International Journal of Automation and Computing

16(6), December 2019, 707-719
DOI: 10.1007/s11633-019-1199-2

# A Survey on 3D Visual Tracking of Multicopters

Qiang Fu[1,2] Xiang-Yang Chen[1,2] Wei He[1,2]

[1] School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing 100083, China

[2] Institute of Artificial Intelligence, University of Science and Technology Beijing, Beijing 100083, China

**Abstract:** Three-dimensional (3D) visual tracking of a multicopter (where the camera is fixed while the multicopter is moving) means continuously recovering the six-degree-of-freedom pose of the multicopter relative to the camera. It can be used in many applications, such as precision terminal guidance and control algorithm validation for multicopters. However, it is difficult for many researchers to build a 3D visual tracking system for multicopters (VTSMs) by using cheap and off-the-shelf cameras. This paper firstly gives an overview of the three key technologies of a 3D VTSMs: multi-camera placement, multi-camera calibration and pose estimation for multicopters. Then, some representative 3D visual tracking systems for multicopters are introduced. Finally, the future development of the 3D VTSMs is analyzed and summarized.

**Keywords:** Multicopter, three-dimensional (3D) visual tracking, camera placement, camera calibration, pose estimation.

## 1 Introduction

Multicopters have been widely used in recent years[1, 2], e.g., in aerial photography, goods transportation and search and rescue. Accurate and robust pose estimation (or motion estimation) of these vehicles is a crucial issue for their autonomous operation. With advantages in the aspects of accuracy, weight, cost, and applicable environment, vision sensors have become a popular choice for providing location (or three-dimensional (3D) tracking) results for multicopters[3, 4].

Note that 3D visual tracking means continuously recovering the six-degree-of-freedom pose of an object relative to the camera (the camera is fixed while the object is moving) or the camera relative to the scene (the scene is fixed while the camera is moving)[5]. Considering that small multicopters often feature CPUs with limited capabilities, this paper focuses on studying the former case. Compared to 3D visual tracking, the traditional 2D visual tracking aims at continuously recovering the size, the centroid or the trajectory of the object in the image[6, 7], but does not involve recovering the 3D position of the object. From the perspective of aims, 3D visual tracking goes further than 2D visual tracking and is more challenging. The relationship between 3D visual tracking and 2D visual tracking is shown in Fig. 1.

Although there are some commercial products for 3D visual tracking, such as Vicon[8] and OptiTrack[9], they are expensive and proprietary. Moreover, these 3D visual tracking systems are not specially designed for multicopters and do not consider the force characteristics of multicopters (the thrust force is perpendicular to the propeller plane). As a result, the robustness of pose estimation for multicopters is limited. Therefore, researchers may want to build their own 3D visual tracking systems for multicopters by using cheap and off-the-shelf cameras.

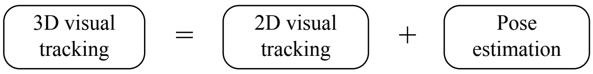

Fig. 1 Relationship between 3D visual tracking and 2D visual tracking

As shown in Fig. 2, we have built a 3D visual tracking system for multicopters by using four MUC36M (MGYYO) infrared cameras[1] equipped with four AZURE-0420MM lenses and four infrared light-emitting diodes (LEDs). These cameras are synchronized by a hardware synchronizer. The markers fixed with the quadrotor do not emit light and just reflect the infrared light emitted by the LEDs so that they will be detected by the cameras. Then, the camera images are transferred to a computer to compute the quadrotor pose. The estimated pose will be sent to another computer to calculate the control command. Based on the above steps, the closed-loop control of the quadrotor is implemented.

Note that there are three key technologies in a 3D VTSMs: 1) Multi-camera placement (how to compute the optimal camera placement off-line); 2) Multi-camera calibration (how to effectively compute the intrinsic and extrinsic parameters of multiple cameras off-line); 3) Pose estimation for multicopters (how to robustly estimate the pose of multicopters on-line). To build a 3D visual track-

Review
Manuscript received December 28, 2018; accepted August 15, 2019
Recommended by Associate Editor Nazim Mir-Nasiri

[1] http://www.catchbest.com/

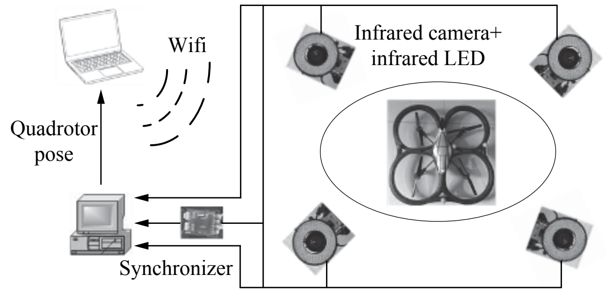

Fig. 2 Structure chart of a 3D VTSMs

ing system for multicopters (VTSMs), researchers need to be familiar with these technologies. The main contribution of this paper is to give an overview of the key technologies. This paper aims to be helpful for researchers who want to build a 3D VTSMs by using cheap and off-the-shelf cameras.

Note that a 3D VTSMs generally consists of multiple fixed cameras because: 1) By fusing the information from multiple cameras, the total field of view (FOV) could be increased and the overall accuracy and robustness of pose estimation could be improved[10]; 2) Compared to onboard vision, using fixed cameras enables the adoption of higher quality imaging devices and more sophisticated vision algorithms (there is no need to consider the constraints related to limited payload and onboard computational capabilities)[11]. The 3D VTSMs could provide accurate six-degree-of-freedom pose of single or multiple multicopters, and it is usually used as a testbed to provide a quick navigation solution for testing and evaluating flight control and guidance algorithms.

This paper is organized as follows. An overview of multi-camera placement methods and multi-camera calibration methods is given in Sections 2 and 3, respectively. Then, in Section 4, a review of pose estimation methods for multicopters is presented, followed by the introduction of some 3D visual tracking systems for multicopters. Finally, challenging issues and conclusions are given in Sections 6 and 7, respectively.

## 2 Multi-camera placement

### 2.1 Camera selection

To build a 3D VTSMs, two types of cameras can be used: visible-light cameras and infrared cameras. The images of visible-light cameras have rich information, but they are sensitive to illumination and do not facilitate marker detection. Therefore, it is better to use infrared cameras (850 nm) together with infrared markers, the same as Vicon and OptiTrack. A sample image of a MUC36M (MGYYO) infrared camera is shown in Fig. 3. By using an infrared camera, infrared markers in Fig. 4 could be easily detected because: 1) The light reflected by the markers lies in the infrared spectrum; 2) The outside light could be minimized by setting the exposure time of the camera to a small value.

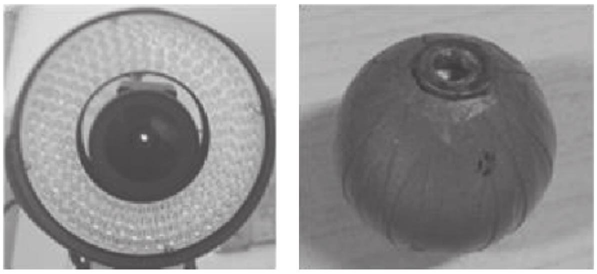

Fig. 3 A sample image of the infrared camera and the infrared marker

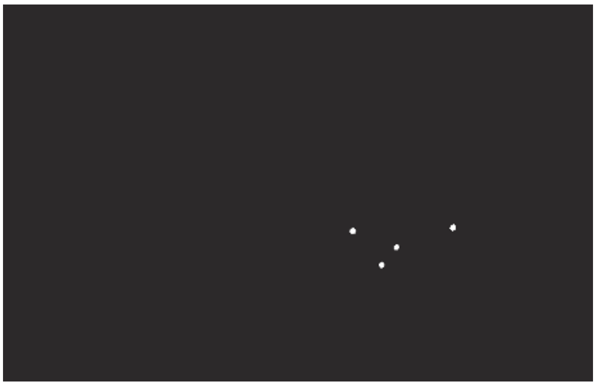

Fig. 4 An image captured by the infrared camera

### 2.2 Existing methods

The placement (position and orientation) of multiple cameras determines the volume of the 3D visual tracking system and 3D reconstruction accuracy of feature points. Therefore, it is very important to optimize the multi-camera placement. The cameras can be placed in the following ways: 1) Attached to the tripods if these cameras are often moved; 2) Attached to the ceiling or a rigid structure if these cameras are rarely moved. The second way is recommended in the real experiments since these camera′s orientation and position are not easy to shift. For the commercial motion-capture systems such as OptiTrack, some multi-camera placement examples are given to the users[12]. For cheap and off-the-shelf cameras, a camera placement example with cameras attached to the ceiling is shown in Fig. 5.

