

Autonomous Hovering of a Vision/IMU Guided Quadrotor *

Tianguang Zhang, Ye Kang, Markus Achtelik, Kolja Kühnlenz and Martin Buss

Institute of Automatic Control Engineering (LSR)

Technische Universität München

D-80290 Munich, Germany

tg.zhang@ieee.org, kangye.kangye@gmail.com, markus@achtelik.net, {kolja.kuehnlenz, m.buss}@ieee.org

Abstract—We describe a multi-sensory control architecture for hovering of a mini four-rotor unmanned aerial vehicle (UAV) over specified markers as an external reference for pose estimation utilizing two on-board sensors: a tiny single camera and Inertial Measurement Units. A high-speed pose estimation approach based on visual information is presented. A closed-loop system is implemented using PID pose-controllers. The control performance is improved by integration of IMU measurement. Results of a real-time experiment are presented.

Index Terms—Vision system, pose estimation, multiple sensing system, UAV/MAV control.

I. INTRODUCTION

In recent years, unmanned aerial vehicles (UAV) has become a major research focus, since they can extend our capability in a variety of areas. A significant challenge in developing UAV is to extract and fuse the useful information in a robust manner and to provide stable flight and an accurate navigation. Strong evidence from biological systems indicates that multiple sensory modalities, e.g. vision, touch and balance etc., are used to guide the movement. Taking the blowfly as example, vision and inertial sensing are critically important to be fused for localization and motion estimation. The individual advantages of these modalities beneficially complement each other [1].

The goal of this work is to control a quadrotor (Fig. 1) using high-frequency feedback of a single on-board camera and measurements of an Inertial Measurement Unit (IMU), so that it can hover stably at a desired position. Some works have been conducted using a vision system for the pose estimation and control of the quadrotor. The VICON tracking system is used in [2] to get the real-time position and orientation of the quadrotor. The whole tracking system is very expensive, not mobile and needs much effort for installation and calibration. Various control techniques are available to find a good approach for mini UAV using on-board multi-sensory information [3] [4]. A novel two-camera method for estimating the full six-degrees-of-freedom of the Quadrotor is proposed in [5]. One camera is mounted in the quadrotor, while the other one is located on the ground. The

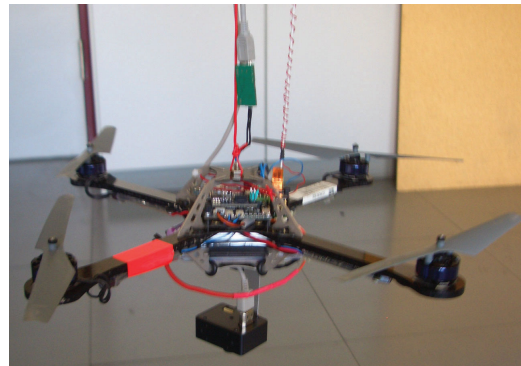


Fig. 1. Quadrotor with camera

feature detection applied in this work is sensitive to lighting conditions and therefore not reliable. In [6] a monocular on-board camera is set to measure the specified moire patterns to get the relative position and orientation of the quadrotor. But the pattern is complex and the update rate is slow. In [4], besides IMU and vision sensor data, several sonars are also used for altitude estimation and obstacle avoidance. The dynamics and the control of quadrotors are analyzed in [7] [8] for multi-agent systems.

In this paper a quadrotor system is described, which can hover autonomously at a desired position using a single on-board camera and IMU. The on-board camera is set to observe the ground in order to compute the relative position and orientation between the quadrotor and the markers on the ground. Using this information, both the position and yaw angle control are implemented based on PID controllers. All the concepts are verified in a real-time experiment.

The remainder of this paper is organized as follows: In Section II, the hardware configuration and software architecture are introduced. Then, the vision based pose estimation concept is presented. The controller design and setting are introduced in Section IV. Experimental results are shown in Section V. Conclusions and future work are given in Section VI.

