The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

Visual Servoing of an Autonomous Micro Air Vehicle for Ground Object Tracking

Syaril Azrad, Farid Kendoul, Dwi Perbrianti and Kenzo Nonami
Robotics and Control Lab, Department of Electronics and Mechanical Engineering
Chiba University, 263-8522, Chiba City, Japan

***Abstract*— This paper describes an object tracking system using an autonomous Micro Air Vehicle (MAV) and demonstrate its potential use for civilian purposes. The vision-based control system relies on a color and feature based vision algorithm for target detection and tracking, Kalman filters for relative pose estimation, and a nonlinear controller for MAV stabilization and guidance. The vision algorithm relies on information from a single onboard camera. An arbitrary target can be selected in real-time from the ground control station, thereby outperforming template and learning-based approaches. Experimental results obtained from outdoor flight tests, showed that the vision-control system enabled the MAV to track and hover above the target as long as the battery is available.**

I. INTRODUCTION

Japan, a country prone to earthquakes attributed to its geographical location has suffered substantially in terms of human lives and economic damage. Inspired by the situation in Japan, we at Chiba University in the Robotics and Control Laboratory, have built a prototype small-size rotorcraft Unmanned Air-Vehicle (UAV) system to address the problem of search and rescue in a disaster area.

There are a lot of systems applying rotorcraft UAVs for civilian applications, industrial based inspections and even similar application to assist search and rescue operations.It is the capabilities of a rotorcraft to stabilize and hover on a place of interest either by controlling manually or autonomously, that enables it to be chosen for such civilian applications.

There are works by Campoy *et. al* in [1] that describes UAV platforms and vision systems on-board developed at Universidad Politécnica de Madrid (UPM) to achieve vision-based tasks such as line inspection or window tracking. The above description also is related to the work by Mejías [2] based on the results for visual servoing in urban areas using feature tracking. There are also works on path planning for multiple UAV cooperation and forest fire detection which are described by Nikolos in [3] [4] and application in traffic monitoring by Puri *et. al* [5].

The works in the above examples are demonstrated by using rotorcraft UAVs, namely radio controlled helicopters that are powered by either battery or gas.

S. Azrad is a PhD student: s.azrad@graduate.chiba-u.jp
F. Kendoul is a PostDoc Research Fellow: fkendoul@restaff.chiba-u.jp
D. Perbrianti is a PhD student: dwipebri@graduate.chiba-u.jp
K. Nonami is a Professor: nonami@faculty.chiba-u.jp

Our focus of research is to build a feasible vision-based UAV system, to assist search and rescue operations during disaster in urban or natural environment. The choice of our air-vehicle platform is a small-size quadrotor UAV, which would be described later as Micro Air-Vehicle or MAV. A small-size autonomous rotorcraft would enable it to be quickly transported around during search and rescue operations. Moreover, a rotorcraft such as the quadrotor type can be launched and deployed easily in the area of disaster.

In our MAV system, we applied visual servoing techniques to control the position of our MAV, in order to hover the rotorcraft on the center of an arbitrarily selected ground target which appears in the camera field of view.

There are lot of works that were published in recent years on visual-servoing of an MAV, that focused more on the theoretical aspects of the control problem. Research by Altug [6] and Bourquardez [7] used quadrotor MAVs and results were presented after indoor tests. Fowers *et. al* [8] described the system to stabilize a hovering quadrotor MAV using vision. Works by Saripalli [9] [10] and Hrabar[11] applied vision-based techniques for UAV guidance by tracking a target used a bigger class helicopter with more payload capability which has enabled sophisticated imaging system or sensors to be mounted on the air vehicle.

Fig. 1. The quadrotor MAV during vision-based autonomous flight. In the right corner, is the quadrotor MAV prototype used in the experiments. The small size and robustness of the vehicle can be potentially utilised in various area of applications

A. Problem Formulation

Our work on vision-based MAV systems described below is based on a strategy to fulfill the requirement of the

practical problem of assisting search and rescue operation based on a scenario to assist an effective rescue and relief operation in an earthquake hit zone. The scenario of this kind operation is simulated in our second experiment.