In the literature on stereo-vision reconstruction, the camera placement problem has been studied and related methods can be roughly divided into the following two categories:

1) Generate-and-test methods[13, 14]. These methods, also called trial-and-error methods, are the original methods of solving the camera placement problem. The principle is to first generate the parameters of the cameras, and then estimate them with respect to task constraints. A target-centric grid sphere (see Fig. 6[15]) is usually used to discretize the observation space. The radius of the

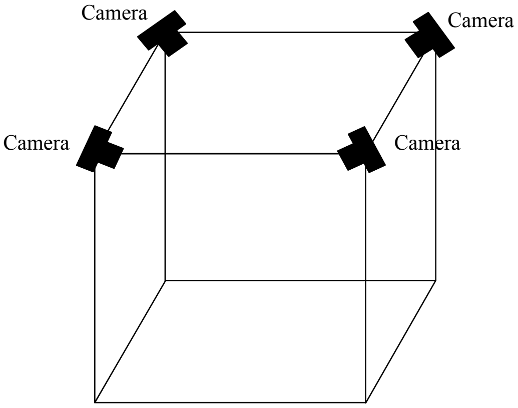

Fig. 5 A camera placement example with cameras attached to the ceiling

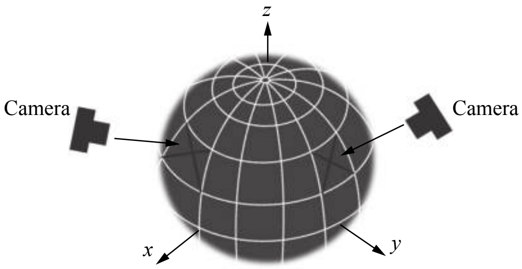

Fig. 6 A target-centric grid sphere used for searching (This figure is recreated from [15])

sphere and the grid resolution are determined by the heuristic search process associated with the task, thus the possible camera viewing positions can be computed in the form of spherical grids. Generate-and-test methods are simple, intuitive and can search the entire sphere. However, calculating and searching the high-dimensional grid parameter space requires a lot of computations, and there is a problem with the grid size ratio (sampling rate).

2) Synthesis methods[16–18]. These methods are also called the constrained optimization methods, which use analytical functions to model constraints (i.e., constructing constraint functions) and task requirements (i.e., constructing the objective function) so that the camera parameters satisfying the constraints can be directly computed. Compared to generate-and-test methods, synthesis methods can be combined with various optimization techniques and actively understand the relationships between the camera parameters to be planned and the task requirements, rather than searching exhaustively.

## 2.3 Discussions

Synthesis methods are now popular for solving the multi-camera placement problem. However, most of the existing synthesis methods focus on optimizing the positioning accuracy of 3D feature points. This is suitable for pose estimation of static rigid objects. But for moving rigid objects, we should not only consider the positioning accuracy of 3D feature points on the rigid objects but also the motion characteristics of rigid objects (including multicopters). In this way, pose estimation accuracy of moving rigid objects (including multicopters) could be further improved.

Note that the camera placement problem still needs to be studied. It is related to many factors, such as the field of view of the camera, the power of the infrared LEDs, the diameter of the marker, etc. Therefore, there is no general solution for the camera placement problem. Based on our experience and the advices given by OptiTrack, the simple and effective method for researchers is to place the cameras uniformly like Fig. 5.

# 3 Multi-camera calibration

## 3.1 Existing methods

Since there are errors in the multi-camera placement in practice and the camera intrinsic parameters are unknown, it is necessary to perform multi-camera calibration to accurately compute the intrinsic parameters (principal point, lens distortion, etc.) and the extrinsic parameters (rotation matrix and translation vector between the camera coordinate system and the reference coordinate system) of each camera. Multi-camera calibration is the basis of pose estimation for multicopters. Pose estimation accuracy of a 3D VTSMs will be determined by the calibration accuracy of multiple cameras directly, so the process of multi-camera calibration is very important. According to the dimension of the calibration object, multi-camera calibration methods can be roughly divided into the following six categories.

### 3.1.1 Three-dimensional calibration methods

Three-dimensional calibration methods require that a 3D calibration object with known geometry in 3D space is used. For example, a calibration object with 3D geometric information known (see Fig. 7) is imaged by a single camera. Note that the 3D calibration object can also be made by using several 2D calibration patterns[19]. Constraint equations are established according to the corresponding relationships between the 3D points of the calibration object and the image points, in order to perform camera calibration. This kind of method can calibrate the intrinsic and extrinsic parameters of multiple cameras simultaneously, but the calibration object is not easy to manufacture and required to be placed in the common field of view of the multiple cameras.

### 3.1.2 Two-dimensional calibration methods

The calibration object commonly used in two-dimensional (2D) calibration methods is a checkerboard pattern with black and white squares (see Fig. 8). Multiple images of the checkerboard pattern are taken from different views, and camera calibration is achieved by establishing constraint equations based on the corresponding relationships between the space points and the image

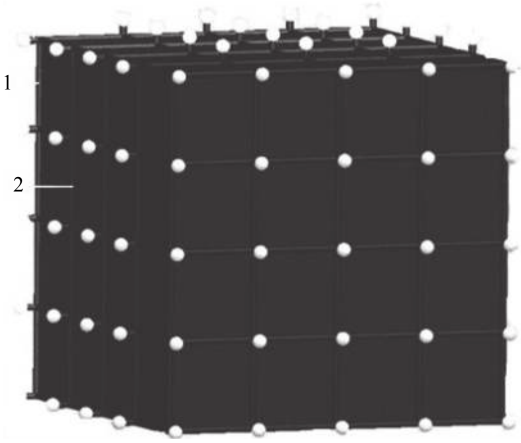

Fig. 7 A sample image of a 3D calibration object

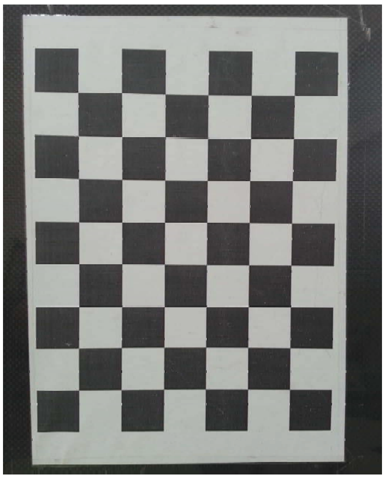

Fig. 8 A sample image of a 2D calibration pattern

points of the planar pattern. This kind of method is easy to use and does not require motion information of the planar pattern. For a monocular camera, the typical calibration method is Zhang′s method[20] that can estimate the intrinsic and extrinsic parameters of the camera with radial distortion. This method requires the camera to take a few (at least two) images of the planar pattern from different orientations, and the intrinsic parameters of the camera are constrained by the homography matrix of each image. Zhang′s method is a two-step method, i.e., first computing the initial values of some parameters linearly, then using the maximum likelihood criterion to optimize the computation results with radial distortion considered. Finally, the extrinsic parameters are obtained by using the camera intrinsic matrix and the homography matrix.

