J Bionic Eng 17 (2020) 215–228
DOI: https://doi.org/10.1007/s42235-020-0017-4

Journal of Bionic Engineering
http://www.springer.com/journal/42235

Natural Feature-based Visual Servoing for Grasping Target with an Aerial Manipulator

Bin Luo[1], Haoyao Chen[1*], Fengyu Quan[1], Shiwu Zhang[2], Yunhui Liu[3]

1. *School of Mechanical Engineering and Automation*, *Harbin Institute of Technology Shenzhen*, *Shenzhen* 518055, *China*
2. *Department of Precision Machinery and Precision Instrumentation*, *University of Science and Technology of China*, *Hefei* 230026, *China*
3. *Department of Mechanical and Automation Engineering*, *Chinese University of Hong Kong*, *Hong Kong* 999077, *China*

Abstract

Aerial transportation and manipulation have attracted increasing attention in the unmanned aerial vehicle field, and visual servoing methodology is widely used to achieve the autonomous aerial grasping of a target object. However, the existing marker-based solutions pose a challenge to the practical application of target grasping owing to the difficulty in attaching markers on targets. To address this problem, this study proposes a novel image-based visual servoing controller based on natural features instead of artificial markers. The natural features are extracted from the target images and further processed to provide servoing feature points. A six degree-of-freedom (6-DoF) aerial manipulator system is proposed with differential kinematics deduced to achieve aerial grasping. Furthermore, a controller is designed when the target object is outside a manipulator's workspace by utilizing both the degrees-of-freedom of unmanned aerial vehicle and manipulator joints. Thereafter, a weight matrix is used as basis to develop a multi-tasking visual servoing framework to integrate the controllers inside and outside the manipulator's workspace. Lastly, experimental results are provided to verify the effectiveness of the proposed approach.

Keywords: unmanned aerial manipulator, visual servoing, aerial grasping, multi-tasking strategy

1 Introduction

Unmanned Aerial Manipulator (UAM), which is a type of Unmanned Aerial Vehicles (UAV) equipped with a multiple Degrees-of-Freedom (DoF) robotic arm, is bio-inspired from flying birds and attracts increasing attention in robotics research. UAMs have immense potential for various applications, including express transportation[1–3], construction and maintenance[4–6], manipulation[7–9], and cooperative operations[10–12] in places that are dangerous and difficult to reach by humans or ground mobile robots. Grasping is an important application for mobile manipulators. Compared with mobile manipulators based on mobile ground robots, UAMs continue to present significant challenges in perception and control mainly because of the considerably complex kinematics/dynamics and motion constraints of the coupled UAV-manipulator system.

Several approaches have been developed to achieve aerial grasping. Garimella *et al.*[13] proposed a nonlinear model–predictive control method to exploit multi-body system dynamics and achieve optimized performance. In Ref. [14], a controller that uses a multi-level architecture was proposed, in which the outer loop is composed of a trajectory generator and an impedance filter that modifies the trajectory to achieve compliant behavior in an end-effector space. Seo *et al.*[13] developed a method of generating locally optimal trajectory for grasping in constrained environments. In Ref. [16], an online active set strategy was applied to generate a feasible trajectory of the manipulator joints and a series of tasks with defined limitations and constraint inequalities were implemented. In Ref. [17], the trajectory planning of aerial grasping was modeled as a multi-objective optimization problem and motion constraints and collision avoidance were considered in the optimization. The aforementioned approaches should provide a priori knowledge on the target position and are difficult to apply when the target's position is unknown. In addition, their reference trajectories are generated in advance, thereby possibly leading to grasping failure due to movements of the target or the actual UAM.

***Corresponding author:** Haoyao Chen
E-mail: hychen5@hit.edu.cn

By contrast, some approaches have been developed for grasping on the basis of visual information, thereby compensating for the disturbance and control dynamic system. The existing approaches generally rely on extracting, tracking, and matching a set of visual features, such as points, lines, circles, or moments. Thomas *et al.*[18,19] proposed a vision-based controller inspired by a rapidly moving hawk in the act of catching fish. The grasping controller was accomplished by mapping the dynamics of a quadrotor to a virtual image plane, thereby enabling dynamically feasible trajectory planning in the image space. The target to be grasped should be covered with pure color. Kim *et al.*[20] developed an Image-Based Visual Servoing (IBVS) controller with image moment to obtain velocity control on the basis of the dynamic model of the entire system. Seo *et al.*[21] formulated the visual servoing problem as a stochastic model-predictive control framework to grasp a cylindrical object using an Aerial Manipulator (AM). Lippiello *et al.*[22,23] proposed a hybrid Visual Servoing (VS) with task priority control by sequentially considering several constraints[24]. In Ref. [25], an uncalibrated IBVS strategy was developed on the basis of a safety-related primary task. In Ref. [26], a complete second-order visual-impedance control law was designed for a dual-arm UAM. The visual information from the camera of one robotic arm was used to assist the assigned task to be executed by the other robotic arm. Fang *et al.*[17] modeled the grasping operation as an optimization problem and utilized visual information to compensate for the motion disturbance from the UAV body.

Note that the existing vision-based approaches for aerial grasping require artificial markers attached on target objects. However, these approaches could not be applied in the majority of the practical applications because of the inconvenience of attaching artificial markers on targets. Accordingly, some approaches have developed direct VS, which does not use classic image features but other image information, such as the luminance of all pixels[27], histograms[28], and deep neural networks[29]. In Ref. [27], the luminance of all pixels in the image was considered and did not require any tracking or matching process. In Ref. [28], the probability of occurrence in intensity, color, or gradient orientation in an image was calculated as a histogram and used for VS. In Ref. [29], a convolutional neural network was used to estimate the position and orientation between the current and desired images and a Position-Based Visual Servoing (PBVS) controller was considered to reach the desired pose. However, these approaches are developed for large targets and suffer from high computation cost. Consequently, directly using them for application in aerial grasping is difficult.

The present study aims to develop a novel VS controller for real-time aerial grasping control using natural features. The features of the VS are extracted in real time from a camera mounted on an end-effector of a manipulator, thereby avoiding the need to attach artificial markers on targets. To further address the limited workspace of the manipulator, we develop a hybrid VS framework on the basis of a weight matrix, thereby achieving VS control either outside or inside the manipulator's workspace. The main contributions of this research are threefold.

(1) The differential kinematics of the proposed AM is derived and describes the relationship between the camera's velocity and UAMs' joint and body velocities. The kinematics is used for the design of the VS controller.

(2) A novel VS controller is designed for aerial grasping based on ORB features, which can be extracted faster than SIFT and SURF. The designed controller can achieve successful grasping of an object without attaching markers. Experiments are performed to illustrate the effectiveness of the new controller compared with the one with attached artificial markers.

(3) A hybrid servoing strategy is further developed to solve the problem of limited workspace of the manipulator. When UAM is distant from the target, one of the manipulator joints is used for servoing to retain the target in the camera view. After the target is within the workspace of its manipulator, only the controller with the manipulator is used to grasp the target. The two control processes are combined into a hybrid formulation by utilizing a weight matrix.

