Toward Visibility Guaranteed Visual Servoing Control of Quadrotor UAVs

Dongliang Zheng, Hesheng Wang, *Senior Member, IEEE*, Jingchuan Wang, Xiufeng Zhang, and Weidong Chen, *Member, IEEE*

***Abstract*—This paper studies the visibility problem in the visual servoing control of quadrotor unmanned aerial vehicles (UAVs). Visual servoing is a useful technique for the inspection, positioning, landing, and tracking of quadrotor UAVs. However, one prerequisite in previous research works is that the visual target stays inside the field of view of the camera for all time. While lose sight of the visual target will result in fatal failure of the task, how to guarantee this prerequisite is often overlooked. In this paper, we aim to achieve reliable visual servoing by deriving a provable solution to guarantee visibility in quadrotor visual serving control. The main idea is to ensure the forward invariance of a set using control barrier functions. The set is constructed as the visible set, which specifies the upper bound of the distance between the center of the image plane and the coordinates of visual features in the image plane. Then, the visibility constraint is developed based on control barrier function. The original control inputs are minimally modified to satisfy the visibility constraint, thus preserving visibility. Finally, some simulations and experimental results are presented to validate this method.**

***Index Terms*—Control barrier function, quadrotor, unmanned aerial vehicle, visibility, visual servoing.**

I. Introduction

VISUAL SERVOING methods are concerned with using visual feedback to control the pose of a robot with respect to the specific visual target [1]. It has been studied extensively over the past two decades. Some examples are robot manipulators object grasping [2], [3], formation tracking of mobile robots [4], and position control, landing, and object tracking for quadrotor unmanned aerial vehicles (UAVs) [5]–[7].

One prerequisite in visual servoing is that the visual target should stay inside the field of view (FoV) of the camera during the servoing process. The guarantee of the visibility of the visual target is vital, as visual servoing needs the real-time visual feedback of the target of interest to generate control inputs. Lose sight of the servoing target will result in failure of the task and even damage the robot. This is especially true for agile quadrotors operating in the air. Therefore, one way to increase the robustness of visual servoing is to explicitly take into account the visibility problem in the controller design.

Due to its three-dimensional movability and agility, the quadrotor is an ideal platform for inspection [8], [9], delivery [10], and moving object tracking [11] tasks and many of these applications can benefit from visual servoing techniques. For example, it is very common to control the position of the quadrotor with respect to some visual targets of interest in autonomous filming, inspection, and surveillance tasks. In addition, visual servoing can exploit real-time visual feedback to facilitate the autonomous landing of quadrotors and moving targets tracking. Visual servoing of quadrotors aim to use onboard vision sensor and onboard computation to achieve autonomous vision-based tasks. A considerable number of image-based visual servoing (IBVS) methods have been proposed considering the underactuation property and the nonlinear dynamics of the quadrotor. However, the visibility problem of ensuring the visual target stay inside the FoV of the camera/quadrotor is often overlooked.

The visibility constraint in the visual servoing of quadrotors can be violated for many reasons. First, even with the right controller, the quadrotor can lose sight of the visual target because of its underactuation. Consider an IBVS task which requires the quadrotor hovering right above the visual feature, see Fig. 1. The red arrows in Fig. 1 represents the image errors. In Fig. 1(a), to reduce the image error in the image plane, the quadrotor needs to move to the left. Because of its underactuation, the quadrotor has to tilt to generate the desired motion [see Fig. 1(b)], which increases the image error instead. If the visual feature is already at the edge of the image plane as in the case of Fig. 1(a), the visibility will be violated with the tilt motion. Second, as the tasks for quadrotors becoming increasingly demanding, there will be multiple objectives in the vision-based control of quadrotors, and they may be conflicted with each other. If one of the objectives is to observing a visual target, following other objectives

Manuscript received March 26, 2018; revised January 18, 2019; accepted March 16, 2019. Date of publication March 20, 2019; date of current version June 14, 2019. Recommended by Technical Editor G. Hu. This work was supported in part by the Natural Science Foundation of China under Grant 61722309 and Grant U1613218, and in part by State Key Laboratory of Robotics and System (HIT). *(Corresponding author: Hesheng Wang.)*

D. Zheng, J. Wang, and W. Chen are with the Key Laboratory of System Control and Information Processing, Ministry of Education of China, and the Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: dzheng@gatech.edu; jchwang@sjtu.edu.cn; wdchen@sjtu.edu.cn).

H. Wang is with the Key Laboratory of System Control and Information Processing, Ministry of Education of China, and the Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China, and also with the State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin 150006, China (e-mail: wanghesheng@sjtu.edu.cn).

X. Zhang is with the National Research Center for Rehabilitation Technical Aids, Beijing 100176, China (e-mail: zhangxiufeng@nrcrta.cn).

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

Digital Object Identifier 10.1109/TMECH.2019.2906430

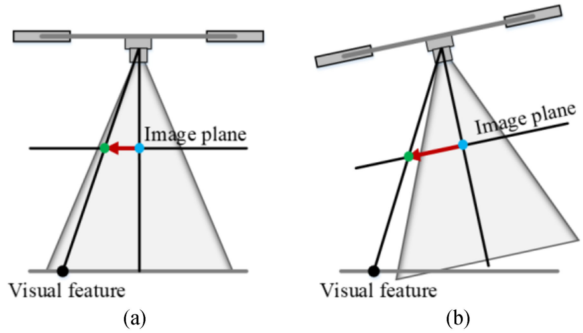


Fig. 1. Visibility violation due to quadrotor underactuation in image-based control. The length of the red arrow represents the magnitude of the image error. The blue dot represents the desired feature position in the image plane. The green dot represents the current image feature position. In (a), the quadrotor needs to move to the left to reduce the image error in the image plane. Because of its underactuation, the quadrotor has to tilt to generate the desired motion, which increases the image error instead as can be seen in (b).

may break the visibility constraint. For example, tracking a desired trajectory while keep observing a target of interest. If the desired trajectory is generated without taking into account the visibility of the target, strictly trajectory tracking may lead to visibility violation. Third, sensor measurement noises and disturbances can also cause visibility violation. For example, camera measure noise can result in false information about the position of the visual features. If the visual measurement is accurate, the properly designed controller will not cause visibility violation. However, since there are errors in the measurements, the visibility constraint may be violated using the controller. In addition, wind gust is a source of disturbance that can cause visibility violation during the visual servoing of quadrotors. The method proposed here mainly deal with the first two cases.

In [12] and [13], input saturated and bounded-input controllers are developed for the IBVS of quadrotors, respectively. Because these methods set upper bounds on the control inputs, the quadrotor is controlled in a less aggressive manner, which will improve visibility for the first case discussed in Fig. 1. However, guarantee visibility in IBVS of quadrotors is still an open problem [13]. In addition, their methods cannot deal with multiobjective cases. Because the visibility violation can happen and will result in fatal failure of the task once it happens, a provable solution to preserve visibility and achieve reliable visual servoing is needed.

