The Aeronautical Journal (2023), **127**, pp. 1048–1064
doi:10.1017/aer.2022.112

REGULAR PAPER

Vision geometry-based UAV flocking

L. Wang* and T. He

Research Institute of Electronic Science and Technology, University of Electronic Science and Technology of China, Chengdu, China
*Corresponding author. Email: wang_lei@uestc.edu.cn
Lei Wang and Tao He are co-first authors.

Received: 6 August 2022; **Revised:** 13 November 2022; **Accepted:** 19 December 2022

Keywords: Vision geometry; Deep learning; Optical flow; Flocking control; UAV

Abstract

A distributed UAV (unmanned aerial vehicle) flocking control method based on vision geometry is proposed, in which only monocular RGB (red, green, blue) images are used to estimate the relative positions and velocities between drones. It does not rely on special visual markers and external infrastructure, nor does it require inter-UAV communication or prior knowledge of UAV size. This method combines the advantages of deep learning and classical geometry. It adopts a deep optical flow network to estimate dense matching points between two consecutive images, uses segmentation technology to classify these matching points into background and specific UAV, and then maps the classified matching points to Euclidean space based on the depth map information. In 3D matching points, also known as 3D feature point pairs, each of their classifications is used to estimate the rotation matrix, translation vector, velocity of the corresponding UAV, as well as the relative position between drones, based on RANSAC and least squares method. On this basis, a flocking control model is constructed. Experimental results in the Microsoft Airsim simulation environment show that in all evaluation metrics, our method achieves almost the same performance as the UAV flocking algorithm based on ground truth cluster state.

Nomenclature

N	number of features
$P^i_{(.)}$	coordinate of feature i in some coordinate system
$\Pi_{(.)}$	orthogonal coordinate system
$R_{(.)}$	rotation matrix between two coordinate system
$T_{(.)}$	translation vector between two coordinate system
H	transformation matrix from time t to $t+1$ of UAV i, equals to $\begin{bmatrix} R_{(.)} & T_{(.)} \\ 0 & 1 \end{bmatrix}$
K	camera intrinsic matrix
Q^i	position vector of UAV i(m)
Q^i_x	x component of position vector of UAV i(m)
Q^i_y	y component of position vector of UAV i(m)
Q^i_Z	z component of position vector of UAV i(m)
V^i	velocity vector of UAV i(m/s)
V^i_x	x component of velocity vector of UAV i(m/s)
V^i_y	y component of velocity vector of UAV i(m/s)
V^i_{xy}	horizontal airspeed of UAV i(m/s)
V^e_{xy}	desired horizontal airspeed(m/s)
efv^i	desired flocking velocity vector of UAV i(m/s)
efv^i_x	x component of desired flocking velocity vector of UAV i(m/s)
efv^i_y	y component of desired flocking velocity vector of UAV i(m/s)
δ^i	desired flocking yaw angle of UAV i(rad)
u^i	control input vector of UAV i

u_x^i	x component of control input vector of UAV i
u_y^i	y component of control input vector of UAV i
u_f^i	flocking geometry control component of UAV i
u_{av}^i	horizontal airspeed alignment control component of UAV i
u_c^i	collision avoidance control component of UAV i
u_{vf}^i	flocking velocity control component of UAV i
C_f	strength coefficient of the flocking geometry control component
C_{av}	strength coefficient of the horizontal airspeed alignment control component
C_c	strength coefficient of the collision avoidance control component
C_{vf}	strength coefficient of the flocking velocity control component
W_j^i	influence weight of UAV j to UAV i for flocking geometry control
W_i^i	influence weight of UAV i to UAV i for flocking velocity control
d^{ij}	horizontal distance between UAV i and j(m)
D_c	maximum horizontal observation distance(m)
D_{l1}	minimum distance between UAVs to avoid collision(m)
D_d	desired horizontal distance between UAVs(m)
obj_1^i	flocking velocity deviation performance metric of UAV i
obj_2^i	flocking shape deviation performance metric of UAV i

1.0 Introduction

UAV flocking has attracted more and more attention because it can solve the problems in many application fields. For example, it can be used to cooperatively transport objects whose weight is not known in advance [1]. What's more, real-time tracking of flood boundary with UAV can speed up the rescue process of survivors and reduce secondary disasters [2]. In addition, UAV flocking can provide temporary network coverage for disaster areas and better serve rescue through captured images, audio and other data [3].

In order to make the UAV flocking fly safely and stably, the relative positioning problem must be solved. A practical approach to solve the relative positioning problem is to use external measurements, such as the Global Positioning System (GPS) [4], the Visual Reference Positioning System [5] and the Fixed Ultra-WideBand (UWB) Communication Module Positioning System [6], to estimate the positions of all participants in the preset reference coordinate system. However, these systems may not always be available, such as GPS for forest and urban environments, or require advance deployment of infrastructure (e.g. multi-camera motion capture system and ultra-wideband communication device), which greatly limits the applicability and ubiquity of systems that rely on this technology. A popular approach to overcoming these limitations is to use vision- or distance-based on-board sensor approaches. Distance-based approaches involve measuring the distance between UAVs and thus recovering the relative positions between them via laser [7] or ultra-wideband [8] on-board sensors. Compared with distance sensors, vision sensors provide richer information and consume less energy because of passive sensing, so it is interesting to study vision-based UAV flocking.

In recent years, significant progress has been made in the research of vision-based UAV flocking. Tang et al. [9] propose a vision-assisted UAV flocking method, which uses correlation filter-based vision detection to track the UAV ahead in order to form a linear formation, and the navigation capability is provided by the LiDAR-based simultaneous localisation and mapping (SLAM), but it fails to implement more complex formations (such as lattice structures) and does not take full advantage of the rich information provided by vision (such as vision-based SLAM). Moshtagh et al. [10] propose a theoretical approach based on visual measurement of orientation, optical flow and collision time, which achieves the directional alignment of individuals, but not the aggregation properties of flocking. Moreover, only numerical simulation experiments were designed, and the algorithm about vision was not verified on pictures. Based on Ref. (10), Moshtagh et al. [11] solve the problem of conflict avoidance among individuals and add parallel flight, circular flight and real flocking test. However, the aggregation nature of the flocking is still not achieved, and their visual perception depends on LED markers. Soria et al. [12] analyse

https://doi.org/10.1017/aer.2022.112 Published online by Cambridge University Press

the effects of field of view and orientation of vision sensors on flocking metrics such as order, union, connectivity and safety, which help to determine vision sensor selection and installation scheme when building UAV flocking system. Hu et al. [13] propose an end-to-end vision-based distributed flocking controller design method, which feeds raw images into a convolutional neural network to obtain feature vectors and propagates them hop-by-hop through the communication network to all UAVs, which selectively receive and process these messages through a graphical neural network to determine the action to be taken. And the whole controller including convolution and graph neural networks are trained together. However, only the results of the effect of different team sizes and feature vector sizes on the flocking stability are given in the paper, and further experiments are needed for other metrics. Schilling et al. [14] propose an imitation learning-based UAV flocking control algorithm. It adopts an end-to-end design in which the visual information of six directions of the UAV is fed into a convolutional neural network to infer control actions, and the training data are obtained from conventional position- and velocity-based flocking control algorithms. The proposed method is validated in both simulation and real environment, but the interpretability of the method is poor. In addition, the experimental site is empty and the more complex environments is not considered, and the grey image with very low resolution is used as the data input, which makes the performance of flocking control poor. Bastien et al. [15] develop a mathematical model of collective behaviour based on purely visual projection fields, which is sufficient to generate organised collective behaviour without spatial representation and distance information. However, this is only a mathematical model, and further research is needed to determine whether it can be finally implemented. Schilling et al. [16] propose a vision detection and tracking algorithm that does not rely on communication or visual markers. It detects nearby UAV by omnidirectional vision based on the YOLOv3-tiny architecture, and estimates the relative positions between drone with known physical size of UAV, which are subsequently fed to a flocking controller to achieve a safe flight of three real UAVs, but the trajectory fluctuates greatly and the relative distance estimation is inaccurate.

