Visual Servoing for Multirotor Precision Landing in Varying Light Conditions

Jesse Wynn
wynnsquad@gmail.com

Tim McLain
Brigham Young University Mechanical Engineering Department, mclain@byu.edu

Follow this and additional works at: https://scholarsarchive.byu.edu/facpub

Part of the Mechanical Engineering Commons

BYU ScholarsArchive Citation
Wynn, Jesse and McLain, Tim, "Visual Servoing for Multirotor Precision Landing in Varying Light Conditions" (2018). Faculty Publications. 2178.
https://scholarsarchive.byu.edu/facpub/2178

This Conference Paper is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in Faculty Publications by an authorized administrator of BYU ScholarsArchive. For more information, please contact ellen_amatangelo@byu.edu.
Visual Servoing for Multirotor Precision Landing in Varying Light Conditions

Jesse Wynn and Timothy McLain
Department of Mechanical Engineering
Brigham Young University

Abstract—The problem of performing a precision landing of an autonomous multirotor UAV in various lighting conditions is studied. A vision-based approach is proposed and consists of varying degree-of-freedom image-based visual servoing (VDOF-IBVS), and a specialized landing marker. The proposed approach is validated through extensive flight testing outdoors in both lighted and dark conditions, and is done using a standard off-the-shelf autopilot system.

I. INTRODUCTION

Precision landing capability for autonomous multirotor vehicles is of utmost importance in many professional applications. Examples include package delivery [1], automatic docking for recharging and data transfer, and in situations where several vehicles need to land within a small area. Most commercial multirotor vehicles available on the market today use GPS data for positioning and waypoint following. While GPS-aided navigation is an effective means for these tasks, it is typically not sufficient for precision landing due to several sources of GPS error which can lead to total positioning error of several meters [2]. The problem of precision landing has been studied extensively and several important contributions have been made. In [3] vision and inertial measurements are fused in an SRUKF to estimate the position of a stationary target for the purpose of performing a precision landing. In [4] a method for precision landing on a vertically oscillating platform is presented. In [5] a different formulation of image-based visual servoing was used to land on a platform with constant horizontal (1D) motion.

In this work we emphasize applicability to real-world scenarios where precision landing is to be implemented using commercially available hardware in outdoor environments. No motion capture data is used, and a single downward facing camera which is rigidly fixed to the multirotor vehicle body is used to enable precise positioning for the landing task. Unique contributions in this work include a “varying degree-of-freedom image-based visual servoing” (VDOF-IBVS) approach specifically for the multirotor case, the design of a specialized landing target that can be detected day or night, and repeatable full-scale outdoor flight test results.

This paper proceeds as follows. In Section II the method for image-based control is presented and includes details about removing unwanted feature motion as a result of using a non-gimbaled camera, and also gives the derivation of image-based visual servoing control specifically for multirotors. Section III gives details regarding applying image-based visual servoing to the precision landing task, introduces the specialized landing marker design, and details the on-board state machine. In Section IV results from hardware flight tests are given to validate the proposed approach. Conclusions are given in Section V.

II. IMAGE-BASED CONTROL

A. Camera Model

The image-based approach presented in this work assumes a pinhole camera that is rigidly attached near the multirotor center of mass (CM) with its optical axis nearly aligned with the vehicle’s body z-axis (see Fig. 2). The camera provides imagery, from which landing target features are extracted and control is computed. Although the actual camera used does not fit the pinhole camera model perfectly, computer vision techniques such as camera calibration and undistorting of feature points are used and are assumed to have been applied to better represent the idealized camera model. The pinhole camera coordinate system is given in Fig. 3.

B. Level-Frame Feature Mapping

With the camera rigidly attached, rolling and pitching of the vehicle leads to undesirable motion of features in the camera frame, $F^c$. To address this problem, we use a virtual camera frame $F^{cv}$ whose origin is at the vehicle’s CM and
whose optical axis is aligned with the gravity vector. Camera features expressed in this frame are not influenced by rolling and pitching of the vehicle and therefore the undesirable motion is removed.

