SCIENCE CHINA
Technological Sciences

• Article •

November 2016 Vol.59 No.11: 1730–1738
doi: 10.1007/s11431-016-6097-z

Verification of monocular and binocular pose estimation algorithms in vision-based UAVs autonomous aerial refueling system

LI Han & DUAN HaiBin*

Bio-inspired Autonomous Flight Systems (BAFS) Research Group, Science and Technology on Aircraft Control Laboratory, School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China

Received January 21, 2016; accepted May 18, 2016; published online September 18, 2016

With the high focus on autonomous aerial refueling (AAR), it becomes increasingly urgent to design efficient methods or algorithms for solving the AAR problems in complicated aerial environments. A vision-based technology for AAR is developed in this paper, and five monocular and binocular visual algorithms for pose estimation of the unmanned aerial vehicles (UAVs) are adopted and verified in this AAR system. The real-time on-board vision system is also designed for precise navigation in the UAVs docking phase. A series of out-door comparative experiments for different pose estimation algorithms are conducted to verify the feasibility and accuracy of the vision algorithms in AAR.

unmanned aerial vehicles (UAVs), autonomous aerial refueling (AAR), pose estimation, monocular vision, binocular vision

Citation: Li H, Duan H B. Verification of monocular and binocular pose estimation algorithms in vision-based UAVs autonomous aerial refueling system. Sci China Tech Sci, 2016, 59: 1730–1738, doi: 10.1007/s11431-016-6097-z

1 Introduction

Unmanned aerial vehicles (UAVs) including rotorcrafts and fixed wing aircrafts have been highly focused on for their various applications in both civilian and military domains [1–3]. Since the rotorcrafts have the advantage of hovering, vertical take-off and landing, the tasks of aerial photography in archaeology [4], topography [5], forest fire detection [6], and geography [7] are easier for them to execute. Generally, such tasks would cost a large deal of labor. In the military domain, missions like surveillance [8] and air combat [9] are more likely to be executed by UAVs.

With the extensive use of UAVs, a problem attracted people's attention: the confined fuel limited the operational radius. Under this circumstance, autonomous aerial refueling (AAR) appears to be especially important. One of the approaches of aerial refueling is that the tanker aircraft and receiver aircraft get close to each other, and then the boom from the tanker aircraft approaches the receiver aircraft and precisely sticks into the refueling port on the receiver aircraft [10]. The prerequisite of accomplishing this mission is to obtain the accurate pose information between the UAVs. Usually, global positioning system (GPS) [11] and inertial navigation system (INS) are used in navigation, while cameras become an important sensor for UAVs [12,13]. In some cases, visual sensors [14–18] and geo-referenced images [19] can be utilized. To detect the drogue, the VisNav sensor is exploited to supervise the beacons mounted on the drogue [15]. The 3D flash lidar is also used for drogue tracking [16]. In ref. [17], the vision-based measurements, inertial and barometric sensor data and global position system (GPS) are fused with unscented Kalman filter (UKF). For boom approach, a visual snakes-based vision sensor is proposed in ref. [18]. Thus, it has been accepted that com-

---

*Corresponding author (email: hbduan@buaa.edu.cn) puter vision is a considerable method for docking phase navigation in the mission of AAR [20].

However, AAR is a truly risky mission, thus the accuracy of visual algorithm is required to be extraordinary high. Hence, both the ground tests and flight tests are necessary. Under this circumstance, a verification system is built and multiple visual algorithms are prepared and operated to fulfill the mission.

A system based on two UAVs is introduced in this paper. The system is used to verify the on-board real-time vision system, which serves for AAR. Moreover, relative control and communication technology is also included in this platform. In this work, an octo-rotor is used as the tanker UAV and a quad-rotor is used as the receiver UAV. An extension-type rod is attached to the tanker UAV served as the boom. Both tanker and receiver UAVs have GPS and Inertial Measurement Unit (IMU) on them to obtain their own flight status. Flight controller certainly is a main part of the UAVs. Particularly, a vision system includes two cameras and an on-board computer is set on the tanker UAV, which is exploited to operate visual measurement. Xbee and image transmitter are utilized to communicate among tanker UAV, receiver UAV and ground station. The binocular cameras on the tanker UAV capture images of the point makers on the receiver UAV. Then the on-board computer processes the images to extract features and estimates the pose of the receiver UAV. The pose information will be transmitted to the control system and used to control the flight status of the UAVs. Besides, boom control also depends on the information from vision system. Figure 1 is a picture of the experimenting platform. The configuration of the platform is shown in Figure 2.

This paper is organized as follows: Section 2 presents the Hardware system of the platform in detail. The feature extraction and pose estimation done with visual system are described in Section 3. Experimental results and comparisons are shown in Sections 4 and 5 gives the conclusion.

Figure 1 (Color online) Experimenting platform. (a) A photo of the flying UAVs; (b) magnified photo of the UAVs close to each other.

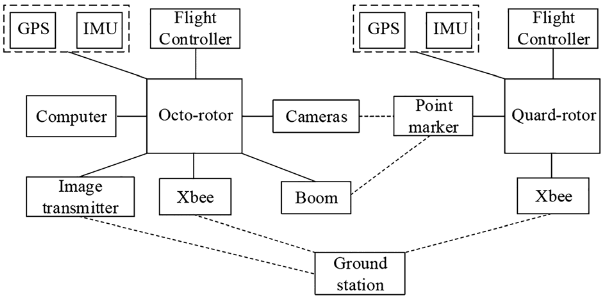

Figure 2 Configuration of the platform.

2 System design

The hardware system of our verification platform consists of three main parts: two UAVs and a visual sensor on the receiver UAV; control and vision processing module; the communication module among two UAVs and UAVs and ground.

