2012 IEEE International Conference on Robotics and Automation
RiverCentre, Saint Paul, Minnesota, USA
May 14-18, 2012

Autonomous Landing of a VTOL UAV on a Moving Platform Using Image-based Visual Servoing

Daewon Lee[1], Tyler Ryan[2] and H. Jin. Kim[3]

Abstract— **In this paper we describe a vision-based algorithm to control a vertical-takeoff-and-landing unmanned aerial vehicle while tracking and landing on a moving platform. Specifically, we use image-based visual servoing (IBVS) to track the platform in two-dimensional image space and generate a velocity reference command used as the input to an adaptive sliding mode controller. Compared with other vision-based control algorithms that reconstruct a full three-dimensional representation of the target, which requires precise depth estimation, IBVS is computationally cheaper since it is less sensitive to the depth estimation allowing for a faster method to obtain this estimate. To enhance velocity tracking of the sliding mode controller, an adaptive rule is described to account for the ground effect experienced during the maneuver. Finally, the IBVS algorithm integrated with the adaptive sliding mode controller for tracking and landing is validated in an experimental setup using a quadrotor.**

I. INTRODUCTION

Vertical take-off and landing (VTOL) unmanned aerial vehicles (UAVs) have been used in many applications such as search and rescue, reconnaissance, surveillance and exploration. Generally, these aircraft have highly coupled nonlinear dynamics and fly in unsteady conditions (e.g. due to turbulence or unmodelled dynamics), which has led to the development of various robust and/or nonlinear control techniques. For example, [1] presents an internal-model based error-feedback dynamic regulator that is robust to parametric uncertainty. Sliding mode disturbance observers ([2]) and adaptive-fuzzy control techniques ([3]) have also been used to make the system more robust to external disturbances. A combined controller and observer using neural networks is used in [4] to track reference trajectories.

One feature becoming more common on miniature VTOL UAVs is an onboard camera, which has resulted in a quickly growing library of research making use of this camera. Common isolated vision tasks include target tracking ([5] and [6]) and simultaneous localization and mapping (SLAM, [7]). Additionally, the onboard camera has been used within the control loop such as [8] and [9] which use the camera for position control.

For this paper, we use the adaptive sliding mode controller described in [10] for precision control of a quadrotor while landing on a moving target (fig. 1). This situation would occur, for example, when landing on a moving ship or truck. While GPS-based control is sufficient for general positioning (assuming the signal is not blocked), the accuracy of GPS receivers fit for use on miniature UAVs is measured in meters, making them unsuitable for precision tasks such as landing. Instead, we demonstrate that image-based visual servoing (IBVS) is capable of providing the necessary precision. IVBS is a technique that performs the majority of the control calculations in 2D image space, rather than the typical 3D body or inertial reference frames. This eliminates the need for expensive 3D position reconstruction calculations allowing for real-time control. Finally, given the need of flying inside ground effect during the landing maneuver, the controller uses a model adaptation technique to dynamically compensate for this system disturbance.

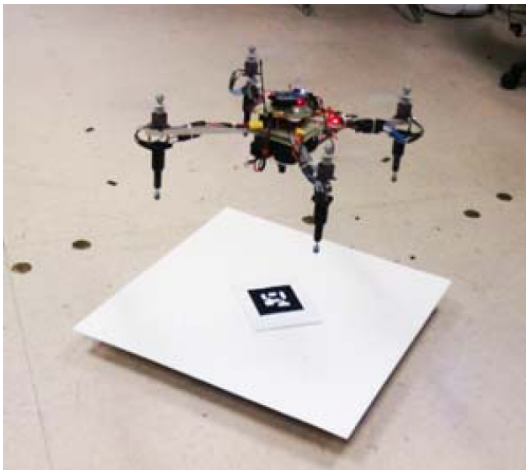

Fig. 1. Quadrotor helicopter and moving ground target under consideration

Most similar to our work is [11] which also calculates control in the image frame thereby avoiding full three dimensional reconstruction. The biggest difference between their work and ours is we adaptively compensate for the ground effect which is quite significant during a landing maneuver. Additionally they use a different image-based method for their formulation.

II. SYSTEM

This section describes the dynamics of the quadrotor and the camera systems.