Therefore, this paper proposes a visual geometry-based UAV flocking control method that does not require inter-UAV communication, prior knowledge of UAV size or special visual markers, and has good accuracy, interpretability and flocking performance.

To summarise, the contributions of the paper are: (1) we propose a novel distributed UAV flocking method based on monocular visual geometry, which combines the advantages of deep learning and classical geometry; (2) we propose a method that allows moving UAVs to estimate the relative velocity and position of neighbouring drones using only RGB images captured by the on-board camera, and give an easy-to-understand derivation procedure; (3) experimental results in the Microsoft Airsim simulation environment show that our method achieves almost the same performance as UAV flocking algorithm based on ground truth cluster state in all evaluation metrics.

The rest of this paper is organised as follows: Section 2 introduces the visual geometry-based UAV flocking algorithm; Section 3 shows the simulation experimental method and results in detail; Section 4 summarises our work, points out the limitations of this paper and provides an outlook for our future work.

2.0 Method

The visual geometry-based UAV flocking algorithm proposed in this paper can be divided into the following steps: 3D feature point pair extraction, transformation matrix estimation and UAV flocking control (see Fig. 1). First, the 3D feature point pairs extraction module (see Fig. 1(a)) acquires two consecutive RGB colour images from the omnidirectional camera configuration, establishes a match by deep optical flow network, segments the match into different categories such as static background, UAV 1, UAV 2, . . . , etc., and then maps the classified match from image to Euclidean space based on the depth map information. Second, the transformation matrix estimation module (see Fig. 1(b)) first estimates the transformation matrix (including rotation matrix and translation vector) of the current UAV based on the 3D feature point pairs of the static background, and then estimates the relative position, rotation matrix and translation vector of the nearby UAV based on the rotation matrix and translation vector of

https://doi.org/10.1017/aer.2022.112 Published online by Cambridge University Press

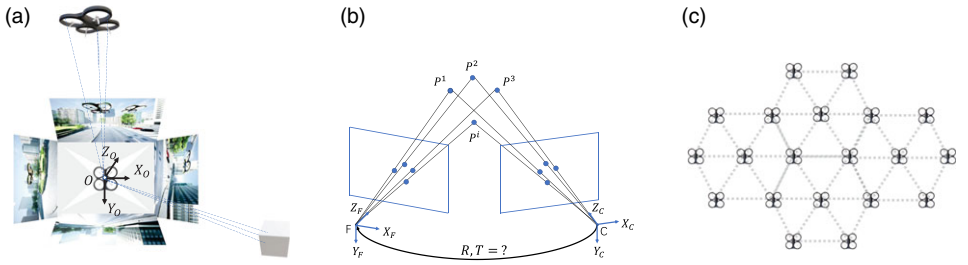

Figure 1. *Processing steps of UAV flocking based on visual geometry in a single sampling interval: (a) 3D feature point pairs extraction, (b) transformation matrix estimation, (c) UAV flocking control.*

the current UAV and the 3D feature point pairs of the nearby UAV. Finally, the estimated state of each UAV is input into the UAV flocking control module (see Fig. 1(c)), and the computer simulation of biological collective behaviour in two-dimensional space is realised according to the three heuristic rules: aggregation, alignment and separation [17]. To obtain richer visual features, we arrange the UAVs at different altitudes and assume that their altitude remains constant throughout the flight.

2.1 3D feature point pairs extraction

2.1.1 Depth estimation

Structure-from-motion (SfM) is the problem of estimating camera motion and scene geometry from monocular image sequences [18]. For decades, numerous researchers have studied this problem intensively. There are two main approaches in existence: classical geometry method and deep learning method. The classical geometry approach first matches features between two images and then infers camera motion and scene geometry from these matches. Its advantage is that the method has very good accuracy and interpretability when there is a set of accurate matching points and scale prior information. However, the traditional feature matching method is not effective in obtaining accurate matching points, and without prior knowledge of the scale or identifiable objects in the scene, two or more views can only estimate relative camera motion and scene geometry, i.e. no absolute scale can be inferred. The deep learning approach uses end-to-end deep neural networks to regress depth and pose from a single image or image pair. It can extract the scale and scene prior knowledge from the training data. Therefore, this paper combines the advantages of both methods. A deep neural network is used to estimate depth map [19, 20] that provides absolute scale information, another deep neural network is used to obtain exact matching points (see the next section for more details), and then camera motion and scene geometry are computed using classical geometry methods.

2.1.2 Optical flow estimation

Feature matching and optical flow estimation are the main methods to obtain feature point pairs. However, due to the following drawbacks of feature matching methods – (1) the number of feature is limited so that the relative position estimation is inaccurate and unstable and (2) in some cases (no texture, occlusion) – there are no or not enough feature point pairs to recover the transformation matrix of the UAV. Therefore, in this paper, we directly utilise an unretrained state-of-the-art deep neural network, RAFT [21], which is a deep learning-based optical flow estimation method. This method builds a feature representation for each pixel, constructs a matching objective function based on the similarity between the current and other pixel feature representations, and trains an update operator by deep learning, which iteratively updates the optical flow field based on the gradient-like information provided by the objective function until the best match is found for each pixel.

https://doi.org/10.1017/aer.2022.112 Published online by Cambridge University Press

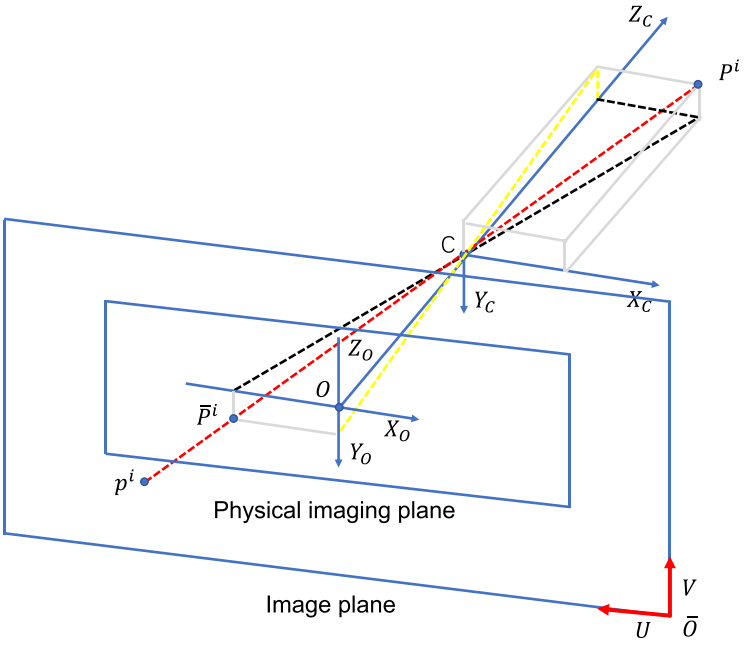

Figure 2. *Pin-hole lens model.*

2.1.3 Segmentation

The motion of a rigid body can be described by the rotation matrix and the translation vector, and it is only necessary to know enough 3D feature point pairs of a particular rigid body in two images to recover its transformation matrix. Therefore, we can use the existing image segmentation technology similar to [22] to segment the image according to different rigid bodies, and recover its transformation matrix according to the 3D feature point pairs corresponding to each rigid body.