2.1 UAV Platform and cameras

The octo-rotor S1000 from DJI technology Inc. is used as the tanker UAV. Its maximum takeoff weight is about 11 kg, and its diameter with eight arms spread is about 150 cm. The 15000 mAh LiPo battery on it can support the octo-rotor fly for 10 to 20 minutes. The receiver UAV is designed based on the quad-rotor X650 pro from XAIRCRAFT Technology Co. Ltd. Human pilot can control the UAVs via a radio control (RC) transmitter, while the simultaneous control signal from on-board computer is the chief command.

The visual sensor on board consists of a binocular camera system including two color video cameras. The camera has a compact size equipped with Mini USB 2.0 interface and has a gross weight around 100 g using a 12 mm lens. Its view field is 40 degree and resolution is up to 1292×964 pixels. While the vision algorithms compute at a rate of 10 frames per second (FPS), the frame rate can reach 30 FPS.

2.2 Control and vision processing module

Since the image processing algorithm occupies a great deal of memory, a separated mini PC, Intel NUC is employed to process the visual information. The PC is equipped with a 4th generation Intel Core i5 processor running at 1.3 GHz and a compact solid-state disk. The total weight is around 650 g. All our visual algorithms such as feature extraction, pose estimation run under Linux on this computer. The communication between vision system with flight controller and vision system with ground station is also coordinated on this computer.

Our ground station is a LENOVO ThinkPad PC, which is employed to display the video captured by the on-board cameras and the position and pose data computed by the vision system and GPS together with IMU. With the ground station, it is easy to monitor the status of the UAVs and to check the image captured by the cameras.

2.3 Communication module

Although the video captured by the on-board camera is directly processed with the on-board computer, the processing result should also be transmitted and displayed in the ground station using the DJI Lightbridge 2.4 G HD digital video downlink. The device can offer 1080p video data transmission from up to 1.7 km away. The air system is connected to the on-board computer by high-definition multimedia interface (HDMI). When transmitting the position and pose data, a pair of Xbee Modules from Digi International are employed based on user data protocol in our system.

The sketch map of the system including subcomponent and connection of the systems is shown in Figure 3.

3 Feature extraction and pose estimation

Generally, image processing procedure includes two major steps: feature extraction and pose estimation. Feature extraction aims to generate the coordinates measured in pixels of the labeled markers. Then, the pose estimation algorithm is employed to calculate the relation matrix between the cameras and the marker coordinate system. Camera calibration should be conducted first to obtain the initial rotation matrix $\boldsymbol{R}$ and translation vector t between different coordinates. The original $\boldsymbol{R}$ and t are given in Table 1. The flowchart of the processing procedure is shown in Figure 4.

3.1 Feature extraction

Feature extraction is the premise of pose estimation including two main procedures, marker detection and point matching. The accuracy of feature extraction has a direct influence on pose estimation, which may lead to the instability of the whole system including the control module.

3.1.1 Marker detection

When the two UAVs are far away from each other, GPS and

Table 1 Initial value of $\boldsymbol{R}$ and t

	Initial value		
	1	0	0
$\boldsymbol{R}$	0	0.7547	0.6561
	0	−0.65	0.75
t (cm)	219.95	7.23	113.11

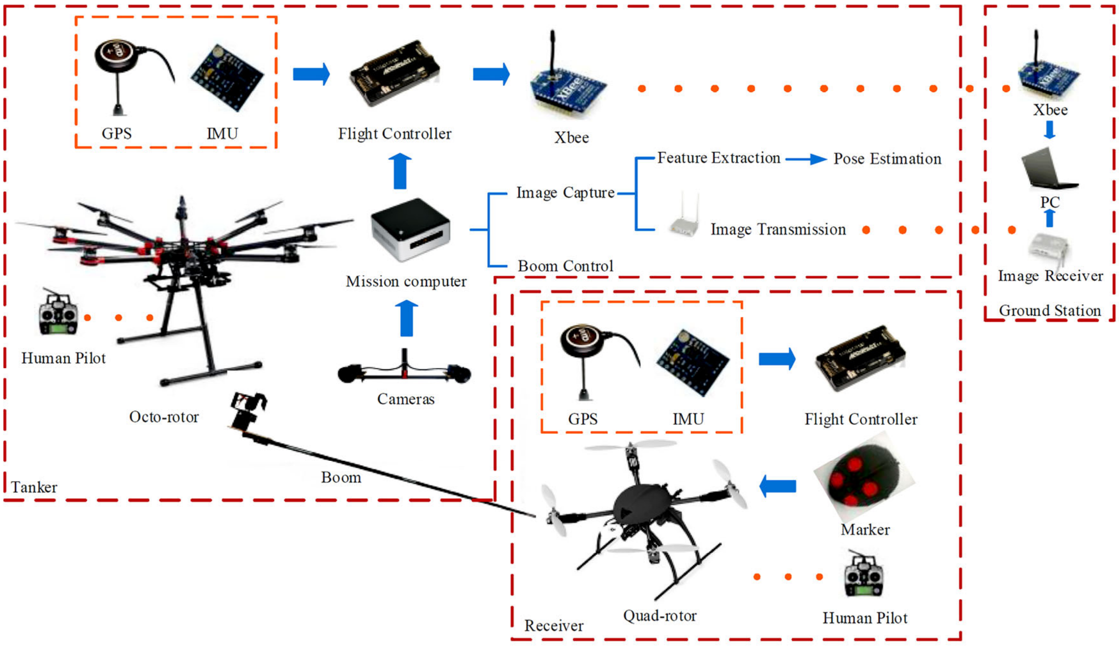

Figure 3 (Color online) Sketch map of the system including subcomponent and connections.