Most of the existing 2D calibration methods for multiple cameras are an extension of Zhang′s method. The problem is that it is not easy for the planar pattern to be observed by all the cameras, so it is difficult to obtain the extrinsic parameters accurately. However, some efforts have been made to solve this problem. As shown in Fig. 9, Theobalt et al.[21] put the planar pattern on the floor in order to make it visible to all the cameras on the ceiling. They chose a corner of the calibration pattern as the origin of the inertial (or world) coordinate system. Then the intrinsic and extrinsic parameters of each camera can be easily obtained by using Zhang′s method.

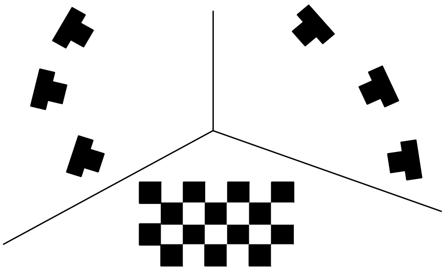

Fig. 9 A sample image of putting a calibration pattern on the floor (This figure is recreated from [21])

If the calibration pattern is not placed to be observed simultaneously by all the cameras, transformations of the extrinsic parameter matrix are needed. The optical center of one camera is usually chosen as the origin of the world coordinate system, and extrinsic parameters of the other cameras are computed relative to the world coordinate system. In order to ensure the accuracy of calibration results, it is necessary to perform a global optimization, or directly establish a multi-camera calibration model[22].

### 3.1.3 One-and-half-dimensional calibration methods

As shown in Fig. 10, Sun et al.[23] used a one-and-half-dimensional (1.5D) calibration object (between one-dimensional calibration object and two-dimensional calibration object) to calibrate the intrinsic and extrinsic parameters of multiple cameras. The 1.5D calibration object has five points in the form of "+", similar with two one-dimensional calibration objects bounded together. In this method, the calibration object moves freely and a linear solution is obtained first. Then, the accuracy of the linear solution is improved by using nonlinear optimization.

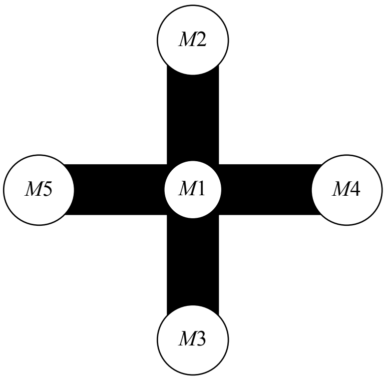

Fig. 10 A sample image of a 1.5D calibration object (This figure is recreated from [23])

However, only simulation experiments are given in the paper.

### 3.1.4 One-dimensional calibration methods

The first one-dimensional (1D) calibration method is proposed by Zhang[24], which uses a calibration object consisting of three or more collinear points with known distances (see Fig. 11). Six or more images of the 1D calibration object are taken to achieve camera calibration. But, this method needs to fix one point, and only allows the 1D calibration object to rotate around the fixed point. To improve the accuracy of Zhang's method[24], Wang et al.[25] proposed a 1D calibration method based on the heteroscedastic error-in-variables (HEIV) model.

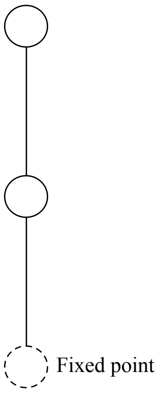

Fig. 11 A sample image of a 1D calibration object (This figure is recreated from [24])

For multiple synchronized cameras, Mitchelson and Hilton[26] proposed a 1D calibration method for calibrating the intrinsic and extrinsic parameters simultaneously, without limiting the motion of the 1D calibration object in the common field of view of all cameras. In this method, stereo calibration is first performed to calculate the initial values of intrinsic and extrinsic parameters, assuming that the principal points of stereo cameras are known or have reasonable values. An iterative bundle adjustment method is then used to optimize the intrinsic and extrinsic parameters of all the cameras. Kurillo et al.[27] studied the problem of initial estimation and global optimization of extrinsic parameters for multiple synchronized cameras with known intrinsic parameters.

In addition, Wang et al.[28, 29] proposed a method to linearly calibrate the intrinsic parameters of multiple synchronized cameras based on a freely-moving 1D calibration object. However, this method requires that the 1D calibration object moves in the common field of view of all the cameras. For synchronized multiple perspective cameras (obeying the pinhole camera model), Fu et al.[30, 31] proposed a calibration method based on a freely-moving 1D calibration object (see Fig. 12). This method can simultaneously compute the intrinsic and extrinsic parameters of each camera and does not need to limit the 1D calibration object moving in the common field of view of all the cameras. They also extend this method to a generic calibration method[32], which is not only suitable for synchronized multiple perspective cameras but also for synchronized multiple fish-eye cameras.

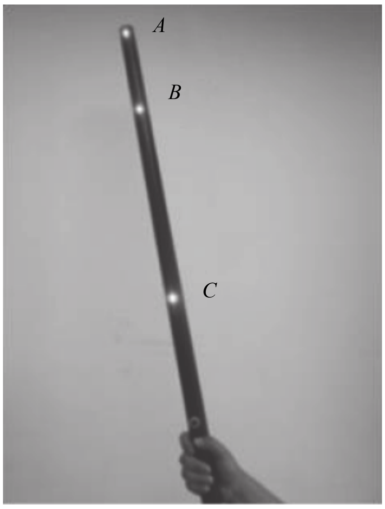

Fig. 12 A sample image of a freely-moving 1D calibration object

### 3.1.5 Point-based calibration methods

Svoboda et al.[33] developed a point calibration toolbox for calibrating the intrinsic and extrinsic parameters of at least three cameras simultaneously. As shown in Fig. 13, the calibration object used is made of a red or green transparent plastic covering a standard laser emitter. The only thing needed is to move the calibration object in the space to be calibrated, and the rest of the work is done automatically by the computer. The calibration object does not need to be observed by all the cameras simultaneously in the process of spatial movement, because in this method there is an algorithm that uses knowledge such as polar line geometry to solve points that cannot be observed. The calibration accuracy is high with about $\frac{1}{5}$ pixel reprojection error, and some researchers have carried out relevant verification work[34].

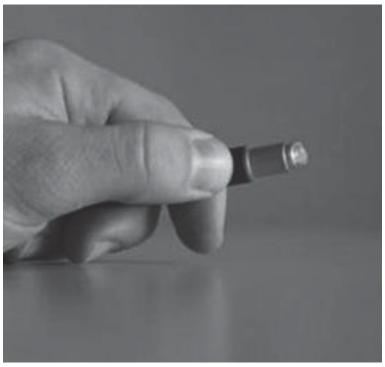

Fig. 13 A sample image of a point calibration object[2]

[2] http://cmp.felk.cvut.cz/svoboda/SelfCal/

The advantage of this method is that the calibration object is simple and the calibration process is highly automatic. The disadvantage is that a relatively dark environment is required so that the calibration object can be easily distinguished from the background.

#### 3.1.6 Self-calibration methods

Self-calibration methods[35–37] usually calibrate the intrinsic and extrinsic parameters of multiple cameras by using the point correspondences among the images without relying on any calibration object. Therefore, they are also called zero-dimensional calibration methods. For example, Bruckner et al.[35] proposed an active self-calibration method for multi-camera systems consisting of pan-tilt zoom cameras, which exploited the rotation information provided by the pan-tilt unit and did not require any artificial calibration object or user interaction. Nguyen and Lhuillier[37] designed a self-calibration method for a moving multi-camera system, which simultaneously estimates intrinsic parameters, inter-camera poses, etc.

Self-calibration methods are more flexible than the other kinds of methods, but these methods are nonlinear and require complex computations without prior knowledge such as geometry information about the scene and motion information about the cameras. The calibration accuracy of these methods is not high (reprojection errors are usually less than 5 pixels). Therefore, self-calibration methods are not suitable for calibrating the cameras in a 3D VTSMs.