In this paper, an IBVS method which takes into account the visibility problem is proposed. The basic idea is to use control barrier functions (CBFs) to derive the visibility constraint. CBFs are widely used to guarantee the states of a system stay inside a set. They have been used for the adaptive cruise control of automated vehicles [14], [15], to impose dynamically-feasible constraints [16], [17], obstacle avoidance for quadrotors [18], and inter collision avoidance for robot team [19], [20]. The underlying idea is to enforce the forward invariance of a set in the state space by imposing a linear constraint on the control inputs [14]. If the state of the system starts inside a set, the states will remain within the set given that the barrier certificates are satisfied.

In order to guarantee that the visual target stays within the FOV of the camera, the states, which are the coordinates of the visual target on the image plane, must stay inside of a set. The set is called the visible set and is corresponding to the image plane of the camera. Visibility constraint is derived using the visible set and CBFs. The Image-based visual servoing method without considering visibility is used to generate control inputs. Then the control inputs are minimally adjusted using quadratic programming to satisfy the visibility constraint. Finally, various simulations and experiments are conducted to prove the validity of the proposed method. By enforcing the visibility constraint, the visibility during the visual servoing process is greatly improved.

The rest of the paper is organized as follows. In Section II, the Image-based visual servoing of a quadrotor with visibility constraint is proposed. Then, simulations and experimental results are detailed in Section III. Finally, some conclusions are drawn in Section IV.

II. IBVS Controller With Visibility Constraint

In this section, we first introduce the IBVS controller which does not consider visibility constraint. Then the visible set is defined, and the visibility constraint is derived using the CBF. Recently, a class of non-conservative control barrier functions is introduced to enforce forward invariance of a set through a linear constraint on the control inputs [14]–[18], [21]. The set represents the allowable region of the states. If the control inputs satisfy the linear constraint, they are comprehended as valid. If the initial states satisfy the constraint and the control inputs are valid, the states will stay within the set for all time. Therefore, by defining the visible set based on CBFs and enforcing the visibility constraint, visibility during IBVS will be greatly improved. After deriving the visibility constraint, we want to modify the original control inputs computed by the IBVS controller in a minimal way, such that the control inputs satisfy the visibility constraint. In this paper, the original control inputs are computed based on an IBVS controller as an example. The visibility constraint can be combined with other controllers to preserve visibility in multiobjective tasks scenarios mentioned in the introduction.

A. IBVS Controller

Before introducing the control barrier certificates, we first present the IBVS controller without visibility constraint. The detailed controller is presented in our previous work [22] and [23].

Inertial frame I and quadrotor body frame B are shown in Fig. 2. They are defined to describe the equation of motion of the quadrotor. The origin of body frame B is located at the center of the quadrotor. Euler angles, roll (φ), pitch (θ), and yaw (ψ) angles are used to describe the attitude of the quadrotor in the inertial frame. The rotation matrix corresponds to the Euler angles is denoted by $\mathbf{R} : B \rightarrow I$.

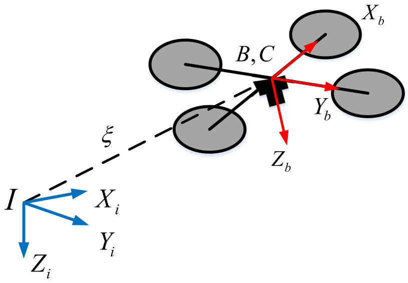
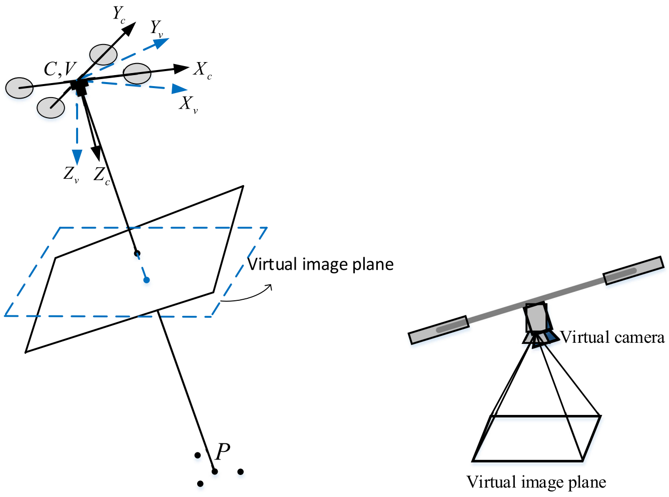


Fig. 2. Quadrotor coordinate frames. Inertial frame I oriented north, east, and down. Quadrotor body frame B and camera frame C are assumed coincide with each other.



(a) (b)

Fig. 3. Virtual camera and its virtual image plane. The defined virtual camera is always parallel to the level ground regardless of the roll and pitch angles of the quadrotor.

The dynamics of the quadrotor is governed by

$$m\dot{\mathbf{V}} = -m\mathbf{\Omega} \times \mathbf{V} + \mathbf{F} \tag{1}$$

$$\mathbf{J}\dot{\mathbf{\Omega}} = -\mathbf{\Omega} \times \mathbf{J}\mathbf{\Omega} + \tau \tag{2}$$

where $\mathbf{F} = -U_1\mathbf{E}_3 + mg\mathbf{R}^T\mathbf{e}_3$, $\mathbf{\Omega} \in R^3$, and $\mathbf{V} \in R^3$ are the linear and angular velocity of the quadrotor with respect to the body frame B, respectively, $\times$ denotes the cross product, m is the mass of the quadrotor, $\mathbf{J}$ is the body frame inertia, $\mathbf{E}_3 = \mathbf{e}_3 = [0\ 0\ 1]^T$ are unit vectors in the body frame and inertial frame, respectively. $\mathbf{F} \in R^3$ and $\tau \in R^3$ are the force and torque vectors act on the quadrotor and expressed in the body frame, respectively, U_1 is the total thrust force generated by the four propellers, g is the gravity acceleration.

In the visual servoing task of quadrotors, the camera is rigidly attached under the quadrotor, facing down (see Fig. 2). We assume that the camera frame C and quadrotor body frame B are aligned for the convenience of presentation. The IBVS controller based on the virtual camera approach is adopted. Following the virtual camera approach [12], [13], [22], [23], the virtual camera is defined as shown in Fig. 3. Virtual camera frame V has the same origin and yaw angles as C and its roll and pitch angles are always zero. The definition of the virtual camera indicates that the virtual camera is always parallel to the level ground, regardless of the roll and pitch angles of the quadrotor. There is also a virtual image plane whose configuration relative to V is the same as actual image plane relative to B. When the camera is rigidly attached under the quadrotor, we can use the roll and pitch angles of the quadrotor to project the visual features in the actual camera plane to the virtual camera plane. If a pan-and-tilt camera is used, the projection is not needed because the pan-and-tilt camera is a physical implementation of the defined virtual camera. The measurements in the pan-and-tilt camera frame are the same as those obtained in the virtual image plane in the fixed camera case. The virtual camera approach is adopted to achieve a simplified image feature dynamics and facilitate the controller design.