II. SYSTEM OVERVIEW

The mechanical structure of a quadrotor is simple, which is lifted and propelled by four fixed-pitch rotors without swashplate. The quadrotor is omnidirectional, dynamically

* This work is supported in part within the DFG excellence initiative research cluster *Cognition for Technical Systems – CoTeSys*, see also www.cotesys.org and the Bernstein Center for Computational Neuroscience Munich, see also www.bccn-munich.de.

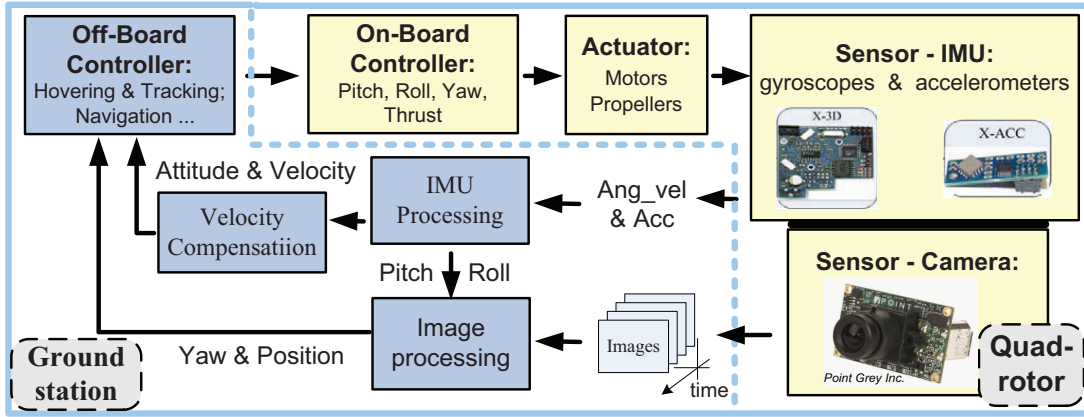


Fig. 2. System overview

elegant, and has almost no constraints on its motion, so it is a good test-bed. The Control of UAV during autonomous flight relies on knowledge of the variables such as position, velocity and orientation. The inertial sensors can measure the axial accelerations and rotating velocities of the quadrotor. But drift of sensor data leads to errors during integration, making a steadily accurate estimation of the absolute pose impossible. A vision system can be utilized to provide accurate position and orientation information.

The entire system (Fig.2) consists of two major parts: 1) the quadrotor itself, consisting of onboard controller, rotors and the multiple sensors e.g. camera and IMU; 2) the ground-station that receives and processes the sensor data, computes the status of the quadrotor, executes the control laws and sends the commands back to the quadrotor.

A. Hardware Description

1) *Quadrotor*: The quadrotor Hummingbird from Ascending Technologies [9] is chosen as the hardware platform for this work. It can offer a 1k Hz control frequency and motor update rate, which enables fast response to changes in the environment. The configuration of the quadrotor is basically the same as described in [2]. Fig. 3 shows the basic schema of the quadrotor. Two pairs of rotors spin clockwise and anticlockwise respectively. They are directly driven by high-torque DC brushless motors that are electronically commutated by optimized controllers to achieve fast responses. The flying motion of a quadrotor is determined by the rotational speed of the four motors. If the rotating velocities of all the four motors are increased at the same amount, the quadrotor will fly upwards. When the left motor is faster than the right one, the quadrotor will tilt around its x axis and fly rightwards. The yaw rotation is caused by the difference between the angular momentum generated by these two pairs of rotors. Therefore, the pitch, roll, yaw angles and thrust can be utilized as the control variables for the on-board control of the quadrotor.

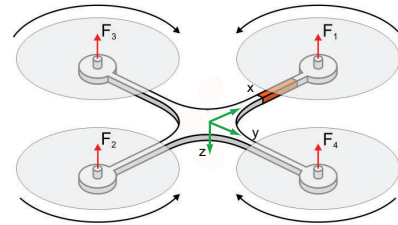


Fig. 3. Quadrotor schema

The inertial sensors on the quadrotor consist of three orthogonal gyroscopes providing the angular velocity for each rotation axis and three orthogonal accelerometers providing the acceleration for each translation axis. The absolute roll- and pitch angles are estimated on-board by fusing the acceleration vector and angular speeds. A relative yaw-angle is estimated by integrating the angular speed measured by the yaw-gyros. With respect to the limited payloads of the quadrotor, a light “Firefly MV” from Point Grey Research Inc. is selected as the on-board camera. The key parameters of the camera are listed as follows:

