

Research Article

Visual Flight Control of a Quadrotor Using Bioinspired Motion Detector

Lei Zhang,¹ Tianguang Zhang,¹ Haiyan Wu,¹ Alexander Borst,² and Kolja Kühnlenz^{1,3}

¹Institute of Automatic Control Engineering (LSR), Technische Universität München, 80290 München, Germany

²Department of Systems and Computational Neurobiology, Max Planck Institute of Neurobiology, Am Klopferspitz 18, D-82152 Martinsried, Germany

³Institute for Advanced Study (IAS), Technische Universität München, 80290 München, Germany

Correspondence should be addressed to Lei Zhang, lei.zhang0918@gmail.com

Received 25 August 2011; Accepted 20 February 2012

Academic Editor: Farid Melgani

Copyright © 2012 Lei Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Motion detection in the fly is extremely fast with low computational requirements. Inspired from the fly's vision system, we focus on a real-time flight control on a miniquadrotor with fast visual feedback. In this work, an elaborated elementary motion detector (EMD) is utilized to detect local optical flow. Combined with novel receptive field templates, the yaw rate of the quadrotor is estimated through a lookup table established with this bioinspired visual sensor. A closed-loop control system with the feedback of yaw rate estimated by EMD is designed. With the motion of the other degrees of freedom stabilized by a camera tracking system, the yaw-rate of the quadrotor during hovering is controlled based on EMD feedback under real-world scenario. The control performance of the proposed approach is compared with that of conventional approach. The experimental results demonstrate the effectiveness of utilizing EMD for quadrotor control.

1. Introduction

Flying insects have tiny brains and mostly possess compound eyes which can get panoramic scene to provide an excellent flying performance. Comparing with state-of-the-art artificial visual sensors, the optics of compound eye provide very low spatial resolution. Nevertheless, the behavior of flying insects is mainly dominated by visual control. They use visual feedback to stabilize flight [1], control flight speed, [2] and measure self-motion [3]. On the other hand, highly accurate real-time stabilization and navigation of unmanned aerial vehicles (UAVs) or microaerial vehicles (MAVs) is becoming a major research interest, as these flying systems have significant value in surveillance, security, search, and rescue missions. Thus, the implementation of a bio-plausible computation for visual systems could be an accessible method to replace the traditional image processing algorithms in controlling flying robots such as a quadrotor.

Most of early applications using insect-inspired motion detector focus on motion detection tasks rather than velocity estimation. In robotics and automation applications, EMDs

are mainly used for a qualitative interpretation of video image sequence, to provide general motion information such as orientation and in front obstacles. In [4], a microflyer with an onboard lightweight camera is developed, which is able to fly indoor while avoiding obstacles by detecting certain changes in optic flow. The recent approach for the navigation in a corridor environment on an autonomous quadrotor by using optical flow integration is shown in [5]. Another example is a tethered optic flow-based helicopter that mimics insect behaviors such as taking off, cruise, and landing [6, 7], and in [8] the EMDs visual sensors were tested and characterized in field experiments under various lighting conditions. Numerous authors have pointed out that the Reichardt model, while sensitive to motion, does not measure velocity [9–11]. However, some efforts have been made, examining the possibility of velocity estimation tasks by introducing elaborated models [12, 13]. In [14], yaw rate estimates on a coaxial helicopter testbed are obtained using a matched filter approach which yet incorporates a virtual 3D environment in the control loop. Although a lot of work has been done on a simulation level or involving simulation

tools, further robotic applications with EMDs considering closed-loop velocity control are still to be investigated under real-world scenarios.

In this paper, a quadrotor system with bioinspired visual sensor is described. The novel image processing methods and the control laws are implemented in real-time experiments. The yaw rate control is totally based on the visual feedback of the on-board camera. The reference velocity value is provided by the on-board inertial measurement unit (IMU). For a 6-DOF (degrees of freedom) flying robot control, a tracking system of multicamera configuration is also utilized to achieve the altitude and attitude stabilization near hover. Due to the noise in real-world, the velocity estimation tasks would be more challenging. In this work, the approach of building an empirical lookup table from open-loop test results is introduced for this task. The Reichardt motion detector is modified which describes, at an algorithmic level, the process of local motion detection in flying behaviors. Certain patterns of receptive fields, which respond to particular optic flow, are utilized to estimate the global ego-motion through the environment. Another main issue which needs to be tackled carefully in this work is that the flying robot should be well stabilized during hovering. By multicamera tracking, the absolute position as well as the pose is determined from the positions of four on-board markers.

The remainder of this paper is organized as follows: in Section 2, we firstly introduce the bioinspired visual image processing methods used in this work. In Section 3, the 3D pose estimation using visual tracking system is described. The control strategy of the system as well as the software structure of algorithms is presented in Section 4. Then in Section 5, an overview of the whole experimental platform is illustrated. The control performance is also evaluated based on the experimental results in this section. Conclusions are given in Section 6, with directions on future works.

2. Bioinspired Image Processing

In this section, we introduce the essential part of this work: using biological models for yaw rate estimation of a quadrotor. The EMDs are utilized for this task. The whole methodology is introduced in detail. To achieve the yaw rate control, the system also requires accurate visual tracking for pose stabilization (Section 3) and efficient controllers (Section 4).

In an insect's perspective, motion information has to be computed from the changing retinal images by the nervous system [15]. For engineering applications, some properties of the biological visual system are converted into computational algorithms.

The elaborated EMD model used in this work is a modified model of the famous Reichardt motion detector [16]. The original Reichardt motion detector (Figure 1(a)) has only low-pass filters and two correlations. In this work, a temporal high-pass filter is added before the low-pass filter to obtain a simple response to step edges [17] (Figure 1(b)). The high-pass filters and low-pass filters in this model are all designed to be of first order.

