2014 International Conference on Unmanned Aircraft Systems (ICUAS)
May 27-30, 2014. Orlando, FL, USA

Outdoor Autonomous Landing on a Moving Platform for Quadrotors using an Omnidirectional Camera

JeongWoon Kim, Yeondeuk Jung, Dasol Lee and David Hyunchul Shim

Abstract — *This paper proposes a vision-based target following and landing system for a quadrotor vehicle on a moving platform. The system is consisted with vision-based landing site detection and locating algorithm using an omnidirectional lens. Latest smartphone was attached on the UAV and served as an on-board image acquisition and process unit. Measurements from the omnidirectional camera are combined with a proper dynamic model in order to estimate position and velocity of the moving platform. An adaptive control scheme was implemented on the flight computer to deal with unknown disturbances in outdoor environment. The system was validated on a quadrotor UAV and the vehicle successfully landed on the moving platform in outdoor flight tests.*[1]

I. INTRODUCTION

In recent years, Unmanned Aerial Vehicles (UAVs) have become important not only in the military, but also in civilian practice. There are many examples where UAVs are successfully applied to help human beings. [1] describes the various civilian applications for the UAVs including search and rescue, border surveillance, communication relay, wild fire suppression, disaster and emergency handling, research, agricultural and industrial applications. The most recent and noticeable contribution was the exploration of damaged nuclear reactors in Fukushima in March 2011 by flying UAVs into the air and monitoring the dangerous parts of the reactors.

GPS receiver is the most popular sensor for navigation of UAVs. A stand-alone GPS receiver, however, cannot afford to provide precise position information required for landing sequence to some miniature UAVs. Thus, the vehicle is hard to land on the landing site precisely even if it has the coordinate of the site in the earth frame. Differential GPS (DGPS) can provide very accurate position information in centimeter units, but the receiver is bulky for attaching on the miniature UAVs.

Vision sensors serve as eyes of unmanned vehicles assigned for reconnaissance missions. Not only the cameras can shoot pictures of the mission area, but they are able to support the vehicle autonomy using visual information.

One of the most representative usages of the visual information in UAV industries is indication of the landing site with several painting patterns. In [2], simulation of the landing logic for a quadrotor UAV is provided.

There were several trials, also, to guide the Vertical Take-Off and Landing (VTOL) UAVs to the moving target through flight tests. [3] describes a landing algorithm for helicopters in outdoors. The experimental results in this paper are acquired by manual flights. In [4], a visual servoing scheme using optical flow of the landing pad helped the multi-rotor UAV land on vertically moving pad in indoors. [5-6] introduces full landing sequence of the quadrotor UAV in indoors. The authors in this paper used visual marker for the moving landing site.

Landing control of the quadrotor UAV in outdoors, however, is difficult due to the absence of the precise position and velocity measurements from the external motion capture system. Rather, wind and gust disturb the landing sequence of the quadrotor.

The objective of this paper is to develop a system capable of following and landing on predefined targets with quadrotor platforms. The main sensor to be used is the bottom facing camera in order to detect the target and calculate relative position with respect to it. Image

[1] J. Kim, Y. Jung and D. Lee are with the Department of Aerospace Engineering, KAIST, Daejeon, South Korea (phone: +82-42-350-3764; fax: +82-42-350-3710; e-mail: yawh03@kaist.ac.kr).

D.H. Shim is with the Department of Aerospace Engineering, KAIST, Daejeon, South Korea (e-mail:shim@kaist.ac.kr).

978-1-4799-2376-2/14/$31.00 ©2014 IEEE

processing algorithms are required to be light enough – also the target itself is needed to be simple – to maneuver the drone with proper performance. Whole procedure including process of the image and control of the drone should robustly deal with various situations, such as following a moving target or landing on a moving platform.

II. Target Detection

In this study, we consider a red target for vision-based tracking using a quadrotor. The methods applied for finding the red target in an image frame in this part uses the same target detection algorithm in [8].

A color filter is an algorithm that collects red pixels from the original RGB image and reforms the image into a binary format. Let us assume that the acquired image has 8-bit, 3-channel RGB format. Algorithm 1 filters out the pixels where one major color is dominant over the other two colors.

Algorithm 2 is the whole procedure of the color filter which has a basis on Algorithm 1. First, run Algorithm 1 three times, one repetition for each of the three colors. The resultant three images are subjected to some binary operations for finding "reddish" pixels in the RGB image and produces final binary image, $\mathbf{I}_{CF}$.

In the next step, contours in $\mathbf{I}_{CF}$ are retrieved using the method proposed in [9]. The biggest contour is finally regarded as our target. The

Algorithm 1.
Input: $(\mathbf{I}_1, \mathbf{I}_2, \mathbf{I}_3)$, where each $\mathbf{I}_k$ is a matrix of an 8-bit, 1 channel image frame.
Output: A filtered binary image frame matrix, $\mathbf{I}_{1b}$

getBinaryImageofColor$(\mathbf{I}_1, \mathbf{I}_2, \mathbf{I}_3)$
  For every pixel value $\mathbf{I}_{k,ij}$ of i-th row and j-th column in the matrix $\mathbf{I}_k$
    If $\mathbf{I}_{1,ij} \geq K_I(\mathbf{I}_{2,ij} + \mathbf{I}_{3,ij}) + C_I$ then
    $\mathbf{I}_{1b,ij} \leftarrow 1$
    Else
    $\mathbf{I}_{1b,ij} \leftarrow 0$
    endIf
  endFor
return $\mathbf{I}_{1b}$

Algorithm 2. Pseudocode for filtering out a 1-channel binary image from a 3-channel image with respect to a specific color.