Using the perspective projection camera model, the coordinates of the kth point in the virtual image plane is denoted by $({}^v u_k\ {}^v n_k)$. Image moments [24], [25] are chose as image features, the features used in the IBVS are defined as follows:

$$q_x = q_z \frac{{}^v u_g}{\lambda}, q_y = q_z \frac{{}^v n_g}{\lambda}, q_z = \sqrt{\frac{a^*}{a}} \tag{3a}$$

$$q_4 = \frac{1}{2}\arctan\left(\frac{2\,{}^v\mu_{11}}{{}^v\mu_{20} - {}^v\mu_{02}}\right) \tag{3b}$$

where λ is the focal length of the camera. ${}^v u_g = \frac{1}{N}\sum_{k=1}^{N} {}^v u_k$ and ${}^v n_g = \frac{1}{N}\sum_{k=1}^{N} {}^v n_k$ are the barycenter of the visual target, $a = {}^v\mu_{20} + {}^v\mu_{02}$, a^* is a constant and it is the value of a at an time instant. At that time instant, the depth of the visual feature with respect to the camera is z^*. In addition, ${}^v\mu_{ij} = \sum_{k=1}^{N} ({}^v u_k - {}^v u_g)^i ({}^v n_k - {}^v n_g)^j$. The image features defined in (3) are functions of the point coordinates in the virtual image plane.

Define $\mathbf{q} = [q_x\ q_y\ q_z]^T$ and $\mathbf{q}_1 = [q_x\ q_y\ q_z\ q_4]^T$. The dynamics of the image features in the virtual image plane are

$$\dot{\mathbf{q}} = -sk\left(\dot{\psi}\mathbf{e}_3\right)\mathbf{q} - \frac{1}{z^*}\mathbf{v} \tag{4a}$$

$$\dot{q}_4 = -\dot{\psi} \tag{4b}$$

where $\mathbf{v}$ is the linear velocity of the virtual camera frame with respect to the virtual camera frame, $\mathbf{v} = \mathbf{R}_{\phi\theta}\mathbf{V}$ and $\mathbf{R}_{\phi\theta} = \mathbf{R}_\theta\mathbf{R}_\phi$. According to (1), we can get translational dynamics of the virtual camera frame

$$\dot{\mathbf{v}} = -sk\left(\dot{\psi}\mathbf{e}_3\right)\mathbf{v} + \mathbf{f} \tag{5}$$

$$\mathbf{f} = \frac{-\mathbf{R}_{\phi\theta}U_1\mathbf{E}_3}{m} + g\mathbf{e}_3. \tag{6}$$

Fig. 4 shows the structure of the proposed visual servoing control method. In the visual servoing control, first, trajectories in terms of the image features in the image space are generated. Next, the image-based visual servoing controller aims to track

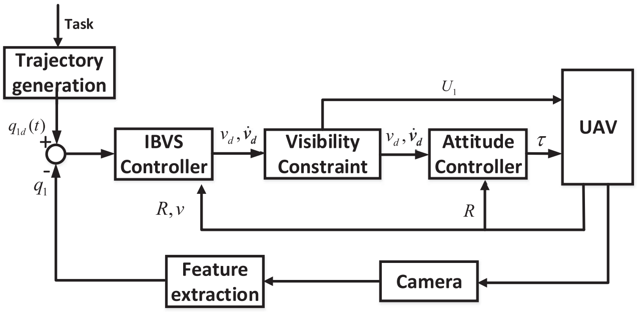


Fig. 4. Structure of the proposed visual servoing control method. The image-based visual servoing controller aims to track the desired trajectories of the image features which is generated according to the specific tasks. Then, in order to ensure the visual target stay within the FOV of the camera, the control inputs generated by the IBVS controller are adjusted in a minimal way based on the visibility constraint and quadratic programming. Finally, the adjusted control inputs are sent to the low-level attitude controller to control the quadrotor.

the desired trajectories. At the same time, the visual target has to stay within the FOV of the camera. The control inputs generated by the IBVS controller are adjusted in a minimal way based on the visibility constraint and quadratic programming. Then, the adjusted control inputs are sent to the low-level attitude controller to control the quadrotor. The desired trajectories in terms of the image features are as follows:

$$\mathbf{q}_{1d} = \begin{bmatrix} q_{xd} & q_{yd} & q_{zd} & q_{4d} \end{bmatrix}^T \tag{7}$$

where $\mathbf{q}_{1d}$ is the desired trajectories of $\mathbf{q}_1$. Time dependency is left out in (7) for simplicity.

The feature error is defined as follows:

$$\mathbf{e} = \mathbf{q}_1 - \mathbf{q}_{1d}.$$

Using (4), the first and second derivative of the feature error $\mathbf{e}$ are

$$\dot{\mathbf{e}} = \mathbf{H}\mathbf{u}_f - \dot{\mathbf{q}}_{1d} \tag{8}$$

$$\ddot{\mathbf{e}} = \dot{\mathbf{H}}\mathbf{u}_f + \mathbf{H}\dot{\mathbf{u}}_f - \ddot{\mathbf{q}}_{1d} \tag{9}$$

where

$$\mathbf{H} = \begin{bmatrix} -1/z^* & 0 & 0 & q_y \\ 0 & -1/z^* & 0 & -q_x \\ 0 & 0 & -1/z^* & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix},$$

$$\mathbf{u}_f = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ \dot{\psi} \end{bmatrix}.$$

Define a new feature tracking error

$$\delta = c_1\mathbf{e} + c_2\dot{\mathbf{e}} \tag{10}$$

with c_1 and c_2 positively defined parameters. Then, the trajectory tracking task is defined to drive the new feature error δ to zero.

The image-based visual tracking controller is designed as follows [23]:

$$\dot{\mathbf{u}}_f = (c_2\mathbf{H})^{-1}(-(c_1\mathbf{H} + c_2\dot{\mathbf{H}})\mathbf{u}_f - c_3\delta + c_1\dot{\mathbf{q}}_{1d} + c_2\ddot{\mathbf{q}}_{1d}) \tag{11a}$$

$$\mathbf{f} = \dot{\mathbf{v}}_d + sk(\dot{\psi}e_3)\mathbf{v} - k_1(\mathbf{v} - \mathbf{v}_d) \tag{11b}$$

where $c_3 > 0$, $k_1 > 0$. $\dot{\mathbf{v}}_d$ and $\mathbf{v}_d$ in (11b) are the first three elements in $\dot{\mathbf{u}}_f$ and $\mathbf{u}_f$, respectively. $\mathbf{u}_f$ can be obtained by solving the differential equation (11a) or by integrating $\dot{\mathbf{u}}_f$. In the controller (11), (11a) designs the desired linear velocity $\mathbf{v}_d$, linear acceleration $\dot{\mathbf{v}}_d$, and angular velocity of the yaw angle. The desired yaw angular velocity is defined by the forth element of $\mathbf{u}_f$. Equation (11b) is a velocity tracking controller.

Based on (11b) and (6), we can get the desired thrust and the desired Euler angles. The desired thrust and the desired roll and pitch angles are computed as follows:

$$U_1 = \mathbf{e}_3^T \mathbf{R}_{\phi\theta}^T (mg\mathbf{e}_3 - \mathbf{f}) \tag{12}$$

$$\phi = \arcsin(f_2/U_1), \theta = \arcsin(-f_1/U_1 c_\phi) \tag{13}$$

where $\mathbf{f} = [f_1\ f_2\ f_3]^T$, the notation c is the shorthand form of cosine. After obtaining the desired Euler angles, an attitude controller [26]–[28] can be used for quadrotor attitude control.