2.1.4 Mapping of image to Euclidean space

In this subsection, we will derive the mathematical relationship between the coordinate of a point in Euclidean space in the camera coordinate system and the coordinate of its projection point in the image coordinate system. As shown in Fig. 2, a point P^i in Euclidean space passes through the pin-hole lens model to get an image point $\bar{P}^i$ in the physical imaging plane and a pixel point p^i in the image plane. The camera coordinate system is fixed on the lens, the origin is C, and the base is $\{X_C, Y_C, Z_C\}$. The coordinate of P^i in the camera coordinate system is (P^i_x, P^i_y, P^i_z). The physical imaging coordinate system is fixed on the physical imaging plane, the origin is O, the base is $\{X_O, Y_O, Z_O\}$, Z_O and Z_C are collinear, and the coordinate of $\bar{P}^i$ in the physical imaging coordinate system is $(\bar{P}^i_x, \bar{P}^i_y, \bar{P}^i_z)$, $\bar{P}^i_z = 0$. Length of the line OC is equal to the focal length f of the camera lens. According to the similarity relation of triangles it is obtained:

$$\frac{-\bar{P}^i_x}{P^i_x} = \frac{f}{P^i_z} \tag{1}$$

$$\frac{-\bar{P}^i_y}{P^i_y} = \frac{f}{P^i_z} \tag{2}$$

The image coordinate system is fixed on the image plane, the origin is $\bar{O}$, the base is $\{U, V\}$, and the pixel point p^i has the homogeneous coordinate $(u, v, 1)$ in the image coordinate system. The transformation of

https://doi.org/10.1017/aer.2022.112 Published online by Cambridge University Press

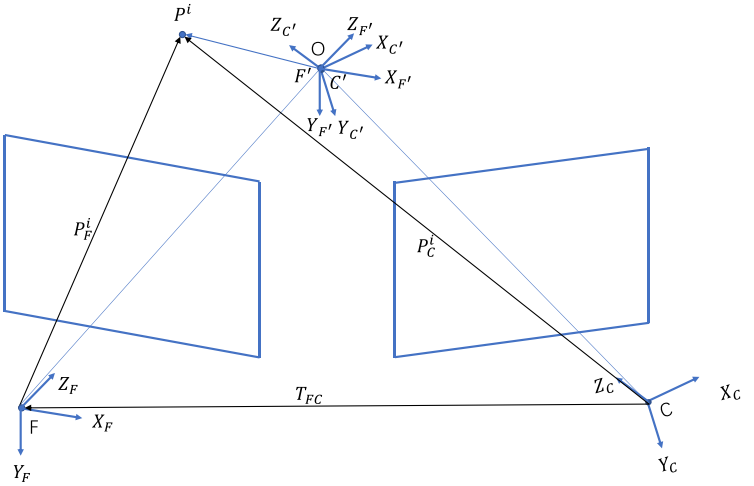

Figure 3. *Camera motion model.*

the physical imaging coordinate system to the image coordinate system requires scaling and translation operations, as follows:

$$u = -a\bar{P}_x^i + t_u \tag{3}$$

$$v = -b\bar{P}_y^i + t_v \tag{4}$$

Organising the above equations into matrix form, we get:

$$P_z^i \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} af & 0 & t_u \\ 0 & bf & t_v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_x^i \\ P_y^i \\ P_z^i \end{bmatrix} \tag{5}$$

Abbreviated as:

$$P_z^i q^i = KP^i \tag{6}$$

K is the camera intrinsic matrix, which is assumed to be known in this paper. If the depth map is known, then P_z^i is known, and the coordinates of a point in Euclidean space expressed in the camera coordinate system can be calculated from the homogeneous coordinate of the point on the image.

2.2 Transformation matrix estimation

2.2.1 Camera motion model

In this subsection, we will derive the mathematical relationship between multiple pairs of static background feature points in two consecutive images and the transformation matrix of the camera coordinate system in this sampling interval. Consider a scenario depicted in Fig. 3 where a moving camera views a static background. At time t, the camera coordinate system Π_C has origin C and basis $\{X_C, Y_C, Z_C\}$. At time $t+1$, the camera coordinate system Π_F has origin F and basis $\{X_F, Y_F, Z_F\}$. $P^i, i = 1, 2, 3, \ldots, N$ denotes the N static background feature points that are captured by the camera at both time t and $t+1$. The coordinates of P^i expressed in Π_C and Π_F are P_C^i and P_F^i respectively, and they are related as follows:

$$P_C^i = R_{FC}P_F^i + T_{FC} \tag{7}$$

https://doi.org/10.1017/aer.2022.112 Published online by Cambridge University Press

where, R_{FC} is the rotation matrix describing the orientation of Π_F with respect to Π_C, and T_{FC} is the position of F with respect to C expressed in Π_C. Solving R_{FC} and T_{FC} according to Equation (7) requires a complex joint optimisation of all the unknown variables, so we try to solve them separately. O is a point in Euclidean space. The coordinate system $\Pi_{C'}$ is obtained by shifting the origin C of Π_C to O, which has origin C' and base $\{X_{C'}, Y_{C'}, Z_{C'}\}$. The coordinate system $\Pi_{F'}$ is obtained by shifting the origin F of Π_F to O, which has origin F' and base $\{X_{F'}, Y_{F'}, Z_{F'}\}$. Coordinates of the vector $\overrightarrow{OP^i}$ expressed in $\Pi_{C'}$ and $\Pi_{F'}$ are $\overrightarrow{OP^i}_{C'}$ and $\overrightarrow{OP^i}_{F'}$ respectively, and they have the following mathematical relation:

$$\overrightarrow{OP^i}_{C'} = R_{F'C'}\overrightarrow{OP^i}_{F'} \tag{8}$$

where, $R_{F'C'}$ denotes the rotation matrix describing the orientation of $\Pi_{F'}$ with respect to $\Pi_{C'}$. From the geometric relationship between the vectors depicted in Fig. 3, the following equality can be obtained.

$$\overrightarrow{OP^i}_{C'} = \overrightarrow{CP^i}_{C'} - \overrightarrow{CC'}_{C'} \tag{9}$$

$$\overrightarrow{OP^i}_{F'} = \overrightarrow{FP^i}_{F'} - \overrightarrow{FF_{F'}}' \tag{10}$$

where, $\overrightarrow{CP^i}_{C'}$ and $\overrightarrow{CC'}_{C'}$ denote the coordinates of vector $\overrightarrow{CP^i}$ and $\overrightarrow{CC'}$ in $\Pi_{C'}$ respectively. $\overrightarrow{FP^i}_{F'}$ and $\overrightarrow{FF_{F'}}'$ denote the coordinates of vector $\overrightarrow{FP^i}$ and $\overrightarrow{FF'}$ in $\Pi_{F'}$ respectively. After substituting Equations (9) and (10) into Equation (8), the following relationship can be developed:

$$\overrightarrow{CP^i}_{C'} - \overrightarrow{CC'}_{C'} = R_{F'C'}(\overrightarrow{FP^i}_{F'} - \overrightarrow{FF_{F'}}') \tag{11}$$

As you know, if there is no rotation between the two coordinate systems, then a vector has the same representation in these two coordinate systems. Since there is no rotation transformation between Π_C and $\Pi_{C'}$, and the same between coordinate systems Π_F and $\Pi_{F'}$, the following equation is true.

$$\overrightarrow{CP^i}_{C'} = P^i_C \tag{12}$$

$$\overrightarrow{FP^i}_{F'} = P^i_F \tag{13}$$

$$\overrightarrow{CC'}_{C'} = \overrightarrow{CC'}_C \tag{14}$$

$$\overrightarrow{FF'}_{F'} = \overrightarrow{FF_{F'}}' \tag{15}$$

where, $\overrightarrow{CC'}_C$ and $\overrightarrow{FF'}_F$ denote the coordinates of point O in the Π_F and Π_C respectively. Because translation does not affect rotation, so the rotation transformation between Π_C and Π_F is the same as that between $\Pi_{C'}$ and $\Pi_{F'}$. The equation is as follows:

$$R_{F'C'} = R_{FC} \tag{16}$$

After substituting Equations (12)–(16) into Equation (11), the following relationship can be developed:

$$P^i_C - \overrightarrow{CC'}_C = R_{FC}(P^i_F - \overrightarrow{FF'}_F) \tag{17}$$