### 3.2 Discussions

Comparison of the multi-camera calibration methods mentioned above is shown in Table 1. The calibration accuracy is evaluated by the camera reprojection error. It can be concluded that compared to the other kinds of methods, 1D calibration methods are very suitable for multi-camera calibration due to their advantages such as being simple to manufacture, not requiring common FOV of all cameras and no self-occlusion problem. In addition, a 1D camera calibration toolbox for generic multiple cameras is published as open-source (available at http://rfly.buaa.edu.cn/resources.html) so that other researchers can use the toolbox. However, most of the existing 1D calibration methods are designed for hardware-synchronized multiple cameras. These methods are no longer suitable for unsynchronized cameras (e.g., wired cameras without a hardware synchronizer or wireless cameras). Practical 1D calibration methods for calibrating unsynchronized multiple cameras need to be proposed in the future.

## 4 Pose estimation for multicopters

### 4.1 Existing methods

According to whether there are markers (point markers are commonly-used) on the rigid object, pose estimation methods for rigid objects can be divided into marker-based pose estimation methods and marker-free pose estimation methods. At present, marker-based pose estimation methods are often used, so this section will focus on reviewing the marker-based methods. In computer vision, estimating the pose of a calibrated camera from $n$ 3D-2D point correspondences is known as the Perspective-n-Point (PnP) problem[38]. It is easy to transform the problem of pose estimation for rigid objects into a PnP problem. Therefore, existing marker-based pose estimation methods for rigid objects (including multicopters) can be roughly divided into the following three categories.

#### 4.1.1 Linear methods

Linear methods used to solve the PnP problem had high computation complexity in the early years, but recently there are some linear methods with computation complexity of $O(n)$, which can handle arbitrary point sets. The first $O(n)$ method is EPnP[39, 40] (Efficient Perspective-n-Point), which converts the PnP problem into the problem of solving the 3D coordinates of four control points. It only considers the distance constraints among the four control points, and finally uses a simple linearization method to solve the derived quadratic polynomial. Then some methods with the computation complexity of $O(n)$ have improved the accuracy of EPnP by replacing the linearization method with a polynomial solver. For example, the Direct-Least-Squares (DLS) method[41] establishes a nonlinear objective function and derives a fourth-order polynomial equation, which is solved by the Macauley matrix method[42]. The main disadvantage of the DLS method is that there are singular points in the parameterization of the rotation matrix. In order to solve

Table 1 Comparison of the multi-camera calibration methods

| Method | Number of cameras needed | Common FOV of all cameras | Calibration accuracy (pixel) |
|---|---|---|---|
| 3D calibration | $\geq 1$ | Required | $< 1$ |
| 2D calibration | $\geq 1$ | Required | $< 1$ |
| 1.5D calibration | $\geq 1$ | Not required | – |
| 1D calibration | $\geq 2$ | Not required | $< 1$ |
| Point-based calibration | $\geq 3$ | Not required | 0.2 |
| Self-calibration | $\geq 2$ | Not required | $< 5$ |

this problem, Zheng et al.[43] proposed an optimal Perspective-n-Point (OPnP) method that adopts a quaternion parameterized rotation matrix and solves the polynomial equations based on Grobner basis.

For multiple cameras, Martinez et al.[44, 45] built a real-time vision system consisting of three ground cameras to estimate the pose of a rotary wing unmanned aerial vehicle and then controlled it to achieve some tasks. The pose estimation method they used is a linear 3D reconstruction method[46] based on the perspective imaging model.

Note that the advantage of linear methods is that they are simple and intuitive. The disadvantage is that they are sensitive to noises.

### 4.1.2 Iterative methods

The iterative methods used to solve the PnP problem are to optimize an objective function involving all the point correspondences. The commonly-used objective function is to optimize a geometric error. For example, Faessler et al.[47] built a monocular-vision pose estimation system to control a quadrotor. The pose estimation method they used is the Perspective-3-Point (P3P) algorithm[48] followed by minimizing reprojection errors. In addition to geometric errors, algebraic errors can be used to make the methods more efficient. For example, Lu et al.[49] proposed an orthogonal iterative method for solving the PnP problem, which minimizes the line-of-sight deviations of 3D-2D point correspondences.

Based on the orthogonal iterative method[49], Xu et al.[50] derived a generalized orthogonal iteration algorithm for multiple cameras. In this method, feature points observed by all the cameras can be used. Assa and Janabi-Sharifi[10] proposed a pose estimation method for multiple cameras based on virtual visual servoing (VVS), and designed two fusion structures, namely centralized and decentralized fusion. They pointed out that the centralized fusion structure offers higher accuracy at the cost of increased computation, while the decentralized fusion structure improves the computation speed at the price of lower accuracy.

Compared to linear methods, iterative methods are more accurate and robust, but, they are computationally more intensive than linear methods and prone to fall into local minima.

### 4.1.3 Recursive methods

Recursive methods depend on time filtering methods, especially extended Kalman filter (EKF) methods (the measurement model is nonlinear in the system states due to the camera imaging model). Wilson et al.[51] designed a position-based robot visual servoing control framework using monocular vision, in which the relative pose between the robot and the work piece is computed in real time based on the traditional EKF. The main problem of the traditional EKF method is that it does not perform well when the statistical characteristics of noises change or the initial state estimation is not good. In order to deal with varying noise statistical characteristics, Ficocelli and Janabi-Sharifi[52] proposed an adaptive extended Kalman filter (AEKF) to update the process-noise-covariance matrix. To facilitate the initialization of EKF, Shademan and Janabi-Sharifi[53] proposed an iterated extended Kalman filter (IEKF) for robotic visual servoing applications. In order to deal with poor noise statistical characteristics and poor initial state estimation simultaneously, Janabi-Sharifi and Marey[54] proposed an iterative adaptive extended Kalman filter (IAEKF) method using monocular vision. Then, Assa and Janabi-Sharifi[55] extended the IAEKF method to the multi-camera case to improve the accuracy of pose estimation and the robustness to camera motion and image occlusion.

In addition, Fu et al.[56] proposed a nonlinear constant-velocity process model featured with the characteristics of multicopters. Based on this process model and monocular vision observations, an EKF pose estimation method is designed. Observability analysis shows that this method is more robust to occlusion than the traditional EKF method (only two feature points are needed to achieve the six-degree-of-freedom pose estimation for multicopters), but, this method is not suitable for multiple cameras. For the optical tracking system with four wireless cameras in Fig. 5, Rasmussen et al.[57] proposed an EKF method to fuse the unsynchronized multi-camera observations.

Note that if pose estimation of multicopters is achieved using fitering methods (e.g., Kalman filter), the filter equations can be written as follows:

$$\Sigma_1 : \begin{cases} \dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t)) \\ \boldsymbol{z}(t) = \boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{p}) \end{cases}$$

where $\boldsymbol{x}(t)$ is the state vector including the pose of multicopters and $\boldsymbol{p}$ is the vector whose elements are the placement (position and orientation) parameters of multiple cameras. It is known from $\Sigma_1$ that the placement parameters of multiple cameras can indeed determine the estimation accuracy of the states (including the pose of multicopters). In fact, the state estimation accuracy of $\Sigma_1$ is related to the degree of observability of the system. The degree of observability is used to describe the observability of a linear or nonlinear system quantitatively. The larger the degree of observability is, the higher the accuracy of state estimation would be. It has been applied to solve the problem of sensor placement in many areas, such as aeronautics and astronautics[58, 59], and power systems[60, 61]. Therefore, it is promising to use the degree of observability as a performance index to optimize the placement parameters of multiple cameras.

