Proceedings of 2016 IEEE
International Conference on Mechatronics and Automation
August 7 - 10, Harbin, China

# Autonomous Landing of a Quadrotor on an UGV*

Mengyin Fu
*Nanjing University of Science and Technology*
*Nanjing, China*
*fumy@bit.edu.cn*

Kuan Zhang, Yang Yi, *Member, IEEE*, and Chao Shi
*Key Laboratory of Complex System Intelligent Control and Decision*
*Beijing Institute of Technology*
*Beijing, China*, *zk1991@bit.edu.cn*

***Abstract*—This paper provides a complete solution for landing a quadrotor Unmanned Aerial Vehicle (UAV) on an Unmanned Ground Vehicle (UGV) in the outdoor environment. An image-based visual servo control law and a GPS-based navigation method are proposed, and the former method acquires the visual feedback from the camera attached to the quadrotor, while the latter strategy is used when the visual feedback is absent. For achieving more precise and stable performance in visually guided landing process, we design a specific landing marker that can provide different features as the distance between the quadrotor and landing target changes. Meanwhile, the corresponding detection and recognition algorithms are raised. Finally, a simulation experiment is carried out to validate the proposed vision algorithms and control strategies.**

***Index Terms*—unmanned aerial vehicle, land, visual servo, unmanned ground vehicle**

## I. Introduction

In recent years, unmanned ground vehicle and vertical take-off and landing (VTOL) unmanned aerial vehicle have attracted lots of research. The combination of both has motivated many novel and challenging applications, as well as research. For example, [1] presents a hybrid system composed of an UAV and an UGV that conduct indoor surveillance cooperatively, while [2] makes multiple unmanned air and ground vehicles work together to track a moving target in urban environment. Imagine the UGV shown in Fig. 1 running on the urban road or in the field environment, the on-board sensors can only obtain the nearby environment information, but a quadrotor UAV equipped with vision sensors can provide more traffic or terrain information [3] to the UGV through inter-vehicle communication. However, landing the quadrotor on the UGV is the foundation of all the subsequent work.

The landing problem of an UAV has been discussed in a lot of papers and a number of methods have been raised. [4], [5] take optical flow measurements as visual feedback, and derive an optical-flow-based control law, while [6] presents an infrared stereo vision system for the purpose of landing an UAV in the Global Navigation Satellite System (GNSS) denied environments. Many papers, such as [7] [8] [9] etc., carry out their work in indoor environments, however the outdoor experiments haven't been conducted.

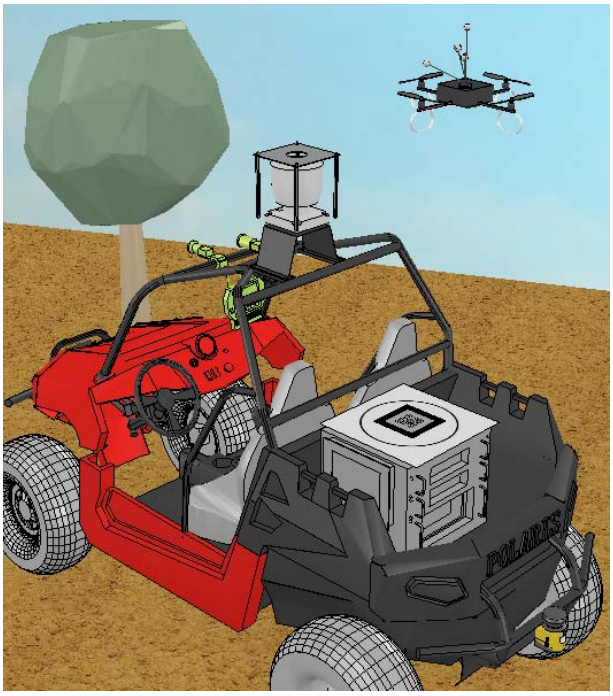

Fig. 1. System components and simulation environment

For this paper, an outdoor landing situation is considered and a complete solution is offered. First we propose the GPS-based navigation algorithm for the purpose of guiding the quadrotor to the UAV. And as the landing target appears in the camera's field of vision, an image-based visual control law is supposed to take effect in landing guidance. However, these proposed control strategies generate velocity command only, the UAV flight control algorithms, such as nonlinear controller with feedback linearization [10] and sliding mode controller [11], are not discussed here. Since the distance between the quadrotor and the landing target becomes smaller and smaller with the landing process, single visual feature with large or small scale is unable to provide a complete visual guide. Considering this case, a specific landing marker is designed and corresponding detection and recognition algorithms have been raised. At last, a simulation environment

