

Autonomous landing of a UAV on a custom ground marker using image-based visual servoing

Ashwin Disa

*Department of Aeronautical and Automobile Engineering
Manipal Academy of Higher Education
Udupi, India
ashwin.disa@gmail.com*

Vishnu G. Nair

*Department of Aeronautical and Automobile Engineering
Manipal Academy of Higher Education
Udupi, India
vishnu.nair@manipal.edu*

Abstract—Landing is a crucial maneuver for a successful flight mission. This paper presents vision-based ground marker detection and autonomous landing of Unmanned Aerial Vehicles (UAVs). The methodology is based on a monocular vision system that detects the predefined ground marker and determines the altitude of the UAV solely from the vision sensor data. The same vision system is also used for positioning and estimating the orientation of the UAV with respect to the ground marker. A proportional controller is implemented for landing the UAV accurately. Image Based Visual Servoing (IBVS) control technique is implemented to reduce the computational cost of the system. The algorithm accounts for abnormal situations such as loss of ground marker in the image frame or a partially visible ground marker. The performance of the landing system is verified through both simulations and outdoor experimentation. The results show the accuracy and robustness of the proposed autonomous landing strategy.

Index Terms—Unmanned aerial vehicle, autonomous landing, image-based visual servoing, software-in-the-loop

I. INTRODUCTION

Over the past few years, Unmanned Aerial Vehicles (UAVs), especially multi-rotors have been rapidly gaining interest with various applications in the field of agriculture, surveillance [1] and emergency response to name a few. Use of aerial vehicles has resulted in substantial development of these applications along with decreased dependency on human labor, thus reducing human intervention while performing some of the dangerous tasks. A typical UAV mission involves take-off, performing tasks and landing. A slight error in the maneuver can damage the UAV and possess a threat due to its high RPM propellers. To safely land the UAV autonomously is the major objective of this paper.

The rest of the paper is organized as follows - Section II and III presents the problem statement and related literature respectively, followed by the implementation methodology in Section IV. Section V is dedicated for results and discussions followed by conclusion.

II. RELATED LITERATURE

Autonomous landing of UAV has been an active topic of research over the past few years [2]. Several methodologies were studied including nonlinear control, intelligent control, vision-based control [3] etc.

Use of ground markers became a common technique in the implementation of vision-based landing algorithms for UAVs on both static as well as dynamic platforms [4] (such as on the deck of a ship). Thomas et al. [5] presented a relative pose estimation and trajectory planning strategy to track a moving sphere with an under-actuated 250 g Micro Aerial Vehicle (MAV) with a downward facing camera. In [6], a closed loop control system is formed in the 2-D image frame, extracting image features from the visual information of the observed object that is the ground marker in the image frame, and calculate the error compared to the desirable position. Here, IBVS (Image Based Visual Servoing) control technique [6], [7] is used. Implementation of IBVS requires few assumptions such as the ground marker should not leave the image frame and the features should remain static with respect to the ground marker.

Accurately estimating the altitude of a UAV is a crucial task for autonomous landing. A common method used for depth estimation is stereo vision [4]. The results are expected to produce more accurate height information than the GPS system. Sereewattana et al. [8] used a monocular camera which captured two consecutive ground images to simulate a pair of stereo images. Guevara et al. [9] considered two ways for altitude estimation. In the first method the altitude is determined by using a predefined relation between the height of the UAV and the radius of the landing area detected by the vision sensor. Another is in case the UAV drifted too much and the landing area is not detected, an ultrasonic sensor is used. A similar technique is followed in this paper to estimate the altitude of the UAV.

III. PROBLEM STATEMENT

This paper focuses on the problems and shortcomings of the previous research works. In this paper we have proposed a system which eliminates the use of an extra sensor such as ultrasonic sensor or a laser rangefinder (LIDAR) for altitude estimation. Also, there is no proper solution investigated in case of partially visible markers which can arise if the landing needs to be done on an irregular terrain or due to varying light conditions and circumstances such that. Improper or unsuccessful detection can lead to failed autonomous landing in case of a vision-based system. All these situations are

considered by the algorithm proposed in this paper. Here, we solely depend on the vision sensor for pose estimation of the UAV with respect to the ground marker along with altitude estimation. This enables the UAV to land autonomously in GPS denied areas [10], [11] such as indoor environments. The system is computationally inexpensive such that it can run on Raspberry Pi (RPi), a low cost onboard computer replacing the high specification and expensive CPUs, by the following ways:

- 1) Simple image processing techniques like thresholding, contour detection and segmentation are implemented using the OpenCV library.
- 2) Image Based Visual Servoing (IBVS) technique in which a real-time feedback is sent from the vision sensor to the controller.
- 3) A proportional controller is implemented to control the motion of the quadcopter.