A. Quadrotor

Define the UAV state vector as $\mathbf{x} = [x, y, z, \phi, \theta, \psi]^T$,

[1] Department of Mechanical and Aerospace Engineering, Ph.D Candidate, Seoul National University, Seoul, Republic of Korea lee.daewon@gmail.com
[2] Department of Mechanical and Aerospace Engineering, Ph.D Student, Seoul National University, Seoul, Republic of Korea ryantr@gmail.com
[3] Department of Mechanical and Aerospace Engineering, Associate Professor Seoul National University, Seoul, Republic of Korea hjinkim@snu.ac.kr

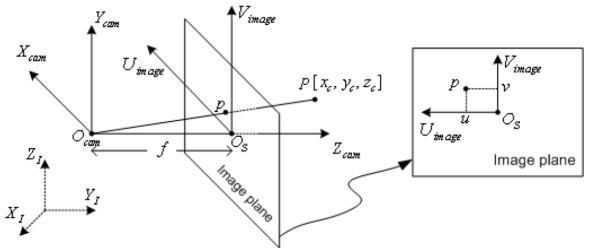

Fig. 2. Geometry and coordinate frames for a pinhole camera model

where x, y and z denote translational position variables and ϕ, θ and ψ denote roll, pitch and yaw, respectively. Define the control input vector as $\mathbf{u} = [u_1, u_2, u_3, u_4]^T$ (inputs for thrust, roll, pitch and yaw respectively). Then system dynamics, including influence of external forces $f_{ex}(\mathbf{x})$, can be represented as eq. (1). (for derivation details, see [10])

$$\ddot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})\mathbf{u} + f_{ex}(\mathbf{x}) \quad (1)$$

where

$$f(\mathbf{x}) = \begin{bmatrix} 0 & 0 & -F_g & 0 & 0 & 0 \end{bmatrix}^T$$

$$g(\mathbf{x}) = \begin{bmatrix} \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & 0 & 0 & 0 \\ \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & 0 & 0 & 0 \\ \cos\phi\cos\theta & 0 & 0 & 0 \\ 0 & l & 0 & 0 \\ 0 & 0 & l & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$f_{ex}(\mathbf{x}) = \begin{bmatrix} 0 & 0 & F_{gr}(z) & 0 & 0 & 0 \end{bmatrix}^T,$$

l denotes the distance from the motor rotational axis to the system center of gravity (assumed equal for all motors), F_g is gravity force and $F_{gr}(z)$ denotes the ground effect force.

B. Vision System

The camera model we consider in this paper is a pinhole camera model as shown in fig. 2 where $[X_I, Y_I, Z_I]^T$ are the axes of the three-dimensional inertial coordinate frame $\mathbb{I}$, $[X_{cam}, Y_{cam}, Z_{cam}]^T$ are the axes of the camera frame $\mathbb{C}$, i.e., the coordinate frame attached to the camera center O_{cam}, $[U_{image}, V_{image}]^T$ are the axes of the image frame $\mathbb{S}$, O_S denotes the point where the z axis of the camera coordinate frame intersects the image plane, and f is the focal length of the camera. The mapping from a point $P = [x_c, y_c, z_c]^T \in \mathbb{C}$ to a point $p = [u, v]^T \in \mathbb{S}$ can be written as eq. (2).

$$p = \begin{bmatrix} u \\ v \end{bmatrix} = \frac{f}{z_c} \begin{bmatrix} x_c \\ y_c \end{bmatrix} \quad (2)$$

Suppose that the camera is moving such that P moves with translational velocity $T = [T_x, T_y, T_z]^T$ and rotational velocity $\Omega = [\omega_x, \omega_y, \omega_z]^T$ both with respect to the camera frame $\mathbb{C}$. The spatial velocity, $\dot{r} \in \Re^6$ can then be defined as (3)

$$\dot{r} = \begin{bmatrix} T \\ \Omega \end{bmatrix} = \begin{bmatrix} T_x, & T_y, & T_z, & \omega_x, & \omega_y, & \omega_z \end{bmatrix}^T. \quad (3)$$

Finally, the dynamics of the image space point p can be expressed as eq. (4) [12].

$$\dot{p} = J_p \dot{r}. \quad (4)$$

where

$$J_p = \begin{bmatrix} -\frac{f}{z_c} & 0 & \frac{u}{z_c} & \frac{uv}{f} & -\frac{f^2+u^2}{f} & v \\ 0 & -\frac{f}{z_c} & \frac{v}{z_c} & \frac{f^2+v^2}{f} & -\frac{uv}{f} & -u \end{bmatrix} \quad (5)$$

is called the image Jacobian matrix associated with the image feature p.