Let \( P^c = [x^c, y^c, z^c]^\top \) represent some feature in \( F^c \). By augmenting \( P^c \) we now leverage the homogeneous transformation matrix \( T \in \mathbb{R}^{4 \times 4} \) to transform features to be expressed in \( F^{cv} \) as

\[
\begin{bmatrix}
P^{cv} \\
1
\end{bmatrix} = T^{cv} \begin{bmatrix}
P^c \\
1
\end{bmatrix},
\]

where

\[
T^{cv} = \begin{bmatrix}
R^{cv} & d^{cv} \\
0 & 1
\end{bmatrix}.
\]

Here \( R^{cv} \in \text{SO}(3) \) is a function of the vehicle’s current roll and pitch angles (\( \phi, \theta \)), as well as any constant rotational mounting offsets that cause misalignment of the camera frame and the vehicle body frame. The vector \( d^{cv} \in \mathbb{R}^{3 \times 1} \) is the translational component of the homogeneous transform and is composed of the constant \( x, y, \) and \( z \) translational offsets that exist between the origin of \( F^c \) and the origin of \( F^{cv} \). Note that \( d^{cv} \) is resolved in \( F^{cv} \).

The point \( P^c \) is found by first obtaining a vector in \( F^c \) that points in the direction of \( P^c \). This can be done using information available on the image plane, and known focal length (usually available from camera calibration) as

\[
\tilde{p} = \begin{bmatrix}
u \\
v \\
f
\end{bmatrix}.
\]

We now normalize \( \tilde{p} \) to obtain a unit vector \( \zeta \) in the direction of \( P^c \) as

\[
\zeta = \frac{\tilde{p}}{\Vert \tilde{p} \Vert}.
\]

Finally, to obtain the point \( P^c \), we multiply \( \zeta \) by the distance to the feature

\[
P^c = D \zeta.
\]

Simple techniques exist for approximating distance to a ground target when the target size is known, but in our case since we are using an ArUco visual fiducial marker to provide image features, we can obtain this distance using the ArUco library [6].

For the image-based visual servoing approach where control is computed in pixel space on the image plane, the last step in the level-frame mapping process is to project \( P^{cv} \) obtained by (1) onto the image plane of the virtual level camera. This is done by applying the pinhole projection relationship and is given by

\[
\tilde{p}^{cv} = \begin{bmatrix}
u \\
v \\
f
\end{bmatrix} = \frac{f}{z^{cv}} \begin{bmatrix}
x^{cv} \\
y^{cv}
\end{bmatrix}.
\]

C. Image-Based Visual Servoing

To drive the vehicle toward the landing target, we implement an image-based visual servoing controls technique [5], [7]. For this approach, the dynamics of a pixel feature on the image plane are given by

\[
\dot{\tilde{p}} = \begin{bmatrix}
\ddot{u} \\
\ddot{v}
\end{bmatrix} = J_p \mathbf{v},
\]

where \( \mathbf{v} = [v_x, v_y, v_z, \omega_x, \omega_y, \omega_z]^\top \) are the linear and angular velocities of the camera frame and \( J_p \) is the image Jacobian given by

\[
J_p = \begin{bmatrix}
-f & 0 & u & \frac{uv}{\tilde{Z}} & \frac{u^2}{\tilde{Z}} - \frac{fu}{\tilde{T}} & -u \\
0 & -f & v & \frac{uv}{\tilde{Z}} & \frac{v^2}{\tilde{T}} - \frac{fu}{\tilde{T}} & -v
\end{bmatrix},
\]

where \( f \) is the focal length of the camera in pixel units, \( u \) and \( v \) are the features pixel coordinates, and \( \tilde{Z} \) is the distance to the feature in \( F^c \). The quadrotor vehicle is under-actuated [8] and the autopilot system can only accept linear reference velocities, and a yaw-rate command. In light of this we modify (8) to suit the quadrotor-specific case as

\[
J_{p,4DOF} = \begin{bmatrix}
-f & 0 & u & \frac{uv}{\tilde{Z}} & v \\
0 & -f & v & \frac{uv}{\tilde{Z}} & -u
\end{bmatrix},
\]

and (7) becomes

\[
\dot{\tilde{p}} = \begin{bmatrix}
\ddot{u} \\
\ddot{v}
\end{bmatrix} = J_{p,4DOF} \mathbf{v}.
\]
where now \( v = [v_x, v_y, v_z, \omega_z]^T \) and we assume that \( p = p^\text{cv} \).