IV. METHODOLOGY

A. Marker Description

Red color for the ground marker is selected as it will be less scattered due to its greater wavelength, thereby aiding the detection from higher altitudes. Also, the marker design is kept simple so that it can be easily identified by the detection algorithm running at a high frame rate on the RPi. A red velvet chart paper is used for experimentation because of its bright texture and visibility in low light conditions. A cyan colored region is added so as to orient the quadcopter accurately with respect to the marker. Cyan color gave better detection results compared to other colors during experimentation.

B. Marker Detection

The image processing algorithm is implemented using the OpenCV library. This algorithm running on the RPi gets video feed from the USB camera attached to the bottom of the quadcopter. In each iteration of the algorithm, one image is processed. A CvBridge is used to convert the ROS image message containing the information into OpenCV image format. This image is used for further processing. The algorithm is divided into two parts: thresholding and filtering, and contour and its center detection.

The upper and lower pixel value of the RGB color format, that are (R_u, G_u, B_u) and (R_l, G_l, B_l) are set for the red and cyan color in the algorithm beforehand. These values are picked by trial and error method over a large data set of ground marker images taken at different locations and varying lighting conditions using the USB camera. Then the captured image is converted from RGB to HSV (Hue, Saturation and Value) format as shown in Fig. 1(b). Thresholding is applied, i.e every pixel in the image frame is checked, if the pixel value is within the given acceptable range, the output will be white, and black otherwise, using a *filtercolor()* function. The output image is in the form of a binary image (black and white) as shown in Fig. 1(c).

Once we have the binary image, A *drawContours()* function is used to detect the white region i.e the ROI (Region of

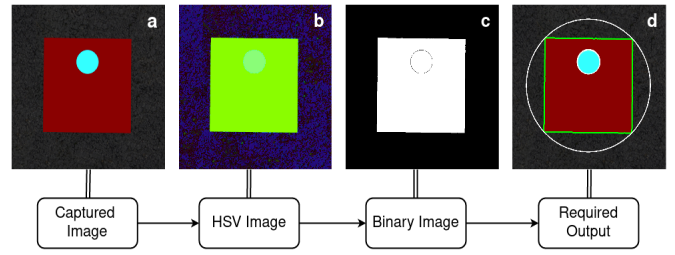


Fig. 1. Illustration of the proposed target detection strategy. The input is the video feed from the camera. At each iteration, (a) one image is processed. The RGB image is converted to HSV format for thresholding and filtering, shown in (b). The output is the binary image as shown in (c). The white circle drawn in (d) is the minimum enclosing circle used for altitude estimation and center detection for the control strategy.

Interest) in the image frame. A *max()* function is implemented to determine the contour with the biggest area. This eliminates errors when the algorithm detects unwanted objects of similar color in the environment. To determine the centroid of the contour of the ground marker, *minEnclosingCircle()* function is used. Fig. 1(d) shows the minimum enclosing circle, whose radius is used for altitude estimation. Here, the 1st order spatial moments about the x and y axes and the zeroth order central moments of the binary image are calculated. Zeroth order central moments are equal to the contour area i.e the white region in the binary image. The x and y coordinates of the ground marker's centroid are given by

$$x_{mar} = 'm10'/'m00' \quad (1)$$

$$y_{mar} = 'm01'/'m00' \quad (2)$$

Where 'm10' and 'm01' are the 1st order spatial moments about x and y-axis respectively and 'm00' is the zeroth order central moments. Along with the centroid coordinates x_{mar} and y_{mar} , the radius of the minimum enclosing circle is determined. The radius is used to calculate the area of this circle which is then used for altitude estimation of the quadcopter. Coordinates of the centroid are used to estimate the relative position of the quadcopter with respect to the ground marker in the image frame. The relative position is denoted by x_{rel} and y_{rel} .

