correspondence-based EKF is employed to estimate the state variables of the UAV in this section. The EKF is a commonly used method for tracking and control of UAVs. The EKF algorithm consists of three stages: initialization, prediction, and update. The procedure of the filter is shown in Table I. There are some practical issues to be noticed. First, the initial values of the system state are important for the filter. The optimal point correspondences between two images are obtained based on the epipolar geometric constraints and a first-order geometric error—Sampson distance [36]; only if the Sampson distance between two image points is small enough, the points are viewed as the matching points. Since point correspondence is required in the filter to constitute the measurement model, a point correspondence search method combining the predicted pose of the filter with the brute-force approach in [37] is proposed. To be more specific, the correspondences are determined by using the prediction from the filter or, in case of failure, using the brute-force search approach. This searching method still takes effect when some markers are missing.

B. Ground Control System Design

The ground control system is designed in a Simulink model. The three-channel control model is proposed first, followed by the controller design based on a PD controller.

1) Three-Channel Control Model: Remote pilots often control multicopters based on open-source semi-autonomous autopilots (SAAs) to execute tasks. They do not need to know

TABLE I
PROCEDURE OF THE EKF IN POSE ESTIMATION

Initialization	The initial state $\hat{\mathbf{x}}_{0\|0}$ and the initial error covariance $\mathbf{P}_{0\|0}$
Prediction	$\hat{\mathbf{x}}_{k\|k-1} = \mathbf{F}\hat{\mathbf{x}}_{k-1\|k-1}$ $\mathbf{P}_{k\|k-1} = \mathbf{F}\mathbf{P}_{k-1\|k-1}\mathbf{F}^T + \mathbf{Q}_{k-1}$
Update	$\mathbf{H}_k = \frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}} \|_{\mathbf{x}=\hat{\mathbf{x}}_{k\|k-1}}$ $\mathbf{K}_k = \mathbf{P}_{k\|k-1}\mathbf{H}_k^T\left(\mathbf{R}_k + \mathbf{H}_k\mathbf{P}_{k\|k-1}\mathbf{H}_k^T\right)^{-1}$ $\hat{\mathbf{x}}_{k\|k} = \hat{\mathbf{x}}_{k\|k-1} + \mathbf{K}_k\left(\mathbf{z}_k - \mathbf{h}\left(\mathbf{x}_{k\|k-1}\right)\right)$ $\mathbf{P}_{k\|k} = \mathbf{P}_{k\|k-1} - \mathbf{K}_k\mathbf{H}_k\mathbf{P}_{k\|k-1}$

the low-level flight control law inside the autopilots. It is applicable to develop special algorithms based on the existing SAAs directly. Besides, There are often three common modes in an SAA: the "stabilize mode," the "altitude hold mode," and the "loiter mode." In this paper, a position controller is designed based on the stabilize mode in an SAA. For the convenience of controller design, the control model of a rigid object can be linearized and simplified to a three-channel control model, i.e., altitude channel, yaw channel, and horizontal position channel. With SAAs, the model is simplified as [24, Ch. 12]

$$\begin{aligned}
\dot{p}_{z_e} &= v_{z_e} \\
\dot{v}_{z_e} &= -k_{v_z} v_{z_e} - k_{u_T} u_T \\
\dot{\psi} &= \omega_z \\
\dot{\omega}_z &= -k_{\omega_z}\omega_z + k_{u_{\omega_z}} u_{\omega_z} \\
\dot{\mathbf{p}}_h &= \mathbf{R}_b^e(\psi)\mathbf{v}_{h_b} \\
\dot{\mathbf{v}}_{h_b} &= -\mathbf{K}_{\mathbf{v}_{h_b}}\mathbf{v}_{h_b} - g\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}\Theta_h \\
\dot{\Theta}_h &= \boldsymbol{\omega}_{h_b} \\
\dot{\boldsymbol{\omega}}_{h_b} &= -\mathbf{K}_{\Theta_h}\Theta_h - \mathbf{K}_{\omega_{h_b}}\boldsymbol{\omega}_{h_b} + \mathbf{K}_{\mathbf{u}_\mathbf{h}}\mathbf{u}_\mathbf{h}
\end{aligned} \tag{13}$$

where $\mathbf{p}_h = [p_{x_e} \ \ p_{y_e}]^T$, $\mathbf{v}_{h_b} = [v_{x_b} \ \ v_{y_b}]^T$, $\Theta_h = [\phi \ \ \theta]^T$, $\boldsymbol{\omega}_{h_b} = [\omega_{x_b} \ \ \omega_{y_b}]^T$, and

$$\mathbf{R}_b^e(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix} \tag{14}$$

defines the horizontal rotation by ψ from the ABCF to the EFCF since the horizontal control signals are with respect to the body frame. The first two equations of (13) represent the altitude channel model, and the following two equations of (13) describe the yaw channel model, while the last three equations of (13) are the horizontal position channel model. Moreover, $k_{v_z}, k_{u_T}, k_{\omega_z}, k_{u_{\omega_z}} \in \mathbb{R}_+$, $\mathbf{K}_{\mathbf{v}_{h_b}}, \mathbf{K}_{\Theta_h}, \mathbf{K}_{\omega_{h_b}}, \mathbf{K}_{\mathbf{u}_h} \in \mathbb{R}^{2\times 2}$ are parameters determined by the selected AAS. There are control command signals from three channels: the altitude channel command u_T, the yaw channel command u_{ω_z}, and the horizontal position channel commands $\mathbf{u}_h = [u_\phi \ u_\theta]^T$.

2) Controller Design: In our testbed, we choose the Parrot Bebop 2.0 as the target UAV due to its small size, low price, sturdiness to crash, and safeness with a protective hull even used indoor or close to people. Above all, it is fairly simple and easy to implement control using the specific SDK, and we need to communicate with the UAVs through wireless LAN.