Given camera velocity \( v \), (10) allows us to compute the resultant pixel velocity for an image feature \( p \) in the virtual camera. This is interesting, but for the precision landing problem we desire to compute the appropriate camera velocities (and subsequently multirotor velocities) to control feature motion in the image plane. Applying the method given in [7] to our four degree-of-freedom case, since (10) is a linear system of matrix equations, we can stack image Jacobians for several feature points that reside on a rigid target body as

\[
\begin{bmatrix}
\dot{p}_1 \\
\vdots \\
\dot{p}_n
\end{bmatrix} = \begin{bmatrix}
J_{p1,4DOF} & \vdots & J_{pn,4DOF}
\end{bmatrix} v.
\]

(11)

For computing \( v \), we substitute the \( \dot{p} \) terms in (11) with a simple linear regulator controller [9] given as

\[
\dot{p} = K (p_{\text{des}} - p),
\]

(12)

where \( p \) is a feature’s pixel coordinate on the virtual image plane, \( p_{\text{des}} \) is the desired feature coordinate, and \( K \) is a tunable gain parameter. Finally, if \( n > 2 \) and image features are not colinear, then the pseudo-inverse can be applied to obtain the desired expression for controlling feature motion and is given by

\[
v = \begin{bmatrix}
J_{p1,4DOF} & \vdots & J_{pn,4DOF}
\end{bmatrix}^+ \begin{bmatrix}
p_{1,\text{des}} - p_1 \\
p_{2,\text{des}} - p_2 \\
\vdots \\
p_{n,\text{des}} - p_n
\end{bmatrix} K,
\]

(13)

where \( K \in \mathbb{R}^{2n \times 2n} \) is a diagonal gain matrix.

Eq. (13) provides reference velocities expressed in \( F^{cv} \) and so \( v \) must be transformed to be expressed in the autopilot’s coordinate frame \( F^a \). After the transformation, the remaining step in preparing the IBVS velocity term is to apply saturation limits. The need for saturation arises from computing image-based visual servoing at large distances (several meters) from the target features and results in excessively large velocity references even when \((p_{\text{des}} - p)\) is relatively small. Therefore the velocity reference that is sent to the autopilot system is given as

\[
v_{\text{sat}} = \text{sat}(v).
\]

(14)

In some situations the full four-degree-of-freedom velocity reference from (13) is not desirable. For example, if the image features appear near the edge, or in the corner of the camera field-of-view, then descending (positive \( v_z \)) or yawing (\( \omega_z \)) could push features out of view. To avoid this situation, we can further modify (8) and remove all but the first two columns to obtain

\[
J_{p,2DOF} = \begin{bmatrix}
-\frac{f}{z} & 0 & 0 \\
0 & -\frac{f}{z}
\end{bmatrix},
\]

(15)

which is a two-degree-of-freedom image Jacobian. Using \( J_{p,2DOF} \) instead of \( J_{p,4DOF} \) in Eq. (10) through Eq. (13) results in \( v = [v_x, v_y]^T \) that can be augmented with zeros and then be passed through (14) in the same manner as was done in the four-degree-of-freedom case. The ability to actively switch between computing \( v \) based on (9) or (15) is what we term, varying degree-of-freedom image-based visual servoing (VDOF-IBVS).

The decision to compute \( v \) based on \( J_{p,4DOF} \) or \( J_{p,2DOF} \) is made by computing the pixel distance \( d \) between the centroids of the desired feature points \( p_{\text{des}} \) and the target feature points \( p \) on the image plane of \( F^a \). Let \( r_1 \) and \( r_2 \) be two pixel distance thresholds where \( r_1 > r_2 \). If \( v \) is currently computed based on \( J_{p,4DOF} \) and \( d > r_1 \), we switch to computing \( v \) based on \( J_{p,2DOF} \). We then stay in two-degree-of-freedom mode until \( d < r_2 \) at which point we resume four-degree-of-freedom IBVS.