Note that the commonly-used process model in the system $\Sigma_1$ is a linear constant-velocity process model applicable to many rigid objects[51, 54, 55], which is not a very appropriate model for multicopters. As shown in Fig. 14 (take quadrotors as an example), multicopters have their

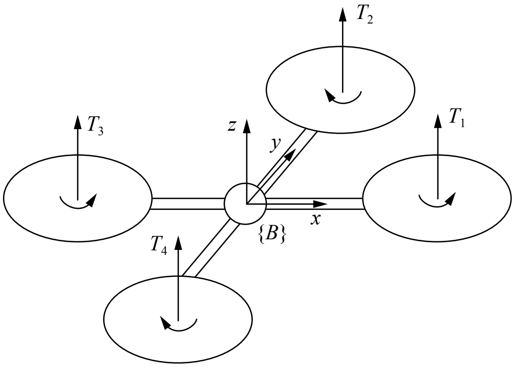

Fig. 14 Force characteristics of quadrotors (This figure is recreated from [1])

own motion characteristics, i.e., they are under-actuated systems with four independent inputs (a thrust force perpendicular to the propeller plane and three moments) and six coordinate outputs[1]. Compared to adopting the linear constant-velocity process model applicable to many rigid objects[51, 54, 55] in the system $\Sigma_1$, it is better to use the nonlinear constant-velocity process model featured with the characteristics of multicopters[56].

### 4.2 Discussions

Comparison of the pose estimation methods for multicopters mentioned above is shown in Table 2. It is found that compared to linear methods and iterative methods, recursive methods are accurate and computationally efficient, and are very suitable for image sequence processing. However, most of the existing recursive methods are designed for general rigid objects and monocular vision. Multicopters are under-actuated systems with four independent inputs (a thrust force perpendicular to the propeller plane and three moments) and six coordinate outputs[1]. Without considering the characteristics of multicopters, the accuracy and robustness of pose estimation for multicopters will be degraded. Therefore, new pose estimation methods based on the process model considering the characteristics of multicopters and synchronized or unsynchronized multiple cameras need to be designed.

Note that the pose estimation results can be sent to the quadrotor by Wifi or Bluetooth communication. The transmission distance and bandwidth of Bluetooth communication are smaller than Wifi communication. However, the power consumption of Wifi communication is higher. The choice of communication depends on the applications.

## 5 3D visual tracking systems for multicopters

### 5.1 Existing systems

There are some representative 3D visual tracking systems for multicopters. The flying machine arena (FMA) is a dual-purpose platform for both research and demonstrations, with fleets of small flying vehicles (mostly quadrotors) at the Swiss Federal Institute of Technology Zurich (ETH Zurich)[62, 63]. The platform is designed similarly to the real-time indoor autonomous vehicle test environment (RAVEN) and the general robotics, automation, sensing and perception (GRASP) testbed at Massachusetts Institute of Technology and the University of Pennsylvania, respectively, where all the agents communicate with a central network consisting of ground-based control computers and the agents. The control computers monitor the states of all the agents, and communicate with them. Based on a motion-capture system consisting of eight 4-megapixel Vicon MX cameras, FMA enables prototyping of new control concepts and implementation of novel demonstrations. It has two versions: a permanent installation version in Zurich with an impressive large size of 10×10×10 $m^3$ and protective netting enclosing the workspace, and a mobile installation version that has been exhibited at some public events. This platform primarily uses the hummingbird quadrotor from ascending technologies as its flight vehicle, but other experimental systems (such as the distributed flight array[64] or the balancing cube[65]) can also be tested in it.

The University of Bologna has developed a multi-agent testbed[66] using an open-source and open-hardware quadrotor, i.e., Crazyflie 1.0[67]. The core elements of the multi-agent testbed are given by the Crazyflie quadrotor, the Optitrack System, the human-machine interface and the ground station. In this testbed, quadrotors can communicate with the ground station computer over Bluetooth. The ground station receives the position information of each quadrotor from a commercial OptiTrack motion capture system (12 infrared cameras are employed). Besides, a human operator can communicate with the ground station using a joystick.

A multi-agent testbed called the Crazyswarm[68] has been developed by the University of Southern California. The testbed adopts the Crazyflie 2.0 quadrotor, which is

Table 2 Comparison of the pose estimation methods for multicopters

| Method | Description | Positioning accuracy |
|---|---|---|
| Linear methods | Simple and intuitive, but sensitive to noises | < 0.5 m |
| Iterative methods | Accurate and robust, but computationally intensive | < 1 cm |
| Recursive methods | Accurate and computationally efficient, and suitable for image sequence processing | < 1 cm |

Table 3 Comparison of some representative 3D visual tracking systems for multicopters

| Name | Institution | Camera | Coverage | Framerate | Camera calibration | Accuracy |
|---|---|---|---|---|---|---|
| FMA | ETH Zurich | Vicon | $10\times10\times10\ m^3$ | 200 Hz | 1D calibration | < 1 mm |
| Multi-agent testbed | University of Bologna | OptiTrack | $4\times4\times2\ m^3$ | 100 Hz | 1D calibration | < 1 mm |
| CrazySwarm | University of Southern California | Vicon | $6\times6\times3\ m^3$ | 75 Hz | 1D calibration | < 20 mm |
| AVRS | Quanser | OptiTrack | $3.5\times3.5\times2\ m^3$ | 120 Hz | 1D calibration | < 1 mm |
| MATRIX | Cranfield University | Firewire CCD | – | 30 Hz | 2D calibration | – |
| – | Beihang University | MUC36M (MGYYO) | $2.5\times2.5\times2\ m^2$ | 40 Hz | 1D calibration | < 1 cm |

the successor to the Crazyflie 1.0 quadrotor used in the multi-agent testbed of University of Bologna. In Crazyswarm, quadrotors communicate with the ground station computer over a Bluetooth radio link with 39 quadrotors on just 3 radios. The pose of the quadrotors is given by a Vicon Vantage motion capture system, which consists of 24 cameras with a working area of $6\times6\times3\ m^3$. However, the tracking software of Vicon is not used because the physical size of the quadrotor makes it difficult to create a lot of different marker arrangements. Instead, a tracking system based on the iterative closest point (ICP) algorithm[69] has been used, which allows every quadrotor to have the same marker arrangements. Reliable flights have been achieved with accurate tracking (less than 2 cm mean position error) by implementing the majority of computation onboard the quadrotor, including sensor fusion, control, and some trajectory planning. The software of Crazyswarm is published as open-source (available at https://github.com/USC-ACTLab/crazyswarm), making this work easily reusable for other researchers.

The Autonomous Vehicles Research Studio (AVRS) developed by Quanser is a good solution for researchers, who want to start a multi-vehicle (quadrotors and ground vehicles) research program in a short time[70]. The quadrotor used is the successor of the QBall 2 and is equipped with powerful on-board Intel Aero Compute Board, multiple high resolution cameras and built-in Wifi capability. AVRS uses a commercial OptiTrack motion-capture system to locate the vehicles. This studio enables researchers to explore topics in advanced flight control, machine vision, simultaneous localization and mapping (SLAM), etc.

There are also some 3D visual tracking systems for multicopters using low-cost off-the-shelf cameras instead of expensive commercial Vicon or Optitrack cameras. For example, the Multi-Agent Test-bed for Real-time Indoor eXperiment (MATRIX) system is developed at Cranfield University to implement the control of an indoor unmanned aerial vehicle (UAV)[71, 72]. It mainly consists of four parts: two firewire charge-coupled device (CCD) cameras, a ground computer, onboard color markers, and quadrotors. Experimental results show that the MATRIX system can provide an accurate and reliable pose estimation so that the pose can be used to control a quadrotor UAV.

In addition, a low-cost 3D visual tracking system for multicopters is developed at Beihang University to implement the indoor control of quadrotors[73]. It consists of three MUC36M (MGYYO) infrared cameras (850 nm), and three infrared LEDs (the power is up to 4 w). Experimental results demonstrate that with the help of this 3D VTSMs, quadrotor hovering and line-tracking control could be achieved.

## 5.2 Discussions

Comparison of the 3D visual tracking systems for multicopters mentioned above is summarized in Table 3. The accuracy in the last column is evaluated by the marker positioning accuracy. It can be found that most of the 3D visual tracking systems are based on commercial motion capture systems (Vicon and OptiTrack), which are proprietary, expensive and not specially designed for multicopters. More effort needs to be put into the research of 3D visual tracking systems for multicopters by using cheap and off-the-shelf cameras. On the other hand, most of the 3D visual tracking systems for multicopters adopte wired and hardware-synchronized cameras. This will not only reduce the system framerate[74] but also make the system layout cumbersome. Therefore, 3D visual tracking systems for multicopters using wireless cameras need to be studied.