B. CBFs and Visibility Constraint

CBFs are widely used to impose the forward invariance of a set. The visibility problem in visual servoing is to ensure that the coordinates of the visual target are inside the border of the image plane of the camera, which can be treated as a set invariant problem. First, the visible set is defined. Then, the visibility constraint based on CBF is obtained to ensure the forward invariant of the visible set. When the states of the system are bounded by a visible set using the CBFs, it is ensured that the visual target stays inside the FOV of the quadrotor.

Consider a point P, its coordinates with respect to the camera frame and virtual camera frame are ${}^c\mathbf{p}(t) = [{}^cx\ {}^cy\ {}^cz]^T$ and ${}^v\mathbf{p}(t) = [{}^vx\ {}^vy\ {}^vz]^T$, respectively. Its projection in the virtual image plane is

$$\begin{cases} {}^vu = \lambda {}^vx/{}^vz \\ {}^vn = \lambda {}^vy/{}^vz \end{cases}.$$

These coordinates can be obtained using the roll and pitch angles of the quadrotor and the measurements from the actual camera if the actual camera is rigidly fixed below the quadrotor, or can be obtained directly from the measurements of a pan-and-tilt camera onboard the quadrotor. The time derivative of the image coordinates is

$$\begin{bmatrix} {}^v\dot{u} \\ {}^v\dot{n} \end{bmatrix} = \begin{bmatrix} -\frac{\lambda}{{}^vz} & 0 & \frac{{}^vu}{{}^vz} \\ 0 & -\frac{\lambda}{{}^vz} & \frac{{}^vn}{{}^vz} \end{bmatrix} \mathbf{v} + \begin{bmatrix} {}^vn \\ -{}^vu \end{bmatrix} \dot{\psi}. \tag{14}$$

To keep the visual point P inside the FOV of the camera, the visible set is defined as

$$C = \{(u,n) \in R^2 | B(u,n) \geq 0\} \tag{15}$$

where

$$B = \rho^2 - u^2 - n^2. \tag{16}$$

From now on, the superscript v in ${}^v u$ and ${}^v n$ is omitted. ρ is the radius of the virtual image plane. Note that (15) is defined using the point coordinate in the virtual image plane, this is the case if a pan-and-tilt camera is used, where the actual image plane of the pan-and-tilt camera is the same as the virtual camera defined before.

In the case where the camera is rigidly attached to the quadrotor, the difference between the virtual image plane and the actual image plane is caused by the roll and pitch angle of the quadrotor according to the definition of the virtual camera. The FOV of the camera can be described by a circle with its radius denoted by π. π is also the radius of the actual image plane, and the center of the circle is the center of the image plane. Then, the radius of the virtual image plane ρ is time-varying, and is related to π, roll, and pitch angle of the camera. This time-varying radius in the virtual image plane case is not considered in the current paper. In this paper, the visible set (15) contains the coordinates of the visual features that are inside the radius of the virtual image plane. If the roll and pitch angles of the quadrotor are small during the visual servoing process, the difference between the actual image plane and the virtual image plane is small.

The Zeroing barriers function (ZBF) introduced in [16] belong to a class of non-conservative control barrier functions. ZBF allows it to decrease, which means that it allows the states inside the set approach the boundary of the set.

Using (14) and (16), we can obtain

$$B^{(1)} = 2\left[u\lambda/z \; n\lambda/z \; -(n^2+u^2)/z\right]\mathbf{v}. \tag{17}$$

Base on (17), if the linear velocity $\mathbf{v}$ is chosen as control input, the system has relative degree 1. The ZBF for system with relative degree 1 is B, and the visibility constraint is

$$B^{(1)} \geq -k_b B \tag{18}$$

where $k_b \geq 0$. Since our goal is to ensure that $B \geq 0$, the visibility constraint (18) is intuitive. Given the visual target starts inside the FOV of the camera, which means $B > 0$ at the beginning, (18) allows B to decrease. When B approaches zero, its derivative also approaches zero. When B is negative, its derivative is positive, which implies that B will increase.

If the linear velocity $\mathbf{v}$ can be used as control command, then (18) is the visibility constraint on the control input $\mathbf{v}$. After the IBVS controller computes the desired linear velocity, visibility constraint (18) can be used to check the validity of the control input in terms of visibility, and modify it in a minimal way [14], if necessary, if $\mathbf{v}$ is treated as control input, a hierarchical control structure is needed, and the modified velocity is set as input to a velocity tracking controller to close the control loop.

In this paper, we choose the acceleration $\dot{\mathbf{v}}$ as control input, which makes the system with relative degree 2. Exponential control barrier function that can deal with high relative-degree systems is introduced in [21]. A system with relative degree 2 is the simplest case among the high relative systems.

Define another function

$$H = B^{(1)} + k_b B \tag{19}$$

where $k_b \geq 0$. If $H \geq 0$, we have

$$B^{(1)} \geq -k_b B.$$

According to visibility constraint (18), if $H \geq 0$for all time and the initial value $B(0) \geq 0$, the visible set C is ensured to be forward invariant.

Then, the goal is to guarantee $H \geq 0$for all time. First, the initial value of H should satisfy $H(0) \geq 0$. Based on the ZBF and (18), the following constraints should be satisfied:

$$H^{(1)} \geq -y_b H \tag{20}$$

where $y_b \geq 0$. Substitute (19) into (20)

$$B^{(2)} \geq -(y_b + k_b)B^{(1)} - k_b y_b B. \tag{21}$$

Using (14) and (16), we can get

$$\begin{aligned} B^{(2)} = {} & 2\left[0 \quad \dot{n}\lambda/z - n\lambda\dot{z}/z^2 \quad n^2\dot{z}/z^2 \quad - 2n\dot{n}/z\right]\mathbf{v} \\ & + 2\left[\dot{u}\lambda/z - u\lambda\dot{z}/z^2 \quad 0u^2\dot{z}/z^2 \quad - 2u\dot{u}/z\right]\mathbf{v} \\ & + 2\left[u\lambda/zn\lambda/z \quad -(u^2+n^2)/z\right]\dot{\mathbf{v}}. \end{aligned} \tag{22}$$

Theorem 1: Choosing acceleration $\dot{\mathbf{v}}$ as control input, if the designed control input $\dot{\mathbf{v}}$ satisfy (21), the initial values of the system satisfy $B(0) \geq 0$ and $H(0) \geq 0$, and parameters $y_b \geq 0, k_b \geq 0$, then the visible set C is forward invariant and the visibility is guaranteed.

From the discussion above, the visibility constraint (17) of the ZBF B is satisfied. Based on the results in [16] and [21], the visible set C is forward invariance. Condition (21) is called the visibility constraint in the IBVS of quadrotors.

C. IBVS With Visibility Constraint

This section combines the visibility constraint and the IBVS controller to ensure visibility during a multiobjective task. As discussed in the introduction section, the visibility in the visual servoing can be violated for many reasons. The control inputs generated by the visual controller need to be checked and adjusted to ensure visibility. We want to adjust the control input in a minimal way, such as the original control objective is maximally preserved and the visibility is ensured by satisfying the visibility constraint.

