

Article

Monocular-Vision-Based Autonomous Hovering for a Miniature Flying Ball

Junqin Lin, Baoling Han * and Qingsheng Luo

School of Mechanical Engineering, Beijing Institute of Technology, 5 Zhongguancun South Street, Haidian District, Beijing 100081, China; E-Mails linjunqin2010@163.com (J.L.); luoqsh@bit.edu.cn (Q.L.)

* Author to whom correspondence should be addressed; E-Mail: hanbl@bit.edu.cn; Tel./Fax: +86-10-6891-8856.

Academic Editor: Gonzalo Pajares Martinsanz

Received: 14 April 2015 / Accepted: 1 June 2015 / Published: 5 June 2015

Abstract: This paper presents a method for detecting and controlling the autonomous hovering of a miniature flying ball (MFB) based on monocular vision. A camera is employed to estimate the three-dimensional position of the vehicle relative to the ground without auxiliary sensors, such as inertial measurement units (IMUs). An image of the ground captured by the camera mounted directly under the miniature flying ball is set as a reference. The position variations between the subsequent frames and the reference image are calculated by comparing their correspondence points. The Kalman filter is used to predict the position of the miniature flying ball to handle situations, such as a lost or wrong frame. Finally, a PID controller is designed, and the performance of the entire system is tested experimentally. The results show that the proposed method can keep the aircraft in a stable hover.

Keywords: monocular-vision sensor; vision measurement; flying height detecting; MAVs; hovering

1. Introduction

The problem of hovering for a micro air vehicle (MAV) is considered here. It would sense the environment based on the information of the vision sensor, breaking the dependence on the Global

Navigation Satellite Systems (GNSS), such as GPS and GLONASS, which will be selectively unavailable and will be disabled in cluttered or indoor environments. The commonly-used range measuring technologies to control the height in hovering, such as radar or laser range finder, are simply impractical for MAVs, because of the size and the weight barriers [1]. In addition to the small volume and light weight, the vision system also has advantages of passive illumination, low power consumption and low cost.

Many research works have been done on controlling the hovering of a vehicle based on vision. In the early years, helicopters were used in experiments because the MAV had not been widely studied. For example, a 67-kg Yamaha R-50 helicopter used a combination of stereo vision and feature tracking to control the hovering [2], and a fly-by-wire helicopter ATTheS (Advanced Technology Testing Helicopter System) used the camera to track a two-dimensional template of *a priori* unknown features [3]. Both of them have the advantages of the sufficient space and carrying capacity of the helicopter. MAVs are characterized as small volume and limited payloads, so it is difficult to miniaturize the equipment for use on MAVs, such as the stereo vision needing a minimum baseline between two cameras.

Vision combined with structured environments has been applied successfully in hovering control [4–6]. For instance, an autonomous quadrotor aircraft can perform stable hovering above a pattern glyph with a data-fusion algorithm using both visual system measurements and inertial sensors [6]. However, the vision system relied on the pattern glyph, which is specially designed to align the aircraft with the glyph. Hence, these methods are still restrained by artificially structured environments in practical applications.

How to sense the height of MAVs by vision for hovering control using natural terrain is also a challenge. In Cherian's work, height is measured by the machine learning to analyze the texture of the image captured by a downwards looking camera [7]. The reliance on the sufficient texture of images limits the application of this method. Optical flow can be used in several movement-control situations, such as terrain following [8], flight in the vicinity of obstacles or over flat terrain [9] and obstacle avoidance [10,11]. As for the control of hovering, optic flow is difficult with respect to measuring the height because it cannot produce a range unless there is a significant motion that goes against the objective of staying at a near perfect hover. Vision has achieved impressive results for the hovering of MAVs, which is fused with other sensors, especially the inertial measurement units (IMUs) [12–14]. These data fusion techniques are usually computationally complex, and the detecting reliability depends on several sensors.

While most of reviewed works have been done using vision assisted with other sensors, the vision-based hovering control proposed in this paper uses only one vision sensor. The MAV does not have any other sensors to measure its position or posture. Our method is inspired by honey bees, which utilize a stored snapshot as a visual marker and keep stable by the disparity between the current retinal image and the snapshot [15,16]. A camera is mounted directly under the MAV, and it takes an image of the ground underneath the MAV as a reference position. An error between the current positions and given reference position in the image plane is computed. When the height of the MFB changes, the distance of two points on the ground is invariable, but the distance of imaging points in the camera changes correspondingly. There is a negative correlation between these two distances. Therefore, we can achieve hovering control of the MFB by maintaining the distance variation near zero.

2. Vision-Based Position Estimation

For the purpose of achieving stable hovering, the position of the MAV, which is relative to an external reference, such as the ground, should be determined at first. We estimate the position of the MAV based on visual information. In addition, a high-speed image processing algorithm also plays an important role to control the performance. Therefore, the image processing algorithm should be as fast as possible.