# 6 Challenging issues

As mentioned above, one typical trend is to design 3D visual tracking systems for multicopters by using wireless cameras. One attempt made in [57] is to design an EKF method to estimate the head and hand pose of users in a virtual environment. However, the multi-camera placement problem and the multi-camera calibration problem are not discussed in this paper. Therefore, it would be promising to study how to effectively solve the optimal placement problem, the camera calibration problem and the robust pose estimation problem for multicopters when multiple wireless cameras are used.

Another research trend is to design a 3D visual tracking system for multicopters that can be used outdoors. Nowadays most of the 3D visual tracking systems for multicopters can only be used indoors because of the interference of sunlight. This largely limits the application scenarios of these systems. Therefore, it would be important to study how to design an outdoor 3D visual tracking system for multicopters.

The third research trend is to design a 3D visual tracking system for multicopters that allows cameras to move. At present, most of the 3D visual tracking systems require a fixed installation with multiple cameras. This will largely limit the application scenarios. So it would be crucial to study how to design a 3D visual tracking system for multicopters with movable multiple cameras.

## 7 Conclusions

Three-dimensional visual tracking of multicopters plays an important role in the design and development of multicopters. This paper gives an overview of the three key technologies of a 3D visual tracking system for multicopters: multi-camera placement, multi-camera calibration and pose estimation for multicopters. Existing problems and development trends of the 3D visual tracking systems are also pointed out. This paper aims to be helpful for researchers who want to build a 3D visual tracking system for multicopters by using cheap and off-the-shelf cameras.

## Acknowledgements

This work was supported by the National Key Research and Development Program of China (No. 2017YFB1300102) and National Natural Science Foundation of China (No. 61803025).

## Open Access

## References

[1] R. Mahony, V. Kumar, P. Corke. Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor. *IEEE Robotics & Automation Magazine*, vol. 19, no. 3, pp. 20–32, 2012. DOI: 10.1109/MRA.2012.2206474.

[2] D. Scaramuzza, M. C. Achtelik, L. Doitsidis, F. Friedrich, E. Kosmatopoulos, A. Martinelli, M. W. Achtelik, M. Chli, S. Chatzichristofis, L. Kneip, D. Gurdan, L. Heng, G. H. Lee, S. Lynen, M. Pollefeys, A. Renzaglia, R. Siegwart, J. C. Stumpf, P. Tanskanen, C. Troiani, S. Weiss, L. Meier. Vision-controlled micro flying robots: From system design to autonomous navigation and mapping in GPS-denied environments. *IEEE Robotics & Automation Magazine*, vol. 21, no. 3, pp. 26–40, 2014. DOI: 10.1109/MRA.2014.2322295.

[3] F. Zhou, W. Zheng, Z. F. Wang. Adaptive noise identification in vision-assisted motion estimation for unmanned aerial vehicles. *International Journal of Automation and Computing*, vol. 12, no. 4, pp. 413–420, 2015. DOI: 10.1007/s11633-014-0857-7.

[4] W. He, Z. J. Li, C. L. P. Chen. A survey of human-centered intelligent robots: Issues and challenges. *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 4, pp. 602–609, 2017. DOI: 10.1109/JAS.2017.7510604.

[5] V. Lepetit, P. Fua. Monocular model-based 3D tracking of rigid objects. *Foundations and Trends in Computer Graphics and Vision*, vol. 1, no. 1, pp. 1–89, 2005. DOI: 10.1561/0600000001.

[6] Z. Q. Hou, C. Z. Han. A survey of visual tracking. *Acta Automatica Sinica*, vol. 32, no. 4, pp. 603–617, 2006. DOI: 10.16383/j.aas.2006.04.016. (in Chinese)

[7] X. Y. Gong, H. Su, D. Xu, Z. T. Zhang, F. Shen, H. B. Yang. An overview of contour detection approaches. *International Journal of Automation and Computing*, vol. 15, no. 6, pp. 656–672, 2018. DOI: 10.1007/s11633-018-1117-z.

[8] VICON Motion Capture Systems, [Online], Available: https://www.vicon.com/, October 3, 2018.

[9] OptiTrack Motion Capture Systems, [Online], Available: https://www.optitrack.com/, October 3, 2018.

[10] A. Assa, F. Janabi-Sharifi. Virtual visual servoing for multicamera pose estimation. *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 2, pp. 789–798, 2015. DOI: 10.1109/TMECH.2014.2305916.

[11] F. Kendoul. Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems. *Journal of Field Robotics*, vol. 29, no. 2, pp. 315–378, 2012. DOI: 10.1002/rob.20414.

[12] OptiTrack Camera Placement, [Online], Available: http://t.cn/EhrxoJk, October 3, 2018.

[13] S. Sakane, T. Sato. Automatic planning of light source and camera placement for an active photometric stereo system. In *Proceedings of IEEE International Conference on Robotics and Automation*, IEEE, Sacramento, USA, pp. 1080–1087, 1991. DOI: 10.1109/ROBOT.1991.131737.

[14] S. K. Yi, R. M. Haralick, L. G. Shapiro. Optimal sensor and light source positioning for machine vision. *Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 122–137, 1995. DOI: 10.1006/cviu.1995.1009.

[15] J. A. Sun, D. H. Lv, A. P. Song, T. G. Zhuang. A survey of sensor planning in computer vision. *Journal of Image and Graphics*, vol. 6, no. 11, pp. 1047–1052, 2001. DOI: 10.3969/j.issn.1006-8961.2001.11.001. (in Chinese)

[16] G. Olague, R. Mohr. Optimal camera placement for accurate reconstruction. *Pattern Recognition*, vol. 35, no. 4, pp. 927–944, 2002. DOI: 10.1016/S0031-3203(01)00076-0.

[17] X. Chen, J. Davis. An occlusion metric for selecting ro-

bust camera configurations. *Machine Vision and Applications*, vol. 19, no. 4, pp. 217–222, 2008. DOI: 10.1007/s00138-007-0094-y.

[18] P. Rahimian, J. K. Kearney. Optimal camera placement for motion capture systems. *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 3, pp. 1209–1221, 2017. DOI: 10.1109/TVCG.2016.2637334.

[19] J. H. Kim, B. K. Koo. Convenient calibration method for unsynchronized multi-camera networks using a small reference object. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, Vilamoura, Portugal, pp. 438–444, 2012. DOI: 10.1109/IROS.2012.6385605.

[20] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000. DOI: 10.1109/34.888718.

[21] C. Theobalt, M. Li, M. A. Magnor, H. P. Seidel. A flexible and versatile studio for synchronized multi-view video recording. *Vision, Video, and Graphics*, P. Hall, P. Willis, Eds., Aire-la-Ville, Switzerland: Eurographics, pp. 9–16, 2003.

[22] T. Ueshiba, F. Tomita. Plane-based calibration algorithm for multi-camera systems via factorization of homography matrices. In *Proceedings of the 9th IEEE International Conference on Computer Vision*, IEEE, Nice, France, pp. 966–973, 2003. DOI: 10.1109/ICCV.2003.1238453.

[23] B. Sun, Q. He, C. Hu, M. Q. H. Meng. A new camera calibration method for multi-camera localization. In *Proceedings of IEEE International Conference on Automation and Logistics*, IEEE, Hong Kong and Macau, China, pp. 7–12, 2010. DOI: 10.1109/ICAL.2010.5585376.

[24] Z. Y. Zhang. Camera calibration with one-dimensional objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 7, pp. 892–899, 2004. DOI: 10.1109/TPAMI.2004.21.