*The work was partly supported by National Natural Science Foundation of China (Grant No. NSFC 61473042 and 61105092) and Beijing Higher Education Young Elite Teacher Project (No.YETP1215).

978-1-5090-2396-7/16/$31.00 ©2016 IEEE

using ROS (Robot Operating System) and v-rep (visual robot experimentation platform) is built so that the control methods and vision algorithms can be tested simultaneously.

## II. MODELS

In this section, the pinhole camera model and quadrotor dynamic model are presented.

### A. Camera Model

The pinhole camera model is the simplest and most common model of a camera. This model is illustrated in a mathematically equivalent way [12] in Fig. 2, in which an interest point Q with coordinates $(X, Y, Z)$ is mapped to a point $q = (x', y', f)$ on the image plane. Both $(X, Y, Z)$ and $(x', y', f)$ are expressed in the camera frame $\mathcal{C}$, then projective geometry yields

$$\begin{cases} x' = f \cdot X/Z \\ y' = f \cdot Y/Z \end{cases} \tag{1}$$

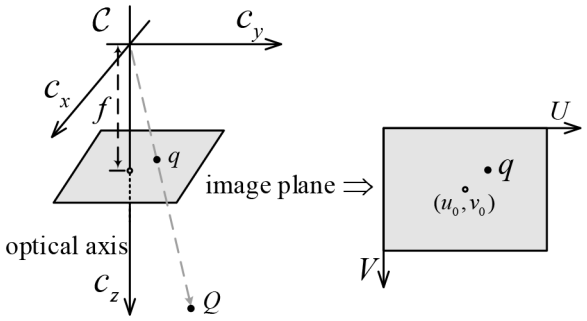


Fig. 2. Pinhole camera model represented in a mathmatically equivalent way and the pixel coordinate system of the image plane

Since $(x', y', f)$ are expressed in physical units, for example millimeters, the value of $x'$ and $y'$ are quite small. In addition, considering the image plane points have the same depth $f$, a new approach is proposed to express the point $q$ using two dimensional coordinates:

$$\begin{cases} x = X/Z = (u - u_0)/p_x \\ y = Y/Z = (v - v_0)/p_y \end{cases} \tag{2}$$

where $(u, v)$ is the pixel coordinates of point $q$ and $P = \{u_0, v_0, p_x, p_y\}$ is the set of camera intrinsic parameters; $(u_0, v_0)$ represent the pixel coordinates of the principal point (i.e., the intersection between the image plane and the optical axis), while $p_x$ and $p_y$ are the ratio between the focal length and the size of a pixel.

### B. Quadrotor Dynamic Model

The quadrotor is actually a nonlinear, highly coupled and underactuated system. It is common to ignore the interactions between rotors and some external uncertainties for establishing a more ideal dynamic model. First, abstract the quadrotor UAV as a rigid body of mass $m$, and use $\mathbf{I}$ to denote the inertia tensor. Secondly, the body-fixed frame $\mathcal{B}$ with the origin at the center of mass and the earth-fixed frame $\mathcal{E}$ are introduced. As shown in Fig. 3, $[e_x\ e_y\ e_z]$ form an orthogonal basis of $\mathcal{E}$, and $[b_x\ b_y\ b_z]$ form an orthogonal basis of $\mathcal{B}$, both of which obey the right-hand rule. And then the linear position and angular position of the quadrotor in reference frame $\mathcal{E}$ are defined as $p$ and $\eta$, the linear velocity and angular velocity in frame $\mathcal{B}$ are denoted by $v_b$ and $\omega_b$

$$p = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}, \eta = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}, v_b = \begin{bmatrix} v_{bx} \\ v_{by} \\ v_{bz} \end{bmatrix}, \omega_b = \begin{bmatrix} \omega_{bx} \\ \omega_{by} \\ \omega_{bz} \end{bmatrix} \tag{3}$$

where $\phi$, $\theta$ and $\psi$ denote the roll, pitch and yaw angle of the quadrotor respectively. Finally, with the attitude angle, the orthogonal rotation matrix $R$ from $\mathcal{B}$ to $\mathcal{E}$ is acquired. Moreover, $R^T$ is the rotation matrix from $\mathcal{E}$ to $\mathcal{B}$.