In this paper, we present an embedded MAV that uses information extracted from real-time images for hovering control. The ball-shaped outer shell of the MFB is lifted and propelled by the coaxial contra rotating twin rotors, which are mounted at the top of the ball (Figure 1). Four control surfaces at the bottom of the ball are used to control the motion direction of the ball freely. Therefore, the MFB almost has no restraints on its movement and can fly omnidirectionally. For the control of the hovering, the aircraft just needs the thrust of the coaxial contra rotating twin rotors, owing to its special power distribution, and it will always keep straight down to the ground.

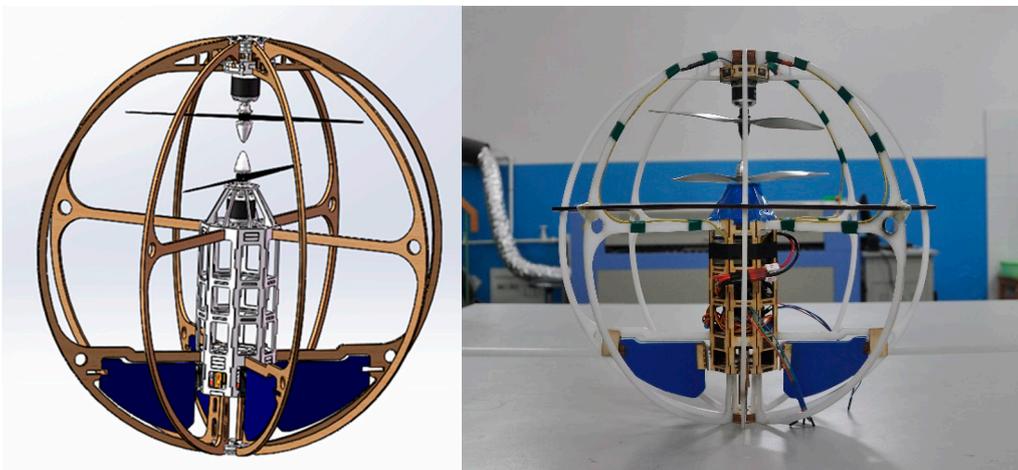


Figure 1. Miniature flying ball.

2.1. Monocular Vision Analysis

The camera is mounted directly under the MFB. In the camera coordinate system, the optical center of the camera is set as the origin, and the Z axis is parallel to the optical axis. The X and Y axes are parallel to the image plane, as shown in Figure 2a. A perspective projection model of a pinhole camera allows the position in a 2D image plane to be inferred from the 3D position [17]. The model projects an arbitrary point $A = (x_w, y_w, z_w)^T$ on the ground to point $a = (u, v)^T$ on the image plane (the camera image) expressed by Equation (1):

$$z_c \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{f}{d_x} & -\frac{f}{d_x} \cot \theta & u_0 & 0 \\ 0 & \frac{f}{d_y \sin \theta} & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \quad (1)$$

where Z_c is an arbitrary positive scalar. The six intrinsic parameters $(f, d_x, d_y, \theta, u_0, v_0)$ can be derived from the calibration of the camera. The extrinsic parameters, the R and T metrics, denote the rotation and translation of the camera. For the control of hovering, the aircraft just needs the thrust upward. It will always keep nearly straight down to the ground, and the image plane of the camera will keep approximately parallel to the ground. Therefore, the extrinsic parameters can be simplified as:

$$R \doteq \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, T \doteq \begin{bmatrix} 0 \\ 0 \\ h \end{bmatrix} \quad (2)$$

where h denotes the height of the camera. Then, we can get Equation (3):

$$u - u_0 = \frac{fx_w}{h}, v - v_0 = \frac{fy_w}{h} \quad (3)$$

Consider the situation of a line AB on the ground and the corresponding line ab on the image plane. When the height of the MFB changes into h' , as shown in Figure 2b, ab on the image plane will change into $a'b'$:

$$h = \frac{f \cdot l_{AB}}{l_{ab}}, h' = \frac{f \cdot l_{AB}}{l_{a'b'}} \quad (4)$$

From Equation (4), the measurement formula is obtained:

$$\frac{h}{h'} = \frac{l_{a'b'}}{l_{ab}} \quad (5)$$

The essential problem of the system is to achieve stable hovering of the MFB by maintaining $h' = h$. According to Equation (5), this condition can be transferred into keeping $l_{a'b'}$ nearly equal to l_{ab} . Therefore, the problem of controlling the height of the MFB is turned into finding the corresponding points ab and $a'b'$.