In Section II, we describe the platform and the embedded architecture. Section III deals with the design of a real-time vision algorithm for object tracking. Section IV briefly described about our nonlinear controller. Experimental results are presented in Section V.

II. Platform,Software Implementation & Ground Station Description

To describe our MAV system, we divide it into three main components.

A. Micro-Air-Vehicle (MAV) Platform

Our MAV system is based on a small-size (53cm) four-rotor X-3D-BL platform, which is highly durable as it is able to sustain minor crashes, and has a high performance in terms of payloads and stability required for vision-based control. The total weight of the original platform with batteries included is about 400 grams, and has a payload up to 300 grams.The MAV is lifted and propelled by four rotors that are fixed on the brushless motors powered by a 2100 mAh 3-cell lithium polymer battery. It can fly for 23 minutes without payload, and about 12 minutes with a 200 grams payload. At high speed it can fly as fast as 10m/s.

For the Flight Control Computer (FCC) we use the Gumstix Connex micro-controller for the autopilot. Gumstix operates on Linux and has a Marvell PXA255 processor embedded with a CPU frequency of 400MHz and a large memory comprising of a 16 MB flash memory and 64MB SDRAM. Despite its powerful computing capabilities, it weighs only 18 g including the interface board. The Gumstix motherboard is expanded to provide two RS-232 ports and WiFi communication. The WiFi communication module provide us with a high bandwidth (about 50 Mbits/s), and a communication range of 500 meters. Communication between the Ground Control Station (GCS) and the FCC is done through UDP protocol using the WiFi communication module.

The autopilot is augmented with the MNAV100CA sensor from Crossbow. The low cost sensor which contains the IMU, GPS and pressure sensor weighs about 35 grams, excluding the GPS antenna. It includes 3-axis accelerometers, 3-axis angular rate sensors and 3-axis magnetometers; static pressure (altitude) and dynamic pressure (airspeed) sensors and a GPS receiver module. The AVR micro controller (Atmega32) is mounted on the system in order to interface our autopilot with the original X-3D-BL board, by generating PPM signal when receiving control inputs from the Flight Control Computer. The PPM interface allows for software interpretation of R/C receiver commands (PPM) and switching between autonomous and manual flight modes. All sensor data and the decoded R/C signals(throttle,pitching/rolling/yawing torques,switch,communication status) from MNAV100CA are sent to the *Gumstix* FCC through the serial port RS-232.*Gumstix* FCC also receives data (flight data/image processing output data) from the Ground Station via WiFi.Our imaging system consists of an analog camera and a wireless 1.3 GHz video transmitter system KX171 from *Range Video*.

The total weight of the MAV system is about 650 grams, which includes the air-vehicle, battery, FCC, sensors and imaging system.

The architecture of the the platform is shown in Fig. 2.

B. On-board real-time software

The implemented software in the embedded system contains the main components which is the navigational software and image processing software for target tracking.

The design of the software is based on the Open Source avionics software developed by Crossbow [12] [13]. The navigational software runs as a process within the Linux Operating System embedded on the FCC. It is designed as a real-time software and comprises of an architecture that can handle multiple events and respond to events within predictable time limits. To prevent the occurrence and propagation of faults or failures throughout the system, the sofware is designed to use 'threads' to implement the hosting of multiple tasks in a timely manner.

C. Ground Control Station (GCS)

The GCS computer communicates with the Micro-Air-Vehicle micro-controller through a wi-fi modem. Besides its main role as a 'Control Station' the GCS also streams videos from the wireless camera embedded on the platform. The user can select the target from the rendered images, and switch to vision-based hovering.The GCS and example of images during vision-based hovering are shown in Fig. 4. Parts of the image processing is done at the Ground Station laptop computer.