In [12], a mathematical analysis of the original Reichardt motion detector is given regarding the response to different images (sinusoidal gratings as well as natural images). Without loss of generality, we firstly consider the response of this modified model to a moving natural image (which possesses energy at all spatial frequencies). Similar to the response of the simplified model [12], for this modified model, the output is

$$R = \int_0^{\infty} \frac{2\pi\tau_L f_s v \sin(2\pi f_s \Delta\phi)}{1 + (\tau_L/\tau_H)^2 + (1/2\pi f_s v \tau_H)^2 + (2\pi f_s v \tau_L)^2} P(f_s) df_s, \quad (1)$$

where $\Delta\phi$ is the angular displacement between the two vision sensors, f_s is the spatial frequency of the image input to the detector, v stands for the velocity of the moving image, τ_L and τ_H are the time constants of the low- and high-pass filters, respectively, and $P(f_s)$ represents the power spectral density. So according to (1), the local motion information is calculated.

To obtain a global ego-motion estimation, certain receptive fields of the motion-sensitive widefield neurons in the fly brain are applied. Considering the specified experimental scenario in this work, two novel templates of receptive fields for rotation detection are utilized (Figure 2), which are proposed in [17].

The algorithms for calculating the rotation global response can be described as (image size: length \times width; R_H : response of local horizontal motion; R_V : response of local vertical motion):

$$\begin{aligned} \text{Horizontal rotation} &= \sum_{i=1}^{\text{width}/2} \sum_{j=1}^{\text{length}} R_H - \sum_{i=\text{width}/2+1}^{\text{width}} \sum_{j=1}^{\text{length}} R_H, \\ \text{Vertical rotation} &= \sum_{i=1}^{\text{width}} \sum_{j=1}^{\text{length}/2} R_V - \sum_{i=1}^{\text{width}} \sum_{j=\text{length}/2+1}^{\text{length}} R_V. \end{aligned} \quad (2)$$

Now we examine the feasibility of using this model for velocity estimation tasks. In [12], two criteria are quantified for an accurate velocity estimation system: (1) image motion at a fixed velocity should always have approximately response; (2) at a given velocity, the response to motion should be unambiguous over certain range. In simulation, we find that by introducing this modified model, the response to a specific velocity of image motion can meet the two basic requirements at low velocities (above which the response output is ambiguous). Thus, for velocity estimation tasks, the motion velocity should be limited in a certain range due to essential property (bell-shaped response) of the Reichardt model. In order to reduce the brightness sensitivity, logarithmic transformation could be also applied (as the modified model in [17]). However, by doing this the discrimination of response is also highly reduced. That means, the response at a given velocity cannot differ significantly from the response at other velocities, which is not appropriate for quantifying velocity. Moreover,

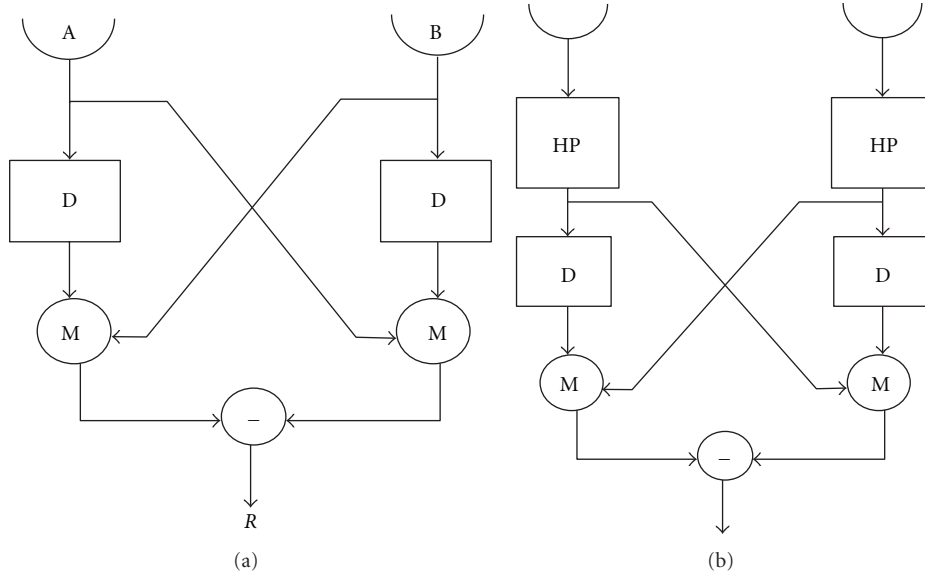


FIGURE 1: The simple Reichardt motion detector (a) and the elaborated EMD model (b). A and B are two visual signal inputs. The moving luminance signal is observed by this pair of visual sensors (such as the ommatidia of a fruit fly). The detector compares visual sensor inputs and generates a direction sensitive response R corresponding to the visual motion. D: delay block. Here it refers to low-pass filter; HP: High-pass filter; M: multiplication.

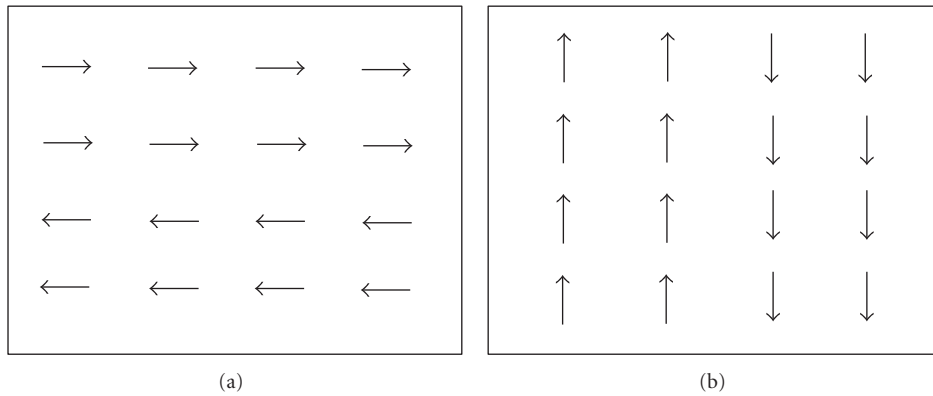


FIGURE 2: Rotation templates: horizontal (a) and vertical (b). The response of horizontal rotation is calculated by subtracting the upper part and lower part of the template of receptive fields, while the vertical rotation is detected by the difference between left part and right part of the global optical flow.

regarding the brightness sensitivity problem, a stable lighting condition is demanded in real-time experiments.