TABLE II
TESTBED SPECIFICATION

Multi − camera system	
Number	Eight smart cameras (Type: SCZE130M-GEHD)
Resolution	1280×720 pixels
Focal length	4 mm
Field of View (FOV)	77.32°
Frames Per Second (FPS)	100 Hz
Coverage area	$7 \times 7 \times 2.2$ m
Infrared light source	Type: Ring100A100-IR; Wavelength: 850 nm;
	Power: 4 W
Infrared-pass filter	Wavelength: 850 nm
Synchronization trigger	Type: CBAT328-IO602
Navigation computer	Intel Core i7, 3.6-GHz, 8-GB RAM
Ground control system	
Control computer	Intel Core i7, 3.6-GHz, 8-GB RAM
	Virtual machine running Ubuntu 16.04 with ROS kinetic
WiFi module	TP-LINK TL-WDR6500, 1300 Mbps
Target UAVs	Quadcopter Parrot Bebop 2.0
	Size 36 cm × 15 cm × 38 cm and weigh 500 g with battery
	three-axis accelerometer, three-axis gyroscope,
	an ultrasonic range finder
Onboard markers	four infrared reflective markers with diameter 15.9 mm

TABLE III
COMPARISON BETWEEN THE ESTIMATED LENGTH AND THE GROUND TRUTH

	AB	BC	CD	AD	AC	BD
True Length (mm)	600.0000	600.0000	600.0000	600.0000	848.5281	848.5281
Reconstruction length (mm)	597.0844	601.5358	602.1676	598.0705	849.7912	846.4435
Reconstruction Error (mm)	2.9156	-1.5358	-2.1676	1.9295	-1.2631	2.0846

Therefore, by comparing the pose information from the visual feedback with the reference position and orientation, we can design a simple PD controller to generate the control signals that are then sent to the target UAV at a fixed frequency. As shown in Fig. 8, the UAV dynamic model, including inner-loop control, is regarded as a black box, which only receives the control commands from three channels.

The objective of this section is to design extra position controllers based on the existing stabilize mode of the SAA (13). Given a desired trajectory $\mathbf{p}_d(t) = [p_{x_e d} \ p_{y_e d} \ p_{z_e d}]^T \in \mathbb{R}^3$ and the desired yaw angle $\psi_d(t)$, it is expected to make $\|\mathbf{x}(t) - \mathbf{x}_d(t)\| \to \mathbf{0}$ through the control inputs u_T, u_{ω_z}, and $\mathbf{u}_h$ as $t \to \infty$, where $\mathbf{x} = [\mathbf{p}^T \ \psi]^T$ and $\mathbf{x_d} = [\mathbf{p}_d^T \ \psi_d]^T$. Therefore, a PD controller is designed for the three-channel control model directly as

$$\begin{aligned} u_T &= -k_{T,P}(p_{z_e} - p_{z_e d}) - k_{T,D}(\dot{p}_{z_e} - \dot{p}_{z_e d}) \\ u_{\omega_z} &= -k_{\psi,P}(\psi - \psi_d) - k_{\psi,D}(\dot{\psi} - \dot{\psi}_d) \\ \mathbf{u}_h &= -\mathbf{K}_{h,P}\mathbf{R}_e^b(\psi)(\mathbf{p}_h - \mathbf{p}_{hd}) - \mathbf{K}_{h,D}\mathbf{R}_e^b(\psi)(\dot{\mathbf{p}}_h - \dot{\mathbf{p}}_{hd}) \end{aligned} \quad (15)$$

where the parameters $k_{T,P}, k_{T,D}, k_{\psi,P}, k_{\psi,D} \in \mathbb{R}_+$, $\mathbf{K}_{h,P}, \mathbf{K}_{h,D} \in \mathbb{R}^{2\times 2}$ are PD coefficients that need to be tuned. Note that the yaw angle can be controlled with pure proportional control, i.e., $k_{\psi,D} = 0$.

C. Software Framework

Based on the multi-camera system design and the ground control system described earlier, the software framework of the indoor flight testbed is shown in Fig. 9. It consists of four main nodes (executable files within ROS packages), i.e., Simulink model node, *bebop_driver* node, *vps_driver* node, and *Bebop_GUI* node. The *vps_driver* node is to communicate with the multi-camera system via user datagram protocol (UDP) and then publish the pose and velocity information by the */vpsnavdata* topic. *bebop_driver* is the ROS driver for Bebop to subscribe the control command by the */bebop/cmd_vel* topic and the basic command, such as takeoff, land, and reset. *Bebop_GUI* is a simple interface to control the Bebop through MATLAB and can implement simple commands, such as taking off, hovering, and landing. After taking off, the Simulink model node to execute visual feedback is to employ the RST of MATLAB to subscribe all the navigation data, generate, and publish control command

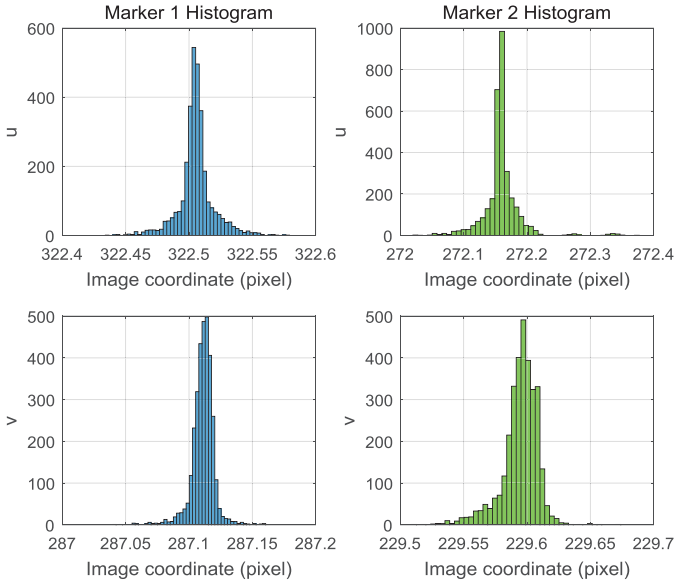


Fig. 16. Histogram of image coordinates of two markers of the UAV at a fixed position. It is calculated that the variances of the image coordinates are marker 1 (2.0457e-04 and 8.3111e-05) pixel and marker 2 (8.3793e-04 and 2.0475e-04) pixel.

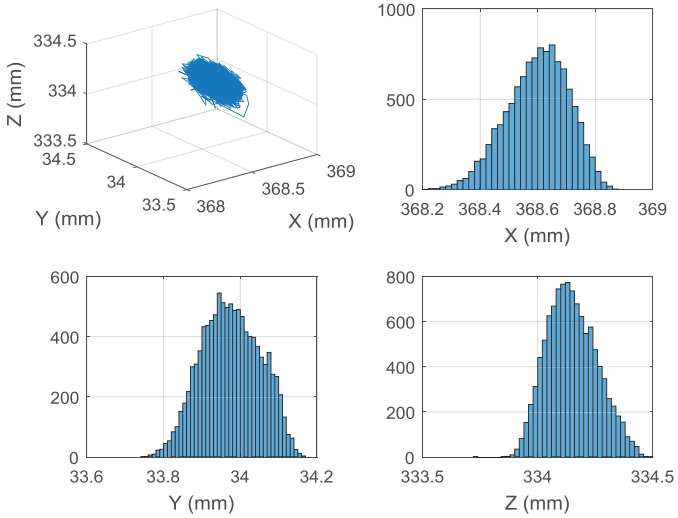
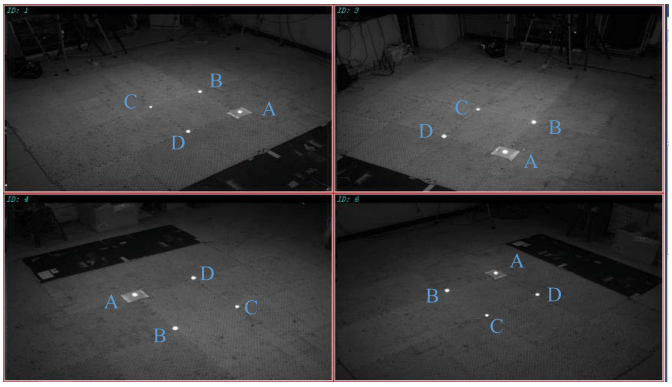