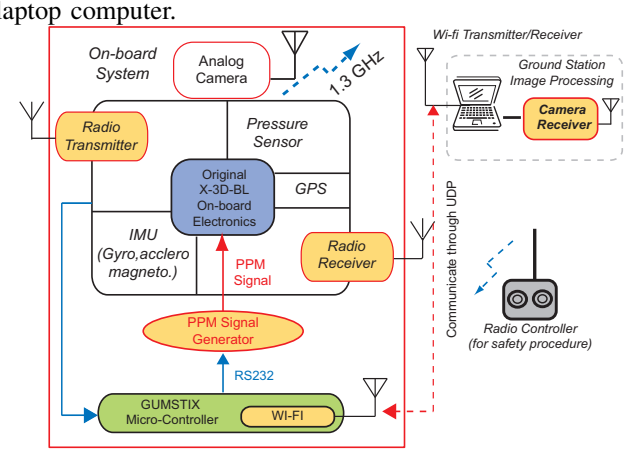

Fig. 2. Ground-station and on-board MAV embedded communication system,sensors and camera setup

III. Image Algorithm for Object Tracking

We applied three approaches for the object tracking algorithm which are (1) Integral Image algorithm (Color-based) and (2)Optic Flow based algorithm (Feature-based) and also (3)Combination between (1) and (2).

A. Integral Image

An integral image is a tool that can be used whenever we have a function from pixels to real numbers (for instance, pixel intensity). Our algorithm compute the sum of this function over a rectangular region of the image.Without an integral image, the sum can be computed in linear time per rectangle by calculating the value of the function for each pixel individually. Integral images, have been applied for example in face detection in images [14], and stereo correspondence [15],[16]. Research on vision-only aircraft control by Proctor *et. al* [17] has also applied the same. In our proposed image algorithm, in order to search the target, we need to compute the sum over multiple overlapping rectangular windows, thus using integral image would reduce the computing time significantly.

B. Probability distributing chart

After the target object is selected,the image algorithm will then extract the color of the object selected and threshold it by using the color probability distribution of the whole image.

This is achieved by converting the original color image to HSV space. The target object histogram is then computed by counting the number of pixel in a region of interest that have a given color. The colors are quantized into bins. This operation allows similar color values to be clustered as a single bin. The quantization into bins reduces the memory and computational requirements.

Back-projection is then used to associate the pixel values in the image with the value of the corresponding histogram bin. This operation replaces the pixel values of the input image with the value of corresponding bin of the histogram. The value of each pixel in the probability image represents the probability that the pixel belongs to the object of interest.

C. Integral Image Based-Tracking

To track the object based on color using integral image, first the sum of the (color)intensities of the selected target is calculated based on the probability distribution image. Then the ratio of the intensity and object size is taken as reference. The search area would be limited to an area of a radius value that changes dynamically according to the last searched object size. The ratio of image integral value of the search box and the size is compared to the reference, and the highest score will be the tracked object.To solve of problems of candidate redundancy located near each other, due to their similar intensities, the size of the search box would be increased/decreased on its four sides after selecting the best target. During this process, the box size will be updated, after comparing the intensity ratio with the reference. The outputted center of the target object,the x and y coordinates are input to the Kalman Filter, considering a linear uniform movement, to improve the algorithm efficiency.

The strategy has its drawbacks due to its simplicity, but we demonstrated in our experiment its good performance for visual servoing, and can be improved by combining with the OF based tracking.

D. Optical flow based object tracking

For optical flow based object tracking we applied the "pyramidal" Lucas-Kanade (LK) method, which is based on the original LK method [18]. Details of this method can be referred to [19], and comparisons between optical flow algorithms can be found in [20].

E. Object tracking assisted by optical flow based algorithm

In this research we also propose to assist the color-based tracking by integral image algorithm by the OF based feature tracking. First the target will be selected by the user from the screen. Both image integral based and optic flow based algorithm will run simultaneously and output the result of the target center.In the LK algorithm for feature tracking, the center of the target would be calculated as the mean of all features' coordinates.