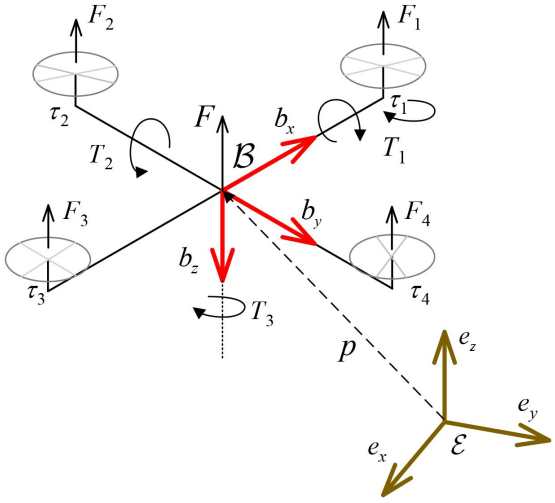


Fig. 3. The dynamic model of a quadrotor

The rotors generate four thrusts $F_i$ $(i = 1, 2, 3, 4)$ in the direction $-b_z$ and four torques $\tau_i$ in the direction that is opposite to the propeller direction of rotation. These thrusts and torques can be replaced by a total thrust force $F$ and three different torques $T_1$, $T_2$ and $T_3$

$$\begin{cases} F = F_1 + F_2 + F_3 + F_4 \\ T_1 = l \cdot (F_2 - F_4) \\ T_2 = l \cdot (F_1 - F_3) \\ T_3 = \tau_1 - \tau_2 + \tau_3 - \tau_4 \end{cases} \tag{4}$$

where $l$ stands for quadrotor's arm length. Define the control input as $F_b = (0, 0, -F)^{\mathrm{T}}$ and $T_b = (T_1, T_2, T_3)^{\mathrm{T}}$ in the body-fixed frame, then Newton's second law of motion yields

$$\ddot{p} = (G + R \cdot F_b + \delta)/m \tag{5}$$

where $G = (0, 0, -mg)^{\mathrm{T}}$ denotes gravity and $\delta$ stands for slowly time-varying unmodeled forces. In the landing procedure, the camera frame $\mathcal{C}$ and the body-fixed frame $\mathcal{B}$ are assumed to be coincide with each other, and the interest points are represented by (2). Hence it is necessary to have a clear view of the dynamic model in the frame $\mathcal{B}$. With all these parameters, Newton's second law of motion and Eular's rotation equation yield:

$$\dot{v}_b = (R^{\mathrm{T}} \cdot G + F_b + R^{\mathrm{T}}\delta)/m \tag{6}$$

$$\mathbf{I} \cdot \dot{\omega}_b = T_b - \omega_b \times (\mathbf{I} \cdot \omega_b) \tag{7}$$

## III. VISION ALGORITHM

This section describes the vision algorithms used in the target detection and recognition. When the quadrotor is hovering above the unmanned ground vehicle with a relatively high altitude, the camera is blind to the detailed features due to the limit of resolution. On the contrary, when the distance between the UAV and the UGV becomes small, a large size marker can't be fully rendered in the camera's field of vision. Therefore a landing marker shown in Fig. 4 is designed to solve this trade-off problem. The landing target is made up of two parts: one is a white square board with a circle inside used as rough features, the other is a QR code pasted on the board providing visual guidance information when the camera is getting close to the target.