Take point O as the centre of all feature points, as shown in the following:

$$\overrightarrow{CC'} = P_C = \frac{1}{N}\sum_1^N P^i_C \tag{18}$$

https://doi.org/10.1017/aer.2022.112 Published online by Cambridge University Press

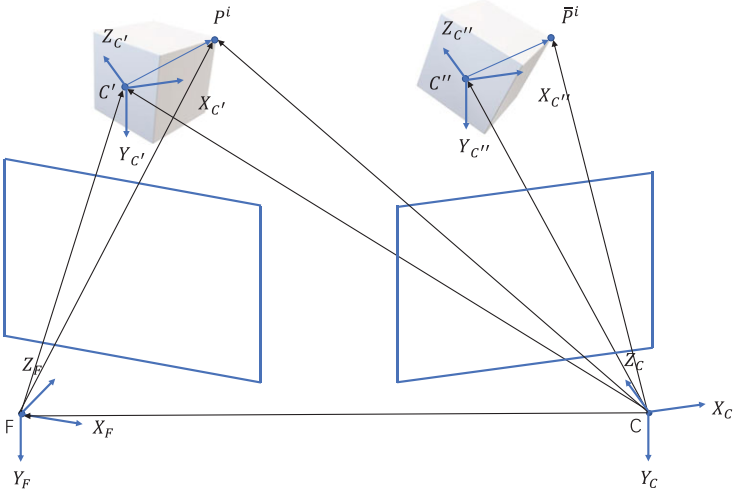

Figure 4. *Object motion model.*

$$\overrightarrow{FF'} = P_F = \frac{1}{N}\sum_{1}^{N} P_F^i \tag{19}$$

After substituting Equations (18) and (19) into Equation (17), the following relationship can be developed:

$$P_C^i - P_C = R_{FC}(P_F^i - P_F) \tag{20}$$

The coordinates of P_C^i and P_F^i can be calculated according to the Equation (6), and according to Equation (20) using the least squares method based on singular value decomposition [23] can be solved to obtain R_{FC}, and T_{FC} can be obtained by substituting R_{FC} into the Equation (7). The velocity V^i of UAV i can be expressed as:

$$V^i = \frac{T_{FC}}{st} \tag{21}$$

Where, st denotes the sampling time.

2.2.2 Object motion model

In this subsection, we will derive the mathematical relationship between multiple pairs of feature points of moving objects in two consecutive images collected by a moving camera and their transformation matrix in the sampling interval. Consider a scenario depicted in Fig. 4 where a moving camera views a moving object. At time t, the camera coordinate system Π_C has origin C and basis $\{X_C, Y_C, Z_C\}$. At time $t+1$, the camera coordinate system Π_F has origin F and basis $\{X_F, Y_F, Z_F\}$. $P^i, i = 1, 2, 3, \ldots, N$ denotes the N feature points on the moving object captured by the camera at the time $t+1$. The coordinates of P^i expressed in Π_C and Π_F are P_C^i and P_F^i respectively, and they are related as follows:

$$P_C^i = R_{FC}P_F^i + T_{FC} \tag{22}$$

Where, R_{FC} is the rotation matrix describing the orientation of Π_F with respect to Π_C, and T_{FC} is the position of F with respect to C expressed in Π_C. The coordinates of P_F^i can be calculated according to the Equation (6), R_{FC} and T_{FC} can be calculated according to the previous section, so P_C^i can be calculated according to the Equation (22). Let $\bar{P}^i$ denote the feature point corresponding to P^i on the moving object captured by the camera at time t. Its coordinates $\bar{P}_C^i$ in the Π_C can also be calculated

https://doi.org/10.1017/aer.2022.112 Published online by Cambridge University Press

according to the Equation (6). Now, the problem we need to solve is to find the transformation matrix between the pose of the moving object at time t and $t+1$ under the condition that P_C^i and $\bar{P}_C^i$ is known. Let C' and C'' denote a certain feature point O on a moving object captured by the camera at time $t+1$ and t respectively. The coordinate system $\Pi_{C'}$ is obtained by shifting the origin C of Π_C to C', which has origin C' and base $\{X_{C'}, Y_{C'}, Z_{C'}\}$. The coordinate system $\Pi_{C''}$ is obtained by shifting the origin C of Π_C to C'', which has origin C'' and base $\{X_{C''}, Y_{C''}, Z_{C''}\}$. $\overrightarrow{C'P^i_{C'}}$ denotes coordinate of the vector $\overrightarrow{C'P^i}$ expressed in $\Pi_{C'}$, $\overrightarrow{C''\bar{P}^i_{C''}}$ denotes coordinate of the vector $\overrightarrow{C''\bar{P}^i}$ expressed in $\Pi_{C''}$, and they have the following mathematical relation:

$$\overrightarrow{C'P^i_{C'}} = R_{C''C'}\overrightarrow{C''\bar{P}^i_{C''}} \tag{23}$$

where, $R_{C''C'}$ denotes the rotation matrix between the pose of the moving object at time t and $t+1$ with respect to the point O. From the geometric relationship between the vectors depicted in Fig. 4, the following equality can be obtained.

$$\overrightarrow{C'P^i_{C'}} = \overrightarrow{CP^i_{C'}} - \overrightarrow{CC'_{C'}} \tag{24}$$

$$\overrightarrow{C''\bar{P}^i_{C''}} = \overrightarrow{C\bar{P}^i_{C''}} - \overrightarrow{CC''_{C''}} \tag{25}$$

where, $\overrightarrow{CP^i_{C'}}$ and $\overrightarrow{CC'_{C'}}$ denote the coordinates of vector $\overrightarrow{CP^i}$ and $\overrightarrow{CC'}$ in $\Pi_{C'}$ respectively. $\overrightarrow{C\bar{P}^i_{C''}}$ and $\overrightarrow{CC''_{C''}}$ denote the coordinates of vector $\overrightarrow{C\bar{P}^i}$ and $\overrightarrow{CC''}$ in $\Pi_{C''}$ respectively. After substituting Equations (24) and (25) into Equation (23), the following relationship can be developed:

$$\overrightarrow{CP^i_{C'}} - \overrightarrow{CC'_{C'}} = R_{C''C'}(\overrightarrow{C\bar{P}^i_{C''}} - \overrightarrow{CC''_{C''}}) \tag{26}$$

As you know, if there is no rotation between the two coordinate systems, then a vector has the same representation in these two coordinate systems. Since there is no rotation transformation between Π_C, $\Pi_{C'}$ and $\Pi_{C''}$, the following equation is true.

$$\overrightarrow{CP^i_{C'}} = P^i_C \tag{27}$$

$$\overrightarrow{CC'_{C'}} = \overrightarrow{CC'_C} \tag{28}$$

$$\overrightarrow{C\bar{P}^i_{C''}} = \bar{P}^i_C \tag{29}$$

$$\overrightarrow{CC''_{C''}} = \overrightarrow{CC''_C} \tag{30}$$

where, $\overrightarrow{CC'_C}$ and $\overrightarrow{CC''_C}$ denote the coordinates of vector $\overrightarrow{CC'}$ and $\overrightarrow{CC''}$ in Π_C respectively. After substituting Equations (27)–(30) into Equation (26), the following relationship can be developed:

$$P^i_C - \overrightarrow{CC'_C} = R_{C''C'}(\bar{P}^i_C - \overrightarrow{CC''_C}) \tag{31}$$