Fig. 17. Histogram of reconstructed UAV position at a fixed position. It is calculated that the variances of the reconstructed 3-D position are (1.1485e-05, 5.7719e-06, and 1.1829e-05) mm.

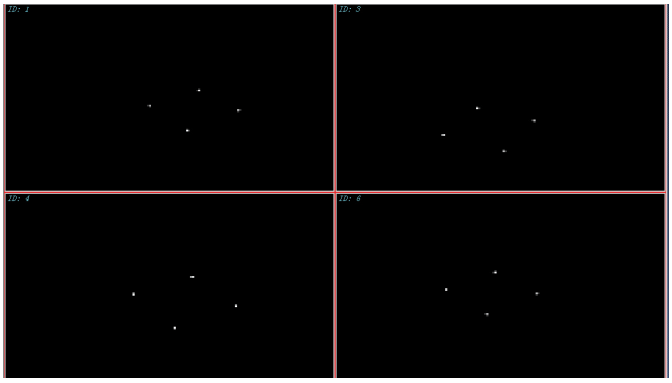
signals to the target Bebops. The diagram of the Simulink model is shown in Fig. 10 with the corresponding source code available at https://github.com/DengMark/BebopTest. Therefore, it is flexible for researchers to test and evaluate any advanced control strategies with a variety of reference trajectories, and they just need to modify the references and controller modules in the Simulink model.

IV. Testbed Hardware Design

In order to test and demonstrate the multi-camera system and ground control system described earlier, an indoor multi-camera-based testbed for 3-D tracking and control of UAVs is constructed by members of our group. Fig. 11 shows the hardware architecture of the proposed testbed comprised of a camera sensor, sensing unit, processing unit, interface unit, and application unit. In the testbed, smart cameras are employed as the main sensors to obtain visual information. The camera sensor, which is implemented in complementary metal–oxide–semiconductor (CMOS) technology, provides the raw image data of the processing pipeline inside a smart camera. The sensing unit reads the raw image data and



(a) The raw image with markers placed on the ground



(b) The corresponding virtual image with markers detected

Fig. 18. (a) Raw and (b) virtual images with markers placed on the ground. Marker A is near the origin of the EFCF.

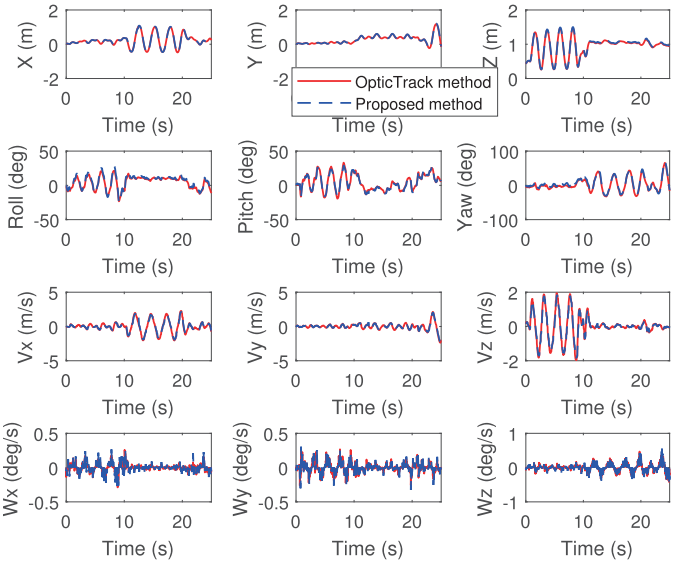


Fig. 19. Comparison of pose estimation results between the proposed multi-camera system and a commercial motion capture system OptiTrack.

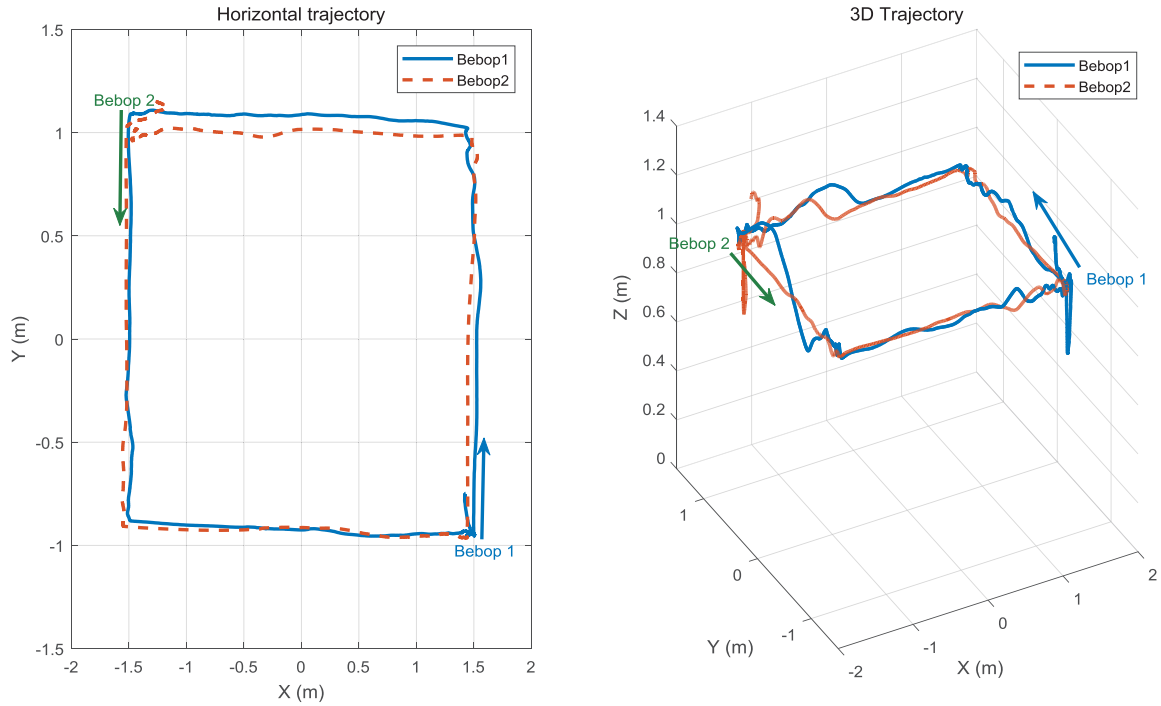


Fig. 20. Horizontal and 3-D trajectory results of waypoint controlling using two bebops.