We firstly examine the open-loop characteristics of the system only for yaw rate estimation. The quadrotor is tethered in the air, with an on-board camera looking directly to the ground texture (the complete system is further introduced in Section 5). This scenario in the indoor environment involves a black-white chessboard ground texture. It is considered to be the most suitable scenario for detecting rotation motion of a flying robot. Compared to other forms of textures, the high image contrast can also help to improve the discrimination for quantifying velocity (due to the characteristics of the biological model itself). The quadrotor is rotated on horizontal level without control, and we can get the relationship between the response output and

the rotation velocity (yaw rate). A lookup table is then built. Due to the system noise and discrimination limitations of the experiments, the curve has some nonmonotonic regions. The polynomial minimum quadric method is used to fit the curve, (where d_i is a residual which is defined as the difference between the predicted value y_i and the actual value $f(x_i)$):

$$\sum_{i=1}^n d_i^2 = \sum_{i=1}^n [y_i - f(x_i)]^2 = \text{minimum}. \quad (3)$$

The experimental results of the open-loop characteristics are shown and further discussed in Section 5.

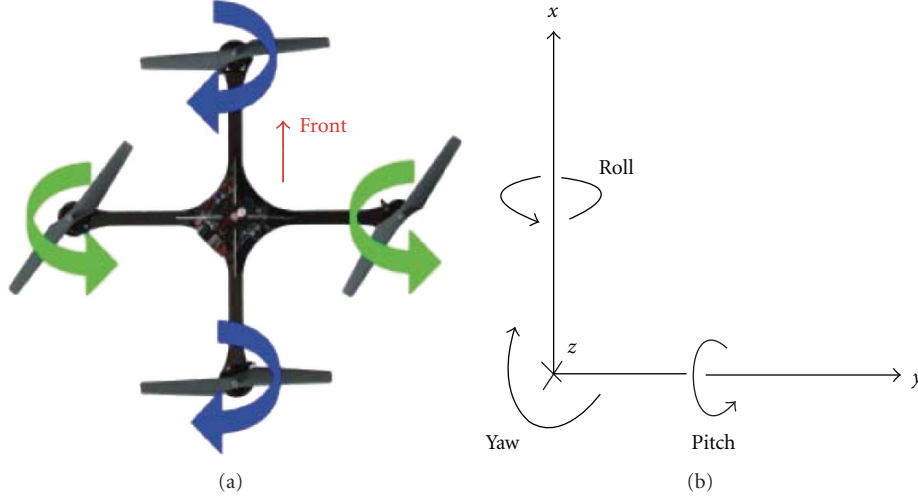


FIGURE 3: Quadrotor dynamics. (a) the overhead view of a quadrotor; (b) six degrees of freedom of the flying robot.

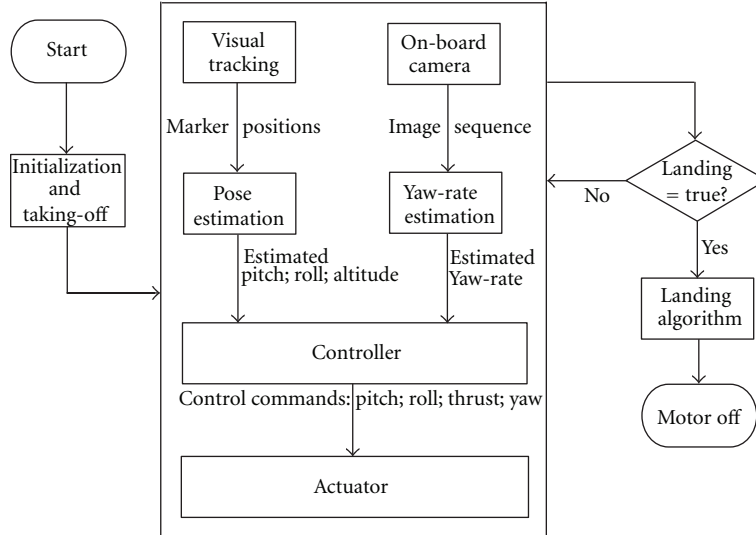


FIGURE 4: Software architecture.

3. Multicamera 3D Pose Estimation

Quadrotor is an underactuated vehicle. The 6 DOFs of the quadrotor are controlled by four motor inputs (pitch, roll, thrust, and yaw) by varying the lift forces and the torque balance through changing the rotating speed of the rotors (see Figure 3(a)). Yaw control is realized by tuning the differential speed between the two counterrotating rotor pairs. Increasing the rotating speed of all the four motors at the same amount will cause an upward movement of the quadrotor. When tuning the differential speed between the two motors of either single rotor pair, the quadrotor will fly sideways.

The work in this section is based on the former related work in [18]. In this work, we set up an indoor GPS system by using multicamera tracking instead of the former two-camera tracking. By tracking the four markers installed on

the axis of the quadrotor, the 3D position as well as the pose of the flying robot can be estimated. The experimental setup of 3D tracking is further introduced in Section 5. The frame of quadrotor dynamics is the same as the in Figure 3(b). We have the following definition:

Marker i position vector: $S_i = (x_i, y_i, z_i)^T$ ($i = 1, 2, 3, 4$).