TABLE I
KEY PARAMETERS OF THE CAMERA

Specification	Description
Mass	14 gram (with microlens)
Dimensions	25 mm × 40 mm
Image	Black-white
Resolution	320 × 240 pixel
Frame Rate	60 FPS
Focal distance	3.8 mm
Field of view	56° × 38°
Interfaces	6-pin IEEE-1394

2) *Ground Station and Communications*: An off-board PC (AMD Athlon 5200+, 2GB RAM) is used for data processing, state estimation and execution of the control laws in our application. Because of the high resolution and frame rate

of the images, the firewire cable (IEEE 1394) is used to send the images captured by the on-board camera to PC. The quadrotor is equipped with XBeePro modules from MaxStream / Digi [10], that transmits IMU data and receives control commands at a rate of 100 Hz wirelessly.

B. Software Architecture

The main software part on PC consists of three blocks, namely calculation of IMU, image processing and controller. They will be described in detail in Sec. III and Sec. IV. Furthermore, there are two other blocks in the whole software structure: a GUI block and a communication block. As illustrated in Fig. 4, GUI is the graphical user interface between the user and the whole system, for example, for parameter tuning and graphic display of sensory data. Communication between the quadrotor and the ground station (PC) is facilitated by the communication block.

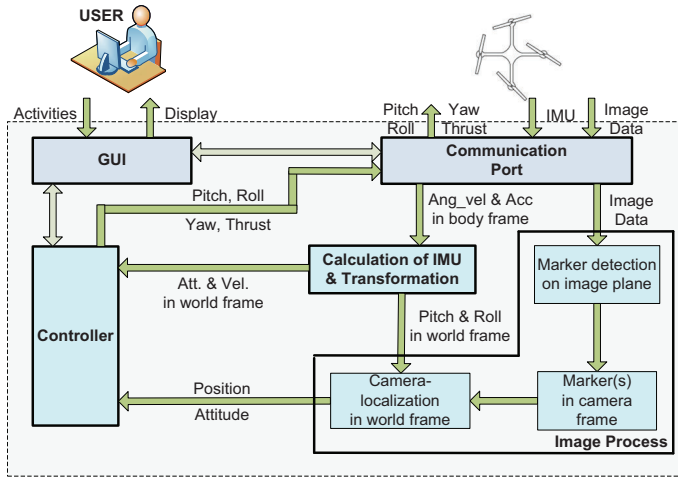


Fig. 4. Software architecture

III. VISION BASED POSE ESTIMATION

Aiming at a stable hovering, the pose of the quadrotor relative to an external reference should be determined at first. We estimate the pose based on visual information. In addition, a high-speed image processing also plays an essential role for the control performance. Therefore, the image processing should be as fast as possible.

A. Marker Design

Two circles with different radii are chosen as the external reference. However, as mentioned in Section II.A.1, the horizontal motion is carried out by roll and pitch angles, resulting in a tilted view of the on-board camera. Due to the limited field of view, the camera may lose the markers, even for a controlled hovering behavior. Therefore, four other markers are introduced and set around the two circles, as

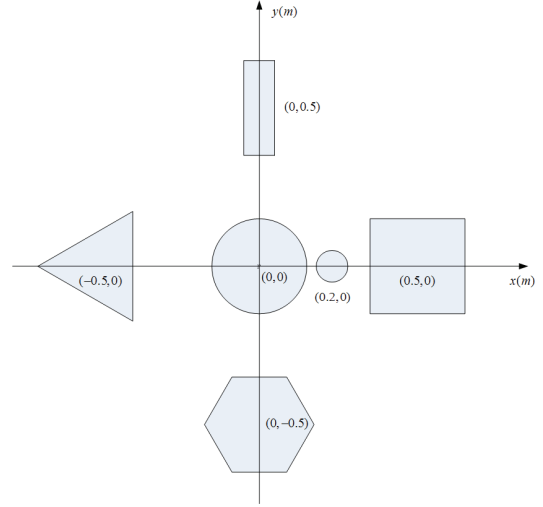


Fig. 5. Layout of the markers

illustrated in Fig. 5, so that the camera can find at least one marker in every image frame.