2 Modeling of AM

Inspired by flying birds, an UAM is developed in the

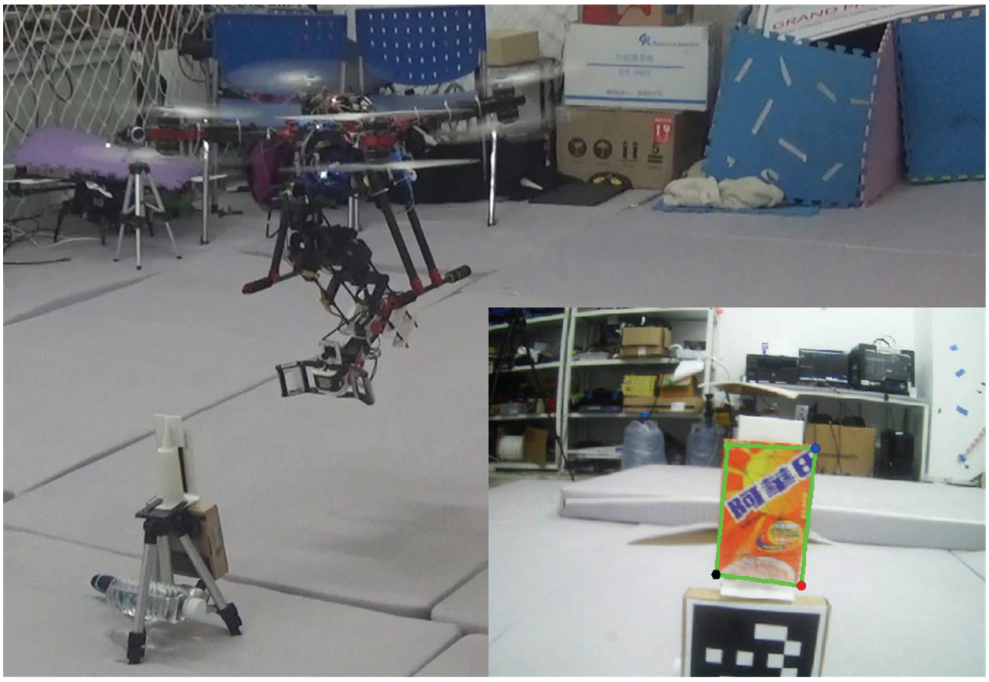


Fig. 1 Aerial manipulator system with a 6-DoF serial manipulator equipped under a co-axis octocopter. The bottom-right sub-figure shows an image of the target from the camera's view.

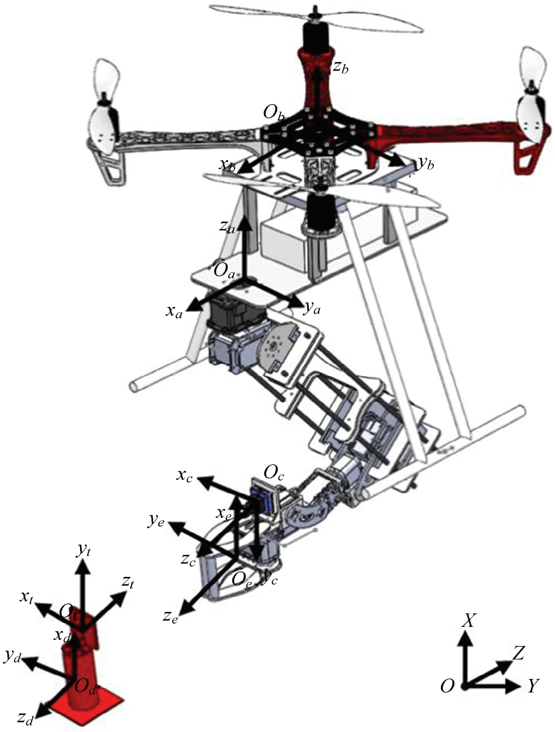


Fig. 2 Coordinate frames of the aerial manipulator system.

current study to achieve aerial grasping (Fig. 1). UAM consists of an X8 coaxial octocopter and a 6-DoF serial manipulator with a camera mounted on the end-effector. The 3D CAD model of the UAM system is labeled with the coordinate frames (Fig. 2). Let Σ_o be the inertial frame, Σ_b the body-fixed frame of the octocopter, Σ_a the base frame of the manipulator, Σ_e the end-effector frame of the manipulator, Σ_c the camera frame, Σ_t the object frame, and Σ_d the object's handle frame. Let $\boldsymbol{o}_b = [x_b, y_b, z_b]^{\mathrm{T}} \in \mathbb{R}^{3\times1}$ and $\boldsymbol{R}_b \in \boldsymbol{SO}(3)$ be the translation vector and rotation matrix of Σ_b with respect to Σ_o, $\boldsymbol{\mu}_b = [\varphi, \phi, \psi]^{\mathrm{T}} \in \mathbb{R}^{3\times1}$ the Eular angles of the octocopter's attitude including pitch, roll and yaw angles, and $\boldsymbol{q} = [\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6]^{\mathrm{T}} \in \mathbb{R}^{6\times1}$ the angle vector of the manipulator joints. We define $\boldsymbol{x}_s = [\boldsymbol{o}_b^{\mathrm{T}}, \boldsymbol{\mu}_b{}^{\mathrm{T}}, \boldsymbol{q}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{3\times1}$ as the system state and $\dot{\boldsymbol{x}}_s = [\dot{\boldsymbol{o}}_b^{\mathrm{T}}, \boldsymbol{\omega}_b^{\mathrm{T}}, \dot{\boldsymbol{q}}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{3\times1}$, $\boldsymbol{\omega}_b = [\omega_x, \omega_y, \omega_z]^{\mathrm{T}} \in \mathbb{R}^{3\times1}$ is the angular velocity vector of frame Σ_b and $\dot{\boldsymbol{q}}$ is the angular velocity vector of the manipulator joints. Fig. 3 shows the coordinate system derived from the Denavit–Hartenberg (DH) parameters of the manipulator. Table 1 provides the DH parameters.

Given the DH parameters, the end-effector pose in the manipulator's base frame Σ_a can be obtained from the following transformation matrix:

$$\boldsymbol{T}_e^a = \boldsymbol{T}_1^a(\theta_1)\boldsymbol{T}_2^1(\theta_2)\dots\boldsymbol{T}_6^5(\theta_6) = \begin{bmatrix} \boldsymbol{R}_e^a & \boldsymbol{t}_e^a \\ \boldsymbol{0}^{1\times3} & 1 \end{bmatrix}, \qquad (1)$$

where

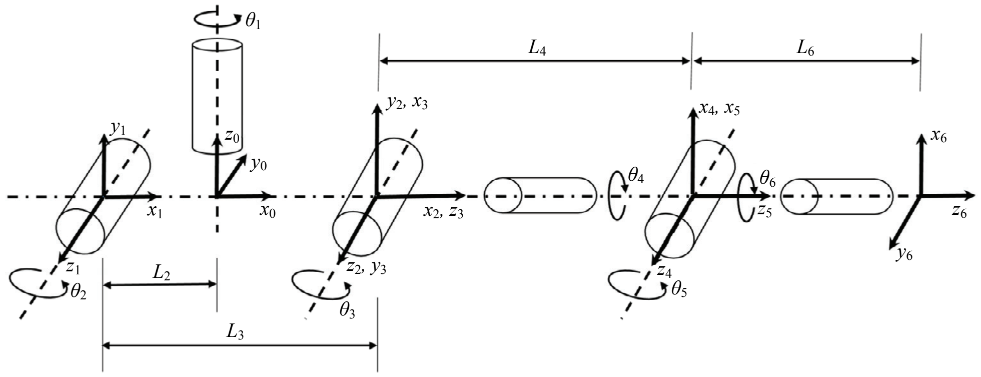


Fig. 3 Coordinate frames of the 6-DoF manipulator.

Table 1 Denavit–Hartenberg parameters of the 6-DoF manipulator

Link i	α_i	a_i	θ_i	d_i
1	$\pi/2$	$-L_2$	θ_1	0
2	0	L_3	θ_2	0
3	$-\pi/2$	0	$\theta_3 + \pi/2$	0
4	$-\pi/2$	0	θ_4	L_4
5	$\pi/2$	0	θ_5	0
6	0	0	θ_6	L_6

$$\boldsymbol{T}_i^{i-1} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

denotes the $\boldsymbol{SE}(3)$ transformation of the coordinate frame i in the coordinate frame $i - 1$, and $\boldsymbol{R}_e^a \in \boldsymbol{SO}(3)$ and $\boldsymbol{t}_e^a$ denote the rotation matrix and translation vector, respectively, of frame Σ_e in frame Σ_a.