Central point vector between two nonadjacent markers: $M_j = (x_{M_j}, y_{M_j}, z_{M_j})^T$ ($j = 1, 2$);

Estimated central point vector of the quadrotor: $M_q = (x_M, y_M, z_M)^T$;

Orientation of marker i : $V_i = (x_{V_i}, y_{V_i}, z_{V_i})^T$ ($i = 1, 2, 3, 4$);

The counting of the markers is clockwise, while the first marker is on the main axis. For 3D pose control, the central point should be used as the reference position of the quadrotor. The central points of the distance between marker 1 and marker 3 as well as between marker 2 and

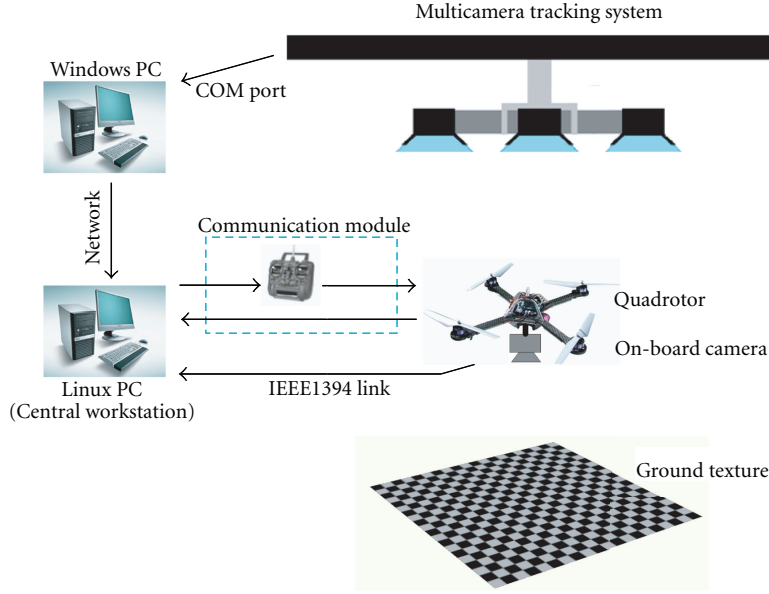


FIGURE 5: Experimental setup.

marker 4 are $M_1 = (1/2)(S_1 + S_3)$ and $M_2 = (1/2)(S_2 + S_4)$. In consequence of the marker's noise through the tracking system, the two central points in the two equations above are not identical. Thus, the central point of the quadrotor is $M_q = (1/2)(M_1 + M_2)$. The vectors between the central point and marker 1 for pitch as well as between the central point and marker 2 for roll are $V_i = S_i - M_q$ ($i = 1, 2$). The values of pitch θ , roll ϕ , and yaw ψ angles can be then calculated, and thus the 3D pose of the quadrotor can be estimated:

$$\theta = \text{sgn}(z_{v_1}) \arccos\left(\frac{\sqrt{x_{v_1}^2 + y_{v_1}^2}}{\sqrt{x_{v_1}^2 + y_{v_1}^2 + z_{v_1}^2}}\right), \quad (4)$$

$$\phi = \text{sgn}(z_{v_2}) \arccos\left(\frac{\sqrt{x_{v_2}^2 + y_{v_2}^2}}{\sqrt{x_{v_2}^2 + y_{v_2}^2 + z_{v_2}^2}}\right), \quad (5)$$

$$\psi = -\text{sgn}(y_{v_1}) \arccos\left(\frac{x_{v_1}}{\sqrt{x_{v_1}^2 + y_{v_1}^2}}\right). \quad (6)$$

4. Controller

At first the quadrotor should be regulated to hover in the air on horizontal plane with little shaking. That means, the stable state commands (u_p^s for pitch, u_r^s for roll, u_y^s for yaw, and u_t^s for thrust) should be adjusted firstly. Basing on these parameters, the control commands can be calculated next. For each controller, we have an output value (u_p^q for pitch, u_r^q for roll, u_y^q for yaw, and u_t^q for thrust) between -1 and 1 , which is then added with the corresponding stable state command. Since we only consider the rotation movement in this experiment, which means, the quadrotor is not always heading with the main axis towards X direction, the pitch

and roll commands should be adjusted in the ψ direction with a rotation matrix:

$$\begin{pmatrix} u_p \\ u_r \\ u_t \end{pmatrix} = \begin{bmatrix} \cos \psi & -\sin \psi & 0 & 0 \\ \sin \psi & \cos \psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} u_p^q \\ u_r^q \\ u_t^q \\ u_y^q \end{pmatrix} + \begin{pmatrix} u_p^s \\ u_r^s \\ u_t^s \\ u_y^s \end{pmatrix}. \quad (7)$$

We choose proportional-integral (PI) controller for the yaw rate control. The pitch, roll, and thrust commands are controlled by proportional-integral-derivative (PID) controllers:

$$\begin{aligned} u_y^q &= k_y^P(\psi_0 - \psi) + k_y^I \int (\psi_0 - \psi) dt, \\ u_p^q &= k_p^P(\theta_0 - \theta) + k_p^I \int (\theta_0 - \theta) dt + k_p^D \frac{d}{dt}(\theta_0 - \theta), \\ u_r^q &= k_r^P(\phi_0 - \phi) + k_r^I \int (\phi_0 - \phi) dt + k_r^D \frac{d}{dt}(\phi_0 - \phi), \\ u_t^q &= k_t^P(z_0 - z) + k_t^I \int (z_0 - z) dt + k_t^D \frac{d}{dt}(z_0 - z). \end{aligned} \quad (8)$$

In this experiment, the reference values θ_0 and ϕ_0 are set to zero. The desired altitude z_0 is 0.35 m near hover. The measured values θ and ϕ are calculated from the received data of the visual multicamera tracking system using (4) and (5), whereas ψ is searched out from a certain empirical lookup table using the response value, which is calculated by insect-inspired motion detectors. The yaw velocity can also be obtained from (6) by time derivative, which is, for the heading stabilization, used as a reference (ground truth). The closed-loop results will be shown and further discussed in Section 5.