$$x_{rel} = x_{mar} - (size_of_image_frame/2) \quad (3)$$

$$y_{rel} = y_{mar} - (size_of_image_frame/2) \quad (4)$$

As the position of the ground marker in the image frame is in terms of pixels, to get the relative position of the quadcopter, 200 is subtracted from x_{mar} and y_{mar} that is the value of $size_of_image_frame/2$. The size of the image frame is set to 400×400 pixels.

C. Altitude Estimation

Area of the minimum enclosing circle estimated as given in Section IV-B is calculated using the radius from the *minEnclosingCircle()* function. This area is plotted against

the actual altitude data from the sensors as shown in Fig. 2. The Curve Fitting Toolbox in MATLAB is used to generate a power equation from the plot, shown in Eq. 5. The generated equation has a R-square value of 0.997 which indicates good correlation between the altitude of the quadcopter and the area of the minimum enclosing circle. The constants in the equation are generated in MATLAB.

$$z_{rel} = 242 * area^{-0.4729} - 0.01454 \quad (5)$$

Where z_{rel} is the altitude of the quadcopter and $area$ is the area of the detected minimum enclosing circle. Therefore whenever the quadcopter detects the ground marker, the area of the circle is given as input to Eq. 5 and the output is the altitude of the quadcopter with respect to the ground marker. This means that the size, shape and color of the ground marker needs to be the same for correct altitude estimation at all times. In case of a partially visible marker, given that any two opposite vertices of the ground marker are visible, the area of the circle obtained from the estimated radius by the $minEnclosingCircle()$ function won't be affected. As stated in Section III, this method eliminates the use of an extra sensor such as an ultrasonic sensor or a laser rangefinder to estimate the altitude of the UAV for autonomous landing.

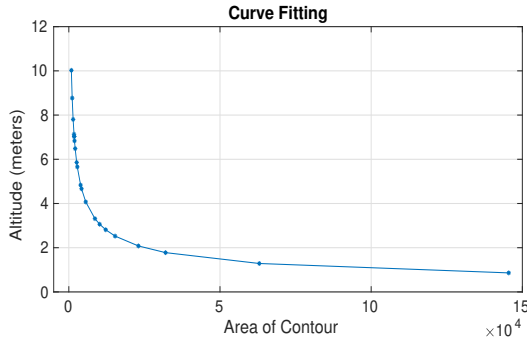


Fig. 2. Graph for altitude of the drone in meters v/s area of the contour detected by the vision system. It is used to derive the equation for altitude estimation given in Eq. 5, using the curve fitting toolbox in MATLAB.

D. Control

The local coordinate system of the quadcopter is denoted by x_{uav} , y_{uav} , z_{uav} . And the origin as the optical center of the camera. x_{uav} in the forward direction, y_{uav} towards the left and z_{uav} pointing in the upward direction. Here the optical center of the camera and the centroid of the quadcopter are considered to be the same. The ground marker coordinate system is denoted as x_{mar} , y_{mar} and z_{mar} , origin being the center of the ground marker. x_{mar} pointing towards the right, y_{mar} towards the front and z_{mar} in the upward direction as shown in Fig. 3(a). Relative position is established in the local coordinate system in order to correctly represent the relative position between the quadcopter and the ground marker, denoted by x_{rel} , y_{rel} and z_{rel} as shown in Eqs. 3, 4 and 5.