The differential kinematics of the manipulator is required to propagate the camera velocity derived from the VS controller to the state velocity $\dot{\boldsymbol{x}}_s$. The definition of the serial manipulator's forward kinematics[30] indicates that the relationship between the velocities of the joints and end-effector is given as:

$$\boldsymbol{v}_e^a = \boldsymbol{J}_e^a \dot{\boldsymbol{q}}, \quad (3)$$

where $\boldsymbol{v}_e^a \in \mathbb{R}^{6\times1}$ denotes the velocity of the end-effector in the base frame Σ_a and $\boldsymbol{J}_e^a$ denotes the geometric Jacobian matrix as:

$$\boldsymbol{J}_e^a = \begin{bmatrix} \boldsymbol{J}_{p1} & \boldsymbol{J}_{p2} & \cdots & \boldsymbol{J}_{p6} \\ \boldsymbol{J}_{o1} & \boldsymbol{J}_{o2} & \cdots & \boldsymbol{J}_{o6} \end{bmatrix}, \quad (4)$$

where $\boldsymbol{J}_{pi} \in \mathbb{R}^{3\times1}$ is the position Jacobian matrix components and $\boldsymbol{J}_{oi} \in \mathbb{R}^{3\times1}$ denotes the orientation of the Jacobian matrix components.

To facilitate the model deduction, the geometric Jacobian matrix is transferred to the analytical Jacobian matrix $\boldsymbol{J}^e(\boldsymbol{q})$ with respect to the end-effector coordinate frame, which is given as:

$$\boldsymbol{J}^e(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{R}_e^a & 0^{3\times3} \\ 0^{3\times3} & \boldsymbol{R}_e^a \end{bmatrix}^{\mathrm{T}} \boldsymbol{J}_e^a(\boldsymbol{q}) = \boldsymbol{J}_t \boldsymbol{J}_e^a(\boldsymbol{q}), \quad (5)$$

where $\boldsymbol{J}_t \in \mathbb{R}^{6\times6}$ denotes the transformation matrix from the geometric Jacobian matrix to the analytical Jacobian matrix.

During aerial manipulations, the movements of the octocopter body and equipped manipulator are combined, thereby affecting each other. The velocity $\boldsymbol{v}^e \in \mathbb{R}^{6\times1}$ of the manipulator's end-effector in its own coordinate frame is equal to the resultant of the velocity vectors derived from the octocopter body and manipulation. The velocity $\boldsymbol{v}^e$ is provided as:

$$\boldsymbol{v}^e = \boldsymbol{v}^{e\_a} + \boldsymbol{v}^{e\_b}, \quad (6)$$

where $\boldsymbol{v}^{e-a} \in \mathbb{R}^{6\times1}$ denotes the velocity component caused by the movement of the manipulator and $\boldsymbol{v}^{e-b} \in \mathbb{R}^{6\times1}$ represents the velocity component caused by the movement of the octocopter. For the special case where the manipulator's joint configuration remains constant and only the octocopter is moving, $\boldsymbol{v}^e$ can be obtained from the octocopter's body velocity as:

$$\boldsymbol{v}^e := \boldsymbol{v}^{e\_b} = \begin{bmatrix} \boldsymbol{R}_b^e & -\boldsymbol{R}_b^e \boldsymbol{S}(\boldsymbol{t}_b^e) \\ \boldsymbol{0}^{3\times3} & \boldsymbol{R}_b^e \end{bmatrix} \boldsymbol{v}^b = \boldsymbol{J}_1 \boldsymbol{v}^b, \quad (7)$$

where $\boldsymbol{R}_b^e \in \mathbb{R}^{3\times3}$ and $\boldsymbol{t}_b^e \in \mathbb{R}^3$ denote the orientation and position, respectively, of the body-fixed frame in the end-effector coordinate frame; $\boldsymbol{v}^b = [\boldsymbol{o}_b^{\mathrm{T}}, \omega_b^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{6\times1}$ denotes the linear and angular velocity vectors in the body-fixed frame of UAV; and $\boldsymbol{S}(*)$ denotes the anti-symmetric matrix operation.

For another special case where the octocopter hovers with only the manipulator moving, $\boldsymbol{v}^e$ is obtained from Eqs. (3) and (5) as:

$$\boldsymbol{v}^e := \boldsymbol{v}^{e\_a} = \boldsymbol{J}_t \boldsymbol{J}_e^a(\boldsymbol{q})\dot{\boldsymbol{q}} = \boldsymbol{J}_2\dot{\boldsymbol{q}}. \tag{8}$$

Combining Eqs. (6) – (8), the differential kinematic model of the UAM system is derived as:

$$\boldsymbol{v}^e = \begin{bmatrix}\boldsymbol{J}_1 & \boldsymbol{J}_2\end{bmatrix}\begin{bmatrix}\boldsymbol{v}^b \\ \dot{\boldsymbol{q}}\end{bmatrix} = \boldsymbol{J}\boldsymbol{v}, \tag{9}$$

where $\boldsymbol{v} = \dot{\boldsymbol{x}}_s$ and $\boldsymbol{J} \in \mathbb{R}^{6\times12}$ is the joint Jacobian matrix.

As an under-actuated system, only the position and yaw angle of the octocopter are controllable. Accordingly, we define the controlled variables as a new vector $\boldsymbol{v}_c = [\dot{\boldsymbol{o}}_b^{\mathrm{T}}, \dot{\psi}_b, \dot{\boldsymbol{q}}^{\mathrm{T}}]^{\mathrm{T}}$ and the uncontrollable angular velocities as $\boldsymbol{v}_{uc} = [\dot{\varphi}_b, \dot{\phi}_b]^{\mathrm{T}}$. The value of $\boldsymbol{v}_{uc}$ can be measured using the onboard Inertial Measurement Unit (IMU). Eq. (9) is rewritten as follows under the assumption of a classical time-scale separation between the attitude controller and velocity controller[22]:

$$\boldsymbol{v}^e = \boldsymbol{J}\boldsymbol{v} = \boldsymbol{J}_c\boldsymbol{v}_c + \boldsymbol{J}_{uc}\boldsymbol{v}_{uc}, \tag{10}$$

where $\boldsymbol{J}_c \in \mathbb{R}^{6\times10}$ and $\boldsymbol{J}_{uc} \in \mathbb{R}^{6\times2}$ are two task Jacobian matrices for the controlled and uncontrolled state variables, respectively. In practical applications, UAV will generally remain stable (*i.e.*, the pitch and roll angles approximate 0). Therefore, we assume that uncontrolled state variables do not affect the velocity. Hence, Eq. (10) is approximated as:

$$\boldsymbol{v}^e = \boldsymbol{J}\boldsymbol{v} = \boldsymbol{J}_c\boldsymbol{v}_c + \boldsymbol{J}_{uc}\boldsymbol{v}_{uc} \approx \boldsymbol{J}_c\boldsymbol{v}_c. \tag{11}$$

3 Visual servoing based on natural features

The VS methodology[26] can continuously and effectively control a robot to approach a target on the basis of the target's image or position information. To control the AM to automatically grasp a target object, VS technology is utilized owing to its high performance. In addition, as we know, the manipulator will degrade the UAV stability due to the force coupling between the UAV body and manipulator, and thus the grasping control needs being carefully designed. This problem will be also addressed by using the VS control technology. Because the tracking errors of feature points in the image frame are considered to achieve the grasping control, the force coupling between the UAV body and manipulator will not affect the grasping process. The traditional IBVS and PBVS approaches[31] typically use artificial patterns, such as color patches or QR tags, to obtain features and location information. Such patterns are required to be attached to a target. However, attaching an artificial pattern on an object is generally impossible in many practical applications. To address this problem, the current study designed a novel approach by using natural features on the targets instead of attaching artificial patterns. Fig. 4 illustrates the algorithm framework of the proposed natural feature-based VS controller for aerial grasping. In the section, the VS controller for aerial grasping is first designed, followed with the feature detection method that provides critical feedback information for the servoing controller.