[25] L. Wang, F. Q. Duan, K. Lv. Camera calibration with one-dimensional objects based on the heteroscedastic error-in-variables model. *Acta Automatica Sinica*, vol. 40, no. 4, pp. 643–652, 2014. DOI: 10.3724/SP.J.1004.2014.00643. (in Chinese)

[26] J. Mitchelson, A. Hilton. Wand-based Multiple Camera Studio Calibration, Technical Report. VSSP-TR-2, Centre for Vision, Speech and Signal Processing, University of Surrey, UK, 2003.

[27] G. Kurillo, Z. Y. Li, R. Bajcsy. Wide-area external multi-camera calibration using vision graphs and virtual calibration object. In *Proceedings of 2nd ACM/IEEE International Conference on Distributed Smart Cameras*, IEEE, Stanford, USA, 2008. DOI: 10.1109/ICDSC.2008.4635695.

[28] L. Wang, F. C. Wu. Multi-camera calibration based on 1D calibration object. *Acta Automatica Sinica*, vol. 33, no. 3, pp. 225–231, 2007. DOI: 10.16383/j.aas.2007.03.001. (in Chinese)

[29] L. Wang, F. C. Wu, Z. Y. Hu. Multi-camera calibration with one-dimensional object under general motions. In *Proceedings of the 11th IEEE International Conference on Computer Vision*, IEEE, Rio de Janeiro, Brazil, pp. 1–7, 2007. DOI: 10.1109/ICCV.2007.4408994.

[30] Q. Fu, Q. Quan, K. Y. Cai. Multi-camera calibration based on freely moving one dimensional object. In *Proceedings of the 30th Chinese Control Conference*, IEEE, Yantai, China, pp. 5023–5028, 2011.

[31] Q. Fu, Q. Quan, K. Y. Cai. Calibration method and experiments of multi-camera's parameters based on freely moving one-dimensional calibration object. *Control Theory & Applications*, vol. 31, no. 8, pp. 1018–1024, 2014. DOI: 10.7641/CTA.2014.31188. (in Chinese)

[32] Q. Fu, Q. Quan, K. Y. Cai. Calibration of multiple fish-eye cameras using a wand. *IET Computer Vision*, vol. 9, no. 3, pp. 378–389, 2015. DOI: 10.1049/iet-cvi.2014.0181.

[33] T. Svoboda, D. Martinec, T. Pajdla. A convenient multicamera self-calibration for virtual environments. *Presence: Teleoperators and Virtual Environments*, vol. 14, no. 4, pp. 407–422, 2005. DOI: 10.1162/105474605774785325.

[34] M. C. Villa-Uriol, G. Chaudhary, F. Kuester, T. Hutchinson, N. Bagherzadeh. Extracting 3D from 2D: Selection basis for camera calibration. In *Proceedings of the 7th IASTED International Conference on Computer Graphics and Imaging*, IASTED, Kauai, USA, pp. 315–321, 2004.

[35] M. Bruckner, F. Bajramovic, J. Denzler. Intrinsic and extrinsic active self-calibration of multi-camera systems. *Machine Vision and Applications*, vol. 25, no. 2, pp. 389–403, 2014. DOI: 10.1007/s00138-013-0541-x.

[36] F. Bajramovic, M. Bruckner, J. Denzler. An efficient shortest triangle paths algorithm applied to multi-camera self-calibration. *Journal of Mathematical Imaging and Vision*, vol. 43, no. 2, pp. 89–102, 2012. DOI: 10.1007/s10851-011-0288-9.

[37] T. T. Nguyen, M. Lhuillier. Self-calibration of omnidirectional multi-cameras including synchronization and rolling shutter. *Computer Vision and Image Understanding*, vol. 162, pp. 166–184, 2017. DOI: 10.1016/j.cviu.2017.08.010.

[38] M. A. Fischler, R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981. DOI: 10.1145/358669.358692.

[39] F. Moreno-Noguer, V. Lepetit, P. Fua. Accurate non-iterative $O(n)$ solution to the PnP problem. In *Proceedings of the 11th IEEE International Conference on Computer Vision*, IEEE, Rio de Janeiro, Brazil, pp. 1–8, 2007. DOI: 10.1109/ICCV.2007.4409116.

[40] V. Lepetit, F. Moreno-Noguer, P. Fua. EP$n$P: An accurate $O(n)$ solution to the PnP problem. *International Journal of Computer Vision*, vol. 81, no. 2, pp. 155–166, 2009. DOI: 10.1007/s11263-008-0152-6.

[41] J. A. Hesch, S. I. Roumeliotis. A direct least-squares (DLS) method for PnP. In *Proceedings of IEEE International Conference on Computer Vision*, IEEE, Barcelona, Spain, pp. 383–390, 2011. DOI: 10.1109/ICCV.2011.6126266.

[42] D. A. Cox, J. Little, D. O'Shea. *Using Algebraic Geometry*, 2nd ed., New York, USA: Springer-Verlag, 2005. DOI: 10.1007/b138611.

[43] Y. Q. Zheng, Y. B. Kuang, S. Sugimoto, K. Astrom, M. Okutomi. Revisiting the PnP problem: A fast, general and optimal solution. In *Proceedings of IEEE International Conference on Computer Vision*, IEEE, Sydney, Australia, pp. 2344–2351, 2013. DOI: 10.1109/ICCV.2013.291.

[44] C. Martinez, P. Campoy, I. Mondragon, M. A. Olivares-Mendez. Trinocular ground system to control UAVs. In

*Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, St. Louis, USA, pp. 3361–3367, 2009. DOI: 10.1109/IROS.2009.5354489.

[45] C. Martinez, I. F. Mondragon, M. A. Olivares-Mendez, P. Campoy. On-board and ground visual pose estimation techniques for UAV control. *Journal of Intelligent & Robotic Systems*, vol. 61, no. 1–4, pp. 301–320, 2011. DOI: 10.1007/s10846-010-9505-9.

[46] R. Hartley, A. Zisserman. *Multiple View Geometry in Computer Vision*, 2nd ed., Cambridge, UK: Cambridge University Press, 2004.

[47] M. Faessler, E. Mueggler, K. Schwabe, D. Scaramuzza. A monocular pose estimation system based on infrared LEDs. In *Proceedings of IEEE International Conference on Robotics and Automation*, IEEE, Hong Kong, China, pp. 907–913, 2014. DOI: 10.1109/ICRA.2014.6906962.

[48] L. Kneip, D. Scaramuzza, R. Siegwart. A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Colorado Springs, USA, pp. 2969–2976, 2011. DOI: 10.1109/CVPR.2011.5995464.

[49] C. P. Lu, G. D. Hager, E. Mjolsness. Fast and globally convergent pose estimation from video images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 6, pp. 610–622, 2000. DOI: 10.1109/34.862199.

[50] Y. X. Xu, Y. L. Jiang, F. Chen. Generalized orthogonal iterative algorithm for pose estimation of multiple camera systems. *Acta Optica Sinica*, vol. 29, no. 1, pp. 72–77, 2009. DOI: 10.3788/AOS20092901.0072. (in Chinese)

[51] W. J. Wilson, C. C. W. Hulls, G. S. Bell. Relative end-effector control using Cartesian position based visual servoing. *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 684–696, 1996. DOI: 10.1109/70.538974.

[52] M. Ficocelli, F. Janabi-Sharifi. Adaptive filtering for pose estimation in visual servoing. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the Next Millennium*, IEEE, Maui, USA, pp. 19–24, 2001. DOI: 10.1109/IROS.2001.973330.

[53] A. Shademan, F. Janabi-Sharifi. Sensitivity analysis of EKF and iterated EKF pose estimation for position-based visual servoing. In *Proceedings of IEEE Conference on Control Applications*, IEEE, Toronto, Canada, pp. 755–760, 2005. DOI: 10.1109/CCA.2005.1507219.

[54] F. Janabi-Sharifi, M. Marey. A Kalman-filter-based method for pose estimation in visual servoing. *IEEE Transactions on Robotics*, vol. 26, no. 5, pp. 939–947, 2010. DOI: 10.1109/TRO.2010.2061290.