After obtaining the desired linear acceleration$\dot{\mathbf{v}}_d$ using the visual servoing controller (11), a quadratic programming is used to minimally adjust $\dot{\mathbf{v}}_d$ such that (21) is satisfied

$$\begin{aligned} \dot{\mathbf{v}}^* = \underset{\dot{\mathbf{v}}}{\operatorname{argmin}} \quad & J(\dot{v}) = \|\dot{\mathbf{v}} - \dot{\mathbf{v}}_d\|^2 \\ \text{s.t.} \quad & \mathbf{A}_c\dot{\mathbf{v}} \leq b_c \end{aligned} \tag{23}$$

where the inequality constraint in (23) is the same as (21). By expanding the terms in (21) using (16), (17), and (22), we have

$$\begin{aligned}
\mathbf{A}_c &= 2\left[u\lambda/z \ n\lambda/z \ -(u^2+n^2)/z\right]\\
b_c &= (y_b + k_b)B^{(1)} + k_b y_b B\\
&\quad + 2\left[0 \ \lambda(\dot{n}z - n\dot{z})/z^2 \ n^2\dot{z}/z^2 - 2n\dot{n}/z\right]\mathbf{v}\\
&\quad +2\left[\lambda(\dot{u}z - u\dot{z})/z^2 \ 0 \ u^2\dot{z}/z^2 - 2u\dot{u}/z\right]\mathbf{v}.
\end{aligned}$$

Solving the optimization problem (23), the control input $\dot{\mathbf{v}}_d$ is minimally adjusted to satisfy the visibility constraint. Then, the adjusted control commands are used in (11b) for velocity tracking.

III. Simulation and Experiment

In this section, various simulations and experiments are conducted to demonstrate the validity and performance of the proposed controller.

In the following examples, two objectives are considered in the IBVS control of quadrotors. One objective is tracking a trajectory defined by the image features, the controller is given by (11); the second objective is to keep the visual feature inside the FOV of the camera. During the IBVS control process, if the visibility is violated, there will be no visual feedback, and the image-based visual tracking will fail. This will happen if the trajectories of the image features are generated inappropriately, in the sense that they are generated without taking into account the visibility constraint. In this case, strictly following the trajectory is destined to violate visibility.

A. Simulation Results

The gains of the controller in (11) are chosen as $c_1 = 0.1$, $c_2 = 0.1$, $c_3 = 1$, $k_1 = 2$. The parameters of the dynamics of the quadrotor used in the simulations are $m = 2$ kg, $g = 9.81$ m/s^2, and $\mathbf{J} = \mathrm{diag}\{0.0081, 0.0081, 0.0142\}$ kg·m^2/rad^2. Focal length of the camera is 3.2 mm and the pixel of the camera is assumed to be square with length 1.4×10^{-6} m. The visual target is composed of four points lying on the level ground. Their coordinates with respect to the inertial frame (0.25, 0.5, 0), (–0.25, 0.5, 0), (–0.25, –0.5, 0), and (0.25, –0.5, 0) m. The desired trajectories of the image features are defined as follows:

$$\begin{array}{lll}
q_x = at^3 + bt^2 + d, & q_y = 0.2 & 0 \le t < 5\\
q_y = a(t-5)^3 + b(t-5)^2 + d, & q_x = -0.2 & 5 \le t < 10\\
q_x = -a(t-10)^3 - b(t-10)^2 - d, & q_y = -0.2 & 10 \le t < 15\\
q_y = -a(t-15)^3 - b(t-15)^2 - d, & q_x = 0.2 & 15 \le t < 20
\end{array}$$

where $a = 0.0064$, $b = -0.048$, $d = 0.2$. The values of q_4 and q_z are controlled to be constant, with $q_4 = 0$, $q_z = 1$. We choose $a^* = 8 \times 10^{-7}$, thus z^* is 4 accordingly, which means that a^* is obtained when the quadrotor is 4 meter above the visual target. Choosing $q_z = 1$ means that the desired height is 4 m.

Simulation results without the visibility constraint are presented in Fig. 5. As shown in Fig. 5, by tracking the desired image features, the quadrotor will follow a square trajectory. In this case, the controller assumes that the FOV of the camera is big enough such that the paths of the image coordinates [see Fig. 5(e)] are all inside the FOV of the camera.

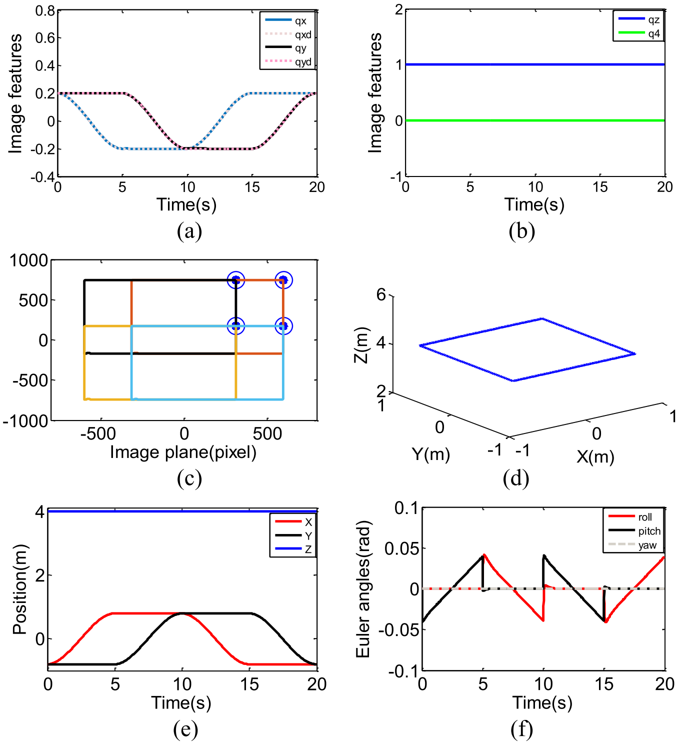
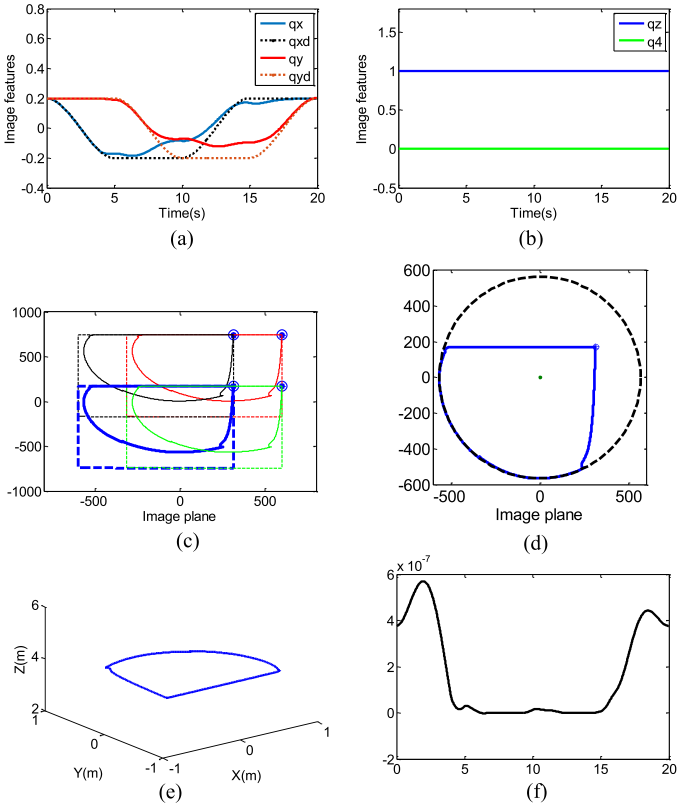


Fig. 5. IBVS without visibility constraint. (a) and (b) Image feature trajectories, dash lines stand for desired trajectories, solid lines stand for actual trajectories. (c) Point coordinates in the virtual image plane. (d) and (e) Time evolution of the UAV's 3-D position. (f) Euler angles.