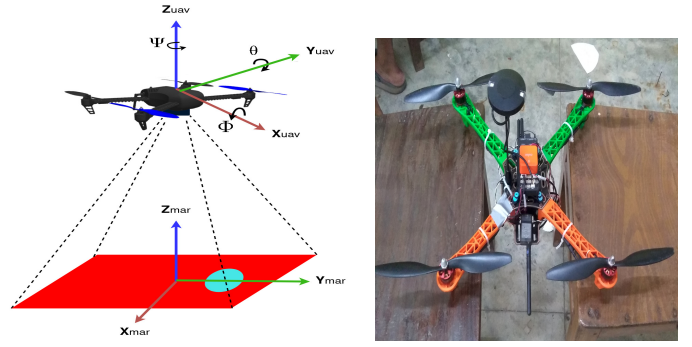


Fig. 3. (a) An illustration of the quadcopter and the ground marker used for the autonomous landing in simulation. It shows the local and ground marker coordinate axes along with the camera projections. ϕ is the roll or rotation along the X_{uav} axis, θ denotes the pitch or rotation along the Y_{uav} axis and ψ is the yaw or rotation along the Z_{uav} axis. (b) Actual image of the quadcopter used for outdoor experiments.

A cyan region is added to the ground marker for the purpose of orienting the quadcopter accurately with the help of the vision system. Same as the red color, the cyan color is filtered using thresholding technique and the center is detected by drawing a contour and using the $minEnclosingCircle()$ function. The center of the cyan region is denoted by x_{cyan} and y_{cyan} in the ground marker coordinate system. The error in orientation ($x_{cyan} - x_{mar}$) as shown in Fig. 4(a), and positional error i.e x_{rel} , y_{rel} and z_{rel} are sent from the detection algorithm to the control algorithm through a ROS topic. A *counter* is introduced which acts as a switch between orientation and translation maneuver. Orientation is given the higher priority. In order to orient the quadcopter accurately such that the center of the cyan region and the ground marker both lie on the vertical axis, the idea is to minimize the term ($x_{cyan} - x_{mar}$) by yawing the quadcopter. This is done by using a proportional controller, shown in Eq. 6, by giving an angular velocity input about the z-axis i.e yaw while maintaining the position of the quadcopter.

$$V_{yaw} = K_{yaw} * (x_{cyan} - x_{mar}) \quad (6)$$

Where K_{yaw} is the proportional gain constant. A condition that the term ($y_{cyan} - y_{mar}$) should always be positive is considered to avoid a situation where the quadcopter is oriented by 180 degrees with respect to the ground marker.

Once the quadcopter is oriented, it starts to translate over to the ground marker. This is done by minimizing the x_{rel} and y_{rel} terms. Fig. 4(b) shows an instance of the image frame after the ground marker has been oriented accurately, and now the objective is to translate over to the marker by minimizing the relative error terms x_{rel} and y_{rel} by applying a velocity input in the local coordinate frame as shown in Eqs. 7 and 8. Along with translation in x and y, the quadcopter starts descending by minimizing the error in altitude i.e ($z_{uav} - z_{mar}$) as shown in Eq. 9. Once the quadcopter is within an acceptable (x_{rel} , y_{rel}) range of ten pixels over the ground marker and below one

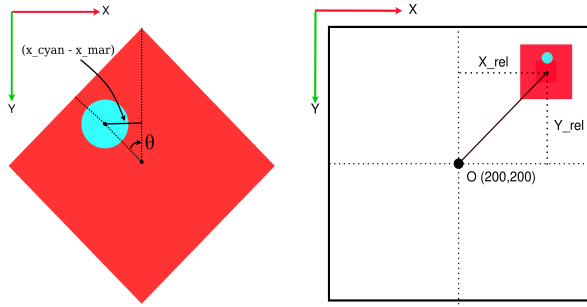


Fig. 4. (a) Enlarged view of the ground marker. Here the orientation of the ground marker is denoted by θ in the global coordinate system. The goal is to minimize the θ , by minimizing the term $(x_{cyan} - x_{mar})$ by an angular velocity input about the z-axis i.e yaw. (x_{cyan} and x_{mar} are the x co-ordinates of the centroid of the cyan and red contours respectively). (b) Ground marker in the image frame of size 400×400 pixels. The objective is to minimize the x_{rel} and y_{rel} terms to accurately position the UAV over the ground marker. O denotes the center of the image frame which coincides with the centroid of the quadcopter in the 2D plane.

meter altitude ($z_{rel} < 1$), the translation maneuver is complete and the quadcopter is ready to land.