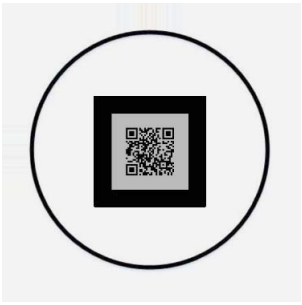

Fig. 4. Landing marker

### A. Landing Target Detection

For landing purpose, the tarmac should be detected first. Since the landing board has a white background which is an obvious characteristic compared to other objects in this scene, a transformation from the original color image to a grayscale image has been done. And then the gray image is binarized using the threshold value division method. After that, the binary image is filtered using morphological operations to remove the noise and then extract contours in the filtered image. Next, for each extracted contour, the matching operation with a square template outline is performed using *Hu moments* which are invariant to scale, rotation and reflection. For detecting the circle in the square contour, the Hough transform method is applied to the region (in the gray image) that exactly contains the found square contour. This practice results in high computational efficiency. Finally, five interest points are extracted from the landing target, which are the center point $P_c = (u_c, v_c)$ (pixel coordinates) of the circle and four corner points $P_s^i = (u_s^i, v_s^i)$ $(i = 1, 2, 3, 4)$ of the square board.

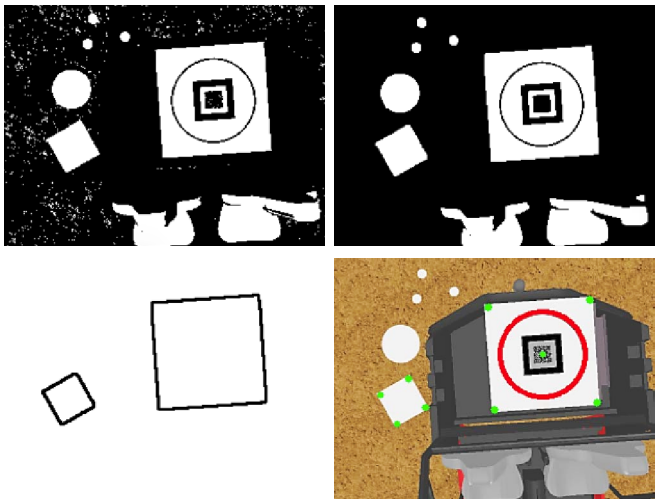

Fig. 5. Landing target detection procedure. (a) Binary image; (b) Filtered binary image using morphological method, such as erosion and dilation; (c) Extracted square contours using the feature of Hu moments; (d) Detected corner points (green) of the squares and the center (green) of the circle (red). The small square without a circle inside doesn't satisfy (8) and is neglected.

As the prior knowledge, $P_c$ is located at the center of the four corner points $P_s^i$, thus (8) is obtained

$$|u_c - \sum_{i=1}^{4} u_s^i/4| + |v_c - \sum_{i=1}^{4} v_s^i/4| < \Delta \tag{8}$$

where $\Delta$ is the error tolerance. In addition, the ratio between the circle's radius and the square's side length should be close to a particular value. At last, these geometric relationships are checked for ensuring correct landing target detection.

### B. QR Code Recognition

As to QR code recognition, an open source ROS package has been used. The recognition algorithm uses moving-edges and keypoint features shown in Fig. 6 and the experiment shows that it satisfies the real-time requirements in the landing process.

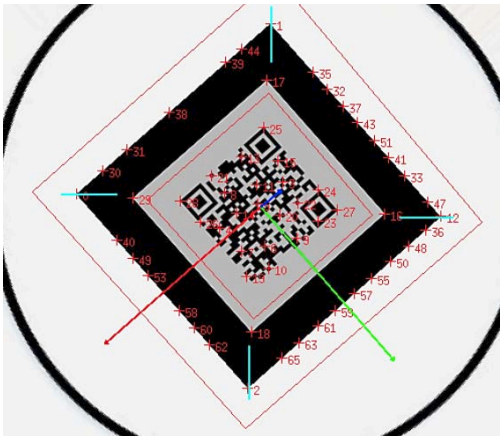

Fig. 6. QR code recognition

## IV. CONTROL

In this section, the landing process is split into two stages. One is the navigation stage using GPS and other sensors, the other is the visual servoing stage. The purpose of the former stage is to guide the quadrotor to the UGV, so that the landing target will be in the camera's field of view. Moreover, the latter stage is divided into two phases using different features for the purpose of acquiring stable and precise performance. In the first phase, the detected landing target gives visual feedback to the controller for achieving a relatively low altitude so that the details of QR code can be seen clearly. And then the QR code leads the quadrotor to complete the landing task in the second phase.