3.1 Design of the visual servoing control law

IBVS is utilized to realize the VS control of aerial grasping, because it exhibits advantages of low computation burden and high accuracy. Similar with the classic controller design[31], we utilize feature points in the image coordinate frame to control the end-effector toward its target pose. The challenge of this study, which differs from the existing research, lies in the natural feature detection and coupled effect between the UAV body and manipulator. From the camera model[31] and given a feature point, we obtain the following equation:

$$\dot{\boldsymbol{x}} = \boldsymbol{L}_x\boldsymbol{v}^e, \tag{12}$$

where

$$\boldsymbol{L}_x = \begin{bmatrix} -1/Z & 0 & x/Z & xy & -\left(1+x^2\right) & y \\ 0 & -1/Z & y/Z & (1+y^2) & -xy & -x \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{2\times6}, \tag{13}$$

denotes the interaction matrix, $\boldsymbol{x} = [x, y]^{\mathrm{T}} \in \mathbb{R}^{2\times1}$ denotes the coordinates of the feature point in the image

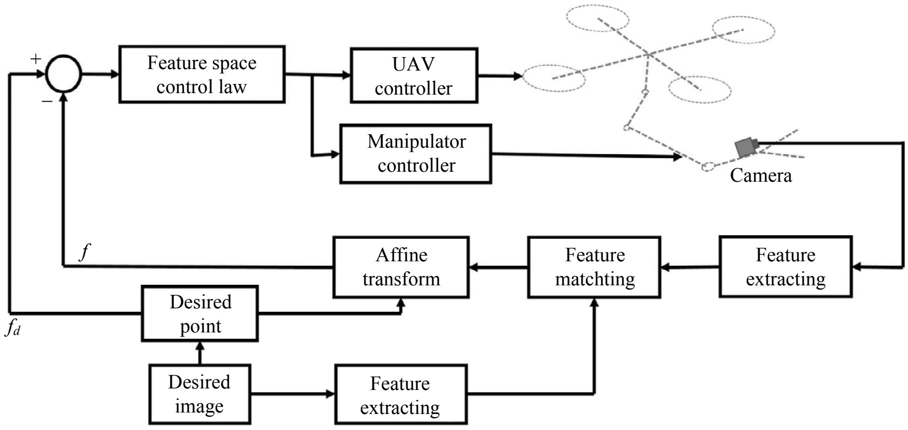


Fig. 4 Control framework of the proposed VS controller.

frame, $\boldsymbol{v}^e \in \mathbb{R}^{6\times1}$ denotes the camera velocity in the camera frame, and Z is the depth value of the feature in the camera frame. Furthermore, the calculation of the camera's 6D velocity requires at least three feature points in an image frame. For redundancy, four servoing point features are used in the current study for VS. By stacking all feature information, Eq. (12) is rewritten as:

$$\dot{\boldsymbol{X}} = \begin{bmatrix} \dot{x}_1 \\ \dot{y}_1 \\ \vdots \\ \dot{x}_4 \\ \dot{y}_4 \end{bmatrix} = \begin{bmatrix} \boldsymbol{L}_1 \\ \vdots \\ \boldsymbol{L}_4 \end{bmatrix} \boldsymbol{v}^e = \boldsymbol{L}_c \boldsymbol{v}^e, \tag{14}$$

where $\dot{\boldsymbol{X}} \in \mathbb{R}^{8\times1}$ denotes the velocity vector of the four points in the image frame; $\boldsymbol{L}_i, i \in (1,\ldots,4)$ is the interaction matrix for the ith feature, as defined in Eq. (13); and $\boldsymbol{L}_c \in \mathbb{R}^{8\times6}$ denotes the stacked interaction matrix.

The error between the current position $\boldsymbol{X}$ and desired constant position $\boldsymbol{X}^*$ is defined as:

$$\boldsymbol{e} = \boldsymbol{X} - \boldsymbol{X}^*. \tag{15}$$

Thereafter, we obtain the following equation:

$$\dot{\boldsymbol{e}} = \boldsymbol{L}_c \boldsymbol{v}^e. \tag{16}$$

We design $\dot{\boldsymbol{e}} = -\lambda e$ to guarantee an exponentially asymptotical convergence of the error. Lastly, the VS control law is designed as:

$$\boldsymbol{v}^e = -\lambda \boldsymbol{L}_c^+ \boldsymbol{e}, \tag{17}$$

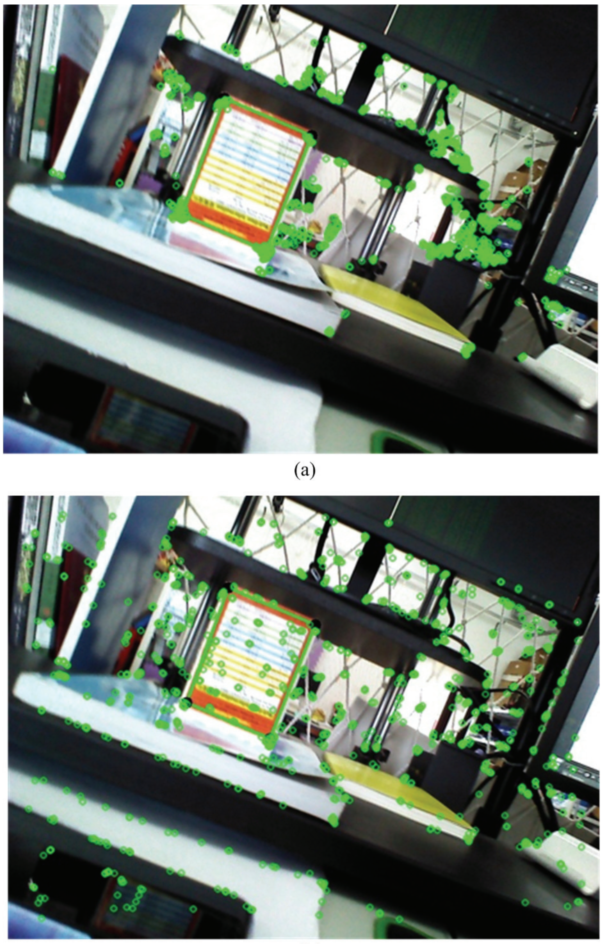
where $\boldsymbol{L}_c^+$ denotes the pseudo-inverse of $\boldsymbol{L}_c$.

Eq. (13) shows that the coordinates of the four features in the image frame for each control loop and their depth values should be obtained. The following subsection presents the calculation of the features' image coordinates and depth values based on natural features.

3.2 Servoing points generation based on the ORB feature extractor

Many feature extraction algorithms have been developed in the field of computer vision[32], where SIFT[33], SURF[34] and ORB[35] are the three most popular algorithms. Although the SIFT and SURF features have good robustness and stability, they are difficult to achieve in terms of real-time performance on feature detection. The ORB features are highly invariant to viewpoint and robust for feature matching among the different visual views. Compared with SIFT and SURF, ORB exhibits faster computation and matching. The existing literature provides a detailed comparison[35]. The ORB feature extractor is applied in the current study to provide the natural features because viewpoint invariance and real-time performance are required for VS. ORB, *i.e.*, oriented FAST and Rotated BRIEF, is developed based on FAST corners[35] but assigning the corners with orientation. Furthermore, the feature descriptor rotated BRIEF is also used to ensure the feature matching process with low computation burden and high matching accuracy.

The original ORB extractor suffers from such problems as uneven distribution and feature redundancy. To address this concern, we utilize the quad-tree homogenization algorithm[36] to sparse and homogenize the



(a)

(b)

Fig. 5 Feature detection under the (a) original and (b) improved ORB.

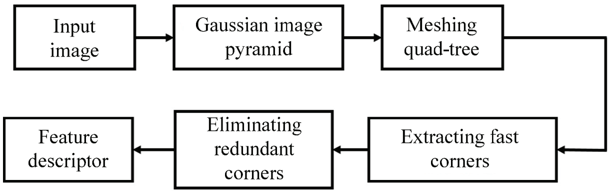


Fig. 6 Flow diagram of ORB.

feature distribution on an image. To detect sufficient feature points in each region on an image, a small threshold is selected to re-detect ORB features on areas with only a few corners. Thereafter, the best key points are selected through non-maximum suppression based on the response values. A demonstration result is shown in Fig. 5b, where the distribution of point features has been improved compared with the original one (Fig. 5a). From the figure, feature points are clustered on objects with uneven distribution under the original method, whereas the distribution of feature points is more uniform under the improved one. The improved distribution benefits the feature matching process. The flow diagram of the feature extraction process is illustrated in Fig. 6.