$$V_x = K_x * (x_{uav} - x_{mar}) \quad (7)$$

$$V_y = K_y * (y_{uav} - y_{mar}) \quad (8)$$

$$V_z = K_z * (z_{uav} - z_{mar}) \quad (9)$$

Where V_x , V_y and V_z are the velocity inputs in the local coordinate system of the quadcopter along the x, y and z axes respectively. K_x , K_y , K_z and K_{yaw} are the proportional gain constants. A comparison of these gain constants is shown in Fig. 5 to get the desired output. The velocities given to the flight controller need to be with respect to the global coordinate system. A 2-D rotation matrix about the z-axis as shown in Eq. 10 is used to convert the velocities in the local coordinate system to the global coordinate system. Here V'_x , V'_y and V'_z are the velocities in the global frame, V_x , V_y and V_z are in the local frame and θ is the orientation angle of the quadcopter in the global frame about the z-axis. Liu et al. [12] used a similar strategy for landing control.

$$\begin{bmatrix} V'_x \\ V'_y \\ V'_z \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} \quad (10)$$

The detection algorithm sends data (x_{rel} , y_{rel} , z_{rel} , yaw_{rel} , $counter$) to the control algorithm once every 100 ms, and the control inputs are sent to the flight controller of the quadcopter to generate appropriate flight control commands every 20 ms. Once the ground marker is visible in the image frame, the quadcopter is set to offboard mode by sending a service request to the PX4 firmware. If during the maneuver, the marker is no longer visible in the image frame, failsafe is enabled automatically and the offboard mode is set off and position mode is set on. Now the pilot can manually take over the control from the auto-pilot and maneuver it so that the ground marker is again visible in the image frame and the

quadcopter can continue the autonomous landing by enabling the offboard mode. Once the orientation and translation is completed and the quadcopter is within an altitude of one meter, the control algorithm sends a service request to the firmware to land and disarm the UAV.

V. EXPERIMENTAL RESULTS

The proposed system consists of the following components for outdoor experiments: a Q450 quadcopter frame, 14.8V 3300 mAh 4S Li-Po battery to power the electronics, 920KV Brushless DC Motors, 40 Amps ESCs with 10 inch length and 4.5 inch pitch propellers. Taranis X7 RC is used for manually controlling the quadcopter with X8R telemetry receiver. PX4 auto-pilot firmware is integrated on the Cube Orange flight controller. PX4 firmware is an open source auto-pilot software written in C++. Raspberry Pi 4 Model B(RPi) is used as the onboard processor as it has better performance than the previous versions and suitable for image processing. A USB camera is connected to the RPi via a USB 2.0 cable and used for sending vision data at 20 fps. The output of the algorithm is sent from the RPi to the flight controller which then generates the appropriate flight control commands.

A. Performance

10 flight test simulations and 5 outdoor flight test experiments are performed to validate the proposed methodology. The mean error in the position of the quadcopter with respect to the centroid of the ground marker in the x direction is found to be about 0.22 meters, and 0.12 meters in the y direction. The mean error in the orientation of the quadcopter is about 7.6 degrees. The average time for landing is about 26.4 seconds from an altitude of about four meters. The mean results of the flight tests conducted in the outdoor environment were evidently less accurate and consumed more time compared to the results from simulation, the results are tabulated in Table I.

TABLE I
EXPERIMENTAL RESULTS

	Simulation	Hardware
Error in x (meters)	0.1	0.22
Error in y (meters)	0.06	0.12
Error in yaw (degrees)	2.3	7.6
Total time (seconds)	12.6	26.4

B. Marker Detection

The main cause of an unsuccessful detection and failure while using a vision-based system is due to the change in external lighting conditions. Therefore, our system has been evaluated by performing few set of experiments in such scenarios. Fig. 6 shows different indoor and outdoor challenging conditions in which our system has been tested. In Fig. 6(a) and Fig. 6(b) the landing marker is deliberately placed to evaluate the behaviour of the vision system in case of shadows due to trees and buildings. The algorithm is robust and detects the whole ground marker irrespective of the different shades