TABLE IV
BIAS ERROR BETWEEN THE PROPOSED ESTIMATION AND THE OPTITRACK ESTIMATION

Bias error	Average	Standard deviation
X (m)	0.0038	0.0147
Y (m)	0.0029	0.0138
Z (m)	0.0071	0.0180
Roll (deg)	1.4713	0.0271
Pitch (deg)	0.0632	0.0262
Yaw (deg)	1.9471	0.0587
Vx (m/s)	0.0053	0.0790
Vy (m/s)	0.0055	0.0707
Vz (m/s)	0.0069	0.1029
Wx (deg/s)	5.4443e-05	1.1559e-05
Wy (deg/s)	2.1968e-05	1.2058e-05
Wz (deg/s)	3.2039e-04	2.0039e-05

TABLE V
VALUES OF THE PD COEFFICIENTS IN THE GROUND CONTROL SYSTEM

Coefficient	$k_{T,P}$	$k_{T,D}$	$k_{\psi,P}$	$\mathbf{K}_{\mathrm{h},P}$	$\mathbf{K}_{\mathrm{h},D}$
Value	0.8	0.1	0.4	diag(0.5,0.5)	diag(0.3,0.3)

performs some preprocessing, such as white balance and color transformations. It also controls some parameters of the camera, such as capture rate, gain, or exposure time. The main image processing tasks (i.e., marker detection and extraction in this testbed) are implemented at the processing unit, which receives the images from the sensing unit, performs real-time image process, and finally transfers the coordinates of all markers to the interface unit. The main processing unit uses a latest Zynq-7000 system on a chip that is equipped with a dual-core ARM Cortex-A9 processor integrated with 28-nm Xilinx-based programmable logic on an FPGA for hardware acceleration and excellent performance. It has 1-GB DDR3 and 4-GB flash memory with the maximum frequency up to 866 MHz. The interface unit provides multiple external input/output interfaces, such as USB and Ethernet. Finally, the processed data are transferred to the application unit via the interface unit. The main algorithms of this testbed, including the multi-camera calibration, visual tracking, and 3-D control, are implemented in the application unit. Camera structure and Ethernet hub are shown in Figs. 12 and 13, respectively.

The indoor flight testbed is developed as in Fig. 14 with specifications shown in Table II. Eight smart cameras with an infrared pass filter and external trigger provide the synchronized images of the target UAV from different fields of view to the ground navigation computer. The navigation computer is to implement image processing, camera calibration, and pose estimation. Besides, a 3-D pose estimation multi-camera ground station is developed in the navigation computer to visualize the current state and operation steps of the system based on a Microsoft Foundation Class (MFC)-based Graphical User Interface (GUI), as shown in Fig. 14. The control computer running ROS and Simulink model receives the pose information from the navigation computer and then generates and transmits the control signals to the UAV through wireless LAN. A low-cost quadcopter Bebop 2.0 with four reflective markers attached is utilized.

TABLE VI
COORDINATES OF THE DESIRED AND ACTUAL WAYPOINTS OF TWO BEBOPS DURING THE TEST (UNIT: M)

	Bebop 1(desired)	Bebop 2(desired)	Bebop 1(actual)	Bebop 2(actual)
Waypoint 1	(1.5,-1,1)	(-1.5,1,1)	(1.45,-0.92,1.02)	(-1.45,1.09,0.96)
Waypoint 2	(1.5,1,1)	(-1.5,-1,1)	(1.47,0.93,1.02)	(-1.48,-0.91,1.01)
Waypoint 3	(-1.5,1,1)	(1.5,-1,1)	(-1.48,1.08,1.03)	(1.44,-0.91,1.02)
Waypoint 4	(-1.5,-1,1)	(1.5,1,1)	(1.50,-0.85,0.99)	(1.49,0.92,1.02)
Waypoint 5	(1.5,-1,1)	(-1.5,1,1)	(1.40,-0.93,0.99)	(-1.44,1.01,1.00)

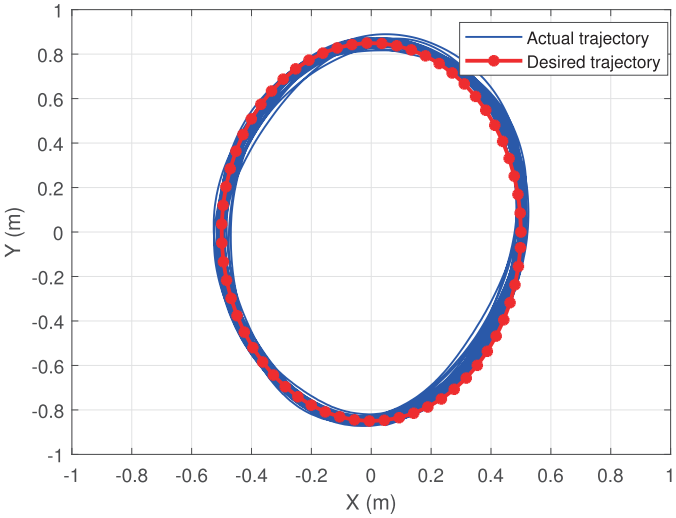


Fig. 21. Horizontal results of an ellipse trajectory.

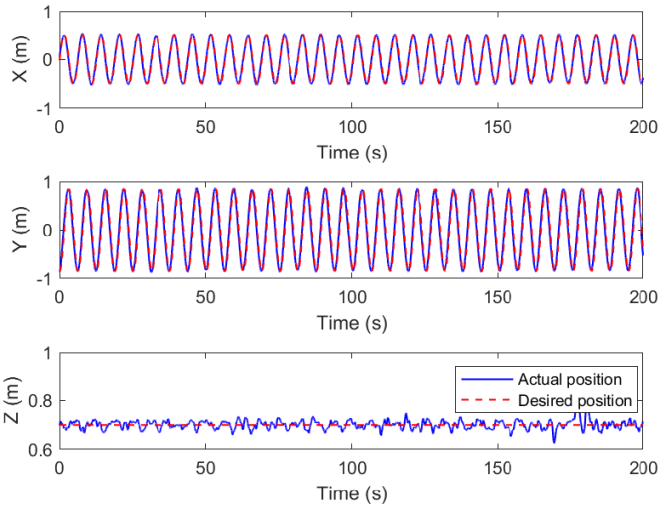


Fig. 22. Comparison results between the desired position and the actual position.