When the object is inside the pre-defined area on the screen, the center of the target will be the average between the coordinate centers outputted by the integral image algorithm and OF based algorithm for simplicity. We pre-defined a border on the screen for the OF based algorithm to prevent errors when tracking features. The OF based algorithm will be initialized to select a "new target" at the center of the screen automatically when the original feature tracked by the OF based algorithm exits a pre-defined border.

The offset distance between the original target and the "new target" center is computed and stored. However the original target center will be taken only from the one computed by the integral image based algorithm, because the target or part of it is still in the field of view (FOV). When the original target completely exits the FOV, the image integral based algorithm will use the offset data from the OF based algorithm to recover the lost target. In other words, the OF based algorithm can be used as an odometer to compute the distance of the MAV and the center of the original target. When the original target is recovered and is inside the pre-defined border, the OF based algorithm will be re-initialized to track the original target, by selecting again the target tracked by the integral image algorithm.

The above strategy will work based on the premise that the background must have enough texture so that new features can be detected by the OF based algorithm. The same applies when re-initializing the OF based algorithm to the center of the screen

Fig. 3 summarizes the combined strategy of using OF based algorithm and the integral image based algorithm explained above.

F. Post image processing relative position estimation

Data from the image processing in pixels is sent to the gumstix micro-controller via wifi. The data from the image frame is then transformed to the Camera frame using the perspecting imaging model as stated in the equations below, where x_i,y_i and x_c,y_c,z_c are the coordinates on the image frame and camera frame respectively, and f is the camera focal length.

$$\frac{x_i}{f} = \frac{x_c}{z_c} \tag{1a}$$

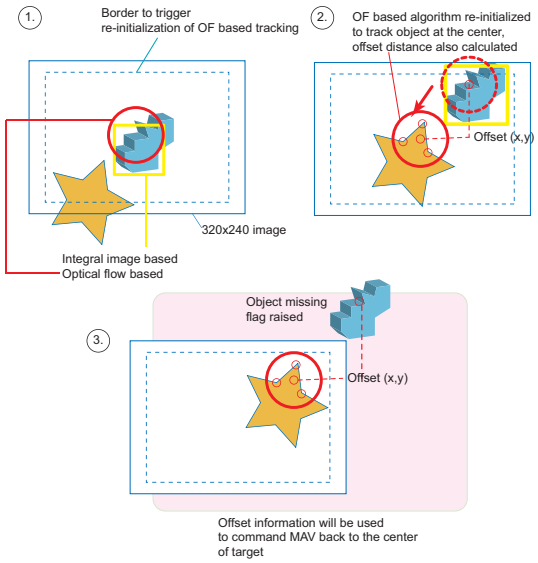

Fig. 3. Using optical flow based algorithm to recover target when it is out of the camera field of view

$$\frac{y_i}{f} = \frac{y_c}{z_c} \tag{1b}$$

The height in the camera frame z_c can be calculated if the size of the target object, S is known in beforehand using the equation below.s_i is the size in the image frame in $pixel \times pixel$.

$$\frac{f}{z_c} = \sqrt{\frac{s_i}{S}} \tag{1c}$$

However the size computation is not so accurate. Furthermore, when tracking some object like a car, the real size of the target is not known. Thus, for accurate height estimation, the altitude controller relied on the Pressure Sensor (PS) data z_h.

$$z_c = z_h \tag{1d}$$

Considering the rotations of the body frame, taken from the IMU unit, the coordinates in the camera frame x_c,y_c and z_c are transformed to the world frame as below,

$$\begin{bmatrix} x_w & y_w & z_w \end{bmatrix} = R \begin{bmatrix} x_c & y_c & z_c \end{bmatrix} \tag{2a}$$

$$R = \begin{pmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{pmatrix} \tag{2b}$$

The main objective for this transformation is to compensate the target displacements in the image that are due to MAV/camera rotation. Now, the computed position vector (x_w, y_w, z_w) represents the relative distances between the target and the MAV. The considered world frame has its origin at the object center, with the X-axis pointing to the North and the Y-axis pointing to the East.