It is noting that ORB is unable to provide 3D information with only a monocular system. In addition, ORB has difficulty in providing life-time stable features through the entire servoing process especially in high dynamic range environments. Therefore, the extracted ORB features cannot be used directly as the servoing points mentioned in Eq. (14). To solve this problem, we develop an efficient method for generating a servoing point based on the homography matrix, which is calculated from the matched ORB features between frames. In a manner similar to that of the classical VS methods, an image containing the target object is selected as the desired image and the target's size in the world coordinate frame is assumed to be a priori known. The ORB features of the target object are extracted from the desired image and stored for further feature matching. Four servoing points are also defined around the ORB features on the target. Note that these points are virtual and undetected by the ORB features.

The ORB features of the consequent images captured from the camera are extracted in real time and matched with the stored ORB features in the desired image. Thereafter, several pairs of the ORB features are obtained on the bases of the matching results. Given the feature pairs, a homography matrix $\boldsymbol{H} \in \mathbb{R}^{3\times 3}$ is obtained as:

$$\boldsymbol{p}_2 = \boldsymbol{H}\boldsymbol{p}_1, \tag{18}$$

where $\boldsymbol{p}_1 = [x_1, y_1, 1]^{\mathrm{T}} \in \mathbb{R}^{3\times 1}$ and $\boldsymbol{p}_2 = [x_2, y_2, 1]^{\mathrm{T}} \in \mathbb{R}^{3\times 1}$ are matched with the feature points in images I_1 and I_2, respectively; and

$$\begin{aligned} \boldsymbol{H} &= \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \\ &= s\boldsymbol{K}\boldsymbol{E} = s\begin{bmatrix} f_x & 0 & 0 & 0 \\ 0 & f_y & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} R_{00} & R_{01} & t_x \\ R_{10} & R_{11} & t_y \\ R_{20} & R_{21} & t_z \\ 0 & 0 & 1 \end{bmatrix}, \end{aligned} \tag{19}$$

where $s \in \mathbb{R}$ denotes the scale factor; $\boldsymbol{K} \in \mathbb{R}^{3\times 4}$ represents the camera projection matrix; f_x and f_y refer to the focal length of the x- and y-axes, respectively; and $\boldsymbol{E} \in \mathbb{R}^{4\times 3}$ denotes the truncated extrinsic matrix, and R_{ij} denotes ith row and jth column element of the rotation part in the extrinsic matrix. The homography matrix contains information on the position and orientation of the target frame relative to the camera frame. The extraction of extrinsic information from the homography matrix requires other parameters, including camera intrinsic matrix and physical size of the object. The extrinsic transformation $\boldsymbol{T} \in \boldsymbol{SE}(3)$ between the current and desired camera frames is calculated without scale problem[37] because the size of the target object is a priori known. RANSAC is applied to produce robust results because outliers may exist in the feature pairs. Thereafter, the corresponding points in the current image are calculated by reprojecting the desired servoing points to the current image using the extrinsic transformation $\boldsymbol{T}$ owing to a priori defined servoing points on the target in the desired image.

4 Multi-task visual servoing strategy

The manipulator is equipped at the bottom of the UAV body. Thus, the motions of the UAV body and manipulator are coupled. Therefore, controlling the aerial grasping is easier than the dynamic grasping when UAV is hovering without motion. However, when UAV is distant from the target, target grasping is outside the workspace of the manipulator, thereby preventing the accomplishment of grasping with UAV hovering. To solve this problem, the VS control methodology is utilized to drive the UAM system to enter the workspace. Moreover, DoF of the UAV and manipulator are simultaneously controlled. The camera at the end-effector is reused as the VS sensor for convenience. If only the DoF of UAV is used for VS, then the target easily leaves the field view of the camera. Thus, we use the manipulator joints to address the aforementioned concern. To maintain the field of view, the idea is to use UAV to provide yaw servoing while using the manipulator to provide pitch servoing. Once the AM system enters into the manipulator workspace, VS (Eq. (17)) is utilized thereafter.

The manipulator has 6 DoFs, which are redundant for achieving pitch servoing. The 2nd, 3rd, and 5th joints can be used to drive the camera and maintain the field of view (Fig. 3). Thereafter, a weight matrix is introduced in Eq. (11) as:

$$\boldsymbol{v}^e = \boldsymbol{J}_c \boldsymbol{W}\boldsymbol{W}^{\mathrm{T}} \boldsymbol{v}_c, \tag{20}$$

where $\boldsymbol{W} \in \mathbb{R}^{10\times n}$ denotes the weight matrix, $\boldsymbol{W}^{\mathrm{T}}\boldsymbol{v}_c$ pertains to the vector of the actual control variable, and n is the number of the actual control variables. Matrix $\boldsymbol{W}$ is used to activate different control variables based on the control strategy. For simplicity, we use only one of the manipulator joints to provide additional freedom for the camera. The 5th joint is selected and used for pitch angle servoing because the 2nd and 3rd joints are distant from the camera's installation position and their movements cause a substantial change in the camera's field view. Thereafter, the corresponding weight matrix $\boldsymbol{W}$ is provided as:

$$\boldsymbol{W} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}^{\mathrm{T}}. \tag{21}$$

In addition, Eq. (20) is simplified as:

$$\boldsymbol{v}^e = \left[\boldsymbol{J}_1(1:3)\boldsymbol{J}_1(6)\boldsymbol{J}_2(5)\right] \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\omega}_z \\ \dot{\theta}_5 \end{bmatrix}, \tag{22}$$

where $\boldsymbol{J}_1(1:3)$ denotes the first three columns of $\boldsymbol{J}_1$; $\boldsymbol{J}_1(6)$ stands for the 6th column of $\boldsymbol{J}_1$; and $\boldsymbol{J}_2(5)$ represents the 5th column of $\boldsymbol{J}_2$. Moreover, $\boldsymbol{J}_1$ and $\boldsymbol{J}_2$ are defined in Eqs. (7) and (8), respectively. With the $\boldsymbol{W}$, UAV's state variables x, y, z, and yaw, and the angular velocity of the 5th manipulator joint are controlled.

On the basis of the controller (Eq. (22)), UAM will approach the target object. Once the target object is located in the workspace of the manipulator, the VS control will be switched to utilize the manipulator only while maintaining UAV hovering. We can design these two control tasks (*i.e.*, VS inside and outside the manipulator's workspace) using a common framework by using a weight matrix. Thus, we design the aforementioned weight matrix as:

$$
\boldsymbol{W}=\begin{bmatrix}0&0&0&0&1&0&0&0&0&0\\0&0&0&0&0&1&0&0&0&0\\0&0&0&0&0&0&1&0&0&0\\0&0&0&0&0&0&0&1&0&0\\0&0&0&0&0&0&0&0&1&0\\0&0&0&0&0&0&0&0&0&1\end{bmatrix}^{\mathrm{T}}. \quad (23)
$$

Lastly, the following hybrid control law is provided on the bases of Eqs. (17) and (20):

$$
\boldsymbol{W}^{\mathrm{T}}\boldsymbol{v}_c=-\lambda\left(\boldsymbol{J}_c\boldsymbol{W}\right)^{+}\boldsymbol{L}_c^{+}\boldsymbol{e}. \quad (24)
$$

The desired VS controls for VS inside and outside the manipulator's workspace are achieved by selecting the corresponding weight matrices in Eqs. (23) and (21), respectively.

5 Experimental result

Several experiments are performed in this section to verify the effectiveness of the proposed system. The proposed UAM platform includes a coaxial eight-propeller structure (Fig. 1). The selected manipulator motors are Dynamixel servo motors and the manipulator links are custom-built using 3D printing for lightweight consideration. The total length of the manipulator is 0.5 m with a total weight of 0.9 kg. An Intel NUC computer is equipped for onboard image processing and control implementation. Our motion capture system, which consists of 10 OptiTrack cameras at 120 Hz, obtains the UAV's pose information. A camera is attached to the end-effector of the manipulator for VS control. Note that the entire control process is automatic except for the start command triggered by a human operator. Fig. 7 shows the signal flow of the developed AM system.