V. EXPERIMENTAL RESULTS

This section presents the performance evaluation of the proposed indoor flight control testbed. First, the multi-camera system is evaluated to check the accuracy and efficiency of pose estimation. Then, based on the pose information, some real indoor flight tests are conducted using the ground control system to control the target UAVs. A video of thorough experimental tests is available at https://youtu.be/k9_u-yvZb1w and our Reliable Flight Control Group website http://rfly.buaa.edu.cn.

A. Pose Estimation Results

First, it is required to obtain the noise characteristic of the multi-camera system to find the accurate values of the filter parameters. Fig. 16 shows the histogram of the image coordinate of the two reflective markers on the UAV at a fixed position after camera calibration. The result depends on the performance of the camera sensors and feature detection capability. Besides, the position of the UAV is obtained by a 3-D reconstruction algorithm discussed earlier, and the histogram of the reconstructed UAV position at a fixed position is shown in Fig. 17. The results indicate that the multi-camera system can provide a high-precision pose estimation, enough to serve as visual feedback to control the UAV.

Then, the pose estimation experiment is conducted in case that there are four reflective markers with fixed positions on the ground composing a square with a length of 60 cm. The placement of the markers and the image captured by four cameras are shown in Fig. 18. Similarly, the global positions of the markers can be reconstructed by the multi-camera system based on the visual algorithm and EKF mentioned earlier, and then, the lengths of each edge of the square are computed.

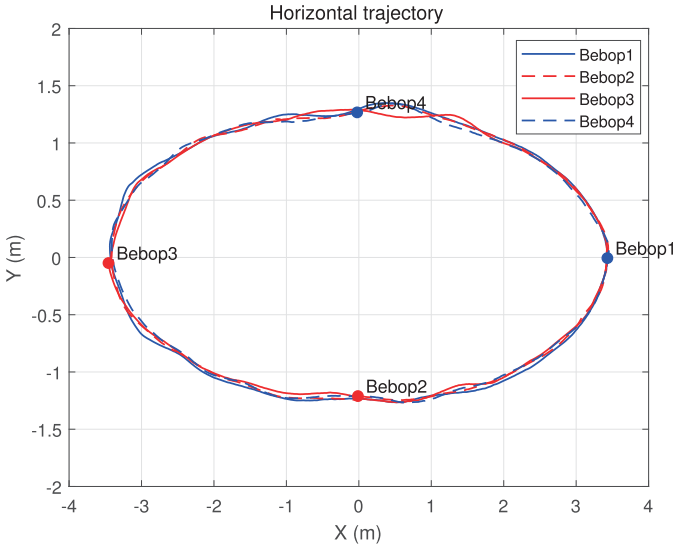


Fig. 23. Horizontal results of four Bebops with an elliptical trajectory. The four markers represent the initial position of four Bebop drones.

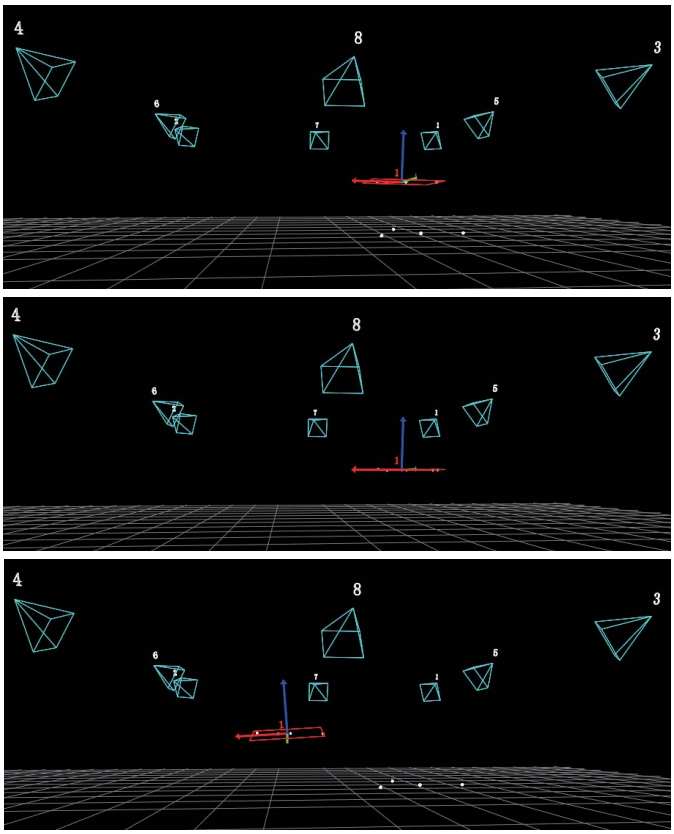


Fig. 24. Samples of the tracking situation. Bebop is tracked and controlled well by the proposed testbed, while some markers are added during flying.

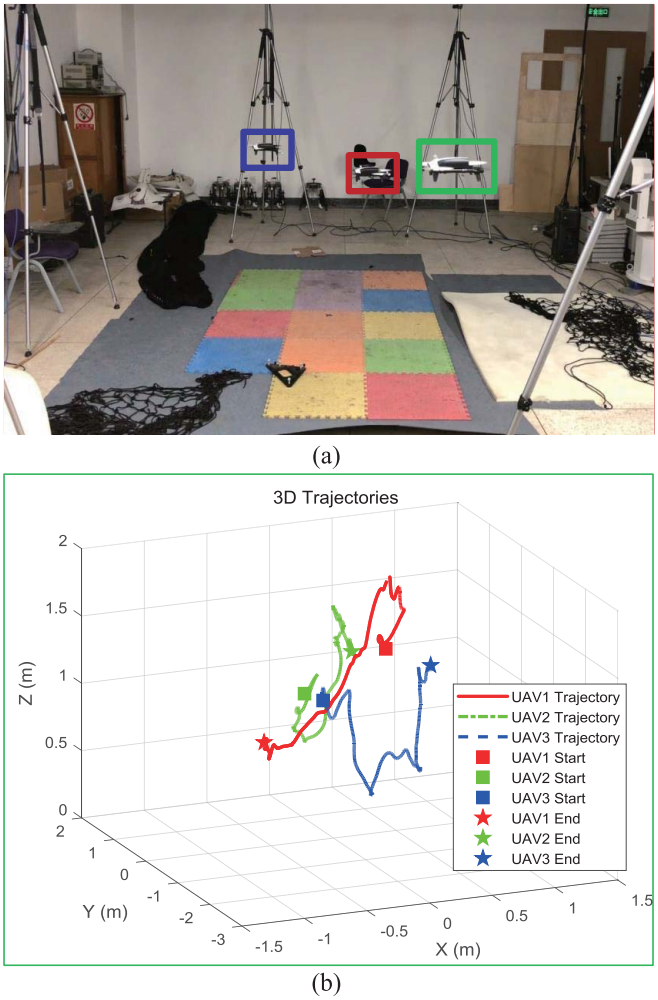
The comparison between the reconstruction length and the corresponding ground truth is shown in Table III. It is indicated that the multi-camera system has high precision with the estimation error of less than 3 mm. Besides, the processing speed is up to 100 frames/s. From the earlier results, the proposed multi-camera system has a good performance, so that it can be employed to control the target UAV within the field of view.