In order to compute the MAV translational velocities and improve the position estimation, we have fused vision-based measurements with the Inertial Navigation System (INS), i.e., accelerations. Second order kinematic Kalman filters were used. The state vector implemented in the calculation is shown below, where $a_{x_{m_k}}$,$a_{y_{m_k}}$ denote acceleration data for each axis in time k.

$$\begin{bmatrix} x_{k+1} \\ v_{x_{k+1}} \\ y_{k+1} \\ v_{y_{k+1}} \end{bmatrix} = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ v_{x_k} \\ y_k \\ v_{y_k} \end{bmatrix} + \begin{bmatrix} \frac{\Delta t^2}{2} & 0 & 0 & 0 \\ 0 & \Delta t & 0 & 0 \\ 0 & 0 & \frac{\Delta t^2}{2} & 0 \\ 0 & 0 & 0 & \Delta t \end{bmatrix} \begin{bmatrix} a_{x_{m_k}} \\ a_{x_{m_k}} \\ a_{y_{m_k}} \\ a_{y_{m_k}} \end{bmatrix} \tag{3a}$$

The measurement vector, denoted by (x_{m_k}, y_{m_k}) includes the position computed in (2a), i.e., (x_w, y_w). The observation model is given by:

$$\begin{bmatrix} x_{m_k} \\ y_{m_k} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_k \\ v_{x_k} \\ y_k \\ v_{y_k} \end{bmatrix} \tag{3b}$$

IV. NONLINEAR VISION-BASED CONTROLLER

Generally, it is difficult to design a reliable autopilot for Rotorcraft UAVs (RUAVs), since the helicopter dynamics is inherently unstable, highly nonlinear, and coupled in inner-axis. The vision system, described in Section III, combined with the Inertial Navigation System (INS) provide the MAV position and velocity with respect to the target. Therefore, the vision-based control for object tracking can be formulated as a stabilization problem.

For the vision-based controller, we proposed a hierarchical nonlinear controller described in [21] which is based on nonlinear model of the RUAV dynamics, thereby considering system nonlinearities and coupling.

V. EXPERIMENTAL RESULTS

We have conducted two experiments to test the performance of the vision algorithm with combined flight control. These experiments were performed in the Chiba University, outdoor athletic field.

The first test was to demonstrate and validate the vision based hovering algorithm performance, and we have shown that as long as the battery that powered the MAV is available, the air-vehicle can hover stably on the object of interest. Fig. 4 shows the object tracking result plotted in the screen. The target is marked by the center dot within the blue circle of dots, which represents the search area. The white box and green box are the tracking results of the raw data and after Kalman filtering respectively.

The results obtained for the first test are presented in Fig. 5. In this test the MAV is controlled manually above a target, and after a target has been chosen, it is switched to vision-based autonomous control. In these experiments, GPS data are only used for comparison with the image data, and not for control. From the results the MAV position and height obtained during the autonomous hovering mode in Fig. 5. The results showed that the MAV succeeded to hover autonomously above the target for about eight minutes. This is a good performance for this MAV category which is

relying on vision for relative position estimation. The control accuracy is about 2 meters. However, higher accuracy can be achieved by adjusting the controller gains. The altitude controller also achieved very good performance since the desired height has been maintained for about 8 minutes. Note that the hovering time is limited by the battery life and not by the vision and control system.

It is observed that the vision/INS estimates are more accurate than the GPS/INS measurements for horizontal position and velocity in Fig. 5. Moreover, the GPS/INS data have a time delay of about 0.6 seconds compared to the vision/INS data.

The inner-loop performance can be evaluated by the MAV orientation in the figure. The desired angles are tracked with high accuracy. This proves the effectiveness of the nonlinear controller.