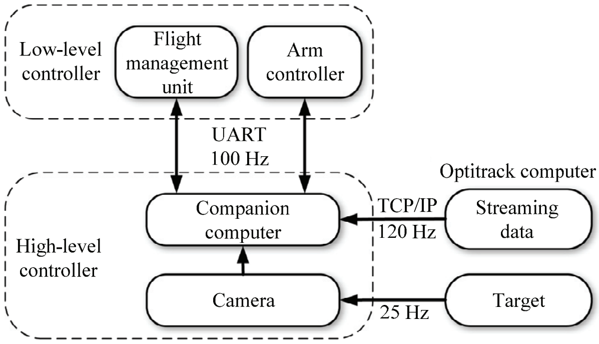


Fig. 7 Signal flow of the aerial manipulator.

5.1 Evaluation of the ORB-based servoing point detection

An experiment was first designed to evaluate the detection of servoing points, which were detected from the target object without attaching artificial markers. A target object was selected for demonstration. The reference image and the detected servoing points are presented in Fig. 8a. The small circles represent the detected ORB features. In addition, the camera was moved 1.5 m away and rotating at different angles (Figs. 8b–8d). The results indicate that the target was successfully matched and detected in different views. Moreover, the servoing points were provided in a stable manner even with complex background and large movements of the camera.

5.2 Aerial grasping experiment

A validation experiment for aerial grasping was further performed on the basis of the robust detection of the servoing points. The grasping experiment was divided into three steps: VS outside the manipulator's workspace, VS inside the manipulator's workspace, and grasping.

Fig. 9 shows the images of the grasping experiment. The AM automatically took off and flew toward the target under the proposed VS controller (Eq. (24)) with the weight matrix (Eq. (21)), where the target is 3 m away from the AM and outside the manipulator's workspace. When the camera was 0.4 m away from the target, the UAV hovered to perform VS and drove the end-effector to approach the target. When the camera moved to a distance of 20 cm away from the target, an open-loop grasping control based on the AM's dynamics was triggered to grasp the target object because the camera was unable to see the target. Lastly, the object was grasped successfully. Refer to the supplementary video for the grasping procedure.

To verify the performance, the VS using artificial markers[17,26] for servoing features was also performed for comparison. Without loss of generality, AprilTag was used as the marker attached on the target. Figs. 10 and 11 illustrate the convergence performance about the camera's translational and rotational velocities, joints' rotation speed, and tracking errors of the AprilTag- and natural feature-based visual servoing approaches, respectively,

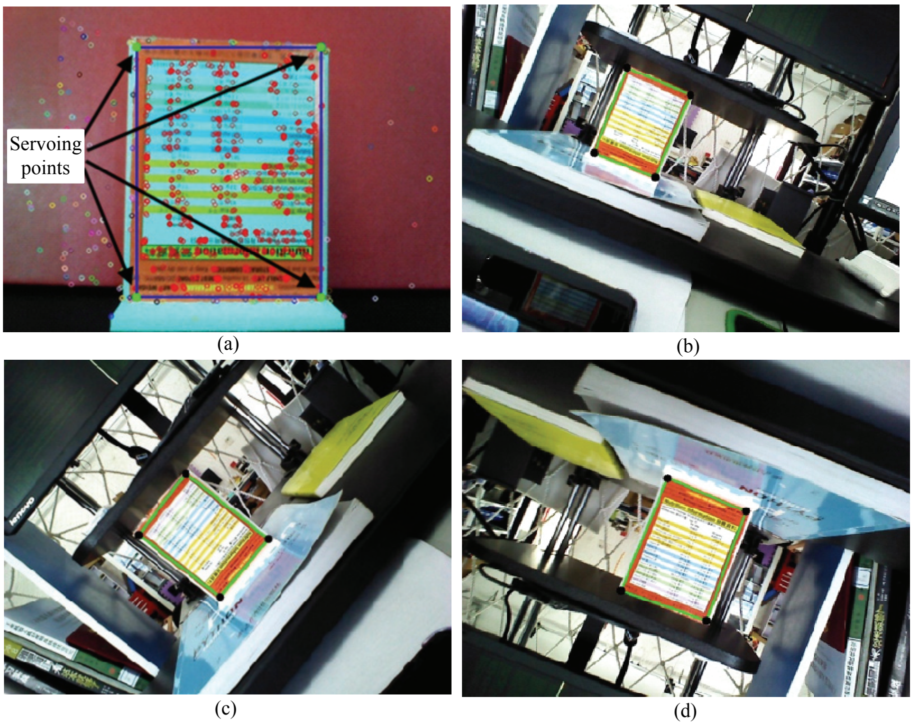


Fig. 8 Experimental results of detecting the servoing points. (a) Reference image of the target object with ORB features and servoing points detected; (b–d) detection results of servoing points with different movements of the servoing camera.

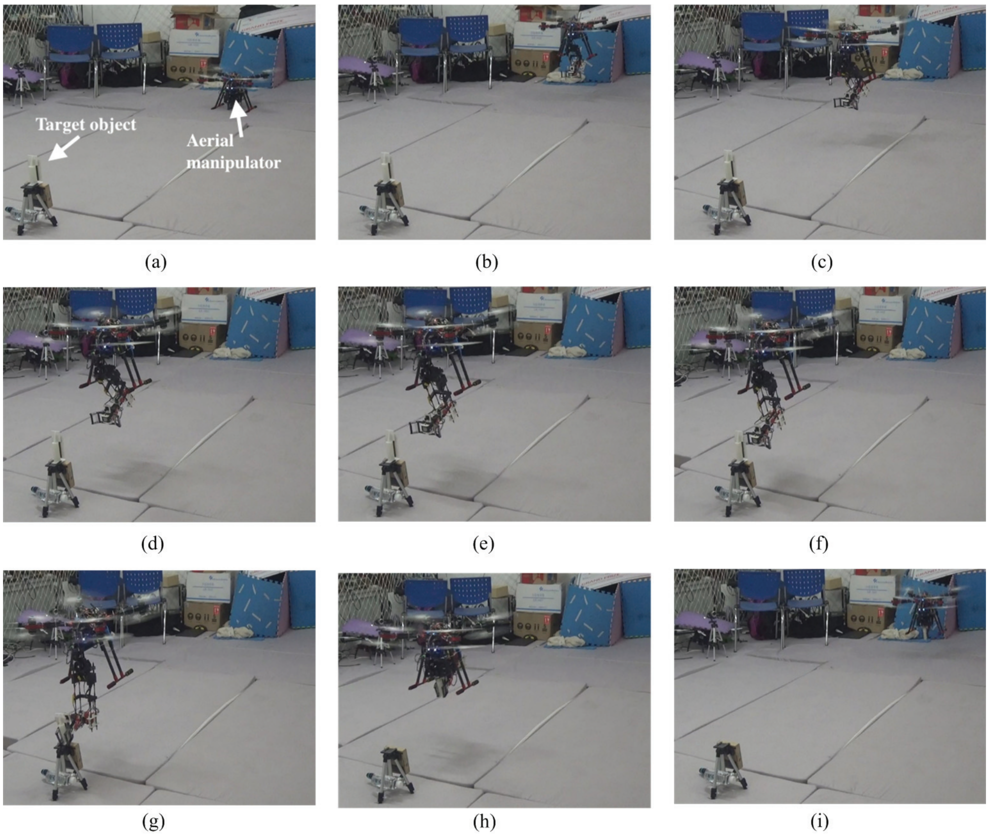


Fig. 9 Snapshots of the aerial manipulator grasping experiment from take-off to landing. (a) Take-off; (b–c) long-distance VS; (d–e) VS inside the manipulator's workspace; (f) grasping; (g) object grasped; (h) object transportation; (i) returning and landing.