Next, the visibility constraint is added to the controller. The visible set (15) specifies a set where the coordinate of a visual feature is inside the FOV of the camera. The visibility constraint can be applied to all the point features to ensure that all the point features stay inside the FOV. To demonstrate the effect of the CBF more clearly, only one of the point features is considered in this example.

Note that a higher altitude of the quadrotor can provide a wider view. Therefore, visibility constraint can be achieved in different ways. The two examples to satisfy the visibility specification are: strictly tracking the desired altitude while adjusting the *X*–*Y* plane motion; strictly tracking the desired *X*–*Y* motion while adjusting the height of the quadrotor. Simulation results corresponding to these two cases are presented in Figs. 6 and 7, respectively. The radius of the image plane is $400\sqrt{2}$ pixels. The goal is to restrict one of the point features, which is shown in Fig. 6(d), inside the radius of the image plane.

Fig. 6(f) is based on the visibility constraint (16). To guarantee visibility, the line shown in Fig. 6(f) should be above zero. Note that the unit of the *Y*-axis in Fig. 6(f) is meter. With the pixel length choosing as 1.4×10^{-6}m, ρ in (16) is $= 6.272 \times 10^{-7}$. In the first and last 4 s of the visual servoing task, the trajectory can be tracked without FOV violation. Therefore, control inputs are not adjusted by the CBF, and reliable trajectory tracking is achieved. However, at around 4 s, the point feature is near



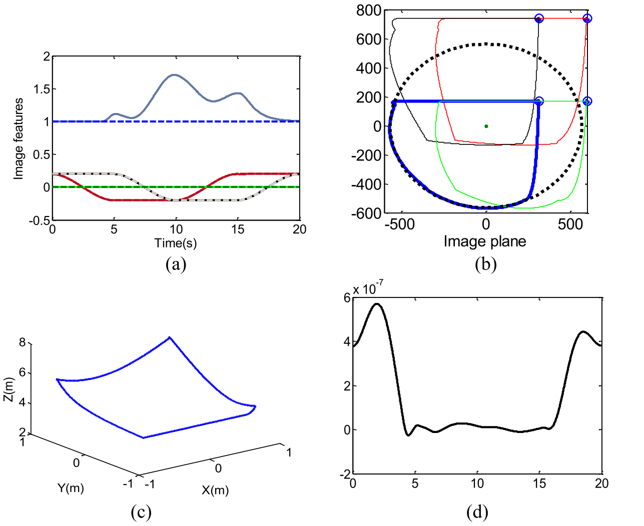
(a) (b) (c) (d) (e) (f)

Fig. 6. IBVS with visibility constraint. (a) and (b) Image feature trajectories, dash lines stand for desired trajectories, solid lines stand for actual trajectories. (c) and (d) Point coordinates in the virtual image plane. (e) 3-D position of the quadrotor UAV. (f) Value of the Zeroing barriers function B.

the edge of the image plane, and the control inputs generated by original IBVS controller will lead to FOV violation. The control inputs are modified by CBF. As can be seen from Fig. 6(f), B is bigger than zero, and point feature is restricted inside the FOV of the camera [see Fig. 6(d)]. In Fig. 7, the vertical motion of the quadrotor is explored to satisfy the visibility constraint and the control inputs with regard to the X–Y plane motion are not affected, as can be seen from Fig. 7(a) and (c).

B. Experimental Results

The micro quadrotor platform used in the experiment is shown in Fig. 8. A monocular camera, a low-cost microcomputer, and a Pixhawk [29] flight controller are equipped onboard the quadrotor to do all the sensing and computation. The overall weight of the platform is 1.25 kg. The visual servoing targets are four static red objects lying on the ground (see Fig. 9). The centroid of each object is extracted using the image processing algorithm. Their coordinates with respect to the inertial frame are (0.15, 0.15, 0), (–0.15, 0.15, 0), (–0.15, –0.15, 0), and (0.15, –0.15, 0) m. Their coordinates in the virtual image plane are used to compute image moments (3). The ground truth position of the quadrotor is recorded using a VICON [30] motion capture system.



(a) (b) (c) (d)

Fig. 7. IBVS with visibility constraint. Adjusting the vertical motion of the quadrotor to satisfy the visibility constraint. (a) Image feature trajectories, dash lines stand for desired trajectories, solid lines stand for actual trajectories. (b) Point coordinates in the virtual image plane. (c) 3-D position of the quadrotor UAV. (d) Value of the Zeroing barriers function B.

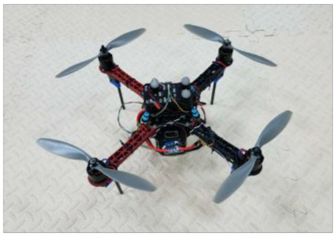
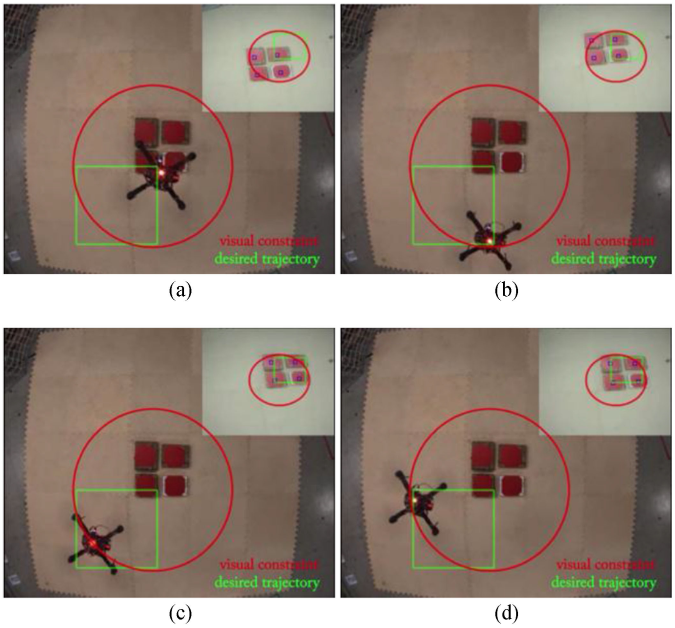


Fig. 8. Quadrotor platform with a microcomputer, a Pixhawk flight controller and a downward facing camera onboard.



(a) (b) (c) (d)

Fig. 9. Snapshots during the visual servoing experiment.