Take point O as the centre of all feature points, as shown in the following:

$$\overrightarrow{CC'_C} = P_C = \frac{1}{N}\sum_1^N P^i_C \tag{32}$$

$$\overrightarrow{CC''_C} = \bar{P}_C = \frac{1}{N}\sum_1^N \bar{P}^i_C \tag{33}$$

https://doi.org/10.1017/aer.2022.112 Published online by Cambridge University Press

After substituting Equations (32) and (33) into Equation (31), the following relationship can be developed:

$$P_C^i - P_C = R_{C''C'}(\bar{P}_C^i - \bar{P}_C) \tag{34}$$

Knowing P_C^i and $\bar{P}_C^i$, Equation (34) can be solved for $R_{C''C'}$ by using the least squares method based on singular value decomposition [23]. The velocity V^j estimated by UAV i for nearby UAV j can be expressed as:

$$T_{C''C'} = P_C - \bar{P}_C \tag{35}$$

$$V^j = \frac{T_{C''C'}}{st} \tag{36}$$

where st denotes the sampling time. $(\frac{x_{min}+x_{max}}{2}, \frac{y_{min}+y_{max}}{2}, \frac{z_{min}+z_{max}}{2})$ is the relative position between UAV i and UAV j. $x_{min}, x_{max}, y_{min}, y_{max}, z_{min}, z_{max}$ are calculated as follows:

$$\begin{cases} x_{min} = min\{P_x^i | i \in inliers\}, \\ x_{max} = max\{P_x^i | i \in inliers\}, \\ y_{min} = min\{P_y^i | i \in inliers\}, \\ y_{max} = max\{P_y^i | i \in inliers\}, \\ z_{min} = min\{P_z^i | i \in inliers\}, \\ z_{max} = max\{P_z^i | i \in inliers\}, \end{cases} \tag{37}$$

where inliers represents the point set on UAV j observed by UAV i. See the next section for its calculation method.

2.2.3 Transformation matrix estimation algorithm

Due to segmentation errors, matching errors and inaccurate depth estimation, there are numerous outliers in the set of 3D feature point pairs, which will lead to a large transformation matrix estimation error. In this paper, the RANSAC [24] algorithm is used to remove the outliers (see Algorithm 1). In each iteration, the rigid body transformation H_t^{t+1} corresponding to three pairs of randomly selected non-collinear 3D feature points is calculated by least squares method based on singular value decomposition (SVD), and the adaptive rejection threshold mechanism is used to select the best transformation matrix to solve the problem of inaccurate rejection threshold settings that cause inability to calculate or insufficient accuracy of the results. Finally, in order to further improve the accuracy of the results, the final transformation matrix is calculated based on the set of inliers.

2.3 UAV flocking control

In this paper, the flocking control model [25] uses the relative position and velocity between UAVs as inputs to calculate the control input $u^i = (u_x^i, u_y^i)$ of UAV i. u^i consists of four components: u_f^i is flocking geometry control component, which is responsible for adjusting the horizontal distance between UAV i and its neighbour so that it is close to the desired horizontal distance. u_{av}^i is horizontal airspeed alignment control component, which regulates the velocity of UAV i to be consistent with its neighbour. u_c^i is collision avoidance control component, which regulates the safe distance among UAVs. u_{vf}^i is flocking velocity control component, which regulates the velocity of UAV i to be consistent with the desired flocking velocity. The formulaic representation of each control component is as follows:

$$u_k^i = u_f^i + u_{av}^i + u_c^i + u_{vf}^i - V_k^i, \quad k = x, y, \tag{38}$$

https://doi.org/10.1017/aer.2022.112 Published online by Cambridge University Press

Algorithm 1. transformation matrix estimation algorithm

Initialize number of iterations $it = 0$, maximum iterations IT_{MAX}, number of point pairs N, P_t^i denotes the set of feature points at time t, P_{t+1}^i denotes the set of feature points at time $t+1$, $i = 1, 2, \ldots, N$, rejection threshold $\delta = 0.01$, transformation matrix $H = None$, $inliers = \emptyset$;
while H *is None* **do**
  while $it < IT_{MAX}$ **do**
    Randomly select 3 pairs of 3D feature points that are not co-linear;
    Calculation of rigid body transformation matrix H_t^{t+1} by least squares method based on singular value decomposition;
    $cur = \emptyset$;
    for $i = 1$ to N **do**
      if $\left\| \begin{bmatrix} P_{t+1}^i \\ 1 \end{bmatrix} - H_t^{t+1} \begin{bmatrix} P_t^i \\ 1 \end{bmatrix} \right\| < \delta$ **then**
        $cur = cur \cup \{i\}$
      end if
    end for
    if $|cur| > 0.2N$ *and* $|cur| > |inliers|$ **then**
      $H = H_t^{t+1}$
      $inliers = cur$
    end if
    $it = it + 1$
  end while
  $\delta = \delta * 2$
end while
Calculate the final rigid body transformation matrix H by *inliers*;
return H

$$u_f^i = C_f \sum_{j \in \left\{ d^{ij} \le D_c \right\}} W_j^i (Q_k^j - Q_k^i) ln \left(\frac{d^{ij}}{D_d} \right) \tag{39}$$

$$u_{av}^i = C_{av} \sum_{j \in \left\{ d_{ij} \le D_c \right\}} W_j^i (V_k^j - V_k^i) \tag{40}$$

$$u_c^i = C_c \sum_{j \in \left\{ d^{ij} \le D_{l1} \right\}} \left(\frac{1}{d^{ij}} - \frac{1}{D_{l1}} \right)^2 \frac{Q_k^i - Q_k^j}{d^{ij}} \tag{41}$$

$$u_{vf}^i = C_{vf} W_i^i efv_k^i \tag{42}$$

where $Q^i = (Q_x^i, Q_y^i, Q_z^i)$ is the position vector of the UAV i, $V^i = (V_x^i, V_y^i)$ is the velocity vector of the UAV i, $C_f = 0.1$, $C_{av} = 0.1$, $C_c = 10$, $C_{vf} = 1$ denote strength coefficient of the corresponding control component, W_j^i denotes influence weight of UAV j to UAV i for flocking geometry control, W_i^i denotes influence weight of UAV i to UAV i for flocking velocity control, $d^{ij} = \sqrt{(X^i - X^j)^2 + (Y^i - Y^j)^2}$ denotes the horizontal distance between UAV i and j, $D_c = 20$m denotes the maximum horizontal observation distance, $D_{l1} = 1.5$m denotes the minimum distance between UAVs to avoid collision, $D_d = 3.5$m denotes the

https://doi.org/10.1017/aer.2022.112 Published online by Cambridge University Press

Table 1. *Initial parameters of the UAVs*

	i	Q_x^i(m)	Q_y^i(m)	Q_z^i(m)	V^i(m/s)
UAV	1	−0.5	0.9	−4.5	(1, 0)
	2	0.5	−2.8	−4.0	(1, 0)
	3	1.2	4.5	−5.0	(1, 0)
	4	3.9	0.5	−3.5	(1, 0)
	5	4.5	1.4	−3.0	(1, 0)

desired horizontal distance between UAVs, $efv^i = (efv_x^i, efv_y^i) = (V_{xy}^e cos\delta^i, V_{xy}^e sin\delta^i)$ denotes the desired flocking velocity vector of UAV i, δ^i denotes the desired flocking yaw angle of UAV i and $V_{xy}^e = 1$m/s is desired horizontal airspeed.