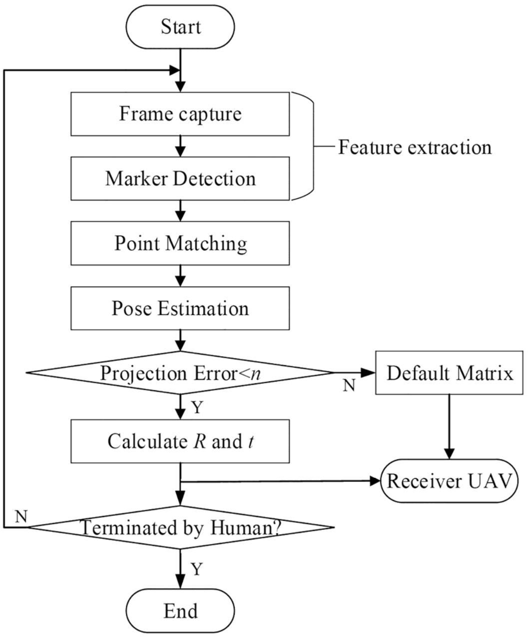

Figure 4 Flowchart of the processing procedure.

IMU are employed to lead the two UAVs getting closer. The marker detection process is implemented when the receiver UAV approaches to the distance where the markers are capable to be viewed clearly. Red markers are set on the receiver UAV, and the color feature is the main characteristic for feature extraction.

Since the resolution of the cameras is high, the captured images are shrunk by half to speed up the following computation. HSV space is always used in image processing for its well performance in describing color features. Thus, the captured images are mapped from RGB space to HSV space. Only the hue and saturation channels are selected to do the threshold segmentation and obtain binary images. Then the morphology methods such as erosion and dilation operators are used to eliminate undesired noise. Connected components from every binary image are extracted and their centers are calculated and grouped by their coordinates. Close centers form one group that corresponds to one blob, which is controlled by a set of parameters. From the groups, we estimate the final centers of blobs and their radiuses, then return them as locations and sizes of marker points. Several filtrations including color, area, and circularity are performed during estimating of blobs.

3.1.2 Point Matching

After the feature extraction procedure, the coordinates of the makers are achieved. However, in order to estimate pose, the detected makers need to be matched with the corresponding actual makers. The detected points are reprojected to the image according to prior pose information. The Euclidean distance matrix of the detected points set $\{p_1, p_2, \cdots, p_n\}$ and reprojected points set $\{\hat{p}_1, \hat{p}_2, \cdots, \hat{p}_m\}$ is denoted as

$$\boldsymbol{Err} = \begin{bmatrix} d(\hat{p}_1, p_1) & d(\hat{p}_1, p_2) & \cdots & d(\hat{p}_1, p_n) \\ d(\hat{p}_2, p_1) & d(\hat{p}_2, p_2) & \cdots & d(\hat{p}_2, p_n) \\ \vdots & \vdots & \vdots & \vdots \\ d(\hat{p}_m, p_1) & d(\hat{p}_m, p_2) & \cdots & d(\hat{p}_m, p_n) \end{bmatrix}. \tag{1}$$

Munkres algorithm [21] is employed to calculate the Euclidean distance matrix of two points sets.

3.2 Pose estimation

Pose estimation calculates the geometry relationship of the actual markers' position and their position in the images. Then the relative position and pose of the tanker UAV and receiver UAV are known as the rotation matrix $\boldsymbol{R}$ (describing the rotation relationship between the camera coordinate and receiver UAV coordinate) and translation vector t (describing the translation relationship between the camera coordinate and receiver UAV coordinate).Both monocular and binocular visions are used in our experiment.

3.2.1 Monocular vision

1) The Efficient Perspective-n-Point (EPnP) algorithm. The EPnP algorithm used in our experiment was proposed as a non-iterative solution to the Perspective-n-Point (PnP) problem by Lepetit et al. [22] in 2008. The world coordinates of the 3D scene points $\boldsymbol{X}_i$ are known and the points are expressed as a weighted sum of four virtual and non-coplanar control points $\boldsymbol{C}_j$. That is

$$\boldsymbol{X}_i = \sum_{j=1}^{4} \alpha_{ij} \boldsymbol{C}_j, \tag{2}$$

where α_{ij} are homogeneous barycentric coordinates. Thus in the world coordinate system, both the points $\boldsymbol{X}_i^W$ and the control points $\boldsymbol{C}_j^W$ are known. Consequently, the points $\boldsymbol{X}_i^C$ can be expressed as

$$\boldsymbol{w}_i \begin{bmatrix} \boldsymbol{X}_i \\ i \end{bmatrix} = \boldsymbol{K}\boldsymbol{X}_i^C = \boldsymbol{K} \sum_{j=1}^{4} \alpha_{ij} \boldsymbol{C}_j^C, \tag{3}$$

where w_i are the projective parameters and $\boldsymbol{K}$ means the camera matrix. Therefore, the expression of 3D points is

$$\boldsymbol{w}_i \begin{bmatrix} \boldsymbol{x}_i \\ \boldsymbol{y}_i \\ \boldsymbol{1} \end{bmatrix} = \begin{bmatrix} f_x & 0 & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \sum_{j=1}^{4} \boldsymbol{\alpha}_{ij} \begin{bmatrix} \boldsymbol{X}_j^C \\ \boldsymbol{Y}_j^C \\ \boldsymbol{Z}_j^C \end{bmatrix}, \tag{4}$$

where f_x and f_y are the focal length coefficients, x_0 and y_0 are the coordinates of the principal points and $\begin{bmatrix} \boldsymbol{X}_j^C & \boldsymbol{Y}_j^C & \boldsymbol{Z}_j^C \end{bmatrix}^{\mathrm{T}}$ are the coordinates of the control points. The solution of this linear system leads to the camera coordinates X_i^C of the 3D points. Namely both the world coordinates and the camera coordinates of the 3D points are known, hence the rotation and translation parameter can be retrieved.