Finally, a comparison of pose estimation results between the proposed multi-camera system and a commercial motion capture system OptiTrack has been made. The target UAV is moved freely by hand. The results are shown in Fig. 19. Furthermore, Table IV shows the average and standard deviation for the bias error between the proposed estimation and the OptiTrack estimation. The results indicate that the proposed multi-camera system is accurate.

B. Flight Test Results

In this section, some closed-loop control flight tests of the target UAV with pose estimation using the proposed multi-camera system are performed. In these tests, the position and yaw are controlled by the proposed PD controller in (15) by using the position, velocity, and attitude from the multi-camera system and an EKF. The visual measurement update rate is 50 Hz, and the process and measurement noises are zero-mean white Gaussian noises with the covariance of 0.0001 and 0.05, respectively. The PD coefficients are shown in Table V.

In the first experiment, two Bebops are controlled to fly a rectangle at the same time. The reference waypoints of the two Bebops are given as in Table VI. Similarly, the horizontal and 3-D trajectories of two Bebops are shown in Fig. 20. The results show that the proposed testbed can control multiple Bebops with a satisfactory tracking performance.



(a)

(b)

Fig. 25. 3-D trajectories of three Bebop drones. The three drones are controlled to fly from the starting waypoint to the end waypoint based on the proposed testbed, and there are cross collisions among their trajectories. (a) Flight scene with cross trajectories among three UAVs. (b) 3D trajectories of three UAVs.

In the second experiment, the Bebop is controlled to fly an ellipse with the desired trajectory defined as

$$x = 0.5\sin(t + \pi/2)$$
$$y = 0.8\sin(t)$$
$$z = 0.7.$$

The horizontal trajectory and the comparison results between the desired position and the actual position are shown in Figs. 21 and 22, respectively. The results imply that the designed controller has an accurate precision.

Then, we tested the proposed testbed on scalability to demonstrate how many targets that the testbed is able to track and control. It is noted that the bandwidth of visual data becomes bigger with more targets and more cameras, which will leave more burden on the processor. Besides, the effective

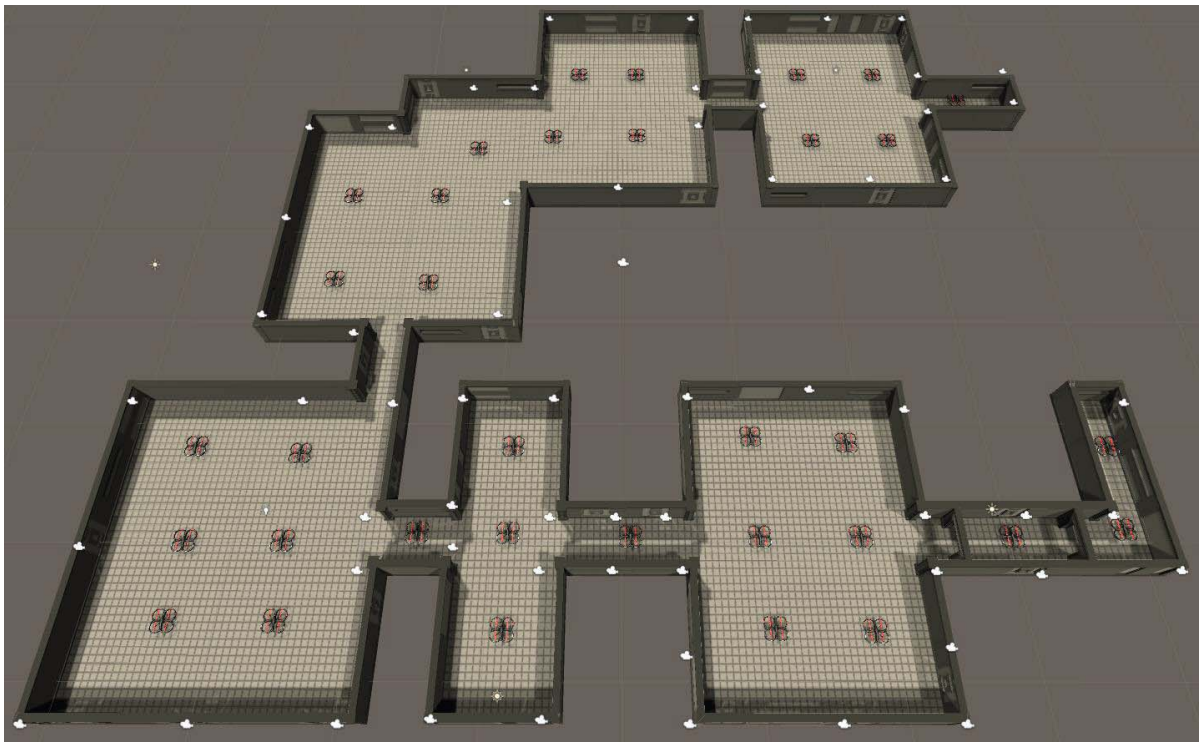


Fig. 26. Example of a large-scale VSN, where the multiple UAVs are coordinated to implement some specific tasks in a large factory.

range of view is limited with a limited number of external cameras. In the final experiment, we have tracked four Bebops and control them to fly an elliptical trajectory. The ellipse center of the referenced ellipse is (0, 0) m. The semimajor and semiminor axes of the referenced ellipse are 3.5 and 1.3 m, respectively. The horizontal trajectory is shown in Fig. 23. The results indicate that the designed testbed can track and control four quadcopters, and the control accuracy is reliable. Moreover, to demonstrate the robustness and reliability of the proposed testbed, we added some markers as outliers and noises, while the quadcopter is flying under control. Some samples of the tracking situations are shown in Fig. 24. The supplementary experimental results of the proposed testbed are available at https://youtu.be/Z99DHUqEERw. In the video, it is demonstrated that the testbed is reliable in the presence of disturbance and noises.