[55] A. Assa, F. Janabi-Sharifi. A robust vision-based sensor fusion approach for real-time pose estimation. *IEEE Transactions on Cybernetics*, vol. 44, no. 2, pp. 217–227, 2014. DOI: 10.1109/TCYB.2013.2252339.

[56] Q. Fu, Q. Quan, K. Y. Cai. Robust pose estimation for multirotor UAVs using off-board monocular vision. *IEEE Transactions on Industrial Electronics*, vol. 64, no. 10, pp. 7942–7951, 2017. DOI: 10.1109/TIE.2017.2696482.

[57] N. T. Rasmussen, M. Storring, T. B. Moeslund, E. Granum. Real-time tracking for virtual environments using SCAAT Kalman filtering and unsynchronised cameras. In *Proceedings of the 1st International Conference on Computer Vision Theory and Applications*, Institute for Systems and Technologies of Information, Control and Communication, Setubal, Portugal, pp. 333–341, 2006. DOI: 10.5220/0001367803330340.

[58] Y. X. Wu, H. L. Zhang, M. P. Wu, X. P. Hu, D. W. Hu. Observability of strapdown INS alignment: A global perspective. *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 1, pp. 78–102, 2012. DOI: 10.1109/TAES.2012.6129622.

[59] Z. S. Yu, P. Y. Cui, S. Y. Zhu. Observability-based beacon configuration optimization for Mars entry navigation. *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 4, pp. 643–650, 2015. DOI: 10.2514/1.G000014.

[60] J. J. Qi, K. Sun, W. Kang. Optimal PMU placement for power system dynamic state estimation by using empirical observability Gramian. *IEEE Transactions on Power Systems*, vol. 30, no. 4, pp. 2041–2054, 2015. DOI: 10.1109/TPWRS.2014.2356797.

[61] K. Sun, J. J. Qi, W. Kang. Power system observability and dynamic state estimation for stability monitoring using synchrophasor measurements. *Control Engineering Practice*, vol. 53, pp. 160–172, 2016. DOI: 10.1016/j.conengprac.2016.01.013.

[62] S. Lupashin, A. Schollig, M. Sherback, R. D'Andrea. A simple learning strategy for high-speed quadrocopter multi-flips. In *Proceedings of IEEE International Conference on Robotics and Automation*, IEEE, Anchorage, USA, pp. 1642–1648, 2010. DOI: 10.1109/ROBOT.2010.5509452.

[63] S. Lupashin, M. Hehn, M. W. Mueller, A. P. Schoellig, M. Sherback, R. D'Andrea. A platform for aerial robotics research and demonstration: The flying machine arena. *Mechatronics*, vol. 24, no. 1, pp. 41–54, 2014. DOI: 10.1016/j.mechatronics.2013.11.006.

[64] R. Oung, R. D'Andrea. The distributed flight array. *Mechatronics*, vol. 21, no. 6, pp. 908–917, 2011. DOI: 10.1016/j.mechatronics.2010.08.003.

[65] S. Trimpe, R. D'Andrea. Accelerometer-based tilt estimation of a rigid body with only rotational degrees of freedom. In *Proceedings of IEEE International Conference on Robotics and Automation*, IEEE, Anchorage, USA, pp. 2630–2636, 2010. DOI: 10.1109/ROBOT.2010.5509756.

[66] M. Furci, G. Casadei, R. Naldi, R. G. Sanfelice, L. Marconi. An open-source architecture for control and coordination of a swarm of micro-quadrotors. In *Proceedings of International Conference on Unmanned Aircraft Systems*, IEEE, Denver, USA, pp. 139–146, 2015. DOI: 10.1109/ICUAS.2015.7152285.

[67] The Crazyflie 1.0, [Online], Available: https://www.bitcraze.io/crazyflie/, October 3, 2018.

[68] J. A. Preiss, W. Honig, G. S. Sukhatme, N. Ayanian. Crazyswarm: A large nano-quadcopter swarm. In *Proceedings of IEEE International Conference on Robotics and Automation*, IEEE, Singapore, Singapore, pp. 3299–3304, 2017. DOI: 10.1109/ICRA.2017.7989376.

[69] P. J. Besl, N. D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992. DOI: 10.1109/34.121791.

[70] Autonomous Vehicles Research Studio, [Online], Available: https://www.quanser.com/products/autonomous-vehicles-research-studio/, October 3, 2018.

[71] H. Oh, D. Y. Won, S. S. Huh, D. H. Shim, M. J. Tahk, A. Tsourdos. Indoor UAV control using multi-camera visual feedback. *Journal of Intelligent & Robotic Systems*, vol. 61, no. 1–4, pp. 57–84, 2011. DOI: 10.1007/s10846-010-9506-8.

[72] D. Y. Won, H. Oh, S. S. Huh, D. H. Shim, M. J. Tahk. Multiple UAVs tracking algorithm with a multi-camera system. In *Proceedings of International Conference on Control Automation and Systems*, IEEE, Gyeonggi-do, South Korea, pp. 2357–2360, 2010. DOI: 10.1109/ICCAS.2010.5669931.

[73] Q. Fu. Research on Robust 3D Visual Tracking of Multirotor Aerial Vehicles, Ph. D. dissertation, Beihang University (formerly Beijing University of Aeronautics and Astronautics), Beijing, China, 2016. (In Chinese)

[74] A. Elhayek, C. Stoll, N. Hasler, K. I. Kim, H. P. Seidel, C. Theobalt. Spatio-temporal motion tracking with unsynchronized cameras. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Providence, USA, pp. 1870–1877, 2012. DOI: 10.1109/CVPR.2012.6247886.

**Qiang Fu** received the B. Sc. degree in thermal energy and power engineering from Beijing Jiaotong University, China in 2009, and the Ph. D. degree in control science and engineering from Beihang University (formerly Beijing University of Aeronautics and Astronautics), China in 2016. He is currently a lecturer in the School of Automation and Electrical Engineering, University of Science and Technology Beijing, China.

His research interests include vision-based navigation and 3D vision.

E-mail: fuqiang@ustb.edu.cn (Corresponding author)

ORCID iD: 0000-0003-0665-4956

**Xiang-Yang Chen** received the B. Sc. degree in the electrical engineering and automation from Soochow University, China in 2017. He is currently a master student in control engineering at the School of Automation, University of Science and Technology Beijing, China.

His research interests include flapping-wing aircraft and machine vision.

E-mail: chenxy0406@outlook.com

**Wei He** received the B. Eng. degree in automation and the M. Eng. degree in control science and engineering, both from College of Automation Science and Engineering, South China University of Technology (SCUT), China in 2006 and 2008, respectively, and the Ph. D. degree in control theory and control engineering from Department of Electrical & Computer Engineering, the National University of Singapore (NUS), Singapore in 2011. He is currently working as a full professor in School of Automation and Electrical Engineering, University of Science and Technology Beijing, China. He has co-authored 2 books published in Springer and published over 100 international journal and conference papers. He has been awarded a Newton Advanced Fellowship from the Royal Society, UK. He is a recipient of the IEEE SMC Society Andrew P. Sage Best Transactions Paper Award in 2017. He is the Chair of IEEE SMC Society Beijing Capital Region Chapter. He serves as an Associate Editor of *IEEE Transactions on Neural Networks and Learning Systems, IEEE Transactions on Control Systems Technology, IEEE Transactions on Systems, Man, and Cybernetics: Systems, IEEE/CAA Journal of Automatica Sinica, Neurocomputing*, and an Editor of *Journal of Intelligent & Robotic Systems*. He is the member of the International Federation of Automatic Control Technical Committee (IFAC TC) on Distributed Parameter Systems, IFAC TC on Computational Intelligence in Control and IEEE Control Systems Society (CSS) TC on Distributed Parameter Systems.

His research interests include robotics, distributed parameter systems and intelligent control systems.

E-mail: weihe@ieee.org