2) The Gaussian Least Square Differential Correction (GLSDC). The problem of pose estimation is always converted to a problem of nonlinear optimization problem, thus nonlinear least square method is used to iterate and solve the problem. The GLSDC algorithm based on the method of Gauss-Newton method has been proposed by Haralick et al. [23].

The estimation $\bar{X}(k)$ of the unknown vector $X(k)$ is updated according to the iteration equation:

$$\bar{X}_{i+1}(k)=\bar{X}_i(k)+R_i^{-1}(k)A_i^{\mathrm{T}}(k)W(k)\Delta G_i(k), \tag{5}$$

where i represents the count of iteration.

The iteration will be terminated until the maximum iteration count is reached or the error is smaller than the set up criterion. Then the rotation matrix and translation vector can be achieved.

3.2.2 *Binocular vision*

1) Lu et al. [24] proposed the algorithm LHM in 2000, which is to formulate pose estimation problem by minimizing an error metric based on collinearity in object space.

Assume that the coordinate of marker point j is given as p_j. From eq. (11), it is obvious that to estimate the optimal translation vector t, the rotation matrix $\boldsymbol{R}$ need to be calculated first. Suppose $Y_j(k)(\boldsymbol{R}_K p_j+t_k)=q_j$, the collinearity error can be defined as

$$\boldsymbol{E}(\boldsymbol{R},T)=\sum_{j=1}^{m}\left\|\boldsymbol{R}p_j+T-q_j\right\|^2. \tag{6}$$

Therefore, the problem is converted to the pose calculation between the known points set $\{p_j\}$ and $\{q_j\}$.Give $\bar{p}=\frac{1}{n}\sum_{j=1}^{n}\{p_j\}$ and $\bar{q}=\frac{1}{n}\sum_{j=1}^{n}\{q_j\}$. Define $p_j'=p_j-\bar{p}$, $q_j'=q_j-\bar{q}$, and $\boldsymbol{M}=\sum_{j=1}^{n}{q_j'}^{\mathrm{T}}P_j'$. After singular value decomposition is done with $\boldsymbol{M}$, it can be obtained that

$$\boldsymbol{M}=\boldsymbol{X}\Sigma\boldsymbol{Y}^{\mathrm{T}}. \tag{7}$$

Consequently

$$\boldsymbol{R}=\boldsymbol{Y}S\boldsymbol{X}^{\mathrm{T}}, \tag{8}$$

where

$$S=\begin{cases} I, & \det\left(\boldsymbol{Y}\boldsymbol{X}^{\mathrm{T}}\right)=1,\\ \operatorname{diag}(1,1,-1), & \det\left(\boldsymbol{Y}\boldsymbol{X}^{\mathrm{T}}\right)=-1. \end{cases} \tag{9}$$

2) Least squares algorithm. This algorithm is a basic and direct method for pose estimation, which calculates the pose information depending on the measured pixel coordinates of the marker points. Suppose that the normalized coordinates of the maker points in camera coordinate system are $\boldsymbol{P}_i^c=\left[\boldsymbol{u}_i^c \quad \boldsymbol{v}_i^c \quad \boldsymbol{1}\right]^{\mathrm{T}}$, in image coordinate system are $\boldsymbol{P}_{ji}{}^c=\left[\boldsymbol{u}_{ji}^c \quad \boldsymbol{v}_{ji}^c \quad \boldsymbol{1}\right]^{\mathrm{T}}$, in machine coordinate system are, $\boldsymbol{P}_i^m=\left[\boldsymbol{X}_i \quad \boldsymbol{Y}_i \quad \boldsymbol{Z}_i\right]^{\mathrm{T}}$ and in virtual camera coordinate system are $\boldsymbol{P}_i=\left[\boldsymbol{x}_i \quad \boldsymbol{y}_i \quad \boldsymbol{z}_i\right]^{\mathrm{T}}$.

The relation of the rotation matrix and translation vector of the marker points to virtual camera system (***R***, *T*), left camera to virtual camera system ($\boldsymbol{R}_1$, T_1) and right camera to virtual camera system ($\boldsymbol{R}_2$, T_2) can be obtained with

$$\begin{cases} \boldsymbol{P}_i=\boldsymbol{R}\boldsymbol{P}_i^m+T,\\ \boldsymbol{P}_i^1=\boldsymbol{R}_1\boldsymbol{P}_i+T_1,\\ \boldsymbol{P}_i^2=\boldsymbol{R}_2\boldsymbol{P}_i+T_2. \end{cases} \tag{10}$$

From eq. (16), it is known that

$$\begin{aligned} \boldsymbol{P}_{ji}{}^c=\boldsymbol{K}_j\boldsymbol{P}_i^c&=\boldsymbol{K}_j\left(\boldsymbol{R}_j\boldsymbol{P}_i+T_j\right)=\boldsymbol{K}_j\left[\boldsymbol{R}_j\left(\boldsymbol{R}\boldsymbol{P}_i^m+T\right)+T_j\right]\\ &=\boldsymbol{K}_j\boldsymbol{R}_j\boldsymbol{R}\boldsymbol{P}_i^m+\boldsymbol{K}_j\boldsymbol{R}_jT+\boldsymbol{K}_jT_j. \end{aligned} \tag{11}$$

Meanwhile, (***R***, *T*) can be calculated from eq. (17).

3) A solution of absolute orientation involving orthonormal matrices. This algorithm was developed by Horn et al. [25]. Since the algorithm supposes that the coordinates of one point in different coordinate systems are known, set the left coordinate system as the camera system and the right coordinate system as the virtual camera system. The coordinates of the maker points in camera system are known as $\boldsymbol{P}_i^C=\left[\boldsymbol{X}_i \quad \boldsymbol{Y}_i \quad \boldsymbol{Z}_i\right]^{\mathrm{T}}$, and the corresponding coordinates in virtual camera system are $\boldsymbol{P}_i=\left[\boldsymbol{x}_i \quad \boldsymbol{y}_i \quad \boldsymbol{z}_i\right]^{\mathrm{T}}$, which can be derived from the least squares algorithm.