3. Simulation results

We evaluate our approach through the round and linear flight experiment of 5 UAVs in the Microsoft Airsim [26] simulation environment, and report the results on estimation error and flocking performance separately. In a 110-s flight test, we take images (resolution 640 * 480) every 0.5s and control drones based on the estimated state to gradually form $\alpha - lattice$ geometry structures and maintain its stability. Initial parameters of UAVs are shown in Table 1 and altitude of drone remains constant throughout the flight. The swarm flies round for the first 66s and linear for the rest of the time, which is achieved by setting the value of δ^i as shown in Equation (43).

$$\delta^i = \begin{cases} \dfrac{V_{xy}^e}{10} t \cdot \cdots \cdot 2\pi, & \text{if } t < 66s, \\ \dfrac{V_{xy}^e}{10} 66 \cdot \cdots \cdot 2\pi, & \text{else}, \end{cases} \tag{43}$$

3.1 Estimation error

In this section, we report the accuracy of transformation matrix and relative distance estimated by visual geometry method where the transformation matrix includes the rotation matrix described by yaw, pitch and roll angles, as well as the X and Y axis of the translation vector. The ground truth UAV state is collected from the simulation environment. As shown in Fig. 5, the estimation error decreases as the distance between drones gets closer and tends to be stable as the flocking converges. Statistical results of estimation error are shown in Table 2. According to data in the table, we can know that the displacement and relative distance error is stable in the order of centimetres, and the Euler Angle error is stable within 1 degree.

3.2 Flocking performance

In order to evaluate the visual geometry-based UAV flocking performance, we introduce two metrics: flocking velocity deviation obj_1^i indicates the degree of difference between the actual velocity of UAV i and the desired flocking velocity, and flocking shape deviation obj_2^i describes the degree of formation and stability of the $\alpha - lattice$ geometry structure. The equation is as follows:

$$obj_1^i = \begin{cases} -\dfrac{(V_x^i, V_y^i).(efv_x^i, efv_y^i)}{2 \left\| efv^i \right\|}, & \text{if } t < 66s, \\ \sum_{k=x,y} \left| efv_k^i - V_k^i \right|, & \text{else}, \end{cases} \tag{44}$$

https://doi.org/10.1017/aer.2022.112 Published online by Cambridge University Press

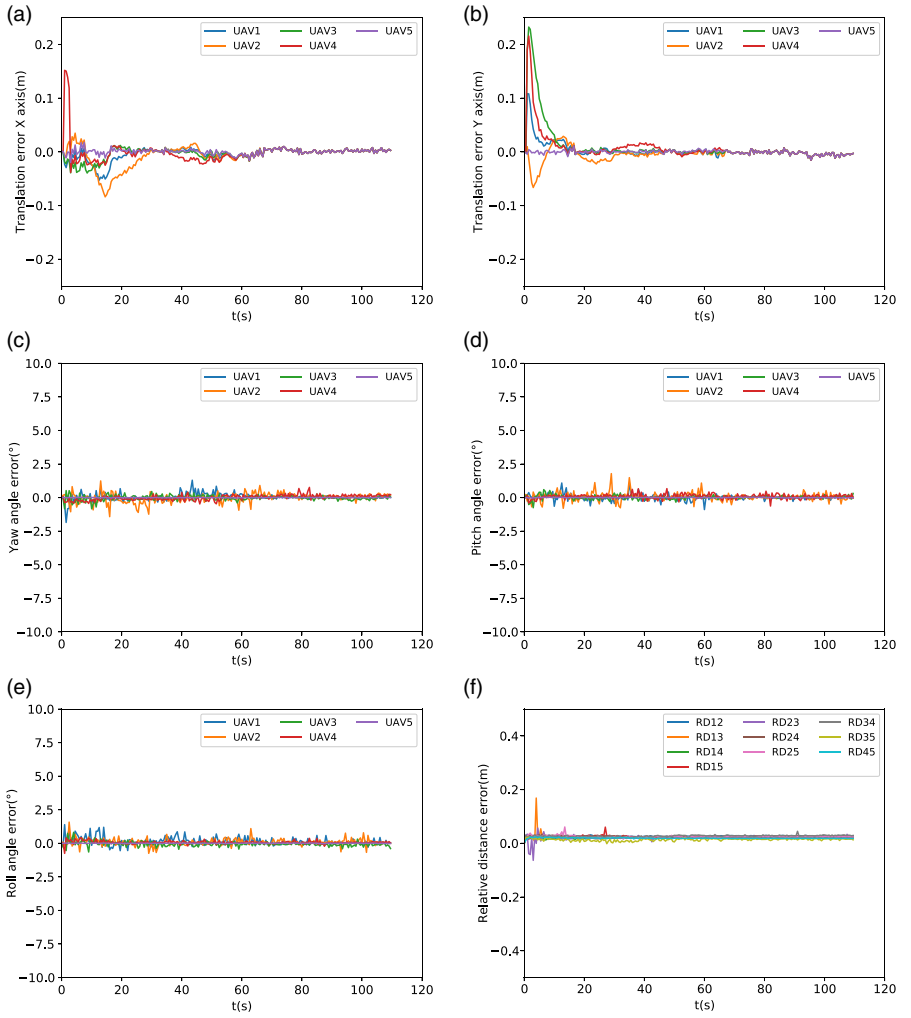

Figure 5. *Estimation error of translation vector, rotation matrix and relative distance: (a) translation error X axis, (b) translation error Y axis, (c) yaw angle error, (d) pitch angle error, (e) roll angle error and (f) relative distance error.*

$$obj_2^i = \sum_{j \in \left\{ d^{ij} \le D_c \right\}} \left[\left| D_d - d^{ij} \right| + \sum_{k=x,y} \left| V_k^i - V_k^j \right| \right] \tag{45}$$

Figure 6 depicts the simulation results of UAV flocking based on visual geometry where RD_{ij} denotes the relative distance between UAV i and UAV j, D_{l1} is the minimum distance to avoid collision between UAVs. As shown in Fig. 6(a), five UAVs successfully flew a very smooth circle, gradually formed an $\alpha - lattice$ geometry during the flight and kept this structure stable. As shown in Fig. 6(b), the difference of horizontal airspeed among UAVs was less than $0.34\,m/s$, and the horizontal airspeed of the UAV flocking quickly converges to the desired horizontal airspeed after fluctuating within the allowable range. As shown in Fig. 6(c), the UAV flocking has a good velocity tracking effect and a slight yaw angle difference. As shown in Fig. 6(d), relative distance between UAVs is converges rapidly and fluctuates less. Moreover, flocking geometry and collision avoidance component in the control input achieve the desired goal, which means that the drones are safe during the whole flight. As shown in Fig. 6(e) and (f), the shape and velocity of the UAV flocking converge rapidly and stably.

https://doi.org/10.1017/aer.2022.112 Published online by Cambridge University Press

Table 2. *Statistical results of estimation error*

Time(s)	TE_x^a(m)	TE_y^b(m)	Yaw^c(°)	$Picth^d$(°)	$Roll^e$(°)	RD^f(m)
< 66	0.1518	0.2324	1.8459	1.7812	1.5739	0.1683
> 66	0.0099	0.0155	0.7560	0.6231	0.7804	0.0333

[a]Maximum of translation error X axis.
[b]Maximum of translation error Y axis.
[c]Maximum of yaw angle error.
[d]Maximum of pitch angle error.
[e]Maximum of roll angle error.
[f]Maximum of relative distance error.