The center of the bigger circle is chosen as the origin of the reference frame, where x-direction points to the center of the smaller circle. Furthermore, the sizes and the positions of the markers are known, shown in table II.

TABLE II
DESCRIPTION OF MARKERS

Number	Name	Size	Position
0	Circle 1	radius = 8cm	(0,0)m
1	Square	16cm × 16cm	(0.5,0)m
2	Rectangle	19cm × 5.5cm	(0,0.5)m
3	Triangle	length of side = 17.5cm	(-0.5,0)m
4	Hexagon	length of side = 20cm	(0,-0.5)m
5	Circle 2	radius = 4cm	(0.2,0)m

B. Marker Detection in 2D Image

To compute the relative pose of the quadrotor, we should distinguish the types of the markers and determine their position in 2D input images. Since the on-board camera only provides black/white images, the geometric shape provides the main features to differentiate the markers.

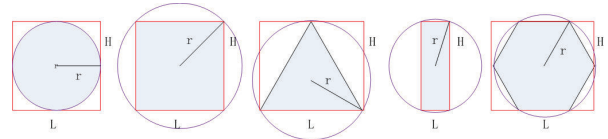


Fig. 7. The geometric features of the markers

A minimal surrounding rectangle and circle around each marker are to be found. For each marker, a feature vector \mathbf{f} is defined as follows:

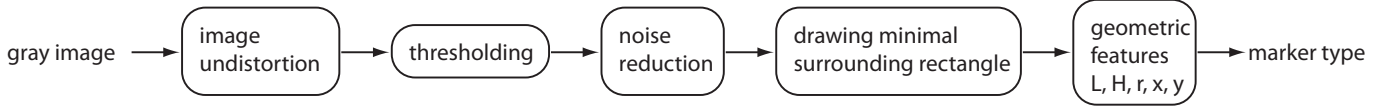


Fig. 6. Image processing procedure

$$\mathbf{f}_i = (L_i, H_i, r_i, x_i, y_i)^T, \quad (1)$$

where $i = 0, 1, 2, 3, 4, 5$. L_i is the length of the minimal surrounding rectangle. H_i is the height of the minimal surrounding rectangle. r_i is the radius of the minimal surrounding circle. (x_i, y_i) is the position of the center of the minimal surrounding circle in 2D input images.

The constrains for the marker discrimination are listed as followings:

- Circle

$$\begin{aligned} 0.8 < L/H < 1.2 \\ L * H > \pi * r^2 \\ \pi * r^2 - S_{circle} < 0.5 * r^2 \\ 40 < x_i < 600, 40 < y_i < 440 \end{aligned} \quad (2)$$

- Square

$$\begin{aligned} 0.8 < L/H < 1.2 \\ L * H - S_{square} < 0.2 * L * H \\ \pi * r^2 - L * H < 0.5 * r^2 \\ 40 < x_i < 600, 40 < y_i < 440 \end{aligned} \quad (3)$$

- Rectangle

$$\begin{aligned} L/H > 2 \text{ or } H/L > 2 \\ \pi * r^2 - L * H > 0.2 * r^2 \\ L * H - S_{rectangle} < 0.2 * L * H \\ 40 < x_i < 600, 40 < y_i < 440 \end{aligned} \quad (4)$$

- Triangle

$$\begin{aligned} L * H > 1.5 * S_{triangle} \\ L * H < \pi * r^2 \\ 40 < x_i < 600, 40 < y_i < 440 \end{aligned} \quad (5)$$

- Hexagon

$$\begin{aligned} L/H < 2 \text{ and } H/L < 2 \\ L * H > \pi * r^2 \\ \pi * r^2 - S_{hexagon} > 0 \\ 40 < x_i < 600, 40 < y_i < 440 \end{aligned} \quad (6)$$

S is the area of each marker.