III. Control

This section describes image-based visual servoing, including image roll and pitch compensation, and the adaptive sliding mode velocity controller.

A. Image-based Visual Servoing

In image-based visual servoing, the task space is defined in two-dimensional image space and the control law is designed as a function of the image error, e, defined as

$$e = p - p_d \quad (6)$$

where p_d is the desired location of the image feature. For the landing maneuver considered in this paper, the desired image feature is static, i.e. $\dot{p}_d = 0$. Using this and eq. (4) we calculate the error rate as,

$$\dot{e} = J_p \dot{r} \quad (7)$$

Define a Lyapunov candidate function as

$$\mathcal{L} = \frac{1}{2} e^T e \quad (8)$$

$$\dot{\mathcal{L}} = e^T J_p \dot{r} \quad (9)$$

Define the desired camera velocity as

$$\dot{r}_d = -W J_p^T e \quad (10)$$

where W is a positive definite weighting matrix. As $\dot{r} \to \dot{r}_d$ (to be achieved by the sliding mode controller described in the next section) then

$$\dot{r} \to -W J_p^T e \quad (11)$$

and eq. (9) becomes

$$\dot{\mathcal{L}} = -e^T (J_p) W (J_p)^T e \leq 0, \quad (12)$$

where $J_p W (J_p)^T$ is a full-rank matrix when the image features are not colinear. This proves the asymptotic convergence of e to zero assuming $\dot{r}_d$ is "well" tracked.

Eq. (10) generates six references (for x, y, z, ϕ, θ, ψ). However, the quadrotor cannot track all the references simultaneously because of its under-actuated property (four motor inputs versus six states). Since the x, y, ϕ and θ channels are highly coupled, we decouple the x and y channels from the ϕ and θ channels.

To do this, we define a virtual coordinate frame $\mathbb{V}$ with the position and yaw angles aligned to match the camera coordinate frame, $\mathbb{C}$, and the z axis is aligned with the inertial

frame z axis, $\mathbb{I}$. If references are generated from an image in $\mathbb{V}$, the desired decoupling is achieved.

To map the captured image into $\mathbb{V}$, first consider an imaginary camera frame which is the same as the true camera frame but rotated to zero roll and pitch. The image features in this imaginary frame become

$$\begin{aligned} P_i^r &= R(1,\phi)R(2,\theta)P_i \quad (13)\\ p_i^r &= \frac{f}{z_{ci}^r}\begin{bmatrix} x_{ci}^r \\ y_{ci}^r \end{bmatrix} \\ &= f\begin{bmatrix} \frac{u_i\cos\theta + f\sin\theta}{-u_i\cos\phi\sin\theta + v_i\sin\phi + f\cos\phi\cos\theta} \\ \frac{u_i\sin\phi\sin\theta + v_i\cos\phi - f\sin\phi\cos\theta}{-u_i\cos\phi\sin\theta + v_i\sin\phi + f\cos\phi\cos\theta} \end{bmatrix} \quad (14)\\ &= \begin{bmatrix} u_i^r \\ v_i^r \end{bmatrix} \end{aligned}$$

where

$$\begin{aligned} R(1,\phi) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \\ R(2,\theta) &= \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}. \end{aligned}$$

Here P_i^r is the i-th feature coordinate defined in the roll and pitch compensated camera frame, and p_i^r is the corresponding image feature coordinate of P_i^r. For most normal flight regimes, roll and pitch angles don't approach 90 deg so the denominators in eq. (14) will be non-zero. Interestingly, so far no depth information is needed to perform the roll and pitch compensation.

Assuming that we know the nominal geometry of the target in the camera frame, we can now estimate depth. First, find the position of another feature in image space, p_j^r. Combined with the already measured point p_i^r we can calculate the distance between these points in the image frame, which we call L_s. Let us call L_c the, already known, corresponding distance in the camera frame. Using these, the depth from the camera to the image features can be simply calculated as

$$z_c = \frac{L_c \cdot f}{L_s} \quad (15)$$

At this point we could, in theory, calculate a full three-dimensional reconstruction of the target and perform control without IBVS. However, this simple depth calculation is only a rough estimate, with a focus on computational efficiency over accuracy, and would not produce an accurate enough reconstruction for standard control algorithms. IBVS, however, is less sensitive to this measurement so having only this rough estimate is sufficient.