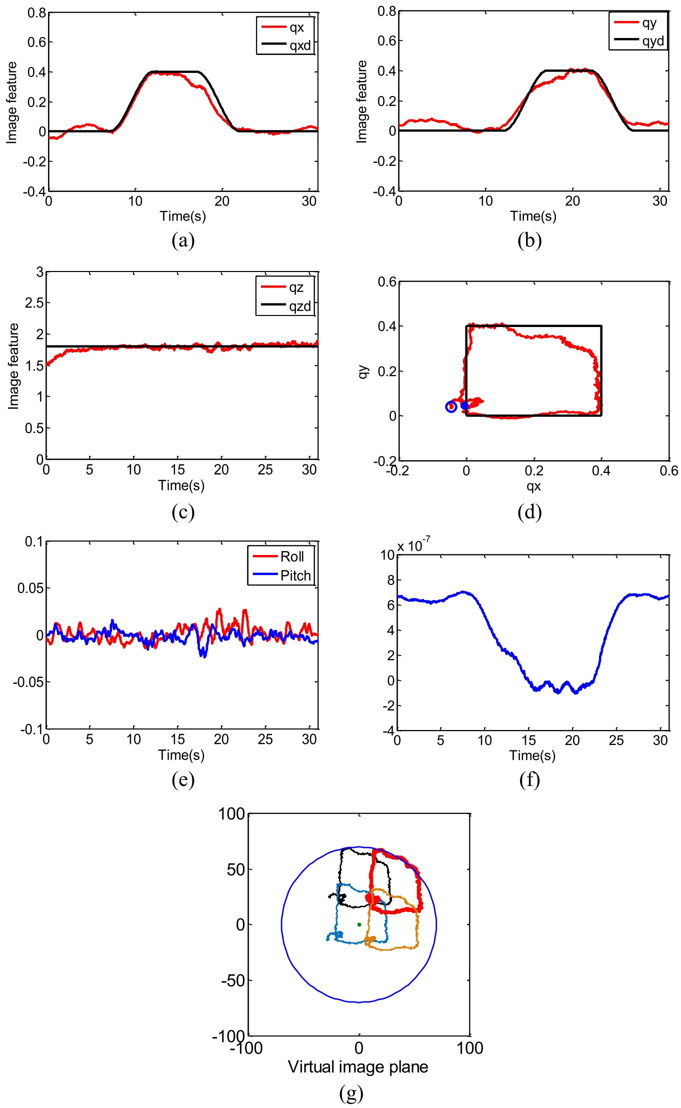


Fig. 10. Experimental result. (a)–(d) Trajectories of the image features. (e) Roll and pitch angles of the quadrotor. (f) Value of the Zeroing barriers function B defined in (16). (g) Paths of the coordinate of the point features in the virtual image plane.

The desired trajectories of the image features are as follows:

$$\begin{array}{lll} q_x = a(t-5)^3 + b(t-5)^2, & q_y = 0 & 5 \le t < 10 \\ q_y = a(t-10)^3 + b(t-10)^2, & q_x = 0.4 & 10 \le t < 15 \\ q_x = -a(t-15)^3 - b(t-15)^2 + d, & q_y = 0.4 & 15 \le t < 20 \\ q_y = -a(t-20)^3 - b(t-20)^2 + d, & q_x = 0 & 20 \le t < 25 \\ q_x = 0, & q_y = 0 & t \in \text{else} \end{array}$$

where $a = -0.0064$, $b = 0.048$, $d = 0.4$. The values of q_4 and q_z are controlled to be constant, with $q_4 = 0$, $q_z = 1.8$.

Experiment results are shown in Figs. 10 and 11. Fig. 10(a) and (b) show the desired image features and the actual image features during the visual servoing. Fig. 10(g) shows the paths of the coordinates of the point features in the virtual image plane. The circle in Fig. 10(g) represents the FOV of the camera. The radius of the virtual image plane ρ in (16) is chosen as 0.00000076, which is 70 pixel. As can be seen from the experimental results, the point coordinates are inside the FOV of the camera. Fig. 10(f) is the value of B defined in (16). In order to guarantee visibility, B should be bigger than or equal to zero. In Fig. 10(f), B is slightly smaller than zero at some points. This is because the ZBF is less conservative than other CBFs, it allows B to decrease to zero to achieve a bigger solution space. Another reason is that the desired acceleration is not the direct control input for quadrotors, the hierarchical control structure may lead B to be slightly smaller than zero.

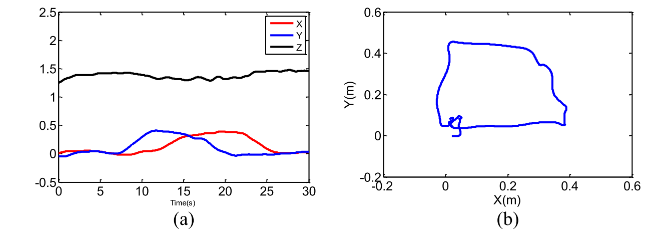


Fig. 11. Position of the quadrotor during the visual servoing.

IV. Conclusion

This paper proposed a method to preserve visibility during the visual servoing of quadrotor UAVs. First, the visible set, which contains the visual features that are inside the FOV of the camera is defined. Then, Control Barrier Function is used to derive the visibility constraint. The satisfaction of the visibility constraint leads to the forward invariance of the visible set, thus ensure visibility. Finally, the original control inputs are modified in a minimal way to satisfy the visibility constraint. With improved visibility, robust visual servoing is achieved. Many applications such as photography, surveillance, and moving objects tracking will benefit from this method. Future extensions including guarantee visibility in moving object tracking, and considering disturbances.

References

[1] F. Chaumette and S. Hutchinson, "Visual servo control. I. Basic approaches," *IEEE Robot. Automat. Mag.*, vol. 13, no. 4, pp. 82–90, Dec. 2006.

[2] V. Lippiello *et al.*, "Hybrid visual servoing with hierarchical task composition for aerial manipulation," *IEEE Robot. Automat. Lett.*, vol. 1, no. 1, pp. 259–266, Jan. 2016.

[3] H. Wang, B. Yang, Y. Liu, W. Chen, X. Liang, and R. Pfeifer, "Visual servoing of soft robot manipulator in constrained environments with an adaptive controller," *IEEE/ASME Trans. Mechatronics*, vol. 22, no. 1, pp. 41–50, Feb. 2017.

[4] H. Wang, D. Guo, X. Liang, W. Chen, G. Hu, and K. K. Leang, "Adaptive vision-based leader-follower formation control of mobile robots," *IEEE Trans. Ind. Electron.*, vol. 64, no. 4, pp. 2893–2902, Apr. 2017

[5] R. Mebarki, V. Lippiello, and B. Siciliano, "Nonlinear visual control of unmanned aerial vehicles in GPS-denied environments," *IEEE Trans. Robot.*, vol. 31, no. 4, pp. 1004–1017, Aug. 2015.

[6] D. Lee, T. Ryan, and H. J. Kim, "Autonomous landing of a VTOL UAV on a moving platform using image-based visual servoing," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2012, pp. 971–976.

[7] H. Jabbari Asl, M. Yazdani, and J. Yoon, "Vision-based tracking control of quadrotor using velocity of image features," *Int. J. Robot. Automat.*, vol. 31, no. 4, pp. 301–309, 2016.