(a) (b)

Fig. 4. (a)Image from on-board camera when hovering over target, top: color (integral image),below: feature (OF) based tracking; (b)Ground control station, with map location of interest to navigate MAV by waypoints

The second test was to demonstrate the feasibility of using our system to assist a search and rescue like operation which is our research motivation. In this test the MAV is controlled autonomously by giving a GPS reference on the screen, by selecting a location of interest on an aerial view map of the area uploaded to the Ground Control Station (GCS). After arriving to the directed GPS location, the MAV will hover autonomously. The operator will select an object of interest from the video image streamed to the GCS. To get the MAV hover on the object of interest, the MAV control is switched to vision based control and the air-vehicle is hovered on the object of interest. It is also possible to land the vehicle near the target as shown in the result of our experiment. All these can be achieved by giving the commands solely from the GCS. The GCS with the uploaded map is shown in Fig. 4 (b).

The results of the second test are displayed in Fig. 6. Note that the results displayed include those of during GPS-based control flight mode to the vicinity of location of interest and after switching to vision-based hovering flight mode. The switch to autonomous flight is also shown to indicate that autonomous flight. After about 56 seconds of the autonomous GPS-based flight, the flight mode is switched to vision-based flight. The reference positions of x and y can also be referred, indicating when the flight is switched between the two modes. During hovering using vision-based control mode, the reference position and velocity of x and y are set to zero. We can see the position error in both axis are between 1 to 2 meters. The velocity error also showed relatively small error between -0.5 to 0.5 m/s. Note that the height is not controlled by vision-based servoing, but instead, data fusion of pressure sensor and INS is used as described earlier. The MAV started to descend after 87 seconds of flight after being commanded from the GCS. This is indicated by the Z-position data of Fig.6. Note that the data from pressure sensor is less accurate, as the height displayed is around 2 meters even though the air-vehicle is already on the ground. However, we successfully landed the MAV near the target after sending landing command from the GCS. Note that during landing vision is not included in the control loop. The performance of the inner loop controller is consistent with the results from the first experiment as shown by the MAV orientation results where the angles are tracked with high accuracy.The video of the first and second test can be seen in the internet links [22] and [23] respectively.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented a vision and control system that can be potentially deployed for search and rescue operations. This is based on the experimental results for hovering an MAV on selected target, in an outdoor environment. Data from flight tests showed that our vision system performs very well, compared to GPS in terms of accuracy and time delay. Indeed, we have succeeded to hover on a target more than 8 minutes (battery life dependent) with very good performance. Furthermore, we have eliminated the need of pre-learning or template approaches for object detection. We have also demonstrated a combination of the visual-based control of the air-vehicle with GPS-based navigation and verified the feasibility for the system to conduct search and rescue operations.

REFERENCES

[1] I. C. M. L. J. P. Campoy, J.F. Correa, "Computer vision onboard uavs for civilian tasks," *Journal Intell Robot Syst*, no. 54, pp. 105–135, 2009.

[2] P. C. G. S. S. Luis Mejías, Srikanth Saripalli, "Visual servoing of an autonomous helicopter in urban areas using feature tracking," *IEEE Transactions on Robotics*, vol. 23, no. 3/4, pp. 185–199, 2006.

[3] K. V. I.K. Nikolos, N.C. Tsourveloudis, "Evolutionary algorithm based path planning for multiple uav cooperation." Springer.

[4] ——, "Multi-uav experiments: application fo forest fires," *Multiple Heterogeneous Unmanned Aerial Vehicles,Springer Tracts in Advanced Robotics*.

[5] M. K. A. Puri, K.P. Valavanis, "Statistical profile generation for traffic monitoring using real-time uav based video data."

[6] E. Altug, J. Ostrowski, and R. Mahony, "Control of a quadrotor helicopter using visual feedback," May 2002.

[7] O. Bourquardez, R. Mahony, T. Hamel, and F. Chaumette, "Stability and performance of image based visual servo control using first order spherical image moments," October 2006.

[8] S. G. Fowers, D.-J. Lee, B. J. Tippetts, K. D. Lillywhite, A. W. Dennis, and J. K. Archibald, "Vision aided stabilization and the development of a quad-rotor micro uav," June 2007.

[9] S. Saripalli, J. Montgomery, and G. Sukhtame, "Visually-guided landing of an unmanned aerial vehicle," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 3, pp. 371–381, June 2003.