The last thing we need to address is distortion due to an offset from the origin of the quadrotor body frame to the origin of the camera frame, which is denoted as $[\delta_x, \delta_y, \delta_z]^T$. For this consider another imaginary camera frame which is translated to compensate for the distance caused by rotation of the offset. Define the i-th feature location in this frame P_i^s, for which following relationship holds:

$$P_i^s = P_i^r + R(1,\phi)R(2,\theta)[\delta_x, \delta_y, \delta_z]^T \quad (16)$$

Now, the only difference between the virtual coordinate frame and the imaginary camera frame is the offset position of the camera. Therefore, the i-th feature coordinate defined in the virtual frame, P_i^v, is

$$\begin{aligned} P_i^v &= P_i^s - [\delta_x, \delta_y, \delta_z]^T \quad (17)\\ p_i^v &= \frac{f}{Z_{ci}^v}\begin{bmatrix} x_{ci}^v \\ y_{ci}^v \end{bmatrix} \quad (18) \end{aligned}$$

where p_i^v is the corresponding image feature coordinate of P_i^v.

B. Adaptive Sliding Mode Controller

To track the translational and rotational velocity references of the UAV given in eq. (10), an adaptive sliding mode controller is adopted. Since $g(\mathbf{x})$ in eq. (1) is a 6-by-4 matrix we need to augment it with a 6-by-2 matrix, $\mathbf{g}_s$, to make it invertible. This also requires, then, that the input vector $\mathbf{u}$ be augmented with a slack variable vector, $\mathbf{u}_s$. Eq. (1) can then be rewritten as

$$\ddot{\mathbf{x}} = f(\mathbf{x}) + G(\mathbf{x})U - \nu + f_{ex}(\mathbf{x}) \ , \quad (19)$$

where $G = [g(\mathbf{x}),\ \mathbf{g}_s]$, $U = [\mathbf{u}^T,\ \mathbf{u}_s^T]^T$, and $\nu = \mathbf{g}_s\mathbf{u}_s$. If we set

$$\mathbf{g}_s = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}^T , \quad (20)$$

then $\nu = \mathbf{g}_s\mathbf{u}_s = \begin{bmatrix} u_5 & u_6 & 0 & 0 & 0 & 0 \end{bmatrix}^T$, so we need to estimate the slack variables u_5 and u_6.

Define the sliding surface as

$$S = [s_1, s_2, s_3, s_4, s_5, s_6]^T = \dot{\mathbf{x}} - \dot{\mathbf{x}}_r \quad (21)$$

where $\dot{\mathbf{x}}_r$ is a velocity reference vector, $[\dot{x}_r, \dot{y}_r, \dot{z}_r, \dot{\phi}_r, \dot{\theta}_r, \dot{\psi}_r]^T$. As previously mentioned, the quadrotor system is under-actuated such that the x and y state variables cannot be controlled directly from the inputs. Therefore, let us define the velocity references for $\dot{\phi}$ and $\dot{\theta}$ as eqs. (22) and (23) to control the x and y positions.

$$\begin{aligned} \dot{\phi}_r &= (\ddot{y} - \ddot{y}_r) + k_\phi\,(\dot{y} - \dot{y}_r) \quad (22)\\ \dot{\theta}_r &= (\ddot{x} - \ddot{x}_r) + k_\theta\,(\dot{x} - \dot{x}_r) \quad (23) \end{aligned}$$

where k_ϕ and k_θ are proportional gains.

Using the IVBS output from eq. (10) the reference velocity state variable, $\dot{\mathbf{x}}_r$, is given by

$$\begin{aligned} \dot{\mathbf{x}}_r &= E_r\dot{r}_d + \dot{\phi}_r E_4 + \dot{\theta}_r E_5 \quad (24)\\ &= -E_r W J_{p^v}^T e + \dot{\phi}_r E_4 + \dot{\theta}_r E_5 \end{aligned}$$

where $E_r = diag[1,1,1,0,0,1]$, $E_4 = [0,0,0,1,0,0]^T$ and $E_5 = [0,0,0,0,1,0]^T$.