Fig. 6 illustrates the procedure how the feature vector \mathbf{f}_i can be computed. Firstly, input images are undistorted using camera intrinsic parameters acquired from calibration process. Then, a threshold gray value is set to distinguish

the markers from the background. Dilation and erosion are executed to reduce noise. After that, the minimal surrounding rectangles and circles are drawn around residuary objects. The geometric features such as L_i, H_i, r_i as well as the marker position (x_i, y_i) are calculated. Using the predefined constraints, the marker type is determined.

C. Pose Estimation

Based on the geometric features and positions of the markers in 2D image, the pose of the quadrotor is computed. The position estimation is derived from the geometric relationship between the on-board camera and the markers, while the yaw angle is derived from the relative position of two arbitrary markers in the image. A scheme showing all the frames of reference is in Fig. 8.

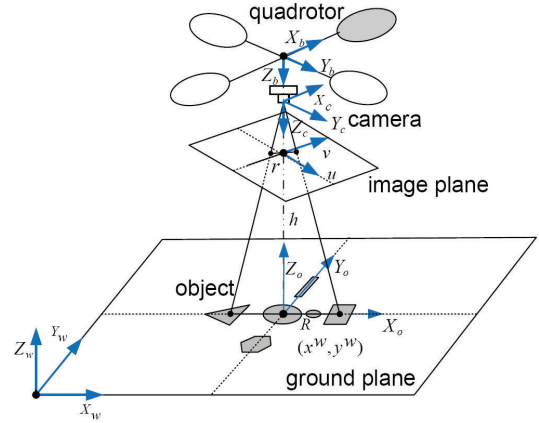


Fig. 8. Model of the camera system

Various frames of reference are defined as follows:

- The world frame S_w
- The object frame S_o is defined as in Fig. 5, where as z_o is the upward vertical axis.
- The body frame S_b has its origin in the center of the gravity of the quadrotor, where as x_b points out the nose of the airframe, y_b points out the right wing, and z_b points out the belly of the quadrotor.
- The camera frame S_c is coincident with the body frame of the quadrotor, while the origin is located at the lens center of the camera.
- The image frame S_i has its origin located at the principle point. The unit vector u along the image vertical direction directs the right rotor of the quadrotor, and the unit vector v points out the front rotor.

The distance between two markers is R with its projection r in 2D image. To decrease the error caused by image pixel discretization, we use the markers with maximum distance. Together with projection of the markers and the absolute pitch, roll value in the world frame provided by the IMU, relative position and orientation between the quadrotor in the world frame are computed using coordinate transformation and basic mathematical relations.

IV. SYSTEM CONTROL

A. Basic Controller Design

To achieve the desired position and orientation, four variables x , y , h and yaw of quadrotor must be controlled with the help of an external reference, the markers on the ground. Couplings between the four channels are ignored for simplification. As mentioned in II.A.1, the interface of the quadrotor offers the following input signals to control the quadrotor:

- Roll : roll-angle \implies acceleration in Y_b -direction
- Pitch : pitch-angle \implies acceleration in X_b -direction
- Thrust : rotating speed of motors \implies acceleration in Z_b -direction
- Yaw : the angular velocity around Z_b -axis

As no dynamic or aerobatic maneuvers should be performed, the control inputs are assumed to be proportional to the resulting accelerations. Four off-board PID controllers are implemented independently for x , y , h (height) and yaw.

Setting the yaw-controller is the easiest task: Since the angular velocity is already controlled by the on-board controller, so only a P-term is needed to compensate drift and integration errors. An I-term is also added to avoid constant errors. D-term can be zero or a small value.

For the position x/y , the controllers are the similar, but independent. The D-terms are calculated from the vision data since there are no sensors directly measure the speed. Time-delay due to bandwidth limitations limits P gain to ensure stability. I-term will reduce steady state errors. Because the visual field of the on-board camera is limited, it is also necessary to set a threshold value for the PID controller output, otherwise the markers will be out of the field of view, and then the quadrotor will be uncontrollable.

For the h -controller, the problems are basically the same but the I-term is needed to provide sufficient thrust for the hovering quadrotor with changing battery status and different payloads.