### A. Navigation Control

A high-quality GPS receiver provides better than 3.5 meters horizontal accuracy. Through the GPS/IMU data fusion using Kalman filter, more accurate position information can be acquired. Compared to the elevation data provided by GPS, the altitude data obtained from a barometer performs better in accuracy. Therefore we use the elevation data measured by the barometer rather than GPS. Define the position coordinates of the quadrotor UAV and the UGV as $p_a = [x_a\ y_a\ z_a]^{\mathrm{T}}$ and $p_g = [x_g\ y_g\ z_g]^{\mathrm{T}}$ in the frame $\mathcal{E}$ respectively, and with a desired altitude offset $p_o = [0\ 0\ h]^{\mathrm{T}}$ $(h > 0)$, the error of the navigation control system is

$$e_p(t) = p_g(t) + p_o - p_a(t) \tag{9}$$

In order to make the error converge to zero, a PID controller is designed

$$u_e(t) = K_p e_p(t) + K_i \int_0^t e_p(t)dt + K_d \frac{d}{dt} e_p(t) \tag{10}$$

where $u_e(t) = [v_{ex}^*\ v_{ey}^*\ v_{ez}^*]^{\mathrm{T}}$ is the reference velocity. For convenience, transform $u_e(t)$ into $u_b(t) = [v_{bx}^*\ v_{by}^*\ v_{bz}^*]^{\mathrm{T}}$ which stands for the desired velocity in frame $\mathcal{B}$ using

$$u_b(t) = R^{\mathrm{T}} u_e(t) \tag{11}$$

### B. Visual Servo Control

As mentioned before, the detected feature points are represented by (2). With a set of interest points $s(t) = [x_1\ y_1\ x_2\ y_2\ \cdots\ x_k\ y_k]^{\mathrm{T}}$ and a vector $s^*(t)$ containing the expected values of the features, the basic idea of visual servo control is to minimize the error defined below

$$e(t) = s(t) - s^*(t) \tag{12}$$

where $s^*(t)$ is a step function for the reason that the desired feature values vary with the landing phase, but it is constant in a certain phase. Thus the relationship between $\dot{e}$ and $\dot{s}$ is obtained

$$\dot{e} = \dot{s} \tag{13}$$

The spatial velocity of the camera in $\mathcal{B}$ is denoted by $\mathbf{v}_c = [v_b^{\mathrm{T}}\ \omega_b^{\mathrm{T}}]^{\mathrm{T}}$, and then the relationship between $\dot{e}$ and $\mathbf{v}_c$ is given by [13]

$$\dot{e} = \mathcal{L}_s \mathbf{v}_c \tag{14}$$

where

$$\mathcal{L}_s = \begin{bmatrix} \mathcal{L}_{q_1} \\ \mathcal{L}_{q_2} \\ \vdots \\ \mathcal{L}_{q_k} \end{bmatrix} \tag{15}$$

is called the interaction matrix with

$$\mathcal{L}_{q_i} = \begin{bmatrix} -1/Z & 0 & x_i/Z & x_i y_i & -(1+x_i^2) & y_i \\ 0 & -1/Z & y_i/Z & 1+y_i^2 & -x_i y_i & -x_i \end{bmatrix} \tag{16}$$

For achieving an exponential convergence rate of the error $e$, which means

$$\dot{e} = -\lambda e = \mathcal{L}_s \mathbf{v}_c \tag{17}$$

this problem is transformed into solving a system of equations with six unknown parameters. Least square estimation gives the best solution shown in (18) that makes the residual norm $\| - \lambda e - \mathcal{L}_s \mathbf{v}_c \|$ reach the minimum.

$$\mathbf{v}_c^* = -\lambda \mathcal{L}_s^{\dagger} e = -\lambda (\mathcal{L}_s^{\mathrm{T}} \mathcal{L}_s)^{-1} \mathcal{L}_s^{\mathrm{T}} e \tag{18}$$

The equation (18) with $\mathbf{v}_c^* = [v_{bx}^*\ v_{by}^*\ v_{bz}^*\ \omega_{bx}^*\ \omega_{by}^*\ \omega_{bz}^*]^{\mathrm{T}}$ is referred to as visual servo control law, and for ensuring $\mathcal{L}_s$ is full column rank, at least four non-colinear interest points should be chosen [14].