In order to cancel nonlinear terms in eq. (19), we need to define the estimated values of ν and $f_{ex}(\mathbf{x})$, which we denote as $\hat{\nu}$ and $\hat{f}_{ex}(\mathbf{x})$, respectively. The uncertainty term is defined as

$$\Delta = -\nu + f_{ex} = [u_5, u_6, F_{gr}(z), 0, 0, 0]^T \quad (25)$$

Then equation (19) becomes,

$$\ddot{\mathbf{x}} = f(\mathbf{x}) + G(\mathbf{x})U + \Delta. \tag{26}$$

We set the Lyapunov candidate function as

$$\mathcal{L} = \frac{1}{2}S^T S + \frac{1}{2\Gamma}\tilde{\Delta}^T\tilde{\Delta} \tag{27}$$

and define the control input as

$$U = G^{-1}(\mathbf{x})[-f(\mathbf{x}) - \hat{\Delta} + \ddot{\mathbf{x}}_r - C_1 S - C_2 \text{sign}(S)] \tag{28}$$

where C_1 is a diagonal weighting matrix and C_2 is a positive constant gain. By setting the uncertainty estimate update law as

$$\dot{\hat{\Delta}} = \Gamma S \tag{29}$$

where Γ is positive semi-definite weighting matrix, the time derivative of the Lyapunov function becomes negative semi-definite.

$$\dot{\mathcal{L}} = -S^T(C_1 S + C_2 \text{sign}(S)) \leq 0\ , \tag{30}$$

Therefore the entire system converges to its desired state.

IV. Experimental Results

In this section, the experimental setup is described and the proposed algorithm for autonomous landing on a moving target is validated.

A. Hardware Description

The quadrotor and landing target are shown in fig. 1. The length from each motor rotational axis to the center of the quadrotor is 30 cm, the weight of the entire UAV system is 1700 grams including two batteries and the height is 19 cm. The onboard camera has a 50 degree of field of view (FOV) and captures 320px $\times$ 240px images at 30fps which are sent to the PC by a 1.2GHz onboard video transmitter. After control calculation, reference commands are sent to the quadrotor by a 2.4GHz radio control (RC) transmitter which is connected to the PC via an Endurance R/C PCTx USB dongle.

The quadrotor carries an inertial measurement unit (IMU) to measure angular rates which are used to estimate the Euler angles. If the target is not detected, a Vicon [13] camera system measures the position from which the translational velocity is calculated. When the target is detected, the Vicon data is used only for translational velocity. The entire control system is depicted in fig 3.

The landing pad is 90 cm long on each side and a fiducial marker is positioned approximately at the center. The four outside corners of the fiducial marker are used as the image features to compute J_p

B. Mission Scenario

The entire flight sequence is depicted in fig. 4. There are three modes for a mission:

1) Patrol Mode: After taking off, the quadrotor patrols over a certain area until the target is detected continuously for a pre-determined time duration (0.2 seconds). In this mode, the position data is obtained from Vicon.

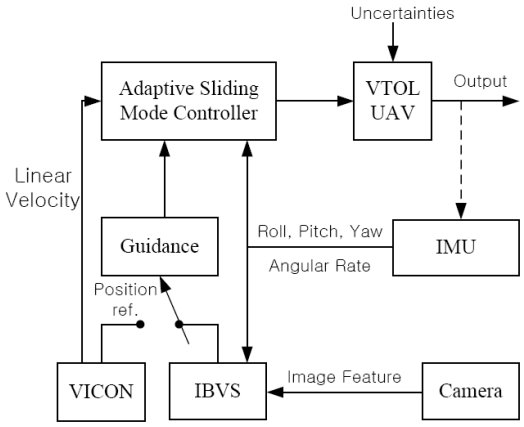

Fig. 3. Structure of the control system for the quadrotor helicopter

2) IBVS-Guided Tracking Mode: After the target is detected for 0.2 seconds, the control logic is switched to tracking mode. The purpose of this mode is to stabilize the quadrotor 0.4m over the target before attempting the landing maneuver (since a one-dimensional vertical maneuver is easier than attempting to land immediately via a three-dimensional maneuver). To do this the desired image feature locations are set to correspond to how they would look if the quadrotor were positioned at the desired location. Since the target usually first appears at the edge of the FOV, this maneuver also moves the target to the center where there is less risk of loosing the target while landing.
3) IBVS-Guided Landing Mode: After the landing mode is turned on, the desired image features are set to correspond to the position 0.20m directly over the target. If the error norm is below a defined threshold (implying that the quadrotor is at the target height and position) the thrust is turned off so the quadrotor can finish the landing maneuver. This small open-loop drop is necessary since the target is now so close to the camera that even small state errors can cause the camera to lose tracking for a number of reasons: the velocity of the features in image space becomes much higher, nonlinear distortions outside the camera calibration range become much more significant, and the FOV is now very small.