B. Improvements by Velocity Compensation Using IMU

To improve the control performance, the measurement provided by the on-board IMU is also integrated to the whole closed-loop control, which is represented as a velocity compensation block in Fig. 2. IMU provides inertial measurements at a higher rate of 100 Hz. By estimating the speed of the quadrotor with these sensors, we achieve better damping

of the oscillations at higher P-terms and faster responses to disturbances. The estimated velocity in world frame is considered as another differential term and weighted by a negative constant factor and added to the control commands sent to the quadrotor every 10 ms.

V. REAL-TIME EXPERIMENTAL RESULTS

To validate the control strategies and the derivative model of the quadrotor we simulated the whole system in MATLAB® 2008a based on simulation model from [11]. After simulations, it is also necessary to set up a safe real-time experiment environment for the quadrotor in order to verify the pose estimation algorithm and test all the control strategies without being damaged. A fishing rod is used to hold the quadrotor and the cable between camera and PC. When the quadrotor hovers at the desired height, the cables should be loose, so that all the six degrees of freedom are autonomously controlled without constraints in this experiment. A real-time implementation of hovering is accomplished with 60 Hz frequency for image processing at a resolution of 320×240 pixel and off-board system control.

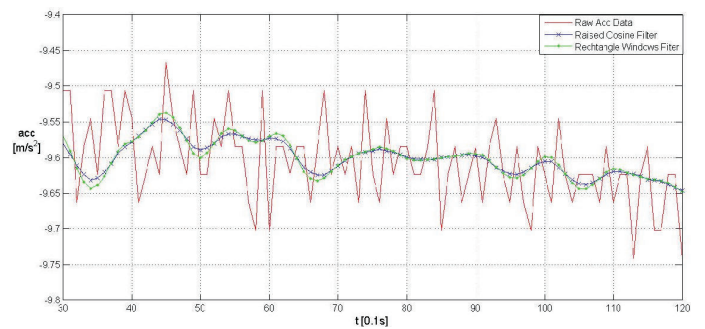


Fig. 9. Low-pass filter for IMU data

The IMU data are integral values and noisy. Therefore, a low-pass filter is essential for the raw IMU data before they are integrated. Fig. 9 shows the outputs of two low-pass filters compared with the raw IMU data. Both filters work at the sample frequency of 100 Hz. Although the outputs of the filters are rather similar, the raised cosine filter has a higher frequency response than the rectangle window low-pass filter. So the raised cosine low-pass filter is chosen for the IMU raw data.

Fig. 10 shows the position of the hovering quadrotor. The desired height for hovering is 1.2 m over the markers. The experiment results show that the quadrotor can hover around the set position with oscillations. The maximum errors in x/y -direction are about 0.2 m, while the maximum error in height is also about 0.2 m. Because the visual field of the on-board camera is limited, the quadrotor should not fly far away from the markers. The thrust forces generated by the four motors at the same rotation velocity are different because of the deformation of the four plastic rotors, so without adequate

control, the quadrotor will fly away from the markers very fast and be out of control.

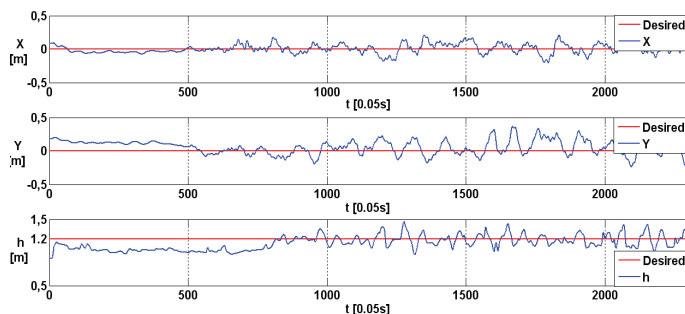


Fig. 10. Position of the hovering quadrotor

The flight of the quadrotor in z-direction is affected by the difference of the thrust forces and the gravity of the quadrotor. Since the thrust forces generated by the four motors are related with the battery voltage, there are some oscillations in the height direction.