Finally, more complicated situations with cross trajectories among multiple UAVs are considered in this paper. Hence, collision avoidance algorithms need to be added to the main navigation and guidance algorithms, which will extend the practical applications of the proposed testbed. As shown in Fig. 25(a), three Bebop drones are controlled to fly from the starting waypoint to the end waypoint based on the proposed testbed, and there are cross collisions among their trajectories. Combined with a collision avoidance algorithm by using an artificial potential field method, flight results are shown in Fig. 25(b). The results show that all three drones fly autonomously without collision. More experimental results are available at https://youtu.be/ShGW6SxSnUI.

Remark 6: During the tests, it is noticed that the altitude channel is not well controlled since the readings of the ultrasonic range finder are not accurate and tend to vary near a fixed position. Therefore, the controller of the altitude channel will respond quickly, and the altitude data will fluctuate. Besides, it is essential to validate the efficiency of the system, the time delays of some major processes, i.e., imaging processing, Kalman filtering, and wireless communication, are measured. Note that the algorithms of the multi-camera system and the ground control system are run in separate computers. The main image processing is also executed inside the FPGA units of smart cameras. Finally, it is measured that the time delay of the imaging processing is about 9 ms, while the filtering costs about 5 ms, and wireless communication delay is $7 \sim 13$ ms for the testbed. The results show that the time delays exist but are relatively small, and the update rate is high enough to control the target vehicles (the control commands are sent to the vehicles at a fixed rate of 50 Hz in the testbed).

C. Discussion

Based on the experimental results, we have verified the feasibility and high-precision accuracy of the proposed indoor multi-camera flight testbed. Although some simple examples are conducted to validate the functionality of the proposed testbed, there exist a great variety of potential applications for the proposed testbed in many fields.

1) In scientific research, as an indoor flight testbed with millimeter positioning accuracy and high frame rate (100 Hz for our testbed), it is very appropriate for testing and evaluation of advanced algorithms, such as multi-agent control strategies, formation control, and obstacle avoidance. The testbed usually serves as a ground-truth reference for localization. Besides,

the testbed can be a testbed for preliminary simulation and validation of some scenarios demanding wide working range and great maneuverability, such as the decision-making of UAV delivery, the control of UAV autonomous refueling docking, and cooperative control of UAV swarm. Furthermore, the testbed can be fused with other sensors, such as active range finders, GPS, and IMU, to estimate states accurately and robustly.

2) In education industry, the testbed can be used as a platform for college curriculum design, based on which we can design some basic experiment courses, such as the design of state filters, the design of positioning controller, the design of tracking controller, the design of path-following controller, the design of obstacle avoidance controller, and the design of formation controller.
3) In robotics industry, the proposed testbed applies to many other kinds of featured robots besides UAVs. It can be extended to the tracking and control of unmanned ground vehicles (UGVs) and robot manipulator. Besides, the ground control system is implemented in a Simulink model, which can be automatically transferred into C++ source code to be directly employed in ROS-enabled robots. This idea is motivated by the model-based design [38] to reduce development time greatly and avoid manually coded errors.

Nevertheless, our proposed testbed still has a few limitations that can be addressed in the future work. The testbed requires all the cameras to be synchronized, and pairwise overlap is necessary. Besides, the working volume of the testbed is small due to the limited number of employed cameras. The applications may be limited due to these limitations in terms of working range and accuracy, and it is hopeful for the testbed to extend to a large-scale visual sensor network (VSN) [39], based on which a large number of topics can be researched further, such as the camera calibration, localization, networking, and collaborative routing. As shown in Fig. 26, a large number of cameras are employed to monitor a large area, where we can control multiple UAVs to implement some specific tasks, such as formation flight and goods transportation. Besides, we are now trying to design an accurate and flexible approach to the geometric calibration of a class of VSNs, and a video is available at https://youtu.be/XmglR4HAsEw.

VI. Conclusion

In this paper, a comprehensive and generic multi-camera based testbed for 3-D tracking and control of UAVs has been proposed. The testbed consists of a multi-camera system, a ground control system, onboard infrared reflective markers, and target UAVs, which performs main algorithms, including image processing, camera calibration, 3-D reconstruction, pose estimation, and motion control. The experimental results show that the proposed testbed provides a comprehensive and complete platform with good scalability applicable for research on a variety of advanced guidance, navigation, and control algorithms. It has extensive potential applications in many fields, such as scientific research, education, and robotics. However, the application range is limited and influenced by the amount and placement of the cameras because of the finite field of view. In the future research, camera placement problem in this multi-camera system needs to be studied to expand the sensing coverage and decrease the reconstruction error. A large-scale VSN is to be designed by using more cameras to track and control more UAVs.

References

[1] T. Tomic *et al.*, "Toward a fully autonomous UAV: Research platform for indoor and outdoor urban search and rescue," *IEEE Robot. Autom. Mag.*, vol. 19, no. 3, pp. 46–56, Sep. 2012.

[2] P. Yao, Z. Xie, and P. Ren, "Optimal UAV route planning for coverage search of stationary target in river," *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 2, pp. 822–829, Mar. 2019.

[3] Y. Zhao, Z. Zheng, and Y. Liu, "Survey on computational-intelligence-based UAV path planning," *Knowl.-Based Syst.*, vol. 158, pp. 54–64, Oct. 2018.

[4] Z. H. Zhiyao, Y. A. Peng, W. A. Xiaoyi, X. U. Jiping, W. A. Li, and Y. U. Jiabin, "Reliable flight performance assessment of multirotor based on interacting multiple model particle filter and health degree," *Chin. J. Aeronaut.*, vol. 32, no. 2, pp. 444–453, Feb. 2019.

[5] Z. Zhao, X. Wang, P. Yao, J. Xu, and J. Yu, "Fuzzy health degree-based dynamic performance evaluation of quadrotors in the presence of actuator and sensor faults," *Nonlinear Dyn.*, vol. 95, pp. 2477–2490, Feb. 2019.

[6] S. Zhu, D. Wang, and C. B. Low, "Ground target tracking using UAV with input constraints," *J. Intell. Robotic Syst.*, vol. 69, pp. 417–429, Feb. 2013.

[7] P. Yao, H. Wang, and Z. Su, "Real-time path planning of unmanned aerial vehicle for target tracking and obstacle avoidance in complex dynamic environment," *Aerosp. Sci. Technol.*, vol. 47, pp. 269–279, Dec. 2015.