The coordinates of three given noncollinear points in the left and right coordinate systems are known as $r_{l,1}$, $r_{l,2}$, $r_{l,3}$ and $r_{r,1}$, $r_{r,2}$, $r_{r,3}$. According to ref. [24], the new left and right coordinate systems can be set as $\hat{x}_l \quad \hat{y}_l \quad \hat{z}_l$ and $\hat{x}_r \quad \hat{y}_r \quad \hat{z}_r$. Define matrixes $\boldsymbol{M_l}$ and $\boldsymbol{M_r}$ as $\boldsymbol{M}_l=\left[\hat{\boldsymbol{x}}_l \quad \hat{\boldsymbol{y}}_l \quad \hat{\boldsymbol{z}}_l\right]$, $\boldsymbol{M}_l=\left[\hat{\boldsymbol{x}}_r \quad \hat{\boldsymbol{y}}_r \quad \hat{\boldsymbol{z}}_r\right]$. Hence the rotation matrix between the two coordinate systems is $\boldsymbol{R}=\boldsymbol{M}_r\boldsymbol{M}_l^{\mathrm{T}}$. The coordinates sets of n points in the left and right coordinate systems are $\{r_{l,n}\}$ and $\{r_{r,n}\}$. One of the points in the right coordinate system can be denoted as

$$r_{r,i}=s\boldsymbol{R}r_{l,i}+T, \tag{12}$$

where s is a scale factor, T is the translational offset, and $\boldsymbol{R}r_{l,i}$ denotes the rotated vector $r_{l,i}$. Since the data are not perfect, a residual error exists

$$e_i = r_{r.i} - \frac{1}{n}\sum r_{r,i} - s\boldsymbol{R}\left(r_{l.i} - \frac{1}{n}\sum r_{l,i}\right) - T. \tag{13}$$

The scale factor s which minimizes the sum of the squares of these errors $\sum_{i=1}^{n}\left\|e_i\right\|^2$ is

$$s = \frac{\sum r'_{r,i}\boldsymbol{R}r'_{l,i}}{\sum \left\|r'_{l.i}\right\|^2}. \tag{14}$$

Thus, the translation vector can be formulated as

$$T = \frac{1}{n}\sum r_{r,i} - s\boldsymbol{R}\left(\frac{1}{n}\sum r_{l,r}\right). \tag{15}$$

4 Experimental results

All the algorithms above are fully tested and proved to be valid in simulations “on land”. A series of experiments are conducted in air to prove and compare the efficiency of the algorithms in the mission of AAR. In addition, the efficiency of the whole system is also verified via the experiments. To compute the relation matrix between the cameras and the marker coordinate system, the camera calibration is conducted first. During flight, all the images are processed onboard; and the estimated pose information is immediately transmitted to the flight controller; and relative flight order is given. The images taken by the cameras are also transmitted to the ground station and are saved for further study.

The independent experimental results obtained with different pose estimation algorithms are shown below. Figures 5 and 6 are the results of monocular algorithms while Figures 7–9 are the results of binocular algorithms.

The images in Figure 5 are the experimental results of the EPnP algorithm. The images on the left show the point detected with the EPnP algorithm and the projection results after pose estimation are shown on the right images. It can be seen from the results that there always exists a small projection error, but in most conditions the error is considered to be acceptable. This algorithm performs considerably well. The results of the GLSDC algorithm in Figure 6 have significant differences in different frames. In some frames, the algorithm performs unstable; however at most time, the results are reliable.

From the images in Figure 7, it is clear that the LHM algorithm performs well. The reprojected points almost cover the real markers. Also the LHM algorithm performs best among binocular algorithms. The experimental result of another binocular algorithm, least square algorithm, is indicated in Figure 8. As the binocular algorithms demand, the markers could be found in both left and right images. There are only few effective experimental results of the least square algorithm. Like the result in the first row, sometimes the feature extraction results appear to be not that accurate. The effect of this algorithm is unstable. The absolute orien-

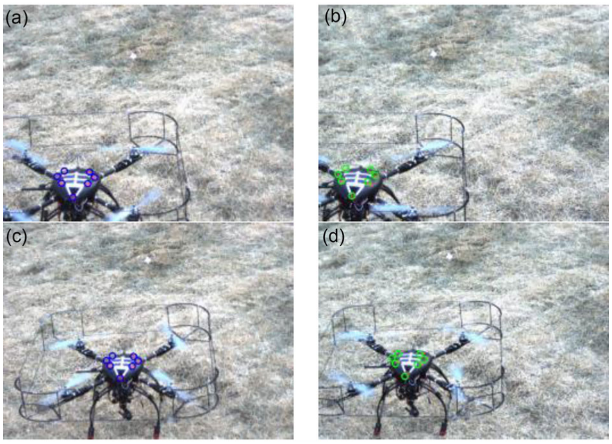

Figure 5 (Color online) The pose estimation result using the EPnP algorithm. (a) and (c) The feature extraction results on images captured with left camera; (b) and (d) Relative projection result of the EPnP pose estimation algorithm on images captured with right camera.