Besides, to apply this control law (18) to a quadrotor UAV, there are still two problems to be addressed, one is the depth $Z$ estimation, and the other is the coupling issue between $(v_{bx}, v_{by})$ and $(\omega_{bx}, \omega_{by})$ caused by the quadrotor's underactuated property.

For problem one, define the distance between two features points as $D$ expressed in frame $\mathcal{E}$, and define the Euclidean

distance of the corresponding image points (expressed in the form shown in (2)) as $d$, then the depth Z can be estimated using the following equation

$$Z = \frac{D}{d} \tag{19}$$

To solve the coupling issue, denote the desired position and attitude as $(p_x^*, p_y^*)$ and $(\phi^*, \theta^*)$ when the error $e$ is approximately zero. As is known to all, the linear velocity $(v_{bx}, v_{by})$ controls the position $(p_x, p_y)$, and the angular velocity $(\omega_{bx}, \omega_{by})$ controls the attitude angle $(\phi, \theta)$. As the quadrotor hovers over the landing target (i.e., $(p_x, p_y) \rightarrow (p_x^*, p_y^*)$), the pitch and yaw angle will be close to zero (i.e., $(\phi, \theta) \rightarrow (\phi^*, \theta^*)$), however the converse is clearly not true. Hence, the control input $(v_{bx}, v_{by})$ plays a more important role than $(\omega_{bx}, \omega_{by})$ in the landing stage. Consequently, the references for linear velocity and yaw rate are tracked by the quadrotor, while $(\omega_{bx}^*, \omega_{by}^*)$ are not.

## V. SIMULATION

The simulation environment is established using ROS and v-rep, both of which are powerful tools in robot development and simulation.

### A. Simulation Environment Description

The UAV model and UGV model used in the simulation scene are shown in Fig. 1. The unmanned ground vehicle equipped with GPS and many other sensors is stationary and is regarded as a landing platform. The quadrotor UAV model is abstracted as a rigid body with the control interface of thrust $F_b$ and torque $T_b$ just as mentioned in section II. However, in reality, there are a number of quadrotor UAV platforms that provide excellent SDK (Software Development Kit) so that users can develop their applications without caring about the flight control. Considering that the control algorithms described in section IV generate linear and angular velocity command only, a PID controller is designed to bridge the reference velocity and control input ($T_b$ and $F_b$) for the quadrotor model.

The size of the landing pad and the QR code are configured as $0.6m \times 0.6m$ and $0.2m \times 0.2m$ respectively. The camera has a resolution of $640 \times 480$ pixels and the angel of view is about 70 degrees. The details of QR code can be clearly seen by the camera in a distance range of 0.15 to 0.7 meters between the quadrotor and the tarmac.

ROS provides an exceptional communication mechanism to transport data and messages between software modules. The communication architecture of this simulation environment is illustrated in Fig. 7.

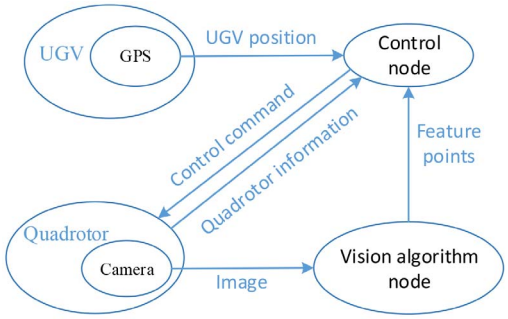


Fig. 7. ROS communication architecture. The quadrotor information includes position and orientation, linear and angular velocity, as well as other physical parameters, such as mass, inertia tensor.