[8] K. Alexis, G. Darivianakis, M. Burri, and R. Siegwart, "Aerial robotic contact-based inspection: Planning and control," *Autonomous Robots*, vol. 40, no. 4, pp. 631–655, 2016.

[9] I. Sa, S. Hrabar, and P. Corke, "Outdoor flight testing of a pole inspection UAV incorporating high-speed vision," in *Field and Service Robotics*. Cham, Switzerland: Springer, 2015, pp. 107–121.

[10] A. Gawel *et al.*, "Aerial picking and delivery of magnetic objects with mavs," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2017, pp. 5746–5752.

[11] J. Chen, T. Liu, and S. Shen, "Tracking a moving target in cluttered environments using a quadrotor," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, 2016, pp. 5310–5317.

[12] H. Xie and A. Lynch, "Input saturated visual servoing for unmanned aerial vehicles," *IEEE/ASME Trans. Mechatronics*, vol. 22, no. 2, pp. 952–960, Apr. 2017.

[13] H. Jabbari Asl and J. Yoon, "Bounded-input control of the quadrotor unmanned aerial vehicle: A vision-based approach," *Asian J. Control*, vol. 19, no. 3, pp. 840–855, 2017.

[14] A. D. Ames, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs with application to adaptive cruise control," in *Proc. IEEE 53rd Annu. Conf. Decis. Control*, Dec. 2014, pp. 6271–6278.

[15] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Trans. Automat. Control*, vol. 62, no. 8, pp. 3861–3876, Aug. 2017.

[16] X. Xu, P. Tabuada, J. W. Grizzle, and A. D. Ames, "Robustness of control barrier functions for safety critical control," in *Proc. IFAC Conf. Anal. Des. Hybrid Syst.*, Oct. 2015, pp. 54–61.

[17] M. Z. Romdlony and B. Jayawardhana, "Stabilization with guaranteed safety using control Lyapunov–barrier function," *Automatica*, vol. 66, pp. 39–47, 2016.

[18] G. Wu and K. Sreenath, "Safety-critical control of a 3d quadrotor with range-limited sensing, " in *Proc. ASME Dyn. Syst. Control Conf.*, 2016, paper DSCC2016-9913.

[19] L. Wang, A. D. Ames, and M. Egerstedt, "Multi-objective compositions for collision-free connectivity maintenance in teams of mobile robots," in *Proc. IEEE Conf. Decis. Control*, 2016, pp. 2659–2664.

[20] L. Wang, A. D. Ames, and M. Egerstedt, "Safety barrier certificates for collisions-free multirobot systems," *IEEE Trans. Robot.*, vol. 33, no. 3, pp. 661–674, Jun. 2017.

[21] Q. Nguyen and K. Sreenath, "Exponential control barrier functions for enforcing high relative-degree safety-critical constraints," in *Proc. Amer. Control Conf.*, 2016, pp. 322–328.

[22] D. Zheng, H. Wang, J. Wang, S. Chen, and W. Chen, "Image-based visual servoing of a quadrotor using virtual camera approach," *IEEE/ASME Trans. Mechatronics*, vol. 22, no. 2, pp. 972–982, Apr. 2017.

[23] D. Zheng, H. Wang, W. Chen, and Y. Wang, "Planning and tracking in image space for image-based visual servoing of a quadrotor," *IEEE Trans. Ind. Electron.*, vol. 65, no. 4, 3376–3385, Apr. 2018.

[24] F. Chaumette, "Image moments: A general and useful set of features for visual servoing," *IEEE Trans. Robot.*, vol. 20, no. 4, pp. 713–723, Aug. 2004.

[25] O. Tahri and F. Chaumette, "Point-based and region-based image moments for visual servoing of planar objects," *IEEE Trans. Robot.*, vol. 21, no. 6, pp. 1116–1127, Dec. 2005.

[26] T. Lee, M. Leok, and N. H. McClamroch, "Nonlinear robust tracking control of a quadrotor UAV on SE (3)," *Asian J. Control*, vol. 15, no. 2, pp. 391–408, 2013.

[27] F. Kendoul, I. Fantoni, and R. Lozano, "Asymptotic stability of hierarchical inner-outer loop-based flight controllers," in *Proc. World Congr. Int. Federation Automat. Control*, Seoul, South Korea, Jul. 2008, pp. 1741–1746.

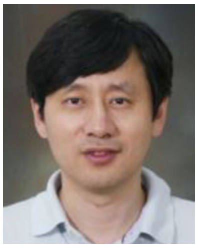
[28] H. Liu, D. Li, Z. Zuo, and Y. Zhong, "Robust three-loop trajectory tracking control for quadrotors with multiple uncertainties," *IEEE Trans. Ind. Electron.*, vol. 63, no. 4, pp. 2263–2274, Apr. 2016.

[29] Pixhawk, 2019. [Online]. Available: http://www.pixhawk.org/

[30] Vicon System, Oxford, U.K., 2019. [Online]. Available: https://www.vicon.com/

Dongliang Zheng received the B.Eng. degree in automation from Northeastern University, Shenyang, China, in 2015, and the M.S. degree in control engineering at Shanghai Jiao Tong University, Shanghai, China, in 2018. He is currently working toward the Ph.D. degree in aerospace engineering in Georgia Institute of Technology, Atlanta, GA, USA.

His current research interests include optimal control, machine learning, motion planning, and robotics.



Hesheng Wang (SM'15) received the B.Eng. degree in electrical engineering from the Harbin Institute of Technology, Harbin, China, in 2002, and the M.Phil. and Ph.D. degrees in automation and computer-aided engineering from the Chinese University of Hong Kong, Hong Kong, in 2004 and 2007, respectively.

From 2007 to 2009, he was a Postdoctoral Fellow and a Researcher Assistant with the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong. Since 2009, he has been with the Shanghai Jiao Tong University, Shanghai, China, where he is currently a Professor with the Department of Automation.

Dr. Wang is an Associate Editor of Assembly Automation, *International Journal of Humanoid Robotics* and IEEE TRANSACTIONS ON ROBOTICS. He is the program Chair of the 2019 IEEE/ASME International Conference on Advanced Intelligent Mechatronics.

Jingchuan Wang received the Ph.D., M.Phil., and B.Eng. degrees in control theory and control engineering from the Shanghai Jiao Tong University (SJTU), Shanghai, China, in 2002, 2005, and 2014, respectively.

He is currently an Associate Professor with the Department of Automation, SJTU. His research interests include service robot, mobile robot's localization and navigation.

Xiufeng Zhang received the Ph.D. degree in mechatronics engineering from the Harbin Institute of Technology, Harbin, China, in 2014.

Since 2014, has been a senior scientist with the National Research Center for Rehabilitation Technical Aids, Beijing, China. His research interests include robotics, pattern recognition, and prosthesis.

Weidong Chen (M'04) received the B.S. and M.S. degrees in control engineering, and the Ph.D. degree in mechatronics, all from the Harbin Institute of Technology, Harbin, China, in 1990, 1993, and 1996, respectively.

Since 1996, he has been at the Shanghai Jiao Tong University, Shanghai, China, where he is currently a Professor with the Department of Automation, and the Director of the Institute of Robotics and Intelligent Processing. His current research interests include autonomous robotics, assistive robotics, and medical robotics.