Algorithm 2.
Input: $(\mathbf{I}_R, \mathbf{I}_G, \mathbf{I}_B)$, where $\mathbf{I}_R, \mathbf{I}_G$ and $\mathbf{I}_B$ are split matrices of 8-bit, 1 channel image frames of red, blue and green color from the 3-channel RGB image.
Output: A filtered binary image frame matrix, $\mathbf{I}_{CF}$

colorFilter$(\mathbf{I}_R, \mathbf{I}_G, \mathbf{I}_B)$
  $\mathbf{I}_{RR} = \text{getBinaryImageofColor}(\mathbf{I}_R, \mathbf{I}_G, \mathbf{I}_B)$
  $\mathbf{I}_{GG} = \text{getBinaryImageofColor}(\mathbf{I}_G, \mathbf{I}_B, \mathbf{I}_R)$
  $\mathbf{I}_{BB} = \text{getBinaryImageofColor}(\mathbf{I}_B, \mathbf{I}_R, \mathbf{I}_G)$
  For every pixel value $\mathbf{I}_{k,ij}$
    If $0.6\mathbf{I}_{R,ij} < \mathbf{I}_{G,ij}$ then
    $\mathbf{I}_{Temp,ij} \leftarrow 1$
    else
    $\mathbf{I}_{Temp,ij} \leftarrow 0$
    endIf
  $\mathbf{I}_{CF,ij} \leftarrow \mathbf{I}_{RR,ij} - \mathbf{I}_{GG,ij} - \mathbf{I}_{BB,ij} - \mathbf{I}_{Temp,ij}$
  endFor
return $\mathbf{I}_{CF}$

Algorithm 1. Pseudocode for filtering out the most "reddish" pixels in an input RGB image frame.

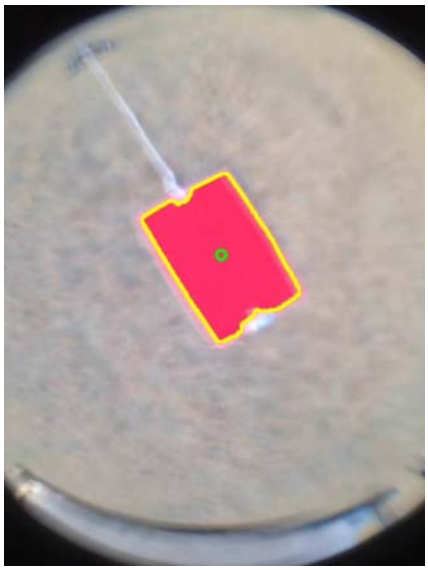

Fig. 1. The detected target in an image frame. The detection algorithm indicates the target with a yellow contour.

Figure 1 depicts the target that is detected in an image frame collected from flight test.

III. Target Geolocation

In order to land on a visual landing pad in the outdoors, the pad needs to be located on the navigation (North-East-Down, NED) frame using outputs of vision sensors. In this chapter we describe the method, so-called geolocation.

A. Fish-eye Lens Model

In order to land on a specific visual landing pad in outdoors, several aspects need to be considered. The most important one is that FOV (Field of View) shrinks while the quadrotor descends above the landing pad. This phenomenon leads the size of visual pad to exceed FOV of vision sensor. Obviously, the landing pad becomes undetectable under certain low altitude. The other aspect we need to consider is that the landing pad might move around. The visual pad has possibility to escape from the image frame and make the UAV lose its landing site.

The omnidirectional camera is a camera with a 360-degree field of view in the horizontal plane, or with a visual field that covers approximately the entire sphere. The problems mentioned above can be efficiently solved by this type of camera. Fish-eye lenses and catadioptric lenses are the most widely used lenses for these cameras. A previous work describes a numerical method to calibrate these lenses to use them for visual odometry by approximating their refraction surface with a polynomial series [7].

First, define the refraction surface as a function of pixel position of the image frame:

$$w = f(u,v) \tag{1}$$

where (u,v) is a pixel coordinate in the camera frame.

Next, the vector which points the target:

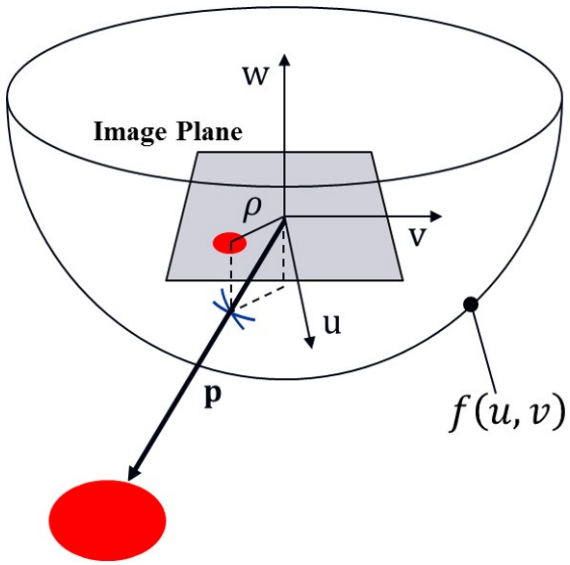

Fig. 2. Geometry between the target and the omnidirectional camera. The refraction surface is approximated with the function $f(u,v)$.

$$\mathbf{p} = \begin{bmatrix} u & v & f(u,v) \end{bmatrix}^T \tag{2}$$

Since the camera is omnidirectional, the refraction surface is symmetrical around w axis. Therefore the refraction surface is described as:

$$w = f(u,v) = f(\rho) \tag{3}$$

where $\rho = \sqrt{u^2 + v^2}$.