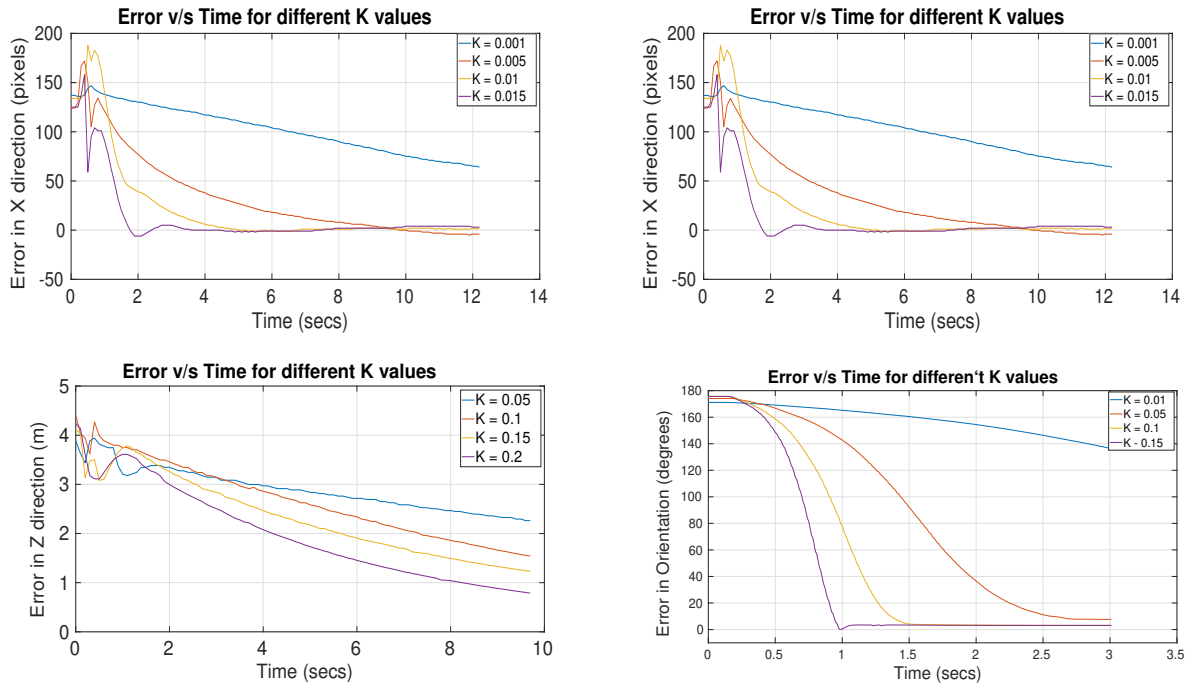


Fig. 5. Independent comparison between different values of proportional gain constants (K_x , K_y , K_z and K_{yaw}) to be applied in the Eqs. 7, 8, 9 and 6 respectively for linear velocity input in x, y, z direction and angular velocity input about the z direction i.e yaw for quadcopter control. (a) shows the error in x direction in pixels v/s time in seconds for different K_x values. (b) shows the the error in y direction in pixels v/s time in seconds for different K_y values. (c) shows the the error in z direction in meters v/s time in seconds for different K_z values. (d) shows the the error in orientation in degrees v/s time in seconds for different K_{yaw} values

of red and cyan color. Fig. 6(c) is taken during low daylight. The ground marker is a velvet chart paper which is bright in color and easily detected in low light conditions. Fig. 6(d) is an indoor instance in which 2 red items are placed in the vicinity of the landing marker to show the vision system is capable of detecting only the areas of interest and no other unwanted items of similar color are detected. This proves the robustness of the target detection system.

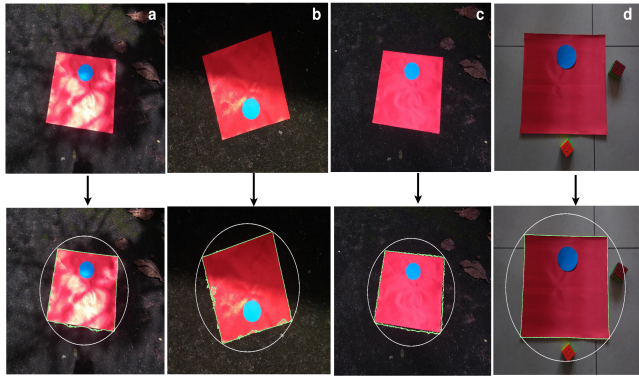


Fig. 6. To evaluate the performance of the detection algorithm in challenging conditions. (a) Under the shadow of a tree (b) Under the shadow of a building. (c) In low daylight (d) Indoor with 2 red items in vicinity to test accurate detection of areas of interest.