### B. Simulation Result

The navigation control simulation shows excellent performance due to high accuracy and precision of GPS in the virtual environments. Therefore we focus on the result analysis of the visually guided landing process simulation.

As shown in Fig. 8, the quadrotor and landing target is initialized with position coordinates $(-1.03, -1, 3)$ and $(-0.4, -0.3, 1.4)$ in $\mathcal{E}$, respectively. After the start of simulation, the quadrotor gradually approaches the target. The trajectory of the quadtotor movement is smooth and precise. Finally, the quadrotor arrives at the position $(-0.4, -0.3, 1.65)$ and hovers over the target. When the error $e$ is lower than a predefined threshold, the rotors are switched off to finish the landing mission .

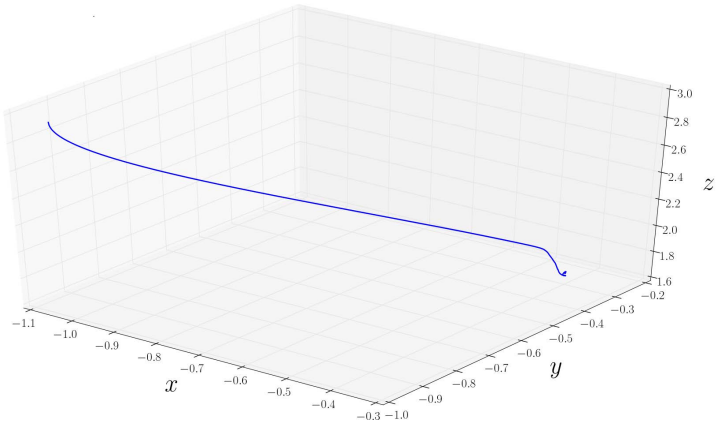


Fig. 8. The trajectory of quadrotor movement in the visual servo control simulation

Just as described before, the simulated landing procedure are divided into two phases (separated by the red solid line shown in Fig. 9): phase one (0-5s) detects the landing target to get the feature points, while phase two acquires visual feedback from the QR code. There are three points to be

illustrated. First, the feature points obtained from landing target detection are centrosymmetric, thus there exist more than one conditons that makes $e = 0$ [15], consequently the reference for yaw rate is ignored in the first phase. That's why the yaw angle remains zero during the time interval 0 to 5 seconds. Secondly, the landing phase shift occurs on the condition that $\|e\|$ is lower than a proper threshold and the QR code is recognised. In this experiment, the phase shift occurs at the position $(-0.47, -0.36, 1.9)$ successfully. Thirdly, roll and pitch angle approach zero when $e \to 0$, and the desired position has been reached, thus the analysis in section IV is verified.

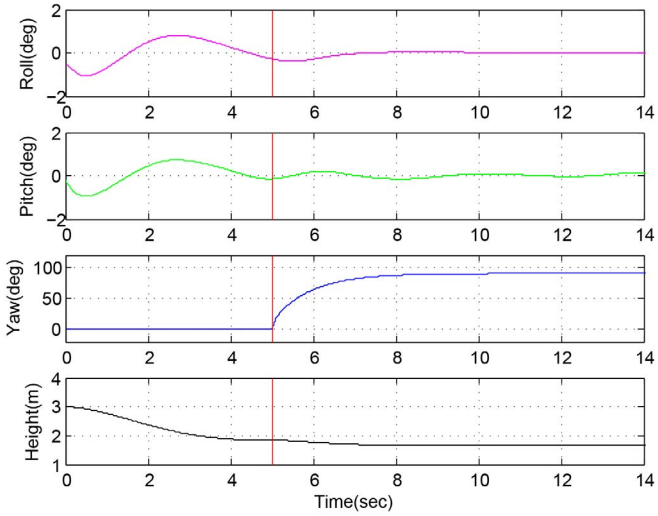


Fig. 9. The attitude and height of the quadrotor in the landing process

Through the experiment, the vision algorithms and visual servo methods for the landing mission were tested and proven to be effective.