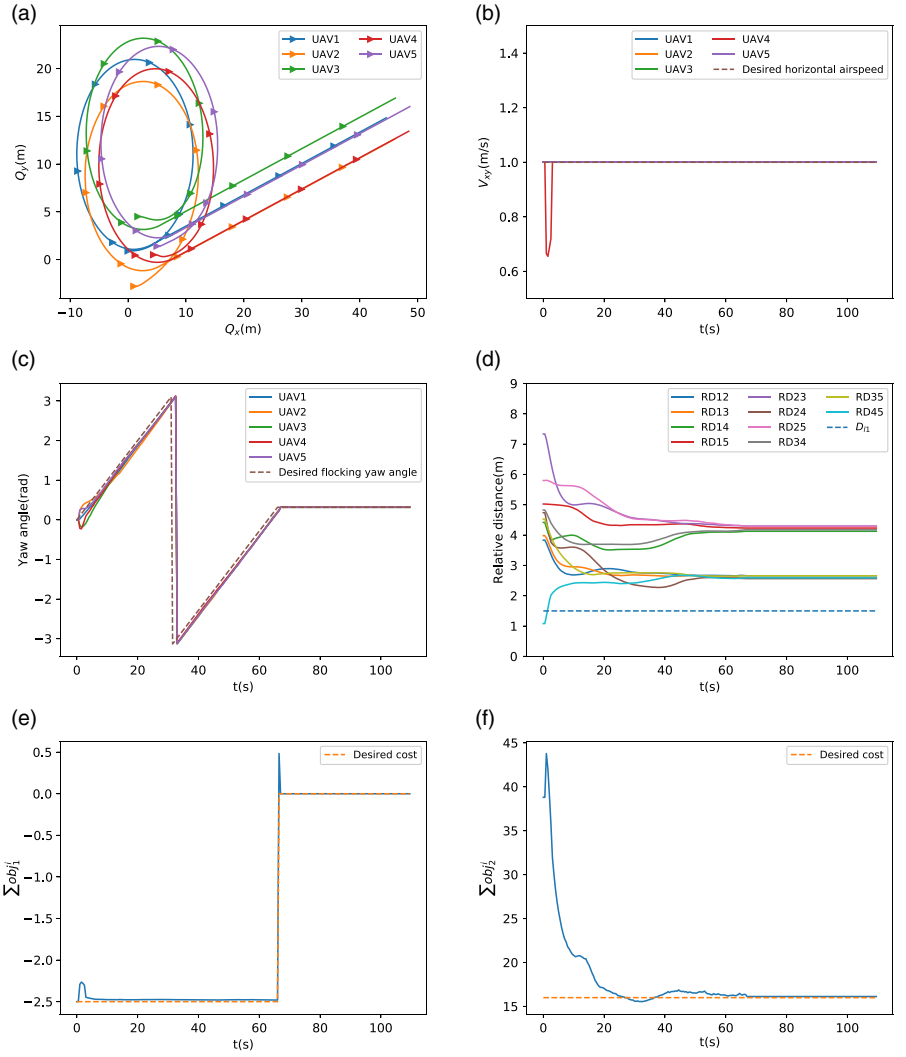

Figure 6. *Simulation results of UAV flocking based on visual geometry: (a) trajectory curves, (b) horizontal airspeed curves, (c) yaw angle curves, (d) relative distance curves, (e)* $\sum obj_1^i$ *curves and (f)* $\sum obj_2^i$ *curves.*

https://doi.org/10.1017/aer.2022.112 Published online by Cambridge University Press

Table 3. *Statistical results of two methods*

Algorithm	CTOBJ1[a](s)	CEOBJ1[b]	CTOBJ2[c](s)	CEOBJ2[d]
Ground truth[e]	67.0	0.0000	67.0	0.0206
Visual geometry[f]	67.0	0.0000	67.0	0.1273

[a]Convergence time of $\sum obj_1^i$.
[b]Convergence error of $\sum obj_1^i$.
[c]Convergence time of $\sum obj_2^i$.
[d]Convergence error of $\sum obj_2^i$.
[e]UAV flocking based on ground truth cluster state.
[f]UAV flocking based on visual geometry.

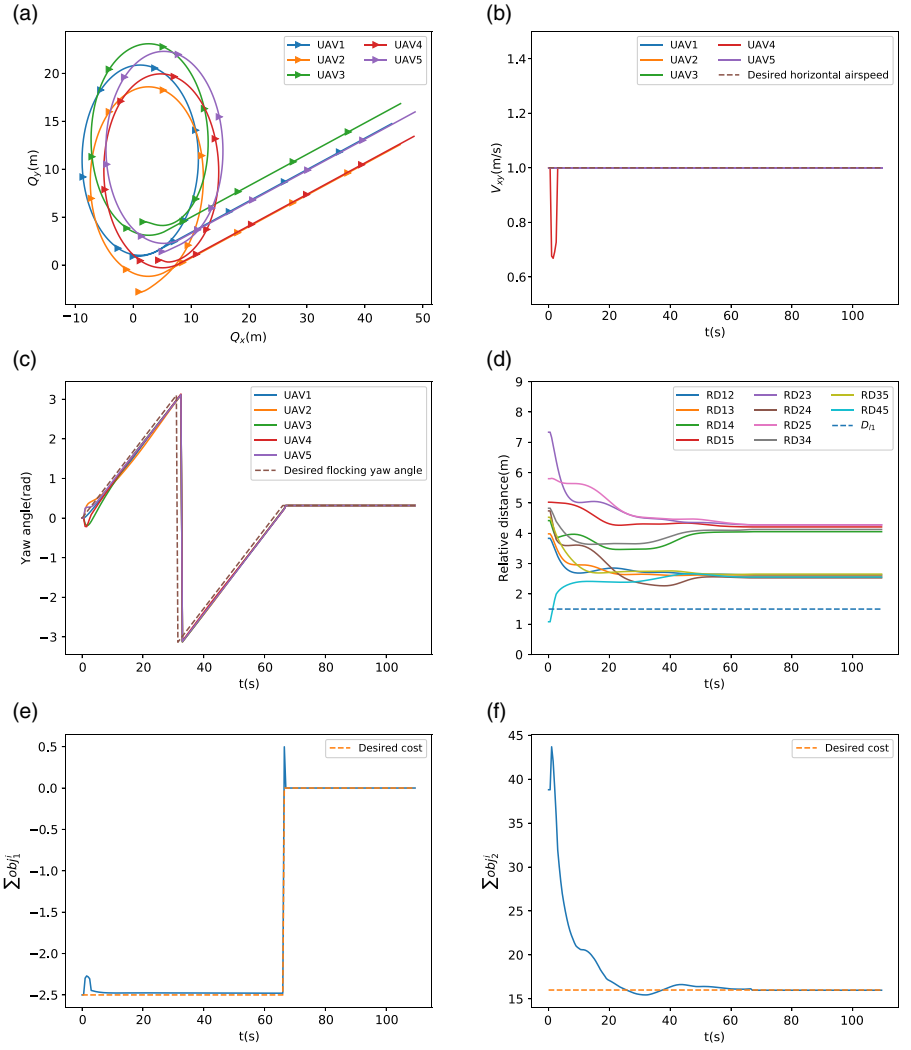

Figure 7. *Simulation results of UAV flocking based on ground truth cluster state: (a) trajectory curves, (b) horizontal airspeed curves, (c) yaw angle curves, (d) relative distance curves, (e) $\sum obj_1^i$ curves and (f) $\sum obj_2^i$ curves.*

https://doi.org/10.1017/aer.2022.112 Published online by Cambridge University Press

To illustrate the effectiveness of the proposed method, we compare it with the ground truth cluster state-based UAV flocking. As you can see from the data in the Table 3 or from the comparison between Figs. 6 and 7, there are small differences between the two results in terms of trajectory curves, horizontal airspeed curves, yaw angle curves, relative distance curves, $\sum obj_1^i$ curves and $\sum obj_2^i$ curves, indicating that the UAV flocking based on visual geometry achieves good results in performance.

4.0 Conclusions

In this paper, we propose a visual geometry-based UAV flocking method, which does not depend on specific visual markers and external infrastructure, nor does it require inter-UAV communication or prior knowledge of UAV size. The proposed approach is fully distributed, as each UAV relies only on the on-board monocular camera to collect images to estimate the transformation matrix of all UAVs and the relative distance between them, and use it for UAV flocking control. It combines the advantages of deep learning and classical geometry, and has good accuracy, interpretability and flocking performance. The deep optical flow network avoids the drawback that the feature-based matching method may not get enough effective matches in the occlusion, no texture, small targets scene and the dense matching obtained is more beneficial to the estimation of the centre of nearby UAV. In addition, depth map estimation method based on deep learning can extract the scale and scene prior knowledge from the training data. Experimental results in the Microsoft Airsim simulation environment show that our method achieves almost the same performance as UAV flocking algorithm based on ground truth cluster state in all evaluation metrics.