[10] S. Saripalli, G. Sukhtame, L. Mejías, and C. Cervera, "Detection and tracking of external features in an urban environment using an autonomous helicopter," 2005.

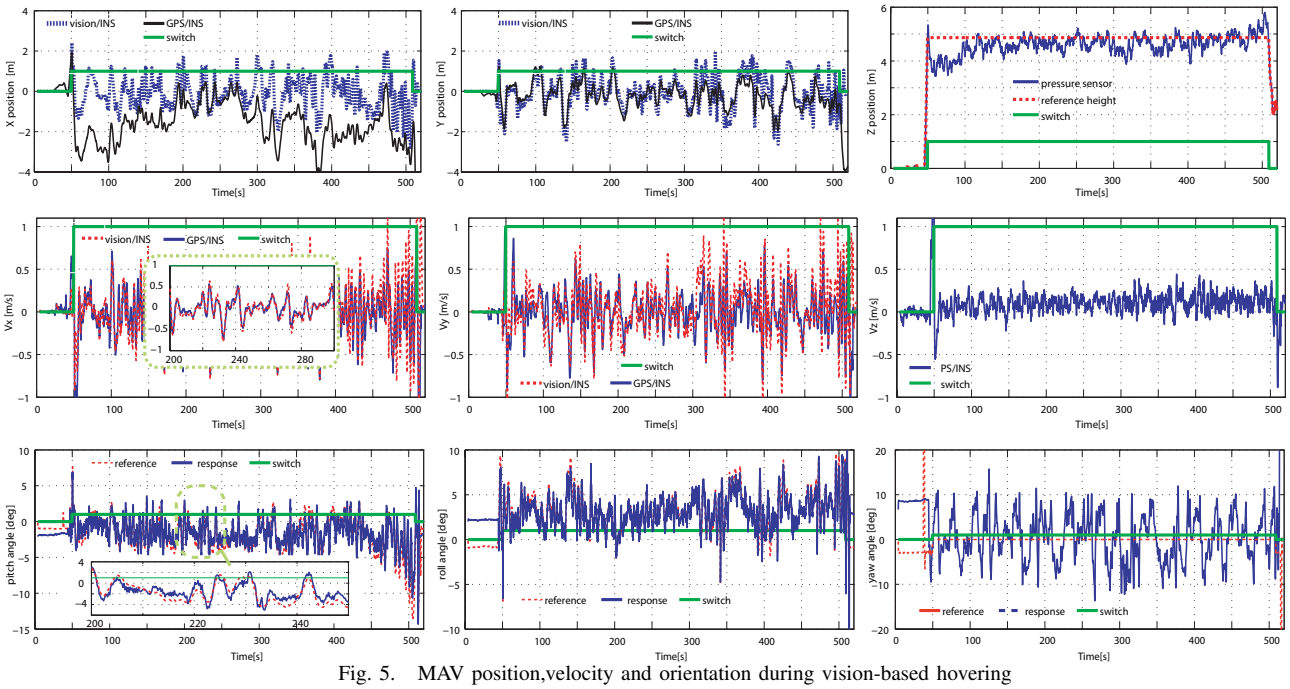

Fig. 5. MAV position,velocity and orientation during vision-based hovering

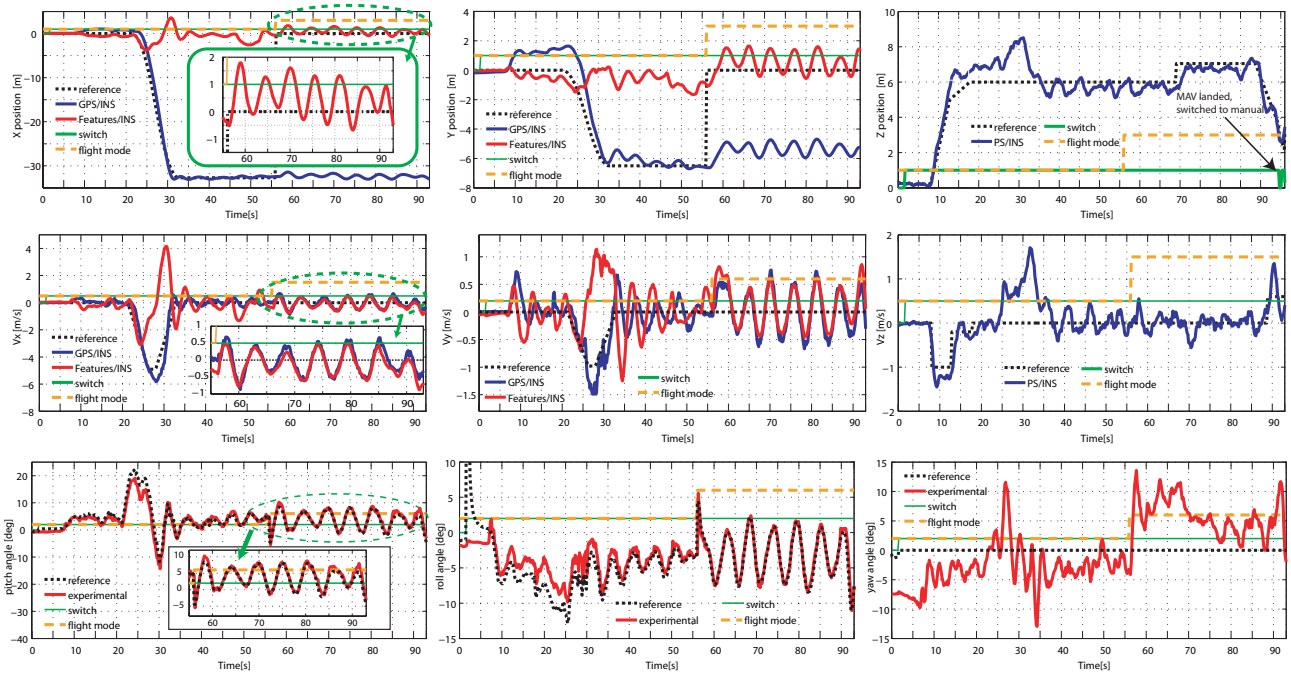

Fig. 6. MAV position,velocity and orientation during search and rescue scenario experiment

[11] S. Hrabar and G. Sukhatme, "Omnidirectional vision for an autonomous helicopter," May 2003, pp. 558–563.

[12] J. S. Jang and D. Liccardo, "Automation of small uavs using a low cost mems sensor and embedded computing platform," in *25th Digital Avionics Systems Conference, 2006 IEEE/AIAA*, Portland, USA, October 2006, pp. 1–9.

[13] [Online]. Available: http://sourceforge.net/project/mnav

[14] P. Viola and M. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[15] O. Veksler, "Fast variable window for stereo correspondence using integral image," 2003, pp. 556–661.

[16] Z. Yu, K. Nonami, J. Shin, , and D. Celestino, "3d vision based landing control of a small scale autonomous helicopter," *International Journal of Advanced Robotic Systems*, vol. 4, no. 1, pp. 51–56, 2007.

[17] A. Proctor, E. Johnson, and T. Apker, "Vision-only control and guidance for aircraft," *Journal of Field Robotics*, vol. 23, no. 10, 2006.

[18] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision."

[19] O. S. C. V. L. OpenCV, "http://www.intel.com/research/mrl/research/opencv."

[20] S. J. Barron, D. Fleet, "Performance of optical flow techniques," *International Journal of Computer Vision*, vol. 12, no. 1, pp. 43–77, 1994.

[21] F. Kendoul, I. Fantoni, and R. Lozano, "Asymptotic stability of hierarchical inner-outer loop-based flight controllers," in *Proceedings of the 17th IFAC World Congress*, Seoul, Korea, July 6-11 2008, pp. 1741–1746.

[22] [Online]. Available: http://www.youtube.com/watch?v=RrSfabBgHiM

[23] [Online]. Available: http://www.youtube.com/watch?v=_NqE9ZOcgIg