The ground target moves from its initial position (x, y and z) [0.1 m, 1 m, 0.07 m] with a velocity of approximately [0 m/s, $-$ 0.07 m/s, 0 m/s]. During the tracking and landing modes, if the target is not detected for 2 seconds continuously, then the control logic is reverted back to patrol mode. The entire sequence from take-off to landing is operated automatically.

C. Experimental Results

The proposed landing algorithm is validated by experiment. The vertical dash-dot line (14.2 second) in each of fig. 5 $\sim$ 11 shows the moment when the control logic switches to IBVS-based tracking mode, the vertical dotted line (18.2 second) shows the moment when logic switches to IBVS-based landing mode, and the vertical dashed line

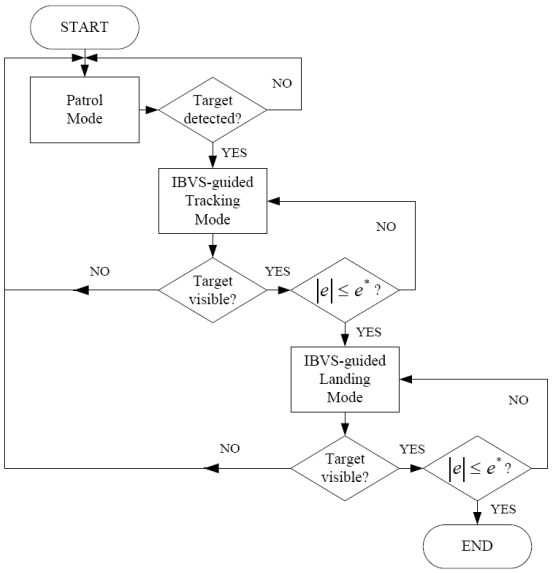

Fig. 4. Autonomous flight sequence

(19.3 second) represents the moment when the mission is completed (i.e. when the thrust is turned off).

Figs. 5 and 6 show position and attitude variables for the quadrotor during the mission. Note that the height of the quadrotor starts from 0.2 meter, which is the height of the markers used by the Vicon system to measure position. Also, the height of the quadrotor at the end of the mission is 0.27 meter, because the height of the target is 0.07 meter.

Fig. 7 shows the four control input signals for thrust, roll, pitch and yaw and fig. 8 shows the reference velocity (eq. (10)) tracking along the x, y and z axes. These reference velocities are generated only once the IBVS-guided control mode is turned on (dash-dot line), so there are no reference velocities (which are recorded as zero) and the controller tracks a guidance command calculated from Vicon position data before this. During the IBVS-guided tracking mode (from dash-dot to dotted line) and landing mode (from dotted to dashed line), the proposed controller shows satisfactory results for tracking performances.

Fig. **??** shows the depth estimation value derived from eq. (15) and the depth value (considered to be the true depth) measured by the Vicon (which is the same as the z position data in fig. 5). Note that eq. (15) calculates the depth between the camera and the target (0.05 m above the camera at rest) while the Vicon measurement is actually the height of the quadrotor markers which are 0.2 meter above the ground at rest. Therefore there is an expected 0.25 m offset between measurements.