In the future, we will continue to deepen the research of vision-based UAV flocking technology and eventually verify the feasibility of the algorithm on large-scale swarm.

Acknowledgements. This research was funded by the Sichuan Science and Technology Program (No.2023YFG0174) and the Fundamental Research Funds for the Central Universities.

References

[1] Klausen, K., Meissen, C., Fossen, T.I., Arcak, M. and Johansen, T.A. Cooperative control for multirotors transporting an unknown suspended load under environmental disturbances, *IEEE Trans Contr Syst Technol*, 2020, **28**, (2), pp 653–660. https://doi.org/10.1109/TCST.2018.2876518

[2] Ma, J., Guo, D., Bai, Y., Svinin, M. and Magid, E. A vision-based robust adaptive control for caging a flood area via multiple UAVs, *18th International Conference on Ubiquitous Robots*, 2021, pp 386–391. https://doi.org/10.1109/UR52253.2021.9494698

[3] Khosravi, M., Enayati, S., Saeedi, H., and Pishro-Nik, H. Multi-purpose drones for coverage and transport applications, *IEEE Trans Wireless Commun*, 2021, **20**, (6), pp 3974–3987. https://doi.org/10.1109/TWC.2021.3054748

[4] Vásárhelyi, G., Virágh, C., Somorjai, G., Tarcai, N., Szörenyi, T., Nepusz, T. and Vicsek, T. Outdoor flocking and formation flight with autonomous aerial robots, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp 3866–3873. https://doi.org/10.1109/IROS.2014.6943105

[5] Lightbody, P., Krajník, T. and Hanheide, M. An efficient visual fiducial localisation system, *ACM SIGAPP Appl Comput Rev*, September 2017, **17**, (3), pp 28–37. https://doi.org/10.1145/3161534.3161537

[6] Ledergerber, A., Hamer, M. and D'Andrea, R. A robot self-localization system using one-way ultra-wideband communication, *IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS)*, 2015, pp 3131–3137. https://doi.org/10.1109/IROS.2015.7353810

[7] Wang, Z. and Gu, D. A local sensor based leader-follower flocking system, *IEEE International Conference on Robotics and Automation*, 2008, pp 3790–3795. https://doi.org/10.1109/ROBOT.2008.4543792

[8] Güler, S., Abdelkader, M. and Shamma, J.S. Infrastructure-free multi-robot localization with ultrawideband sensors, *American Control Conference (ACC)*, 2019, pp 13–18. https://doi.org/10.23919/ACC.2019.8814678

[9] Tang, Y., Hu, Y., Cui, J., Liao, F., Lao, M., Lin, F. and Teo, R.S. Vision-aided multi-UAV autonomous flocking in GPS-denied environment, *IEEE Trans Ind Electron*, 2019, **66**, (1), pp 616–626. https://doi.org/10.1109/TIE.2018.2824766

[10] Moshtagh, N., Jadbabaie, A. and Daniilidis, K. Vision-based control laws for distributed flocking of nonholonomic agents, *IEEE International Conference on Robotics and Automation*, 2006, pp 2769–2774. https://doi.org/10.1109/ROBOT.2006.1642120

[11] Moshtagh, N., Michael, N., Jadbabaie, A. and Daniilidis, K. Vision-based, distributed control laws for motion coordination of nonholonomic robots, *IEEE Trans Robot*, 2009, **25**, (4), pp 851–860. https://doi.org/10.1109/TRO.2009.2022439

https://doi.org/10.1017/aer.2022.112 Published online by Cambridge University Press

[12] Soria, E., Schiano, F. and Floreano, D. The influence of limited visual sensing on the reynolds flocking algorithm, *IEEE International Conference on Robotic Computing(IRC)*, 2019, pp 138–145. https://doi.org/10.1109/IRC.2019.00028
[13] Hu, T.K., Gama, F., Chen, T., Wang, Z., Ribeiro, A. and Sadler, B.M. Vgai: End-to-end learning of vision-based decentralized controllers for robot swarms, *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp 4900–4904. https://doi.org/10.1109/ICASSP39728.2021.9414219
[14] Schilling, F., Lecoeur, J., Schiano, F. and Floreano, D. Learning vision-based flight in drone swarms by imitation, *IEEE Robot Autom Lett*, 2019, **4**, (4), pp 4523–4530. https://doi.org/10.1109/LRA.2019.2935377
[15] Bastien, R. and Romanczuk, P. A model of collective behavior based purely on vision, *Sci Adv*, 2020, **6**, (6), pp 1–10. https://doi.org/10.1126/sciadv.aay0792
[16] Schilling, F., Schiano, F. and Floreano, D. Vision-based drone flocking in outdoor environments, *IEEE Robot Autom Lett*, 2021, **6**, (2), pp 2954–2961. https://doi.org/10.1109/LRA.2021.3062298
[17] Reynolds, C.W. Flocks, Herds, and Schools: A distributed behavioral model, *ACM SIGGRAPH Comput Graph*, July 1987, **21**, pp 25–34. https://doi.org/10.1145/280811.281008
[18] Wang, J., Zhong, Y., Dai, Y., Birchfield, S., Zhang, K., Smolyanskiy, N. and Li, H. Deep two-view structure-from-motion revisited, *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp 8949–8958. https://doi.org/10.1109/CVPR46437.2021.00884
[19] Ming, Y., Meng, X., Fan, C. and Yu, H. Deep learning for monocular depth estimation: A review, *Neurocomputing*, 2021, **438**, pp 14–33. https://doi.org/10.1016/j.neucom.2020.12.089
[20] Zhao, C., Sun, Q., Zhang, C., Tang, Y. and Qian, F. Monocular depth estimation based on deep learning: An overview, *Sci China Technol Sci*, September 2020, **63**, (9), pp 1612–1627. https://doi.org/10.1007/s11431-020-1582-8
[21] Teed, Z. and Deng, J. RAFT: Recurrent all-pairs field transforms for optical flow, *Computer Vision – ECCV 2020*, November 2020, pp 402–419. https://doi.org/10.1007/978-3-030-58536-5_24
[22] Asgari Taghanaki, S., Abhishek, K., Cohen, J.P., Cohen-Adad, J. and Hamarneh, G. Deep semantic segmentation of natural and medical images: A review, *Artif Intell Rev*, 2021, **54**, pp 137–178. https://doi.org/10.1007/s10462-020-09854-1
[23] Arun, K.S., Huang, T.S. and Blostein, S.D. Least-squares fitting of two 3-D point sets, *IEEE Trans Pattern Anal Machine Intell*, 1987, **PAMI-9**, (5), pp 698–700. https://doi.org/10.1109/TPAMI.1987.4767965
[24] Fischler, M.A. and Bolles, R.C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, Readings in *Computer Vision*, Morgan Kaufmann, San Francisco (CA), 1987, pp 726–740. https://doi.org/10.1016/B978-0-08-051581-6.50070-2
[25] He, T. and Wang, L. Neural network-based velocity-controllable UAV flocking, *Aeronaut J*, 2022, pp 1–16. https://doi.org/10.1017/aer.2022.61
[26] Shah, S., Dey, D., Lovett, C., and Kapoor, A. AirSim: High-fidelity visual and physical simulation for autonomous vehicles, *Spr Tra Adv Robot*, 2017, arXiv: 1705.05065. https://arxiv.org/abs/1705.05065.

Cite this article: Wang L. and He T. (2023). Vision geometry-based UAV flocking. *The Aeronautical Journal*, **127**, 1048–1064. https://doi.org/10.1017/aer.2022.112