[8] F. Borrelli, T. Keviczky, and G. J. Balas, "Collision-free UAV formation flight using decentralized optimization and invariant sets," in *Proc. 43rd IEEE Conf. Decis. Control*, Dec. 2004, pp. 1099–1104.

[9] H. Rezaee, F. Abdollahi, and H. A. Talebi, "H_∞ based motion synchronization in formation flight with delayed communications," *IEEE Trans. Ind. Electron.*, vol. 61, no. 11, pp. 6175–6182, Nov. 2014.

[10] D. H. Won *et al.*, "Selective integration of GNSS, vision sensor, and INS using weighted DOP under GNSS-challenged environments," *IEEE Trans. Instrum. Meas.*, vol. 63, no. 9, pp. 2288–2298, Sep. 2014.

[11] X. Gong, J. Zhang, and J. Fang, "A modified nonlinear two-filter smoothing for high-precision airborne integrated GPS and inertial navigation," *IEEE Trans. Instrum. Meas.*, vol. 64, no. 12, pp. 3315–3322, Dec. 2015.

[12] D. Floreano and R. J. Wood, "Science, technology and the future of small autonomous drones," *Nature*, vol. 521, nos. 75–53, p. 460, May 2015.

[13] A. Jiménez-González, J. R. Martinez-de Dios, and A. Ollero, "Testbeds for ubiquitous robotics: A survey," *Robot. Auton. Syst.*, vol. 61, no. 12, pp. 1487–1501, Dec. 2013.

[14] M. Valenti, B. Bethke, G. Fiore, J. How, and E. Feron, "Indoor multi-vehicle flight testbed for fault detection, isolation, and recovery," in *Proc. AIAA Guid., Navigat., Control Conf. Exhibit*, Aug. 2006, p. 6200.

[15] L. C. Mak, M. Whitty, and T. Furukawa, "A localisation system for an indoor rotary-wing MAV using blade mounted LEDs," *Sensor Rev.*, vol. 28, no. 2, pp. 125–131, Mar. 2008.

[16] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, "The GRASP multiple micro-UAV testbed," *IEEE Robot. Autom. Mag.*, vol. 17, no. 3, pp. 56–65, Sep. 2010.

[17] H. Oh, D.-Y. Won, S.-S. Huh, D. H. Shim, M.-J. Tahk, and A. Tsourdos, "Indoor UAV control using multi-camera visual feedback," *J. Intell. Robotic Syst.*, vol. 61, nos. 1–4, pp. 57–84, Jan. 2011.

[18] S. Tomer *et al.*, "A low-cost indoor testbed for multirobot adaptive navigation research," in *Proc. IEEE Aerosp. Conf.*, Mar. 2018, pp. 1–12.

[19] S. Parrot. (2016). *Parrot Bebop 2 FPV*. [Online]. Available: https://www.parrot.com/global/drones/parrot-bebop-2-fpv

[20] N. Lepisto, B. Thornberg, and M. O'Nils, "High-performance FPGA based camera architecture for range imaging," in *Proc. NORCHIP*, Nov. 2005, pp. 165–168.

[21] B. Rinner and W. Wolf, "An introduction to distributed smart cameras," *Proc. IEEE*, vol. 96, no. 10, pp. 1565–1575, Oct. 2008.

[22] E. Gudis, G. van der Wal, S. Kuthirummal, and S. Chai, "Multi-resolution real-time dense stereo vision processing in FPGA," in *Proc. IEEE 20th Int. Symp. Field-Program. Custom Comput. Mach.*, Apr. 2012, pp. 29–32.

[23] C. Bobda and S. Velipasalar, *Distributed Embedded Smart Cameras*. Singapore: Springer, 2014.
[24] Q. Quan, *Introduction to Multicopter Design Control*. New York, NY, USA: Springer, 2017.
[25] M. Quigley *et al.*, "ROS: An open-source robot operating system," in *Proc. ICRA Workshop Source Softw.*, May 2009, pp. 1–6.
[26] K. A. Tarabanis, P. K. Allen, and R. Y. Tsai, "A survey of sensor planning in computer vision," *IEEE Trans. Robot. Autom.*, vol. 11, no. 1, pp. 86–104, Feb. 1995.
[27] V. Akbarzadeh, C. Gagne, M. Parizeau, M. Argany, and M. A. Mostafavi, "Probabilistic sensing model for sensor placement optimization based on line-of-sight coverage," *IEEE Trans. Instrum. Meas.*, vol. 62, no. 2, pp. 293–303, Feb. 2013.
[28] P. Rahimian and J. K. Kearney, "Optimal camera placement for motion capture systems," *IEEE Trans. Vis. Comput. Graphics*, vol. 23, no. 3, pp. 1209–1221, Mar. 2017.
[29] 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.
[30] R. I. Hartley, "In defense of the eight-point algorithm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 6, pp. 580–593, Jun. 1997.
[31] G. H. Golub and C. Reinsch, "Singular value decomposition and least squares solutions," *Numerische Mathematik*, vol. 14, no. 5, pp. 403–420, Apr. 1970.
[32] 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, 2014.
[33] M. I. A. Lourakis, "Sparse non-linear least squares optimization for geometric vision," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2010, pp. 43–56.
[34] W. J. Wilson, C. 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.
[35] 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.
[36] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, U.K.: Univ. Press, 2003.
[37] M. Faessler, E. Mueggler, K. Schwabe, and D. Scaramuzza, "A monocular pose estimation system based on infrared LEDs," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Jun. 2014, pp. 907–913.
[38] F. Paterno, *Model-Based Design and Evaluation of Interactive Applications*. New York, NY, USA: Springer, 1999.
[39] S. Soro and W. Heinzelman, "A survey of visual sensor networks," *Adv. Multimedia*, vol. 21, Mar. 2009, Art. no. 640386.

Heng Deng received the B.S. degree in control science and engineering from the Beijing Institute of Technology, Beijing, China, in 2015. He is currently pursuing the Ph.D. degree in guidance, navigation, and control with the School of Automation Science and Electrical Engineering, Beihang University, Beijing.

His current research interests include vision-based navigation, vision measurement, and unmanned aerial vehicles.

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, Beijing, in 2016.

He is currently a Lecturer with the School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing. His current 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, Beijing, China, in 2004 and 2010, respectively.

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

Kun Yang received the B.S. degree from the School of Control and Computer Engineering, North China Electric Power University, Beijing, China, in 2018. He is currently pursuing the master's degree in guidance, navigation, and control with the School of Automation Science and Engineering, Beihang University, Beijing.

His research interests include visual navigation and unmanned aerial vehicle (UAV) cluster.

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

He has been a Full Professor with 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 current research interests include software testing, software reliability, reliable flight control, and software cybernetics.