As the landing system completely rely on the ground marker, any changes to the marker or the environment might

result in algorithm to misbehave. Therefore the detection algorithm needs to be robust to avoid unwanted scenarios. The proposed algorithm accounts for a partially visible ground marker, say due to dust or any other phenomenon. From the simulation results it is inferred that, even if the marker is only 50 percent visible, the algorithm estimated the altitude accurately, given that any two diagonally opposite vertices of the marker are visible. Fig. 7(a) shows an instance when the marker is completely visible and Fig. 7 (b) shows an instance when the ground marker is partially visible. As it can be seen, both the images indicate the altitude of the quadcopter as 5.93 meters with a 0.01 meter error in estimation when it was partially visible.

C. Altitude Estimation

To test the accuracy of the altitude estimation, a simple maneuver is performed over the ground marker. Altitude data from the algorithm is plotted against time along with the real-time altitude data from the sensors on the quadcopter, as shown in Fig. 8(a). The difference in altitude measurements obtained from the sensor data and from the proposed algorithm is shown in Fig. 8(b). The maximum error is slightly more than 0.2 meters with a mean error of about 0.07 meters. The results suggests that the altitude estimation is reliable and has good accuracy.

VI. CONCLUSION

This paper presents a vision-based ground marker detection and control algorithm with altitude estimation for autonomous

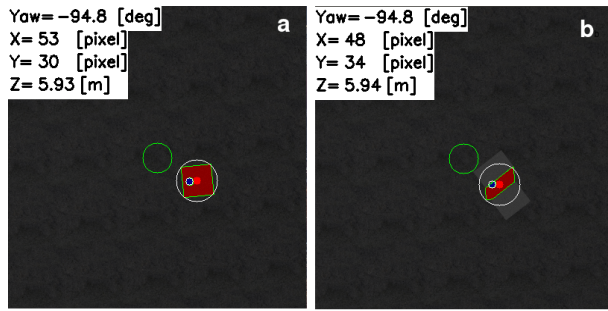


Fig. 7. Detection of the ground marker at two instances from the same altitude. (a) with a completely visible marker and (b) with a partially visible marker. Both indicating the same altitude with an error of 0.01 meters. The white box in the top left corner of each image shows the error in yaw (degrees), error in x and y (image pixels) and the altitude (meters).

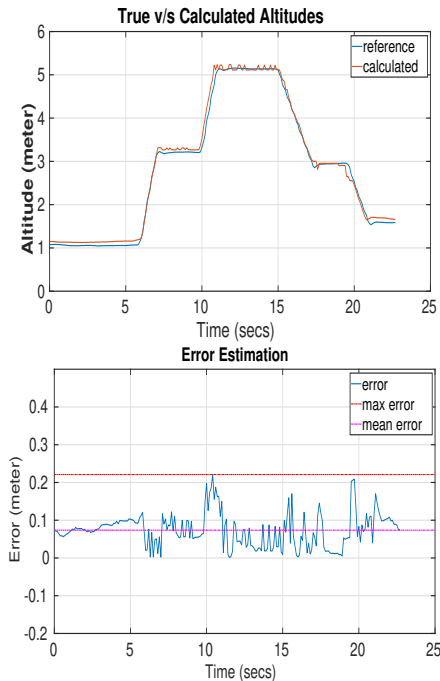


Fig. 8. (a) Comparison of altitude from the real-time sensor data and altitude estimated from our vision system in meters v/s Time in seconds. (b) Error in the estimated altitude from the vision system, compared to the actual altitude data from the sensors. Maximum error recorded is about 0.2 meters and mean error about 0.07 meters.

landing of UAVs on a custom ground marker. The vision system is equipped with a USB camera attached to the bottom of the quadcopter. The altitude is estimated accurately and robustly when the ground marker is partially visible. Considering the computational cost of the system and its performance on relatively cheap embedded system in real-time application, image based visual servoing technique is implemented which sends data directly from vision to the control system. Proportional controller is implemented for the control of the quadcopter. The proposed control strategy involves two steps, orient the quadcopter with respect to the ground marker, and translate as well as descend to land accurately.