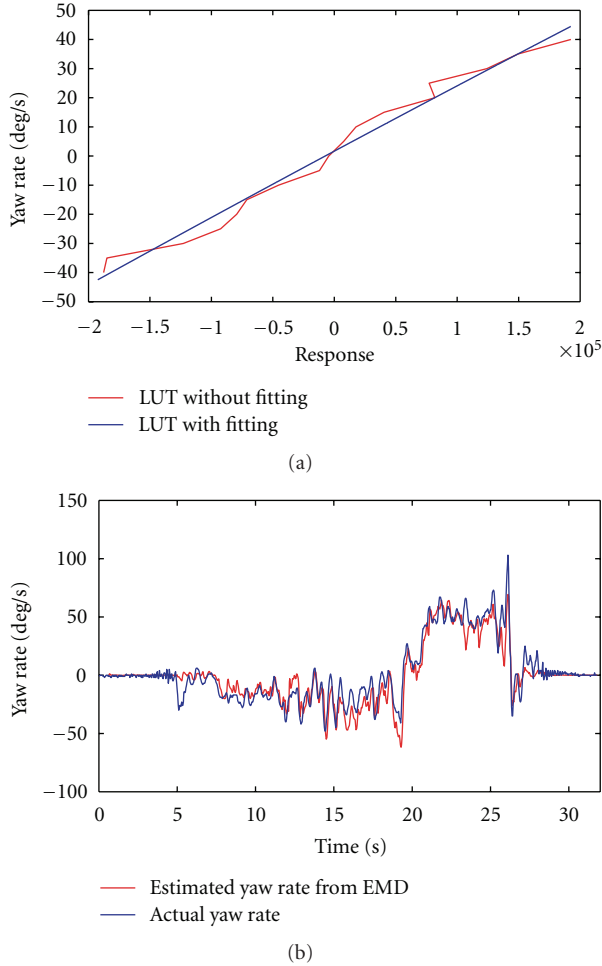


FIGURE 6: (a) lookup Table (LUT) in this work. This empirical LUT shows the relationship between yaw rate (in low speed) and visual response under the experimental environment; (b) open-loop characteristics for rotation motion near hover (without control).

The main loop in the whole software architecture (Figure 4) consists of two simultaneous processes: yaw rate control using on-board camera visual feedback and X, Y, and Z position/poses control using visual tracking system. A graphical user interface (GUI) is developed basing on Qt cross-platform application, which provides data visualization (e.g., 3D trajectory of the quadrotor, battery voltage and sensory data information), commands input and real-time online configuration of control parameters.

5. Experiments and Results

5.1. Experimental Setup. The whole experimental platform is shown in Figure 5. It mainly consists of a quadrotor testbed, off-board workstation, and video camera tracking system.

(1) *Quadrotor.* The miniquadrotor used in this work is a “Hummingbird” with an “AutoPilot” central control board from ascending technologies. It offers a 1 kHz control frequency and motor update rate, which guarantees fast

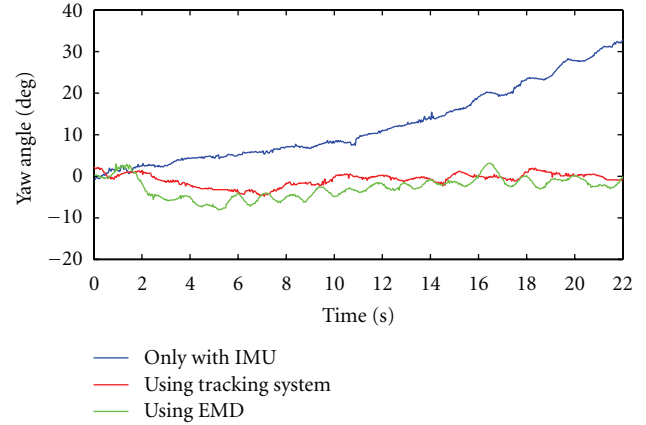


FIGURE 7: Heading stabilization.

response to changes in the environment. The size of the whole quadrotor testbed is 36.5 cm in diameter, and the four rotors (each with a propeller size of 19.8 cm) are directly driven by four high-torque DC brushless motors respectively. Powered by a state-of-the-art 3-cell 2100mAh lithium-polymer battery, the vehicle is able to hover up to 15 minutes (with about 120 g of payload in this work).

(2) *On-Board Camera.* Considering the limited payloads of the quadrotor, the PointGrey Firefly MV CMOS camera which has a light weight (14 g) and a tiny size (25 mm × 40 mm) is selected as the on-board camera. We choose a standard resolution of 640 × 480 (pixels), and the frame rate is 60 Hz. The camera is equipped with a 6 mm microlens, providing a viewing angle of 56 deg and 38 deg in the length and width directions, respectively. It uses a 5-pin USB 2.0 digital interface with a 480 Mb/s transfer rate and 8-bit raw Bayer data format (connected through IEEE 1394 to workstation). The camera is mounted under the base board of the quadrotor, looking directly down to the ground texture.

(3) *Workstation and Communication Module.* An off-board Linux PC (AMD Athlon 5200+; 2 GB RAM) is used for image data processing, 3D pose estimation and control law execution in this case. The quadrotor is equipped with XBeePro wireless communication module from MaxStream/Digi, which enables the data transmission from the on-board inertial measurement unit (IMU) and the control command reception (with R/C transmitter enabled) from workstation at a rate of 100 Hz.