Fig. 10 shows the error distance of the four image features in the image frame. Before the dash-dot line the image is not visible (i.e. Vicon control) so the error is set to zero. Between the dash-dot line and the dotted line the control logic is in tracking mode and the IBVS algorithm successfully tracks the image until it achieves the pre-landing setpoint of 0.4m above the target. The sudden change in tracking error at the dotted line is due to the image feature desired location changing once the control logic enters the landing mode,

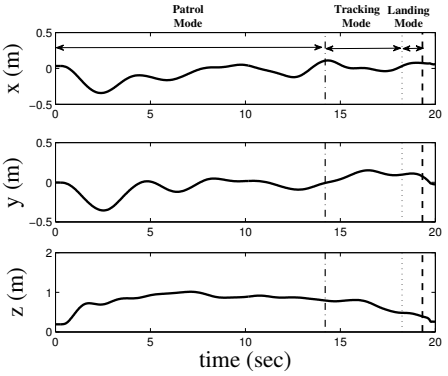

Fig. 5. Time history of the position of the UAV (x, y and z)

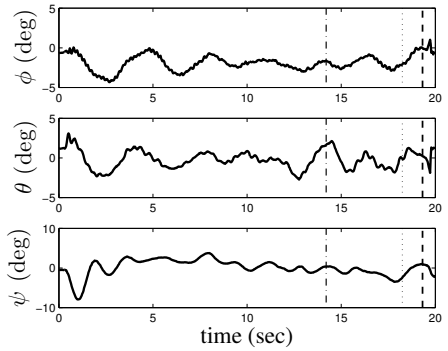

Fig. 6. Time history of the attitude of the UAV (ϕ, θ and ψ)

which has a larger desired feature size.

Fig. 11 represents the estimated ground effect as shown in eq. (29). As expected the estimated effect increases rapidly as the quadrotor approaches the ground.

V. Conclusion

In this paper, an autonomous landing control algorithm using image-based visual servoing (IBVS) for vertical take-off and landing (VTOL) unmanned aerial vehicles (UAVs) is described. The ground effect experienced during this maneuver is also compensated for by an adaptive control law. The algorithm is also validated by experiment where three control modes are defined: a patrol mode, a tracking mode and a landing mode. During the patrol mode, since the target is not detected yet, an external position sensor (i.e. Vicon) provides position guidance to the UAV. After

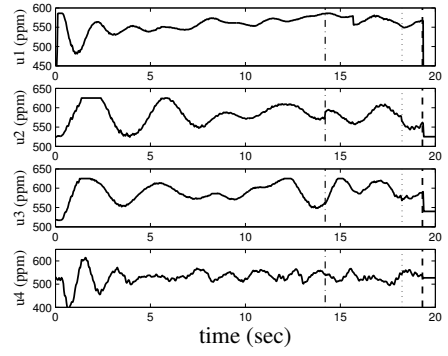

Fig. 7. Time history of the control inputs

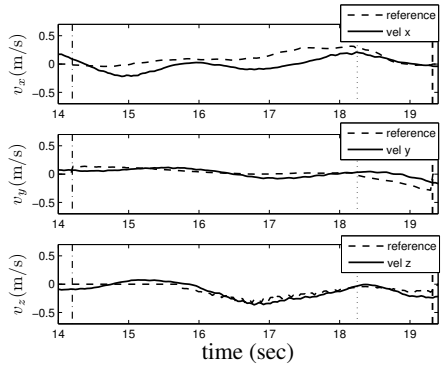

Fig. 8. Velocities of the quadrotor helicopter (solid line) and its references (dashed line) during the IBVS-guided control

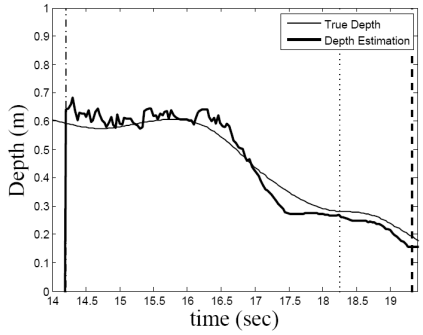

Fig. 9. Depth estimation and its true values

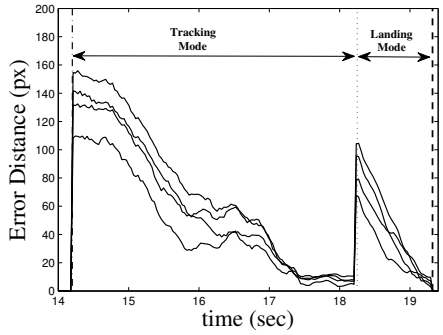

Fig. 10. Error distance of the four corners in the image frame. The sudden change at 18.2 seconds is due to the difference in desired feature locations between tracking mode and landing mode.