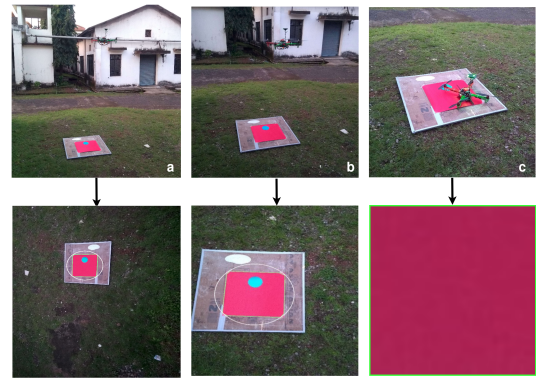


Fig. 9. Pictures taken during outdoor experiments at 3 stages and their respective outputs of the detection algorithm at that instance.

REFERENCES

- [1] L. Geng, Y. F. Zhang, J. J. Wang, J. Y. H. Fuh, and S. H. Teo. Mission planning of autonomous uavs for urban surveillance with evolutionary algorithms. In *2013 10th IEEE International Conference on Control and Automation (ICCA)*, pages 828–833, 2013.
- [2] Alvika Gautam, P.B. Sujit, and Srikanth Saripalli. A survey of autonomous landing techniques for uavs. In *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1210–1218, 2014.
- [3] R Om Prakash and Chandran Saravanan. Autonomous robust helipad detection algorithm using computer vision. In *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, pages 2599–2604, 2016.
- [4] CUI XU, MING LIU, BIN KONG, and YUNJIAN GE. Stereo vision-based estimation of pose and motion for autonomous landing of an unmanned helicopter. *International Journal of Information Acquisition*, 03(03):181–190, 2006.
- [5] Justin Thomas, Jake Welde, Giuseppe Loianno, Kostas Daniilidis, and Vijay Kumar. Autonomous flight for detection, localization, and tracking of moving targets with a small quadrotor. *IEEE Robotics and Automation Letters*, 2(3):1762–1769, 2017.
- [6] Yongwei Zhang, Yangguang Yu, Shengde Jia, and Xiangke Wang. Autonomous landing on ground target of uav by using image-based visual servo control. In *2017 36th Chinese Control Conference (CCC)*, pages 11204–11209, 2017.
- [7] Miguel Saavedra-Ruiz, Ana Maria Pinto-Vargas, and Victor Romero-Cano. Monocular visual autonomous landing system for quadcopter drones using software in the loop. *IEEE Aerospace and Electronic Systems Magazine*, pages 1–1, 2021.
- [8] Montika Sereewattana, Miti Ruchanurucks, Panjavee Rakprayoon, Supakorn Siddhichai, and Shoichi Hasegawa. Automatic landing for fixed-wing uav using stereo vision with a single camera and an orientation sensor: A concept. In *2015 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 29–34, 2015.
- [9] Gervin Ernest C. Guevarra, Ilen Rafael Tatsuya S. Koizumi, John Nicholas B. Moreno, Jeremy Christian B. Reccion, Carl Michael O. Sy, and Jay Robert B. del Rosario. Development of a quadrotor with vision-based target detection for autonomous landing. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, 10(1-6):41–45, Feb. 2018.
- [10] Chang Liu, Stephen D. Prior, and James P. Scanlan. Design and implementation of a low cost mini quadrotor for vision based maneuvers in gps denied environments. *Unmanned Systems*, 04(03):185–196, 2016.
- [11] Sven Lange, Niko Sunderhauf, and Peter Protzel. A vision based onboard approach for landing and position control of an autonomous multirotor uav in gps-denied environments. In *2009 International Conference on Advanced Robotics*, pages 1–6, 2009.
- [12] Rong Liu, Jianjun Yi, Yajun Zhang, Bo Zhou, Wenlong Zheng, Hailei Wu, Shuqing Cao, and Jinzhen Mu. Vision-guided autonomous landing of multirotor uav on fixed landing marker. In *2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, pages 455–458, 2020.