(4) *Visual Tracking System and Marker Placement.* The tracking system VisualeyzeII VZ4000 from Phoenix Technologies Incorporated is used to get the absolute position of the quadrotor. It has three cameras inside which can capture the certain markers installed on the four axes of the quadrotor in an accuracy of millimeter level. In this work, the tracking system is installed on the ceiling of the lab (Figure 5). The software VZSoft is installed in another Windows PC

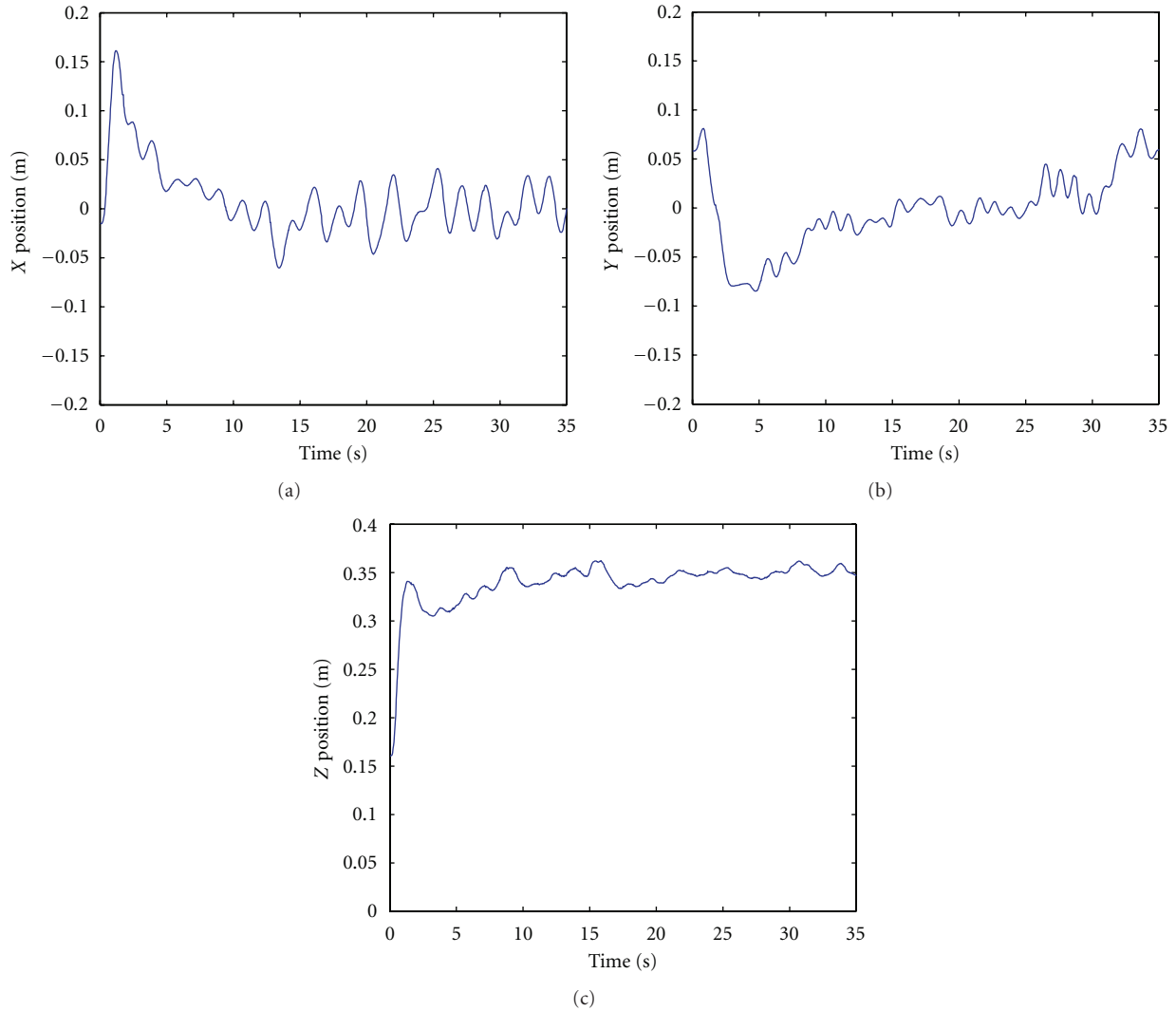


FIGURE 8: 3D position measurements when the quadrotor is in stabilized hover ((a), (b), and (c) for X, Y, and Z positions, resp.). The 3D position is stabilized by multicamera tracking system exclusively. The X and Y positions are set to zero, which means the on-board camera is heading directly to the central point of the ground texture as well as the origin of world reference coordinate. The altitude (Z position) is set to 0.35 m.

(AMD Athlon XP 3000+; 2.1 GHz). It gets the data from the tracking system through a COM interface. The data will be then sent to the workstation with the interface Babelfish which is developed by the Institute of Autonomic Control Engineering, Technical University Munich, using Internet Communications Engine (ICE).

5.2. Velocity Estimation. To validate the designed templates of receptive fields for rotation detection, the bioinspired image processing algorithm is implemented with C++ language using Open CV.

Under low velocities and within certain altitude range, the response can be regarded as monotonic and near linear from the test results. In this work, the yaw rate is under 100 deg/s and the altitude value is set to 0.35 m. The lookup table is shown in Figure 6(a), with a polynomial curve fitting. From the comparison in Figure 6(b), this approach provides

a fairly accurate yaw rate estimation (the mean error is 1.85 deg/s and the standard deviation of error is 10.22 deg/s). This lookup table could be then used in the closed-loop control under the same light condition.