Then the surface equation can be approximated to 4th order polynomial:

$$\begin{aligned} w &= f(u,v) \\ &= f(\rho) = a_0 + a_1\rho + a_2\rho^2 + a_3\rho^3 + a_4\rho^4 \end{aligned} \tag{4}$$

where a_i $(0 \le i \le 4)$ are coefficients which are approximated by using the calibration method in[7]. Knowing that the coordinate (u, v) of the target in the image frame, we can evaluate the projection vector $\mathbf{p}$.

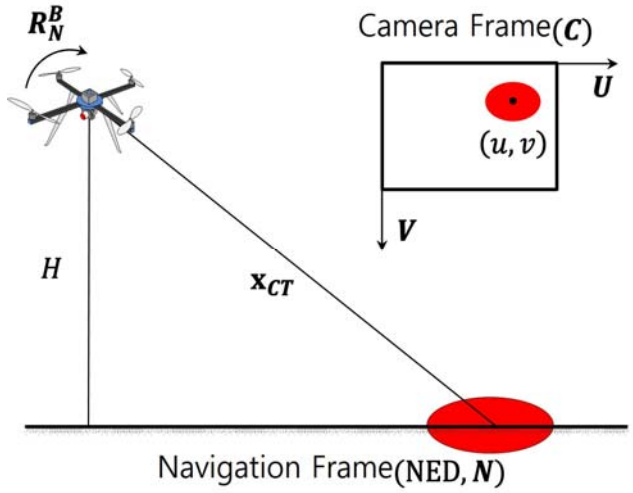

Fig. 3. Geometry between the UAV and the target in the navigation frame.

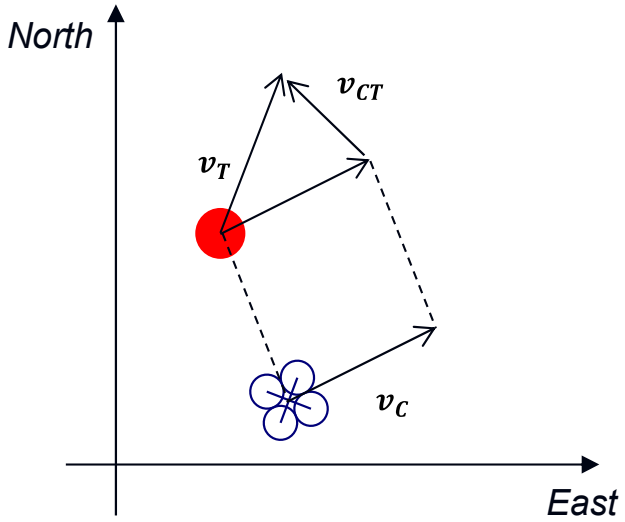

Fig. 4. Horizontal geometry between the UAV and the target in the navigation frame.

B. Geolocation Algorithm

The fish-eye lens model derived in the previous subsection is:

$$s\mathbf{R}_N^C\mathbf{x}_{CT} = \begin{bmatrix} u & v & f(u,v) \end{bmatrix}^T \tag{5}$$

where $\mathbf{R}_N^C$ is a rotation matrix from the navigation frame to the camera frame, $\mathbf{x}_{CT} = \begin{bmatrix} x & y & H \end{bmatrix}^T$ is a position difference of the camera and the target in the navigation frame and s is a scale factor. (u,v) is a detected position of a target in the camera frame and s is a scale factor.

$\mathbf{x}_{CT} = \begin{bmatrix} x & y & H \end{bmatrix}^T$ is unknown, but H is known by assuming flat earth model. This means the relative altitude of the UAV from the ground becomes the position difference between the UAV and the target in vertical direction in the navigation frame. By substituting $\mathbf{M}^{-1} = \mathbf{R}_N^C$ into Equation (5), we get:

$$\begin{aligned} \mathbf{x}_{CT} &= s^{-1}\mathbf{M}\begin{bmatrix} u & v & f(u,v) \end{bmatrix}^T \\ \Leftrightarrow \begin{bmatrix} x \\ y \\ H \end{bmatrix} &= s^{-1}\begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix}\begin{bmatrix} u \\ v \\ f(u,v) \end{bmatrix} \end{aligned} \tag{6}$$