### III. Precision Landing

#### A. Visual Servoing For Precision Landing

When equipped with a downward-facing camera, the image-based visual servoing approach can be used effectively for precision landing tasks. The left frame in Fig. 5 gives a hypothetical view of the landing marker through the downward-facing camera where the vehicle is roughly positioned above the marker at some altitude. The right frame in Fig. 5 shows the hypothetical result of image-based visual servoing where \( p_{\text{des}} - p = 0 \) and the vehicle is now precisely positioned above the marker at a lower altitude. This is the basic approach that we implement to accomplish the precision landing.
B. Nested Landing Marker Design

To facilitate full-scale outdoor precision landing tasks, a specialized landing marker was developed. Requirements for the marker were established as: robustly detectable at high rate, visible from afar, visible up-close, and visibility during the day and at night. To achieve robust target detection, we leverage the open-source ArUco project and compose our landing marker using ArUco fiducials. ArUco was chosen due to its robustness and low processing time [6].

To enable visibility from afar we made our landing target and ArUco fiducial relatively large and the primary ArUco marker has a side length of 0.7 m. Due to its size however, the large ArUco marker is not detectable up close. This is because at small distances, the ArUco marker fills and even exceeds the camera’s field-of-view. To solve this problem, we nest a small ArUco marker at the center of the larger marker. The smaller (inner) marker was sized to have a 0.12 m side length which made it detectable when the larger (outer) marker was exceeding the camera field-of-view, and was also selected such that its presence wouldn’t hinder detection of the larger marker. A depiction of the developed nested ArUco marker is given in Fig. 6.

The ArUco landing marker is visible during the daytime with the greatest challenge being glare from the sun. This problem was mitigated by lightly sanding the surface of the marker such that it would be less reflective. To enable detection at night, the marker surface material was chosen to be a translucent white acrylic that is then back-lit by an array of 264 infrared (IR) LEDs. The acrylic marker surface serves as a light diffuser to give even lighting of the white portions marker, and also is sturdy enough to be landed on. When paired with a standard RGB camera whose IR filter has been removed, the marker becomes visible at night. Views of the constructed marker are given in Fig. 7, and views of the marker taken from a flying multirotor at night are given in Fig. 8. Fig. 9 gives results displaying the average detection rate of both the outer and inner ArUco markers in the range of 0 to 20 m with a camera rate of 30 Hz. Notice that there is a region of overlap in the range of 1 to 2.5 m where both markers are detected at full rate. This region of overlap enables a smooth transition from computing control based on the outer marker to the inner marker during the precision landing maneuver.

C. Landing Procedure

The precision landing maneuver is guided by a state machine that begins with the vehicle having completed its intended mission, and ready to begin execution of a precision landing. The landing state machine consists of three finite states: Rendezvous, IBVS, and Landing with transition between states occurring when certain criteria are met. Descriptions of each mode are now given:

1) Rendezvous: In this mode a position reference which is directly above the marker at an altitude of approximately 13 m is sent to the autopilot. The vehicle navigates to this position and measurements are supplied by the on-board GPS module. Once the vehicle is within one meter of the rendezvous location, it attempts to detect the landing marker. If the marker is detected at least
five times within a one second interval, then the marker is considered visible and the state machine switches to IBVS mode.

2) IBVS: In IBVS mode, image-based visual servoing commands are computed according to (13), (14) and are then sent to the autopilot as velocity references. This mode has two stages of its own: IBVS Outer and IBVS Inner. In the IBVS Outer stage, the corners of the outer marker are used as image feature points, and desired feature points are defined such that visual servoing will drive the vehicle to a location which is directly above the marker at a distance of approximately two meters. At this location, the vehicle is near the center of the overlap region where both markers are detectable at full frame rate. If $\|p_{des} - p\|$ is sufficiently small for the outer marker, and the inner marker is visible, the switch from computing control based on the outer marker to control based on the inner marker is performed. In the IBVS Inner stage, desired feature locations are set to position the vehicle directly over the center marker at a distance of approximately 30 cm. If at any time while the vehicle is in IBVS mode the landing marker is lost for more than 1 second, the vehicle returns to Rendezvous mode and begins ascending to the Rendezvous location. We call this a go-around maneuver. If during the ascent the outer or inner marker again becomes visible, the vehicle quickly re-enters IBVS mode and need not return all the way to the rendezvous location.