5.3. Heading Stabilization. In this experiment, we compare the heading control performance using EMD with those using IMU or tracking system respectively. At first the stable commands (u_p^s , u_r^s , u_y^s , and u_t^s) should be determined experimentally, so that without any controllers off board, the quadrotor can be hovering in the air nearly on a horizontal level and rotating as little as possible, with all the payloads mounted (in this experiment, with on-board bread board for tracking system using TCM8 mode, and with cable power supply instead of battery). The X, Y, and Z positions should be further controlled using the feedback from the tracking system, while the yaw position has no controllers

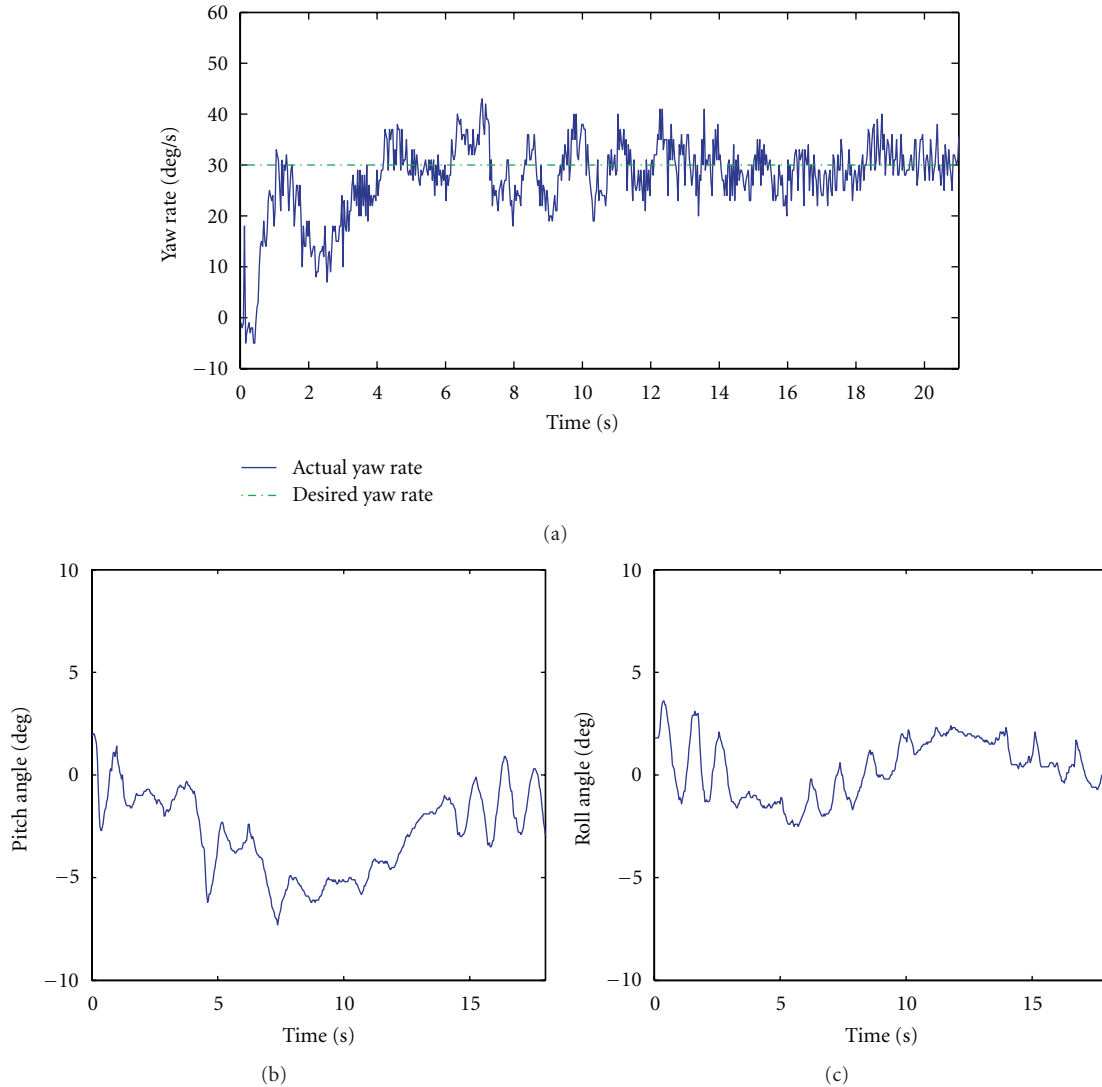


FIGURE 9: Attitude measurements of the flying robot near hover with yaw-rotation (a): yaw rate control. After takeoff, the quadrotor is switched to a desired yaw rate of 30 deg/s; (b) and (c): pitch and roll angles of the quadrotor.

except the on-board IMU at the first attempt. The IMU controller is already integrated on the base board so that no other off-board controller is needed for IMU controlling. A major disadvantage of using IMUs for navigation is that they typically suffer from accumulated error (see Figure 7 blue curve). In this case, in about 20 seconds the yaw position will deviate by 30 degrees if only IMU is used for the heading stabilization. The second reference is the yaw position with the control using tracking system, but without using EMD (the red curve in Figure 7). In Figure 8, the 3D position when using EMD for heading stabilization is shown. By using tracking system and EMDs (the green curve in Figure 7) a satisfying performance could be both achieved. Despite some deviation (for tracking system maximal ± 5 degrees and for EMD maximal ± 7 degrees), the quadrotor can hover very well with straight heading direction.

5.4. Yaw Rate Control. The next step is to achieve the velocity control using EMDs. The yaw rate should be set in a low-speed area considering a monotonic relationship between response and velocity. In this case, the desired velocity is 30 degrees/s. The results are shown in Figure 9(a). The settling time is about 4 seconds and the maximal error is about ± 10 degrees/s. The inflight performance of 3D pose is shown in Figures 9(b) and 9(c). Since the IMU provides only the angular velocity values, the angle positions are integrated by the base control board on the quadrotor in order to get the angle positions, which are sent to the central workstation.