Therefore, we get the following equations:

$$s = H^{-1}\left(M_{31}u + M_{32}v + M_{33}f(u,v)\right) \tag{7}$$

$$\begin{aligned} x &= s^{-1}\left(M_{11}u + M_{12}v + M_{13}f(u,v)\right) \\ &= \frac{M_{11}u + M_{12}v + M_{13}f(u,v)}{M_{31}u + M_{32}v + M_{33}f(u,v)} \end{aligned} \tag{8}$$

$$\begin{aligned} y &= s^{-1}\left(M_{21}u + M_{22}v + M_{23}f(u,v)\right) \\ &= \frac{M_{21}u + M_{22}v + M_{23}f(u,v)}{M_{31}u + M_{32}v + M_{33}f(u,v)} \end{aligned} \tag{9}$$

IV. Estimation Filter

Uncertainties from the target position in the camera frame are suitable to model as a non-biased Gaussian noise. They sometimes cause noisy output from geolocation algorithm, so they must be reduced by using proper Kalman filter. On the other hand, the Kalman filter can help seeking the target when it is missing temporarily by propagating the target motion model. In this section, we construct nonlinear Kalman filter to estimate position and velocity of our target in the navigation frame.

A. Dynamic model

We assume that our target is moving on the earth with maintaining its velocity constant. Therefore in continuous time domain we can say:

$$\dot{\mathbf{v}}_T = 0 \tag{10}$$

where $\mathbf{v}_T$ is velocity of the target on the navigation frame. Above equation can be also expressed in discrete time domain:

$$\mathbf{v}_{T,k} = \mathbf{v}_{T,k-1} \tag{11}$$

where $\mathbf{v}_{T,k}$ denotes velocity of the target at specific moment k. Considering that our geolocation algorithm provides relative position between the target and the camera (or UAV), we need to reproduce our state equation from the above equation. By substituting $\mathbf{v}_{T,k} = \mathbf{v}_{C,k} + \mathbf{v}_{CT,k}$ to the equation, we can get:

$$\mathbf{v}_{CT,k} = \mathbf{v}_{CT,k-1} - \Delta\mathbf{v}_C \tag{12}$$

where $\mathbf{v}_{C,k}$ is the velocity of the camera at the specific moment k and $\Delta\mathbf{v}_C$ indicates the velocity difference between time k and $k-1$.

B. Nonlinear Discrete Kalman Filter Formulation

Now we can formulate the discretized version of nonlinear Kalman filter:

$$\begin{aligned} \mathbf{x}_k &= \mathbf{f}\left(\mathbf{x}_{k-1}, \mathbf{u}_k\right) \\ \mathbf{y}_k &= \mathbf{h}\left(\mathbf{x}_k, H_k\right) \end{aligned} \tag{13}$$

where $\mathbf{x}$ is a state vector of our filter, $\mathbf{u}$ is a input vector of state prediction equation and $\mathbf{y}$ is a measurement vector. Let us define $\mathbf{x}$ as:

$$\begin{aligned} \mathbf{x} &= \begin{bmatrix} \mathbf{x}'_{CT} & \mathbf{v}'_{CT} & \boldsymbol{\eta}_C \end{bmatrix}^T \\ &= \begin{bmatrix} x & y & v_x & v_y & \phi_C & \theta_C & \psi_C \end{bmatrix}^T \end{aligned} \tag{14}$$

where $\mathbf{x}'_{CT} = \begin{bmatrix} x & y \end{bmatrix}^T$ and $\mathbf{v}'_{CT} = \begin{bmatrix} v_x & v_y \end{bmatrix}^T$ are horizontal position distinction and its derivative between the target and the UAV. $\boldsymbol{\eta}_C = \begin{bmatrix} \phi_C & \theta_C & \psi_C \end{bmatrix}^T$ is an Euler angle vector of the camera frame with respect to the navigation frame. Next, input of state propagation equation is defined:

$$\mathbf{u} = \begin{bmatrix} 0 \\ -\Delta\mathbf{v}'_C \\ \boldsymbol{\eta}_C \end{bmatrix} \tag{15}$$

where $\Delta\mathbf{v}'_C$ is the horizontal velocity difference between time k and $k-1$.

According to Equation (12), the state prediction equation is finalized as below:

$$\mathbf{x}_k = \mathbf{f}\left(\mathbf{x}_{k-1}, \mathbf{u}_k\right) = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{u}_k \tag{16}$$

where $\mathbf{A} = \begin{bmatrix} \mathbf{I}_{2\times2} & \Delta t\mathbf{I}_{2\times2} & \mathbf{0}_{2\times3} \\ \mathbf{0}_{2\times2} & \mathbf{I}_{2\times2} & \mathbf{0}_{2\times3} \\ \mathbf{0}_{3\times2} & \mathbf{0}_{3\times2} & \mathbf{0}_{3\times3} \end{bmatrix}$.