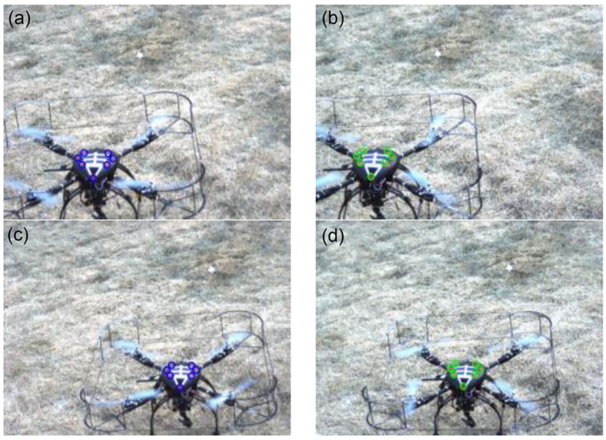

Figure 6 (Color online) The pose estimation result using the GLSDC algorithm. (a) and (c) The feature extraction results on images captured with left camera; (b) and (d) Relative projection result of the GLSDC pose estimation algorithm on images captured with right camera.

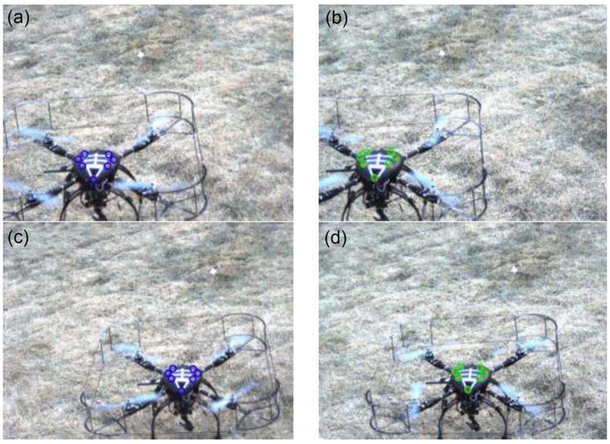

Figure 7 (Color online) The pose estimation result using the LHM algorithm. (a) and (c)The feature extraction results on images captured with left camera; (b) and (d) Relative projection result of the LHM pose estimation algorithm on images captured with right camera.

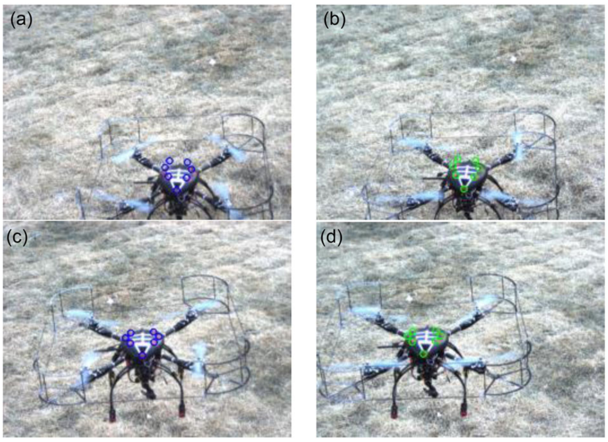

Figure 8 (Color online) The pose estimation result using least squares algorithm. (a) and (c) The feature extraction results on images captured with left camera; (b) and (d) Relative projection result of the least squares pose estimation algorithm on images captured with right camera.

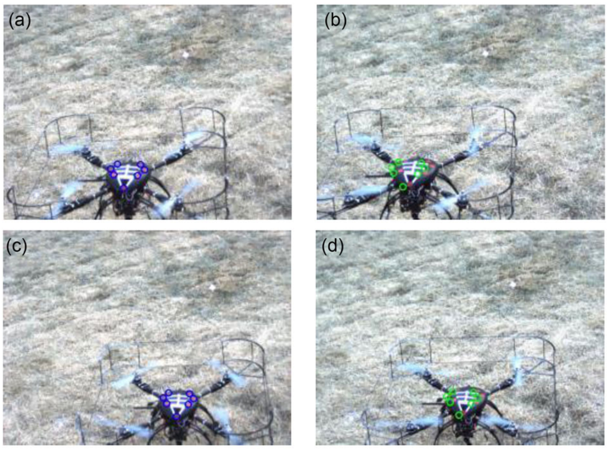

Figure 9 (Color online) The pose estimation result using absolute orientation algorithm. (a) and (c) The feature extraction results on images captured with left camera; (b) and (d) relative projection result of the absolute orientation pose estimation algorithm on images captured with right camera.

tation algorithm is accurate and easy to realize "on land". However, if the coordinates matching of the feature point exists bulky error, the algorithm would not work. The experimental results of the absolute orientation algorithm are shown in Figure 9. As is illustrated, the algorithm has a poor performance.

Since different algorithms obtain different pose estimation results, the consequence of flight status is rather dissimilar. To compare the algorithms, a same sequence of 100 frames taken during flight is used. Figure 10 displays a comparison of the relative distance estimation obtained with different algorithms. Since the real value of the distance between two UAVs is unknown, it is unreasonable to confirm "the most" reliable algorithm. However, from the curves below it is rational to infer that the least square algorithm performs far from satisfactory, while there is no big difference between the other algorithms. Figure 11 presents a comparison of the rotation angle between two UAVs obtained with different algorithms. Theta represents the pitch angle and phi represents the roll angle while psi represents the yaw angle. Since the UAVs are flying forward, change of theta is comparably fierce. While the flight does not conduct a distinct change in orientation, psi keeps steady. Figure 11 gives a likely information from Figure 10 that the least square algorithm is not stable at all. Figure 12 shows the comparison of the reprojection error computed with different algorithms. It is the error between the reprojection results on the right images and the makers detected from the right images. The error value more than one hundred is set to be zero. Together with the figures above, it is obvious and reliable that when the UAVs fly unstable, the error is correspondingly increased a great deal. It can also be inferred that the algorithms of EPnP, GLSDC and LHM perform better. The absolute orientation algorithm is proposed to use three of the seven feature points in the pose estimation process. Thus, fewer information is taken into consid eration, which could cause a larger error. The least squares algorithm does not iterate during calculating. As a result, it