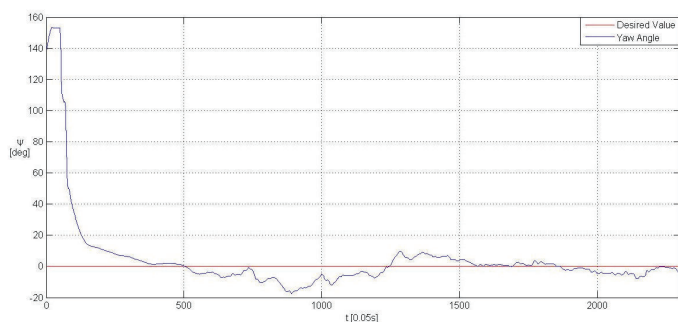


Fig. 11. Yaw angle of the hovering quadrotor

Fig. 11 shows the yaw angle of the quadrotor during the flight. The yaw angle is about 150° when the quadrotor takes off. In about 10 seconds the quadrotor can rotate to the set point hovering with the desired yaw angle.

The quadrotor oscillates around the set position, although the oscillations have already been reduced by using the low-pass filter and the velocity compensation. The video can be found on the web page ¹.

Evaluating the experimental results various challenges remain. Due to the limited camera field of view, the marker can be easily lost during horizontal flight by rotating the pitch and roll angles. Furthermore, the raw IMU measurements are very noisy. A small error can cause a large displacement because of the high-speed motion of the quadrotor. Time delay due to information transfer can render control unstable.

VI. CONCLUSIONS AND FUTURE WORK

The realization of a control architecture for autonomous hovering of a mini quadrotor is described in this paper,

¹<http://www.lsr.ei.tum.de/team/zhang/ICMA2009zhang.mpg>

which consists of marker design, image processing, IMU processing, pose estimation and control. A high-speed pose estimation based on visual pose information is implemented. Using pose estimation feedback and four PID controllers, a closed-loop system is implemented. The control performance is improved by integration of IMU measurement. An autonomous hovering at a desired altitude over the markers is evaluated in a real-time experiment.

Although there is still some oscillations during the hovering, the main contribution consists in system integration and experiments in real physical environments. Future works are concerned with multi-modal data fusion based on stochastic observers.

REFERENCES

- [1] P. Corke, J. Lobo, and J. Dias, "An introduction to inertial and visual sensing," *The International Journal of Robotics Research*, vol. 26, pp. 219–535, 2007.
- [2] D. Gurdan, J. Stumpf, M. Achtelik, K.-M. Doth, G. Hirzinger, and D. Rus, "Energy-efficient autonomous four-rotor flying robot controlled at 1 khz," in *Proceedings of the 2007 IEEE International Conference on Robotics and Automation(ICRA)*, Roma, Italy, Apr. 2007.
- [3] P. Corke, "An inertial and visual sensing system for a small autonomous helicopter," *Journal of Robotic Systems*, vol. 21, p. 4351, 2004.
- [4] S. Bouabdallah, "Design and control of quadrotors with application to autonomous flying," Ph.D. dissertation, Lausanne, EPFL, 2007.
- [5] E. Altug, J. P. Ostrowski, and C. J. Taylor, "Control of a quadrotor helicopter using dual cameravisual feedback," *The International Journal of Robotics Research*, vol. 24, pp. 329–341, 2005.
- [6] G. P. Tournier, M. Valenti, and J. P. How, "Estimation and control of a quadrotor vehicle using monocular vision and moire patterns," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2006.
- [7] G. Hoffmann, D. G. Rajnarayan, S. L. Waslander, D. Dostal, J. S. Jang, and C. J. Tomlin, "The stanford testbed of autonomous rotorcraft for multi-agent control," in *Digital Avionics System Conference*, 2004.
- [8] G. Hoffmann, H. Huang, S. L. Waslander, and C. J. Tomlin, "Quadrotor helicopter flight dynamics and control: Theory and experiment," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, 2007.
- [9] Ascending Technologies GmbH. <http://www.asctec.de>.
- [10] "Maxtream / digi," <http://www.digi.com>.
- [11] <http://www.et.byu.edu/groups/ece490quad/>, brigham Young University.