Now let's define measurement vector as:

$$\mathbf{y}_k = \mathbf{h}\left(\mathbf{x}_k, H_k\right) \tag{17}$$

Considering that the state update equation needs to update the horizontal position of the target in the NED frame $\mathbf{x}'_{CT} = \begin{bmatrix} x & y \end{bmatrix}^T$ using fish-eye lens model and the measurement vector $\mathbf{y} = \begin{bmatrix} u & v \end{bmatrix}^T$, the nonlinear function $\mathbf{h}\left(\mathbf{x}_k, H_k\right)$ can be simplified as:

$$\mathbf{y}_k = \mathbf{h}\left(x_k, y_k, H_k\right) \tag{18}$$

From Equation (5),

$$\begin{bmatrix} u \\ v \\ f(u,v) \end{bmatrix} = s\mathbf{R}_N^C \mathbf{x}_{CT}$$
$$= s\begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix}\begin{bmatrix} x \\ y \\ H \end{bmatrix} \quad (19)$$
$$= s\begin{bmatrix} R_{11}x + R_{12}y + R_{13}H \\ R_{21}x + R_{22}y + R_{23}H \\ R_{31}x + R_{32}y + R_{33}H \end{bmatrix}$$

Now, let us split the matrix equation (19) into three individual equations:

$$u = s\left(R_{11}x + R_{12}y + R_{13}H\right) \quad (20)$$

$$v = s\left(R_{21}x + R_{22}y + R_{23}H\right) \quad (21)$$

$$f(u,v) = f(\rho)$$
$$= a_0 + a_1\rho + a_2\rho^2 + a_3\rho^3 + a_4\rho^4 \quad (22)$$
$$= s\left(R_{31}x + R_{32}y + R_{33}H\right)$$

again, $\rho = \sqrt{u^2 + v^2}$.

By concatenating Equations (20) and (21),

$$s = \frac{\rho}{\sqrt{\sum_{i=1}^{2}\left(R_{i1}x + R_{i2}y + R_{i3}H\right)^2}} \quad (23)$$

By substituting s in Equation (22) by Equation (23),

$$a_0 + a_1\rho + a_2\rho^2 + a_3\rho^3 + a_4\rho^4 = \rho K_{\mathbf{x}_{CT}} \quad (24)$$

where, $K_{\mathbf{x}_{CT}} = \dfrac{\left(R_{31}x + R_{32}y + R_{33}H\right)}{\sqrt{\sum_{i=1}^{2}\left(R_{i1}x + R_{i2}y + R_{i3}H\right)^2}}$

Now Equation (24) becomes a quartic equation with respect to ρ. Fortunately, the general formula for four roots of quartic equation is well-known. Therefore we can evaluate the four roots $\rho_i \ (0 \le i \le 3, i \in Z)$. Among them, we need to select the most reliable root. First, the root should be positive real number. If multiple roots satisfy the condition, Euclidian distances between each roots ρ_i and $\rho_{prev} = \sqrt{u_{prev}{}^2 + v_{prev}{}^2}$ calculated from previous (most recent) measurement would be evaluated and the one $\rho_{nearest}$ which has the nearest distance are finally selected for the most suitable root.

Then from Equation (23),

$$s_r = \frac{\rho_{nearest}}{\sqrt{\sum_{i=1}^{2}\left(R_{i1}x + R_{i2}y + R_{i3}H\right)^2}} \quad (25)$$

By substituting $s = s_r$ in Equation (20) and (21),

$$u = \rho_{nearest}\frac{\left(R_{11}x + R_{12}y + R_{13}H\right)}{\sqrt{\sum_{i=1}^{2}\left(R_{i1}x + R_{i2}y + R_{i3}H\right)^2}} \quad (26)$$

$$v = \rho_{nearest}\frac{\left(R_{21}x + R_{22}y + R_{23}H\right)}{\sqrt{\sum_{i=1}^{2}\left(R_{i1}x + R_{i2}y + R_{i3}H\right)^2}} \quad (27)$$

The two Equations (26) and (27) formulates the nonlinear state update vector function $\mathbf{h}(x, y, H)$ from Equation (18). Since the function $\mathbf{y}_k = \mathbf{h}(x_k, y_k, H_k)$ is highly nonlinear, we used an Unscented Kalman Filter (UKF) to concatenate the linear process model and the nonlinear observation model.

V. OUTDOOR FLIGHT TESTS

The nonlinear target geolocation filter from Chapter 5 and the control strategy from Chapter 6 are integrated into one single UAV system and tested in outdoors. The tests not only include revision of moving target following but also

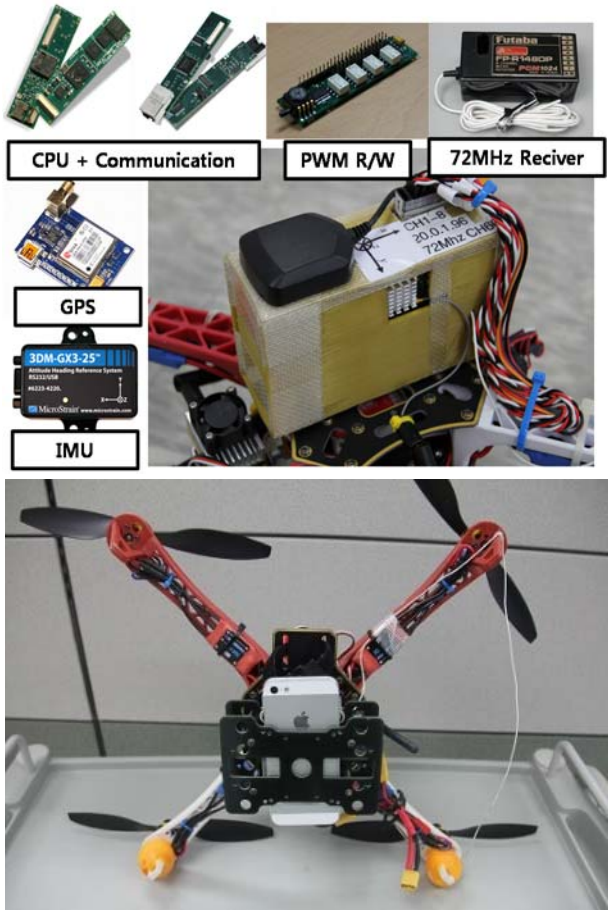

Fig. 5. DJI NAZA F450 equipped with a customized FCC box and a smartphone.

contain trials of landing on a moving pad. This chapter describes the tests and their results.