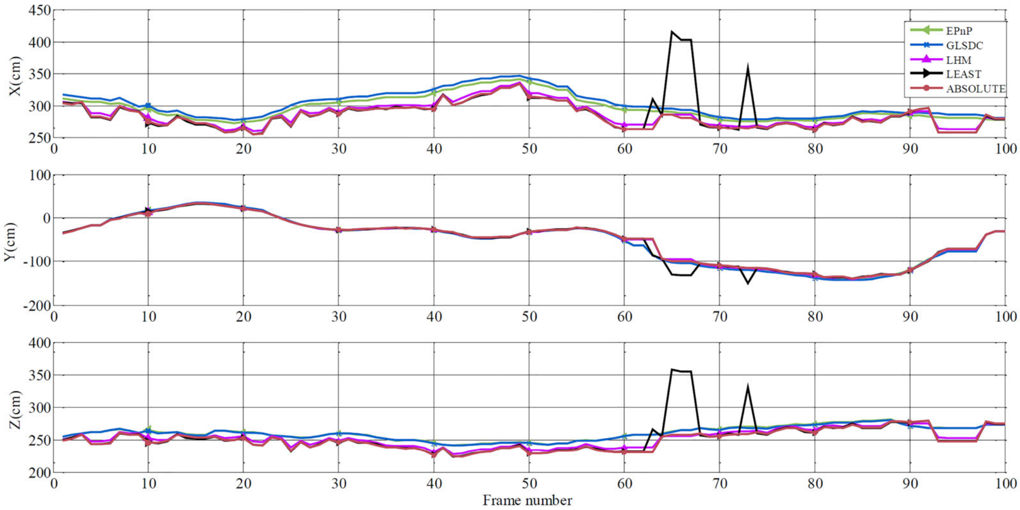

Figure 10 (Color online) Comparison of the relative distance estimation obtained with different algorithms.

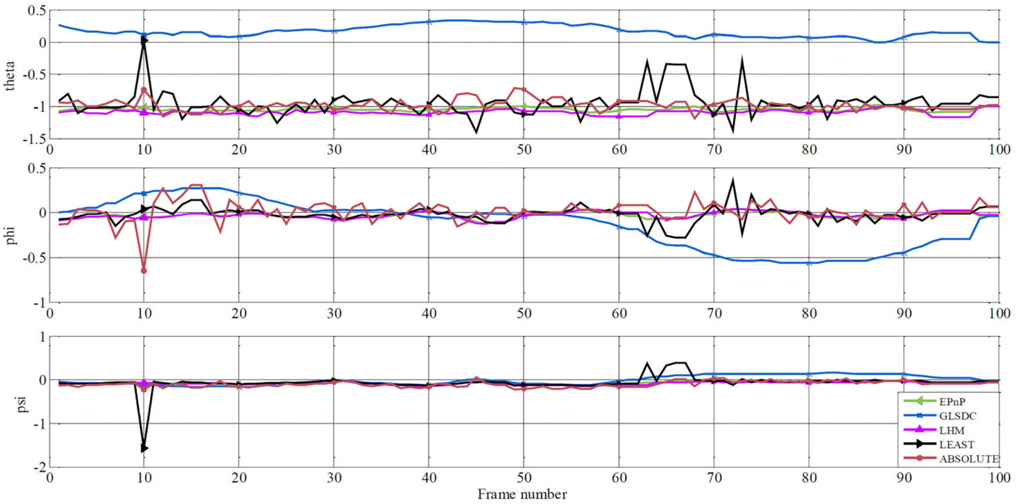

Figure 11 (Color online) Comparison of the rotation angle between two UAVs obtained with different algorithms.

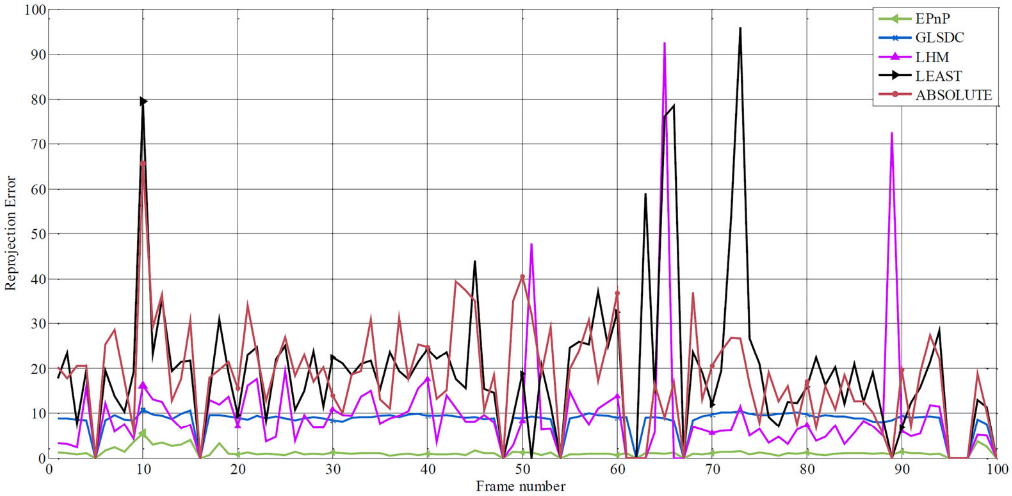

Figure 12 (Color online) Comparison of the reprojection error computed with different algorithms.

is sensitive to the parameters in the equations, including the intrinsic and extrinsic parameters of the cameras, and the markers coordinates in the image coordinate system and world coordinate system. While an inevitable error exists in the measurement of the parameters above, the least squares algorithm would enlarge it, especially in the estimation of the rotation angle. The average performance of the binocular algorithms is worse than the monocular algorithm, which is considered to be caused by the relative movement of the cameras derived from UAV's shaking. Also the possibility of inaccurate binocular camera calibration should not be ignored.