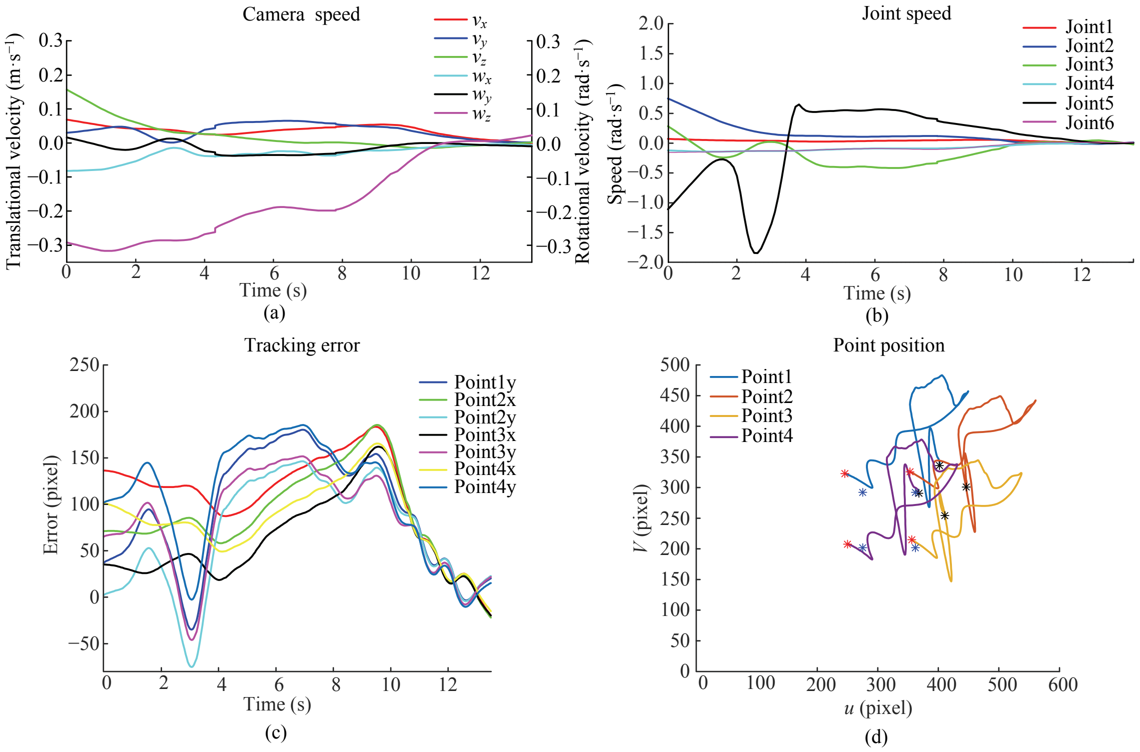


Fig. 10 Experimental results under the AprilTag-based VS for aerial grasping. (a) Camera velocity *vs* time; (b) joint angular velocity *vs* time; (c) tracking error in image plane for each servoing point *vs* time; (d) servoing points' trajectories in the image plane, where the blue, black, and red points indicate the desired, initial, and converged positions, respectively.

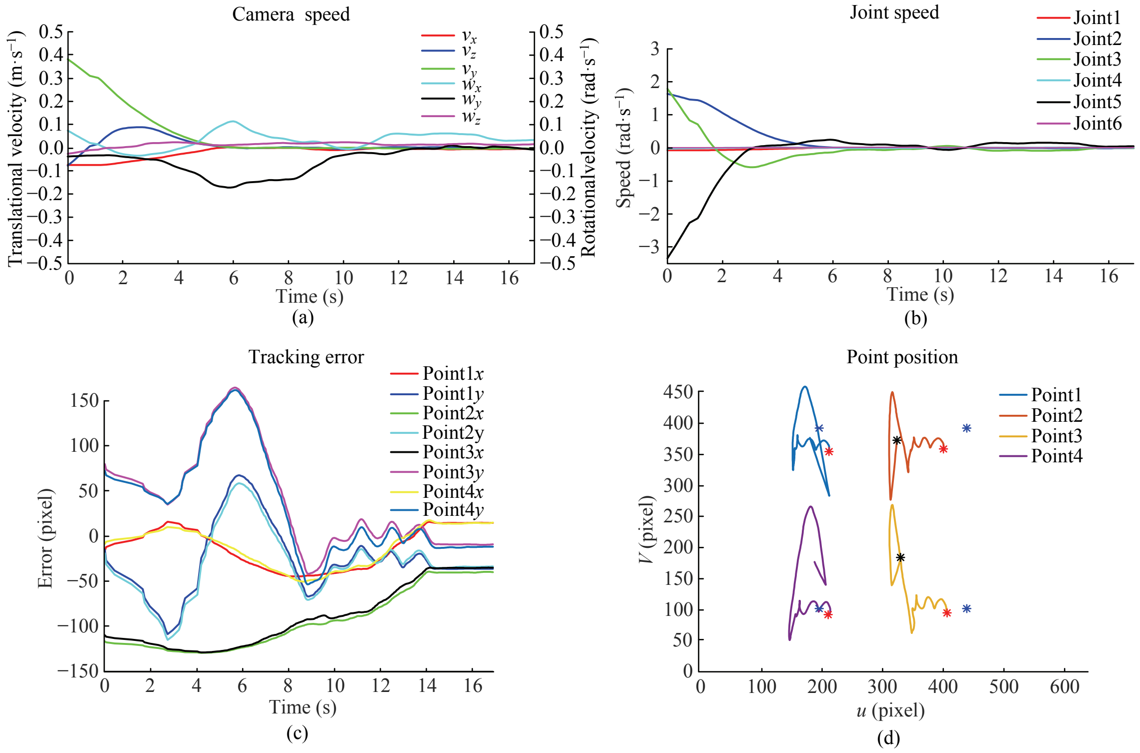


Fig. 11 Experimental results under the proposed approach for aerial grasping. (a) Camera velocity *vs* time; (b) joint angular velocity *vs* time; (c) tracking error in image plane for each servoing point *vs* time; (d) servoing points' trajectories in the image plane, where the blue, black, and red points indicate the desired, initial, and converged positions, respectively.

when the UAV was hovering within the AM's workspace. The servoing points' trajectories in the image plane were also illustrated. From the figures, the convergence results of the servoing points in the image plane and joint angles in the joint space are similar. Natural feature-based VS can achieve the same control effect with the Apriltag-based approach, but without attaching a pattern on the target. Based on Fig. 11, we find that the camera and joint velocities converged to zero. Figs. 11c and 11d show that the position of the point features eventually converged to the desired position. Some errors, which may be caused by such as hand–eye calibration, camera calibration, and static error, were observed. The errors in the image frame denote small position errors of the end-effector in the world frame. Therefore, the end-effector was able to grasp the target object successfully.

6 Conclusion

This study develops an AM system that achieves object grasping without artificial landmarks on the target. The kinematic model of the proposed system is first deduced as the basis of the design of the VS controller. Thereafter, a novel VS controller is designed by utilizing ORB features detected from the captured images. This natural feature-based method does not need to attach artificial markers on targets. In addition, a VS controller when the AM is outside the manipulator's workspace is developed by utilizing the DoFs of the UAV and the manipulator joints. By involving a weight matrix, the two VS controllers inside and outside the manipulator's workspace is further designed into a common framework. Lastly, experiments are carried out to verify the effectiveness of the proposed approach.

The paper considers only target objects with rich texture; however, the targets may be lack of texture on the surface in practical applications. In addition, the motion capture system is used for the UAV stability control; however, it is generally unavailable. To realize the fully autonomous aerial grasping, there still needs deep research on the UAV localization and environmental 3D perception, robust object detection, grasping force control, and safety mechanism, etc. Therefore, our future work will be the development of algorithms to promote the autonomous ability of the aerial manipulator system.

7 Acknowledgment

This study was partially supported by grants from the National Natural Science Foundation of China (Nos. U1713206, 61673131 and 51975550), the Bureau of Industry and Information Technology of Shenzhen (No. 20170505160946600), and Hong Kong Research Grant Council (No. 14204814).

* All supplementary materials are available at https://doi.org/10.1007/s42235-020-0017-4.

References

[1] Lee H, Kim H J. Estimation, control, and planning for autonomous aerial transportation. *IEEE Transactions on Industrial Electronics*, 2017, **64**, 3369–3379.

[2] Lee H, Kim S, Kim H J. Control of an aerial manipulator using on-line parameter estimator for an unknown payload. *Proceedings of the* 2015 *IEEE International Conference on Automation Science and Engine*ering, Gothenburg, Sweden, 2015, 316–321.