A. Quadrotor Testbed

We organized a quadrotor UAV system by using customized avionics box. The base quadrotor platform is DJI NAZA F450. It has attitude stabilization board inside to feedback angular rates. FCC (Flight Control Computer) provides PWM (Purse Width Modulation) commands of control surfaces such as an aileron, an elevator and a rudder to the stabilization board. The FCC of the UAV is Gumstix Verdex Pro composed of ARMv5 600MHz processor with 128MB DDR RAM. GPS and IMU sensors are equipped and packed in a box in the simple manner. The quadrotor can be manually operated with a controller with 72MHz RF signal.

In order to provide nearly real-time target

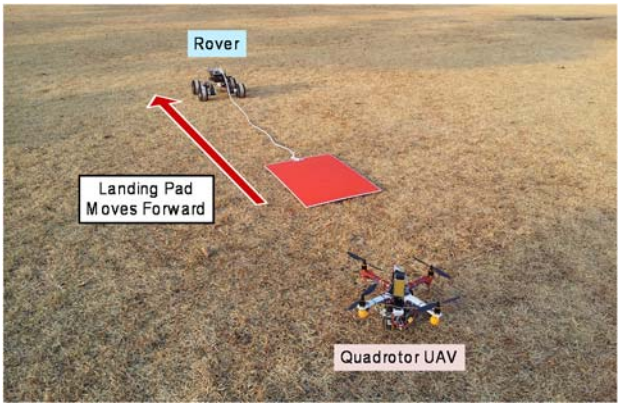

Fig. 6. Illustration of the test environment.

detection result, a smartphone is equipped on the bottom of the quadrotor. The smartphone takes real-time video pictures by using its onboard camera and process them with its own processor. Results from the video processing algorithm are passed to the FCC to generate guidance commands.

B. Test Environment

Due to extremely large FOV of fish-eye lens, shape of the target becomes unidentifiable above certain altitude. Hence, the target detection is only done by the color filter algorithm from [8], which finds a biggest red contour in the camera frame.

A rover platform drags red target toward and is controlled manually. The quadrotor UAV is commanded to maintain its horizontal position right above the target while descending.

The UAV faces certain interruptions including wind disturbances and model uncertainty while it is in flight. Therefore, we use L_1 adaptive augmentation loop based on an output feedback method to improve the accurate path tracking. The adaptive manner can supplements the linear controller to guarantee the flight performance in the presence of the uncertainties [9].

C. Flight Test Results

We tried several landing tests on the moving visual pad with the test environment and we got successful results.

The rover maintained its movement direction, so the landing pad only moved along the straight line

while maintaining nearly constant velocity.

Figure 8 shows the two-dimensional horizontal trajectory of the UAV and the landing pad in the NED frame. The horizontal position difference between the drone and the pad becomes smaller while descending because geolocation estimation error is proportional to the altitude. The error eventually converged to zero, which indicates successful landing.

Figure 7 shows time domain plots of the NED position and the horizontal velocity of the drone and the marker. We can notice that the vehicle speed tends to converge to the estimated speed of the marker with about 2 seconds time delay. Reference altitude command was sent to the quadrotor as a ramp input, which has -0.1m/s slope. The altitude output of the vehicle had about a half meter error at 20s to 45s, but it eventually converged to the zero.

Figure 9 illustrates vehicle attitude and body velocity plots. The attitude angle graph showed about 1 degree offset error from in whole flight. The body velocity has some amount of high frequency noise, but it almost converged to the reference.

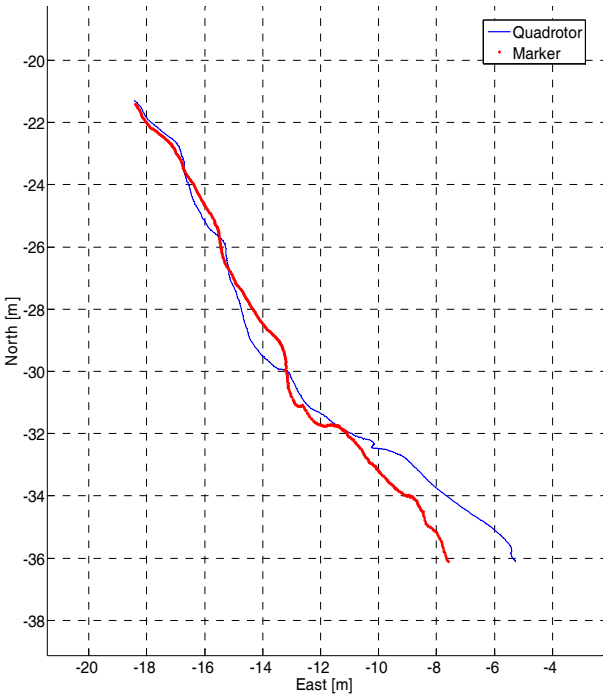

Fig. 8. 2D Trajectory of the UAV and the estimated position of the landing pad.