## VI. CONCLUSION AND FUTURE WORK

The main contributions of this paper are summarized as follows: (1) Proposed a complete solution for landing a quadrotor UAV on an UGV, which provides the foundation for the combination of unmanned aerial and ground vehicles, as well as the following work; (2) Designed a compound landing marker with both large and small scale features that can offer visual feedback during the entire landing process, moreover, the corresponding detection and recognition algorithm are developed and proved to be effective and reliable; (3) Established a simulation environment using ROS and v-rep and validated the control strategies and vision algorithms simultaneously. The result showed good performance and indicated that the issue is worthy of further research and extension.

We plan to finish two tasks in the near future: one is the implementation of the vision algorithms and landing control strategies on the realistic platform, the other is designing a mechanical device used for fixing the quadrotor on the ground vehicle.

## ACKNOWLEDGMENT

The authors would like to show the deepest gratitude to Hao Wang, Zhen Wang, Meiling Wang and Jianqing Zhang for their valuable guidance and enlightening instruction. We'd like to extend our appreciation to all other members of BIT ININ Lab for their help and kindness.

## REFERENCES

[1] M. Saska, T. Krajník, and L. Pfeucil, "Cooperative $\mu$UAV-UGV autonomous indoor surveillance," in *Systems, Signals and Devices (SSD), 2012 9th International Multi-Conference on*. IEEE, 2012, pp. 1–6.

[2] H. Yu, K. Meier, M. Argyle, and R. W. Beard, "Cooperative path planning for target tracking in urban environments using unmanned air and ground vehicles," *Mechatronics, IEEE/ASME Transactions on*, vol. 20, no. 2, pp. 541–552, 2015.

[3] J. Artieda, J. M. Sebastian, P. Campoy, J. F. Correa, I. F. Mondragón, C. Martínez, and M. Olivares, "Visual 3-D SLAM from UAVs," *Journal of Intelligent and Robotic Systems*, vol. 55, no. 4-5, pp. 299–321, 2009.

[4] B. Herisse, T. Hamel, R. Mahony, and F.-X. Russotto, "The landing problem of a VTOL unmanned aerial vehicle on a moving platform using optical flow," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 1600–1605.

[5] B. Herissé, T. Hamel, R. Mahony, and F.-X. Russotto, "Landing a VTOL unmanned aerial vehicle on a moving platform using optical flow," *Robotics, IEEE Transactions on*, vol. 28, no. 1, pp. 77–89, 2012.

[6] W. Kong, D. Zhang, X. Wang, Z. Xian, and J. Zhang, "Autonomous landing of an UAV with a ground-based actuated infrared stereo vision system," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 2963–2970.

[7] S. Yang, S. A. Scherer, and A. Zell, "An onboard monocular vision system for autonomous takeoff, hovering and landing of a micro aerial vehicle," *Journal of Intelligent & Robotic Systems*, vol. 69, no. 1-4, pp. 499–515, 2013.

[8] S. Lange, N. Sünderhauf, and P. Protzel, "A vision based onboard approach for landing and position control of an autonomous multirotor UAV in GPS-denied environments," in *Advanced Robotics, 2009. ICAR 2009. International Conference on*. IEEE, 2009, pp. 1–6.

[9] D. Lee, T. Ryan, and H. J. Kim, "Autonomous landing of a VTOL UAV on a moving platform using image-based visual servoing," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 971–976.

[10] I. Voos, "Nonlinear control of a quadrotor micro-UAV using feedback-linearization," in *Mechatronics, 2009. ICM 2009. IEEE International Conference on*. IEEE, 2009, pp. 1–6.

[11] D. Lee, H. J. Kim, and S. Sastry, "Feedback linearization vs. adaptive sliding mode control for a quadrotor helicopter," *International Journal of control, Automation and systems*, vol. 7, no. 3, pp. 419–428, 2009.

[12] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc.", 2008, ch. 11, pp. 371–373.

[13] F. Chaumette and S. Hutchinson, "Visual servo control, Part I: Basic approaches," *Robotics & Automation Magazine, IEEE*, vol. 13, no. 4, pp. 82–90, 2006.

[14] H. Michel and P. Rives, "Singularities in the determination of the situation of a robot effector from the perspective view of 3 points," 1993.

[15] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.