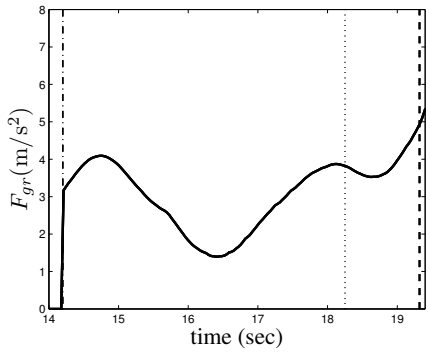

Fig. 11. Estimated ground effect value with the adaptive sliding mode controller

the target is detected, control logic switches to the tracking mode where position control is accomplished by tracking image features in the image frame. This mode positions the UAV above the target where the UAV is now ready to perform the landing maneuver. Now in landing mode, another desired image feature is defined to bring the UAV to the landing target. Once the image feature has converged to its desired position, the throttle is turned off to allow the UAV to land on the target. If the target leaves the FOV for longer than a couple seconds during either tracking or landing modes, the control logic is switched back to the patrol mode where the sequence can be started again once if the target can be found. The experimental results show satisfactory performance with the proposed IBVS in the nonlinear control loop. A video of this can be seen at http://www.youtube.com/watch?v=qrbVVJJzy_Q.

VI. ACKNOWLEDGMENTS

This work was supported by KARI-University Partnership Program (grant No. 2009-09-sunggwa-7) from the Korea Aerospace Research Institute (KARI) and Basic Science Research Program Through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology (2011-0003656).

REFERENCES

[1] L. Marconi, A. Isidori, and A. Serrani, Autonomous vertical landing on an oscillating platform: An internal-model based approach, *Automatica, vol. 38, pp. 2132, 2002.*

[2] L. Besnard, Y. Shtessel and B. Landrum, "Control of a Quadrotor Vehicle Using Sliding Mode Disturbance Observer," Proceedings of the 2007 American Control Conference*, pp. 5230-5235, 2007.*

[3] C. Coza and C.J.B. Macnab, "A New Robust Adaptive-Fuzzy Control Method Applied to Quadrotor Helicopter Stabilization," NAFIPS 2006 Annual meeting of the North American Fuzzy Information Society*, pp. 454-458, 2006*

[4] T. Dierks and S. Jagannathan, "Output Feedback Control of a Quadrotor UAV Using Neural Networks", IEEE Transaxtions on Neural Networks, VOL. 21, NO. 1, 2010, pp. 50-66

[5] S. Saripalli, J. Montgomery and G. Sukhatme, Visually-Guided Landing of an Unmanned Aerial Vehicle. IEEE Transactions on Robotics and Automation, 19(3) pp. 371-81, 2003.

[6] S. Lange, N. Sunderhauf, and P. Protzel, A Vision Based Onboard Approach for Landing and Position Control of an Autonomous Multirotor UAV in GPS-Denied Environments, in Proceedings of the International Conference on Advanced Robotics (ICAR), 2009.

[7] M. Blosch, S. Weiss, D. Scaramuzza, and R. Siegwart, "Vision based mav navigation in unknown and unstructured environments", in International Conference on Robotics and Automation (ICRA), 2010.

[8] L. R. G. Carrillo, E. Randon, A. Sanchez, A. Dzul and R. Lozano, "Stabilization and trajectory tracking of a quad-rotor using vision", JournalL of Intelligent and Robotic Systems, 61(1-4), 2011, pp 103-118.

[9] E. Altug, J. P. Ostrowski and R. Mahony, "Control of a Quadrotor Helocopter using Visual Feedback," Proceedings of the 2002 IEEE International Conference on Robotics and Automation, pp. 72-77. 2002.

[10] D. Lee, H. J. Kim, S. Sastry, "Feedback linearization vs. adaptive sliding mode control for a quadrotor helicopter", Int. J. Control Autom. Syst. 7(3), 419428 (2009)

[11] B. Herisse, F.-X. Russotto, T. Hamel, and R. Mahony, Hovering flight and vertical landing control of a vtol unmanned aerial vehicle using optical flow, in Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2008, pp. 801-806.

[12] Hutchinson, S., Hager, G., and Corke, P., "A tutorial on visual servo control", Robotics and Automation, IEEE Transactions on, 12(5) 1996, pp. 651-670.

[13] www.vicon.com