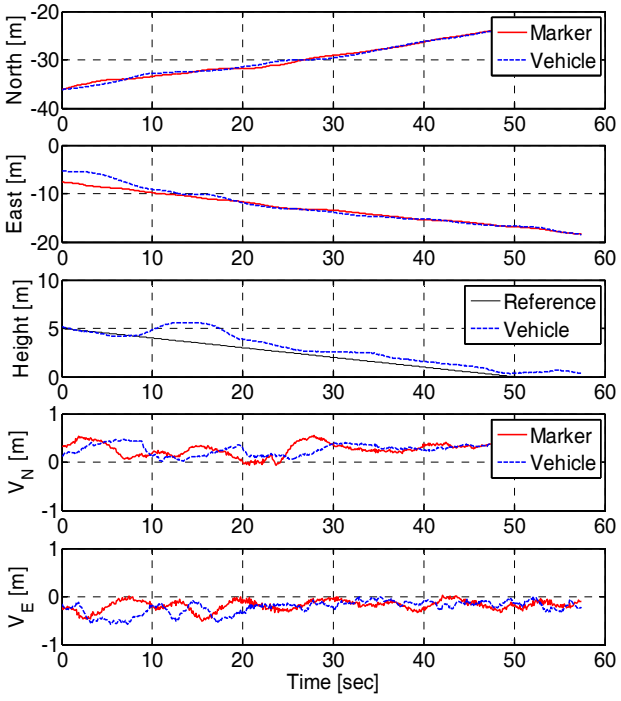

Fig. 7. Time history of the UAV, the estimated position and velocity of the landing pad.

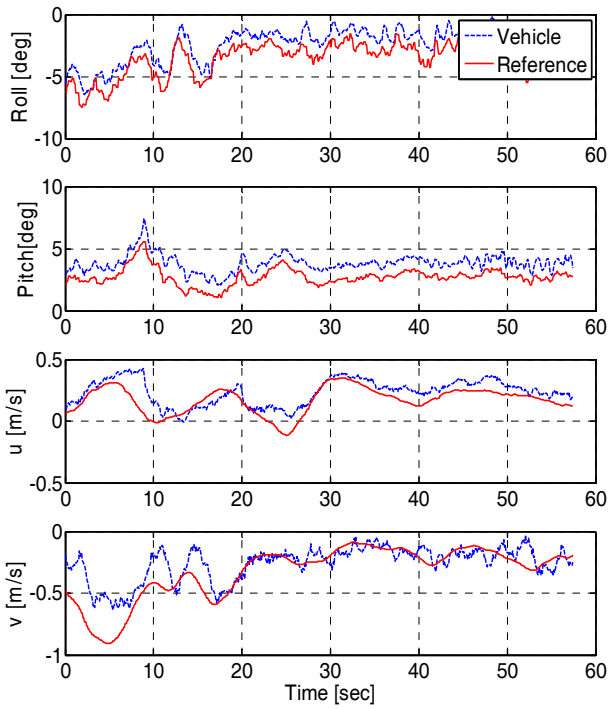

Fig. 9. Time history of the attitude angles and body velocity with their reference inputs.