For velocity estimation, although the EMD is not a pure velocity detector, a closed-loop control of yaw rate is achieved with restrictions of the structured environment and the limitation of velocity in low-speed area. Including image translation delay through IEEE1394/USB cable, image

processing by CPU costs 10–20 ms (the program takes nearly 10 ms to wait for images from the camera, while the actual computing time is only several milliseconds). So the EMD computing is extremely fast, which provides an evidence of the efficiency when using biological models.

6. Conclusions and Future Works

In this work, the closed-loop control of a flying robot is achieved by using bioinspired image processing method, which proves to be an effective approach with low computational cost. For real-time implementation, the experimental results of heading stabilization show that, by using EMD response as a feedback, the accumulating drift from the on-board IMU is compensated. Another trial regarding the EMDs as a velocity sensor has realized a low-speed control of the yaw rate on the quadrotor in real-world scenario. In the future, all the 6 DOFs should be controlled by using bioinspired image processing exclusively, without relying on any off-board visual sensors or GPS. The absolute position should be determined by certain advanced algorithms. For the future works, some efforts should be put in the development of novel approaches for highly robust flying performance.

Acknowledgments

This work is supported in part by the DFG excellence initiative research cluster Cognition for Technical Systems-CoTeSys, see also www.cotesys.org, the Bernstein Center for Computational Neuroscience Munich, see also <http://www.bccn-munich.de/>, and the Institute for Advanced Study (IAS), Technische Universität München, see also <http://www.tum-ias.de/>.

References

- [1] M. Egelhaaf and A. Borst, "Motion computation and visual orientation in flies," *Comparative Biochemistry and Physiology A*, vol. 104, no. 4, pp. 659–673, 1993.
- [2] M. V. Srinivasan and S. W. Zhang, "Visual navigation in flying insects," *International Review of Neurobiology*, vol. 44, pp. 67–92, 2000.
- [3] H. G. Krapp and R. Hengstenberg, "Estimation of self-motion by optic flow processing in single visual interneurons," *Nature*, vol. 384, no. 6608, pp. 463–466, 1996.
- [4] A. Beyeler, J.-C. Zufferey, and D. Floreano, "3D vision-based navigation for indoor microflyers," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '07)*, pp. 1336–1341, April 2007.
- [5] J. Conroy, G. Gremillion, B. Ranganathan, and J. S. Humbert, "Implementation of wide-field integration of optic flow for autonomous quadrotor navigation," *Autonomous Robots*, vol. 27, no. 3, pp. 199–200, 2009.
- [6] F. Ruffier and N. Franceschini, "Optic flow regulation: the key to aircraft automatic guidance," *Robotics and Autonomous Systems*, vol. 50, no. 4, pp. 177–194, 2005.
- [7] N. Franceschini, F. Ruffier, and J. Serres, "A bio-inspired flying robot sheds light on insect piloting abilities," *Current Biology*, vol. 17, no. 4, pp. 329–335, 2007.
- [8] F. Expert, S. Viollet, and F. Ruffier, "Outdoor field performances of insect-based visual motion sensors," *Journal of Field Robotics*, vol. 28, no. 4, pp. 529–541, 2011.
- [9] E. Buchner, "Behavioural analysis of spatial vision in insects," in *Photoreception and Vision in Invertebrates*, M. A. Ali, Ed., pp. 561–621, Plenum Press, New York, NY, USA, 1984.
- [10] M. Egelhaaf and W. Reichardt, "Dynamic response properties of movement detectors: theoretical analysis and electrophysiological investigation in the visual system of the fly," *Biological Cybernetics*, vol. 56, no. 2-3, pp. 69–87, 1987.
- [11] M. V. Srinivasan, S. W. Zhang, M. Lehrer, and T. S. Collett, "Honeybee navigation en route to the goal: visual flight control and odometry," *Journal of Experimental Biology*, vol. 199, no. 1, pp. 237–244, 1996.
- [12] R. O. Dror, D. C. O'Carroll, and S. B. Laughlin, "Accuracy of velocity estimation by Reichardt correlators," *Journal of the Optical Society of America A*, vol. 18, no. 2, pp. 241–252, 2001.
- [13] S. Rajesh, D. O'Carroll, and D. Abbott, "Elaborated Reichardt correlator for velocity estimation tasks," in *Proceedings of the Biomedical Applications of Micro- and Nanoengineering*, pp. 241–253, December 2002.
- [14] S. Han, A. Straw, M. Dickinson, and R. Murray, "A real-time helicopter testbed for insect-inspired visual flight control," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Kobe, Japan, 2009.
- [15] A. Borst, C. Reisenman, and J. Haag, "Adaptation of response transients in fly motion vision. II: model studies," *Vision Research*, vol. 43, no. 11, pp. 1311–1324, 2003.
- [16] B. Hassenstein and W. Reichardt, "Systemtheoretische Analyse der Zeit-Reihenfolgen, und Vorzeihenauswertung bei der Bewegungsperzeption des Ruesselflaefers *Chlorophanus*," *Naturforsch*, vol. 11, pp. 513–524, 1956.
- [17] T. Zhang, H. Wu, A. Borst, K. Kuehnlentz, and M. Buss, "An FPGA implementation of insect-inspired motion detector for high-speed vision systems," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Pasadena, Calif, USA, May 2008.
- [18] M. Achtelik, T. Zhang, K. Kuehnlentz, and M. Buss, "Visual tracking and control of a quadcopter using a stereo camera system and inertial sensors," in *Proceedings of the IEEE International Conference on Mechatronics and Automation*, Changchun, China, 2009.