3) Land: The vehicle enters Land mode when $\|p_{des} - p\|$ for the inner marker is sufficiently small, and the distance above the marker is less than a certain threshold. In Land mode, attitude references are sent to the autopilot and the vehicle is set to hold its current attitude while the motors are quickly ramped down to idle. Holding attitude rather than setting roll and pitch angles to zero is done so that the vehicle will not drift significantly with the wind. The final landing maneuver takes place in less than a second, after which the vehicle is automatically disarmed. Although this stage of the precision landing is done open-loop, little extra error is accumulated because the maneuver happens quickly and occurs over a small distance.

IV. HARDWARE FLIGHT TEST RESULTS

A. Vehicle Hardware Description

The vehicle used for these tests is a 3DRobotics X8 multirotor which has four sets of coaxial motors. The motor-to-motor diagonal distance is 0.57 m and the vehicle’s ready-to-fly weight is 3.29 kg. The downward-facing camera is a FLIR® Chameleon3 global shutter sensor which has a resolution of 1288×964 and is fitted with a 3.6 mm M12 lens. The field-of-view of the camera is 1.36 rad (78 deg). Image processing, and image-based control is computed on-board in real-time using the NVIDIA® Jetson TX2. The autopilot system is a Pixhawk 2.1 flashed with PX4 firmware and the ROS package MAVROS provides a communications bridge between the TX2 and the Pixhawk. A system communications diagram is given in Fig. 10.

B. Flight Results

The precision landing process was validated through hardware flight testing both during the day and at night. For these tests the nested ArUco landing target was placed at some location near the take-off site, and the vehicle was placed on the target to measure its location. The vehicle was then moved back to the take-off site and the test was carried out. For simplicity and safety, the vehicle was taken-off manually by a safety pilot, and brought to an altitude a few meters off the ground before switching the vehicle into autonomous mode. Fig. 11 gives the NED location of the vehicle while autonomous mode was active. The vehicle starts at $[0.5, -0.1, -8.3]^T$ and finishes the landing maneuver at $[2.1, -9.6, -0.1]^T$. The vertical lines in the plots signify when certain events took place. The dotted line at $t = 29$ s indicates when IBVS mode became active. The dashed line at $t = 46$ s is when the vehicle switched to computing control based on the inner marker. Finally, the dash-dot line at $t = 50$ s and the solid line at $t = 52$ s indicate when Land mode and touchdown occurred respectively.

Fig. 12 gives the pixel error distance for each of the four corner feature locations while IBVS mode was active. Here we observe that the pixel error was driven towards zero while IBVS control was executed. Note that the rate of convergence toward zero increases significantly around $t = 42$ s. This is consistent with the dynamics of the image Jacobian where each term is a function of distance to the target. In Fig. 13, The raw distance measurements from ArUco are given for the duration of the test.

To show consistent performance of the developed landing technique, the flight experiment was repeated consecutively five times at night, and then again in daylight. The results of these tests are summarized in Tables I and II. Landing Error is the measured horizontal distance between the vehicle CM and the center of the ArUco target, and Landing Time is the time between when IBVS became active, and when touchdown
occurred. The combined average landing error for these tests was 0.07 m and the average landing time was 26.4 s.

V. CONCLUSION

Here we have presented a vision-based approach to enable precision landing for a multirotor vehicle operating during the day or at night. A varying-degree-of-freedom version of image-based visual servoing is used to compute velocity references, and these references are given as commands to the off-the-shelf autopilot system. A specialized IR-illuminated nested landing target was developed which enables day and night operations. The whole landing procedure is guided by a simple state machine which integrates well with a generic autopilot infrastructure. Finally, results validating the proposed approach were given.

The approach has been shown to be effective, consistent, and far more accurate than GPS-only landing approaches. A merit of this work is that it was done using only off-the-shelf equipment, and required no external infrastructure such as a motion capture system. This aspect makes this work implementable and relevant to many real-world applications. A natural extension to this work would be to consider a moving landing target to enable land-based or maritime applications where it is desirable to perform a precision landing while on the go.

REFERENCES