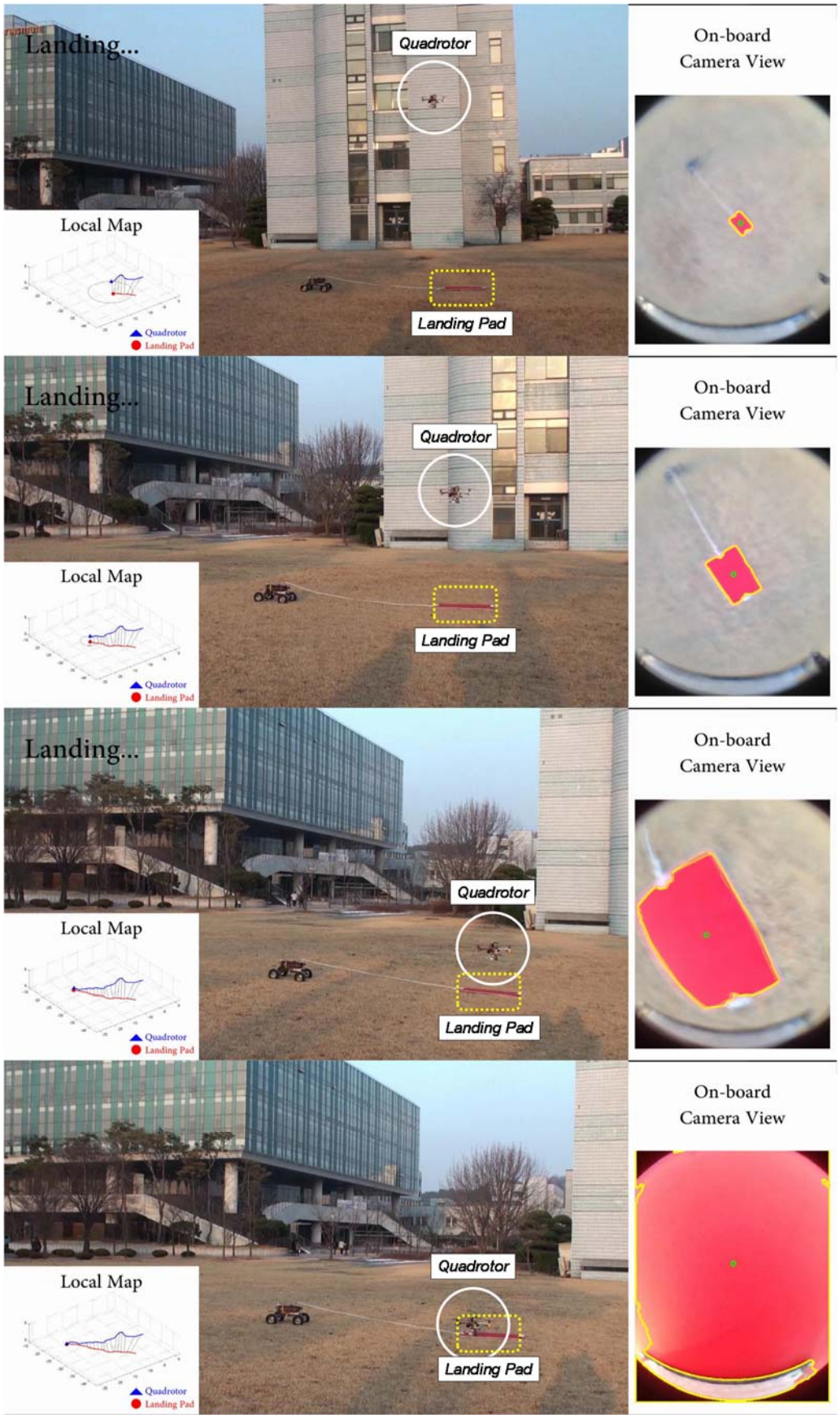

Fig. 10. Still Pictures from the Landing Test on a Moving Pad

VI. CONCLUSION

This paper proposed a landing algorithm for a quadrotor UAV. A set of algorithms were developed for detecting the target based on the color appearing on the image. In order to land on the specific visual landing pad in outdoors, fish-eye lens and its calibration model helped the shrinking FOV problem while descending above the visual pad. The nonlinear observation model and the constant velocity model in the NED coordinate frame were concatenated to formulate nonlinear estimation model. The model were properly evaluated and the state vector is estimated by using the Unscented Kalman Filter.

The geolocation filter and the rotorcraft controller were integrated and implemented into one single system and the system was validated by several flight tests. Especially, the UAV successfully landed on the moving pad in outdoors without help of external motion capture system. The resultant flight test graphs showed satisfactory performance.

The proposed target observation and tracking algorithm can be used for following or landing on specific platforms such as ground carrier vehicle or shipboards with visual markers. Furthermore, the possibility of using smartphones as a viable on-board image acquisition and computation platform and potent computing platform for realistic application was demonstrated.

ACKNOWLEDGMENT

Authors are gratefully acknowledging the financial support by Agency for Defense Development funded by the Korean government (No.UE124026JD).

REFERENCES

[1] Z. Sarris, “Survey of UAV applications in civil markets,” 9th Mediterranean Conference on Control and Automation, Dubrovnik, Croatia, June 27–29, 2001.

[2] H. Voos, H. Bou-Ammar, “Nonlinear tracking and landing controller for quadrotor aerial robots,” 2010 IEEE International Conference on Control Applications, pp.2136-2141, Sep 8-10, 2010.

[3] S. Saripalli, G. Sukhatme, "Landing a Helicopter on a Moving Target," 2007 IEEE International Conference on Robotics and Automation, pp.2030-2035, April 10-14, 2007.

[4] B. Herissé, T. Hamel, R. Mahony, F-X. Russotto, "Landing a VTOL Unmanned Aerial Vehicle on a Moving Platform Using Optical Flow," IEEE Transactions on Robotics, Vol.28, no.1, pp.77-89, Feb. 2012.

[5] D. Lee, T. Ryan, H.J. Kim, "Autonomous landing of a VTOL UAV on a moving platform using image-based visual servoing," 2012 IEEE International Conference on Robotics and Automation (ICRA), pp.971-976, May 14-18, 2012.

[6] K.E. Wenzel, A. Masselli, A. Zell, “Tracking and Landing of a Miniature UAV on a Moving Carrier Vehicle,” Journal of Intelligent and Robotic systems, Vol. 61, Issue 1-4, pp.221-238, Jan. 2011.

[7] D. Scaramuzza, A. Martinelli and R. Siegwart, "A Flexible Technique for Accurate Omnidirectional Camera Calibration and Structure from Motion," Proceedings of IEEE International Conference of Vision Systems, Jan 5-7, 2006.

[8] J. Kim, D. H. Shim, J. R. Morrison, “Tablet PC-based Visual Target-Following System for Quadrotors,” Journal of Intelligent and Robotic systems, Vol. 74, Issue 1-2, pp.85-95, Apr. 2014.

[9] Jung, Y.D., and Shim, D.H., “Development and Application of Controller for Transition Flight of Tail-Sitter UAV”, Journal of Intelligent and Robotic systems, Vol. 65, pp.137-152, 2012.