[3] Kondak K, Huber F, Schwarzbach M, Laiacker M, Sommer D, Bejar M, Ollero A. Aerial manipulation robot composed of an autonomous helicopter and a 7 degrees of freedom industrial manipulator. *Proceedings of the* 2014 *IEEE International Conference on Robotics and Automation*, Hong Kong, China, 2014, 2107–2112.

[4] Laiacker M, Huber F, Kondak K. High accuracy visual servoing for aerial manipulation using a 7 degrees of free-

dom industrial manipulator. *Proceedings of the* 2016 *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Daejeon, Korea, 2016, 1631–1636.

[5] Korpela C, Orsag M, Oh P. Towards valve turning using a dual-arm aerial manipulator. *Proceedings of the* 2014 *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Illinois, USA, 2014, 3411–3416.

[6] Ruggiero F, Lippiello V, Ollero A. Aerial manipulation: A literature review. *IEEE Robotics and Automation Letters*, 2018, **3**, 1957–1964.

[7] Munoz-Morera J, Maza I, Fernandez-Aguera C J, Caballero F, Ollero A. Assembly planning for the construction of structures with multiple UAS equipped with robotic arms. *Proceedings of the* 2015 *International Conference on Unmanned Aircraft Systems*, Colorado, USA, 2015, 1049–1058.

[8] Kim S, Choi S, Kim H J. Aerial manipulation using a quadrotor with a two dof robotic arm. *Proceedings of the* 2013 *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Tokyo, Japan, 2013, 4990–4995.

[9] Sanchez M I, Acosta J Á, Ollero A. Integral action in first-order closed-loop inverse kinematics. Application to aerial manipulators. *Proceedings of the* 2015 *IEEE International Conference on Robotics and Automation*, Washington, USA, 2015, 5297–5302.

[10] Kim S, Seo H, Shin J, Kim H J. Cooperative aerial manipulation using multirotors with multi-DOF robotic arms. *IEEE/ASME Transactions on Mechatronics*, 2018, **23**, 702–713.

[11] Kim H, Lee H, Choi S, Noh Y K, Kim H J. Motion planning with movement primitives for cooperative aerial transportation in obstacle environment. *Proceedings of the* 2017 *IEEE International Conference on Robotics and Automation*, Singapore, Singapore, 2017, 2328–2334.

[12] Lee H, Kim H, Kim W, Kim H J. An integrated framework for cooperative aerial manipulators in unknown environments. *IEEE Robotics and Automation Letters*, 2018, **3**, 2307–2314.

[13] Garimella G, Kobilarov M. Towards model-predictive control for aerial pick-and-place. *Proceedings of the* 2015 *IEEE International Conference on Robotics and Automation*, Washington, USA, 2015, 4692–4697.

[14] Cataldi E, Muscio G, Trujillo M A, Rodríguez Y, Pierri F, Antonelli G, Caccavale F, Viguria A, Chiaverini S, Ollero A. Impedance control of an aerial-manipulator: Preliminary results. *Proceedings of the* 2016 *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Daejeon, South Korea, 2016, 3848–3853.

[15] Seo H, Kim S, Kim H J. Locally optimal trajectory planning for aerial manipulation in constrained environments. *Proceedings of the* 2017 *IEEE/RSJ International Conference on Intelligent Robots and Systems*, British Columbia, Canada, 2017, 1719–1724.

[16] Rossi R, Santamaria-Navarro A, Andrade-Cetto J, Rocco P. Trajectory generation for unmanned aerial manipulators through quadratic programming. *IEEE Robotics and Automation Letters*, 2016, **2**, 389–396.

[17] Fang L X, Chen H Y, Lou Y J, Li Y J, Liu Y H. Visual grasping for a lightweight aerial manipulator based on NSGA-II and kinematic compensation. *Proceedings of the* 2018 *IEEE International Conference on Robotics and Automation*, Queensland, Australia, 2018, 3488–3493.

[18] Thomas J, Loianno G, Sreenath K, Kumar V. Toward image based visual servoing for aerial grasping and perching. *Proceedings of the* 2014 *IEEE International Conference on Robotics and Automation*, Hong Kong, China, 2014, 2113–2118.

[19] Thomas J, Loianno G, Polin J, Sreenath K, Kumar V. Toward autonomous avian-inspired grasping for micro aerial vehicles. *Bioinspiration* & *Biomimetics*, 2014, **9**, 025010.

[20] Kim S, Seo H, Choi S, Kim H J. Vision-guided aerial manipulation using a multirotor with a robotic arm. *IEEE/ASME Transactions on Mechatronics*, 2016, **21**, 1912–1923.

[21] Seo H, Kim S, Kim H J. Aerial grasping of cylindrical object using visual servoing based on stochastic model predictive control. *Proceedings of the* 2017 *IEEE International Conference on Robotics and Automation*, Singapore, Singapore, 2017, 6362–6368.

[22] Lippiello V, Cacace J, Santamaria-Navarro A, Andrade-Cetto J, Trujillo M A, Esteves Y R, Viguria A. Hybrid visual servoing with hierarchical task composition for aerial manipulation. *IEEE Robotics and Automation Letters*, 2015, **1**, 259–266.

[23] Santamaria-Navarro A, Lippiello V, Andrade-Cetto J. Task priority control for aerial manipulation. *Proceedings of the* 2014 *IEEE International Symposium on Safety*, *Security*, *and Rescue Robotics*, Hokkaido, Japan, 2014, 1–6.

[24] Muscio G, Pierri F, Trujillo M A, Cataldi E, Antonelli G, Caccavale F, Viguria A, Chiaverini S, Ollero A. Coordinated control of aerial robotic manipulators: Theory and experiments. *IEEE Transactions on Control Systems Technology*, 2017, **26**, 1406–1413.

[25] Santamaria-Navarro A, Grosch P, Lippiello V, Solà J, An-

drade-Cetto J. Uncalibrated visual servo for unmanned aerial manipulation. *IEEE/ASME Transactions on Mechatronics*, 2017, **22**, 1610–1621.

[26] Lippiello V, Fontanelli G A, Ruggiero F. Image-based visual-impedance control of a dual-arm aerial manipulator. *IEEE Robotics and Automation Letters*, 2018, **3**, 1856–1863.

[27] Collewet C, Marchand E. Photometric visual servoing. *IEEE Transactions on Robotics*, 2011, **27**, 828–834.

[28] Bateux Q, Marchand E. Histograms-based visual servoing. *IEEE Robotics and Automation Letters*, 2016, **2**, 80–87.

[29] Bateux Q, Marchand E, Leitner J, Chaumette F, Corke P. Training deep neural networks for visual servoing. *Proceedings of the* 2018 *IEEE International Conference on Robotics and Automation*, Queensland, Australia, 2018, 1–8.

[30] Bruno S, Lorenzo S, Luigi V, Giuseppe O. Robotics: Modelling, planning and control. *Advanced Textbooks in Control and Signal Processing Series*, Springer, London, UK, 2009.

[31] Chaumette F, Hutchinson S. Visual servo control. II. Advanced approaches [Tutorial]. *IEEE Robotics & Automation Magazine*, 2007, **14**, 109–118.

[32] Li Y L, Wang S J, Tian Q, Ding X Q. A survey of recent advances in visual feature detection. *Neurocomputing*, 2015, **149**, 736–751.

[33] Lowe D G. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2004, **60**, 91–110.

[34] Bay H, Tuytelaars T, Van Gool L. Surf: Speeded up robust features. *Proceedings of the European Conference on Computer Vision*, Graz, Austria, 2006, 404–417.

[35] Rublee E, Rabaud V, Konolige K, Bradski G. ORB: An efficient alternative to SIFT or SURF. *Proceedings of* 2011 *International Conference on Computer Vision*, Barcelona, Spain, 2011, 2564–2571.

[36] Meagher D. Geometric modeling using octree encoding. *Computer Graphics and Image Processing*, 1982, **19**, 129–147.

[37] Hartley R, Zisserman A. *Multiple View Geometry in Computer Vision*, Cambridge University Press, UK, 2003.