5 Conclusion

In this paper, a vision-based technology for AAR is developed in this paper. The monocular vision algorithms of EPnP, GLSDC and binocular vision algorithms of LHM, least squares algorithm and absolute orientation algorithm for pose estimation of UAVs are adopted and verified in this AAR system. The real-time on-board vision system is also designed for precise navigation in the UAVs docking phase. The experimental results indicate that the system is capable of verifying the algorithms expected to be applied to AAR. Besides, the results also give some hints about the matters that need to be paid attention to, like the shaken of the UAV, the influence of the UAVs on each other, the collision avoi-dance etc. Since the rapid and fierce movement of the UAVs is a challenge for visual algorithms, our future research will focus on improving the robust of the visual system in both hardware and software.

This work was supported by the National Natural Science Foundation of China (Grant Nos. 61425008, 61333004 & 61273054), and the Aeronautical Foundation of China (Grant No. 2015ZA51013).

1 Sun C H, Duan H B. Markov decision evolutionary game theoretic learning for cooperative sensing of unmanned aerial vehicles. Sci China Tech Sci, 2015, 58: 1392–1400

2 Wang Y, Wang D B. Variable thrust directional control technique for plateau unmanned aerial vehicles. Sci China Inf Sci, 2016, 59: 033201

3 Qiu H X, Wei C, Dou R, et al. Fully autonomous flying: from collective motion in bird flocks to unmanned aerial vehicle autonomous swarms. Sci China Inf Sci, 2015, 58: 128201

4 Hill A C, Rowan Y, Kersel M M. Mapping with aerial photographs. Near Eastern Archaeology, 2014, 77: 182–186

5 Mancini F, Dubbini M, Gattelli M, et al. Using unmanned aerial vehicles (UAV) for high-resolution reconstruction of topography: the structure from motion approach on coastal environments. Remote Sens, 2013, 5: 6880–6898

6 Honma T, Kaku K, Usup A, et al. Trop Peatland Ecosyst. Japan: Springer Japan, 2016. 397–406

7 Barrell J, Grant J. High-resolution, low-altitude aerial photography in physical geography: a case study characterizing eelgrass (zostera marina l.) and blue mussel (mytilus edulis l.) landscape mosaic structure. Prog Phys Geog, 2015, 8: 440–459

8 Gallagher A. Surveillance UAV. WPI Qualifying Report. 2014

9 Borck H, Karneeb J, Alford R, et al. Case-based behavior recognition in beyond visual range air combat. In: Proceedings of the 28th International Florida Artificial Intelligence Research Society Conference, Hollywood: AAAI, 2015. 1–7

10 Zhu Z S, Su A, Liu H B, et al. Vision navigation for aircrafts based on 3D reconstruction from real-time image sequences. Sci China Tech Sci, 2015, 58: 1196–1208

11 Zhang J J, Xue M, Xie J. Research on assessment method of intersystem and intersystem of the global navigation satellite system. Sci China Tech Sci, 2015, 58: 1672–1681

12 Hua C S, Qi J T, Shang H, et al. Detection of collapsed buildings with the aerial images captured from UAV. Sci China Inf Sci, 2016, 59:1–15

13 Chen Z, Di S, Cao Z X, et al. A 256×256 time-of-flight image sensor based on center-tap demodulation pixel structure. Sci China Inf Sci, 2016, 59: 1–10

14 Liu X C, Wang H, Fu D, et al. An area-based position and attitude estimation for unmanned aerial vehicle navigation. Sci China Tech Sci, 2015, 58: 916–926

15 Valasek J, Gunnam K, Kimmett J, et al. Vision-based sensor and navigation system for autonomous air refueling. J Guid Control Dynam, 2005, 28: 979–989

16 Chen C I, Stettner R. Drogue tracking using 3D flash lidar for autonomous aerial refueling. In: Proceedings of Defense, Security, and Sensing. Orlando: SPIE, 2011. 80370Q–80370Q-11

17 Campa G, Fravolini M L, Ficola A, et al. Autonomous aerial refueling for UAVs using a combined GPS-machine vision guidance. In: Proceedings of Guidance, Navigation, and Control Conference and Exhibit, Providence: AIAA, 2004. 1–11

18 Doebbler J, Spaeth T, Valasek J, et al. Boom and receptacle autonomous air refueling using visual snake optical sensor. J Guid Control Dynam, 2007, 30: 1753–1769

19 Kimmett J, Valasek J, Junkins J L. Autonomous aerial refueling utilizing a vision based navigation system. Proceedings of the 2002 Guidance Navigation and Control Conference, Monterey: AIAA. 2002, 1–11

20 Duan H B, Li H, Luo Q N, et al. A binocular vision-based UAVs autonomous aerial refueling platform. Sci China Inf Sci, 2016, 59: 053201

21 Munkres J. Algorithms for the assignment and transportation problems. J Soc Ind Appl Math, 1957, 5: 32–38

22 Lepetit V, Moreno-Noguer F, Fua P. Epnp: An accurate o (n) solution to the pnp problem. Int J Comput Vision, 2009, 81: 155–166

23 Haralick R M, Joo H, Lee C N, et al. Pose estimation from corresponding point data. IEEE T Syst Man Cy, 1989, 19: 1426–1446

24 Lu C P, Hager G D, Mjolsness E. Fast and globally convergent pose estimation from video images. IEEE T Pattern Anal, 2000, 22: 610–622

25 Horn B K P, Hilden H M, Negahdaripour S. Closed-form solution of absolute orientation using orthonormal matrices. J Opt Soc Am A, 1988, 5: 1127–1135