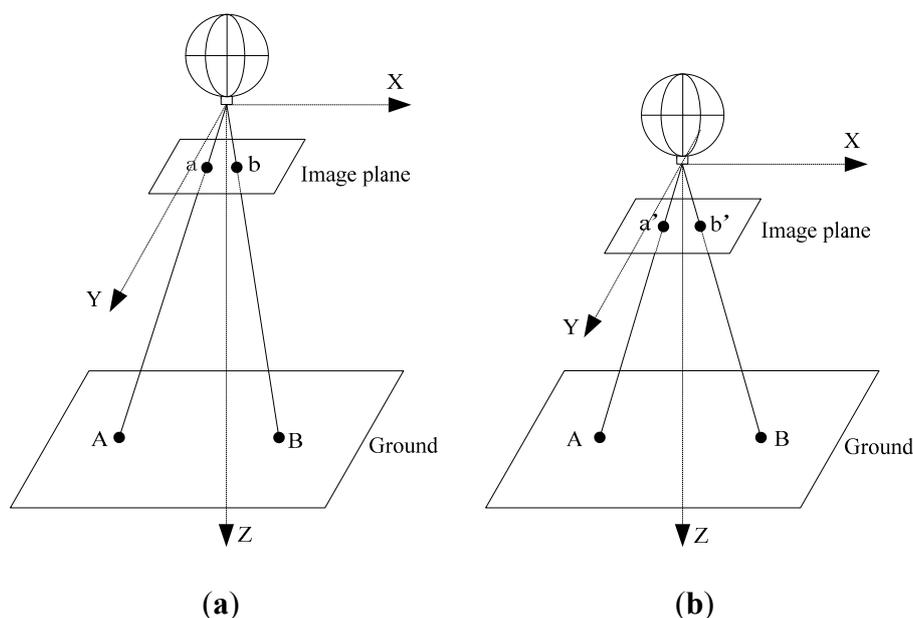


Figure 2. The monocular vision model and coordinate system used.

2.2. Measurement from Images

In order to reduce computing requirements, it is desirable to keep the detection region of images as small as possible. This also indicates that not all of the visual angles in the field of view have the same relationship for flight control. Visual angles pointing at the margin of images correspond to regions that may disappear during flight and, thus, do not require computation. For visual angles close to zero (*i.e.*, close to the center of the visual field), the measurable magnitude tends to be zero. These two limits suggest that the region of interest lies around the middle (Figure 3), where measurements are both reliable and relevant for controlling the aircraft.

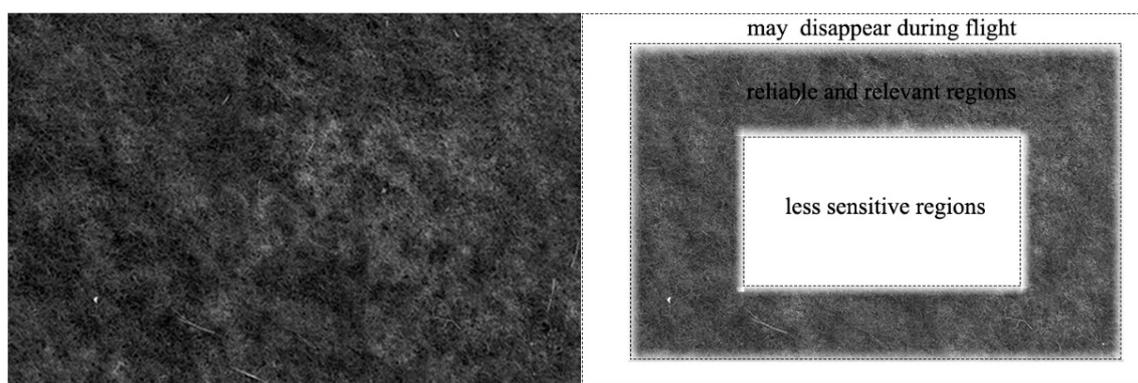


Figure 3. (Left) An image captured from a downwards looking camera of the miniature flying ball (MFB); (Right) Representation of the regions where measurements are both reliable and relevant.

2.2.1. Feature Extraction

The camera was calibrated to get the intrinsic parameters before use. Then, we can use the parameters to remove the geometrical distortion of the camera and obtain the accurate distances on the image plane.

In the follow-up process, we should find out the corresponding points between subsequence frames and the reference frame. The first step is to extract the features in the frames. There are some common features, such as point, region and contour. Generally speaking, feature point extractions, such as Harris, SUSAN (Smallest Univalued Segment Assimilating Nucleus) and DOG (Difference of Gaussian), are relatively easy in their calculation and are invariable against the rotation, translation and illumination variation.

In the application of our system, the sequence images may change not only in the rotation, translation and illumination variation, but also in the scale and the viewpoint. Therefore, we adopt the SIFT algorithm proposed by Lowe [18], which can deal with these situations. However, the algorithm of SIFT is too time-consuming and is impractical in our system. Therefore, we use the SURF algorithm [19], the improved version of SIFT, to achieve quicker speed and more feature points. The extraction result of SURF is shown in Figure 4.

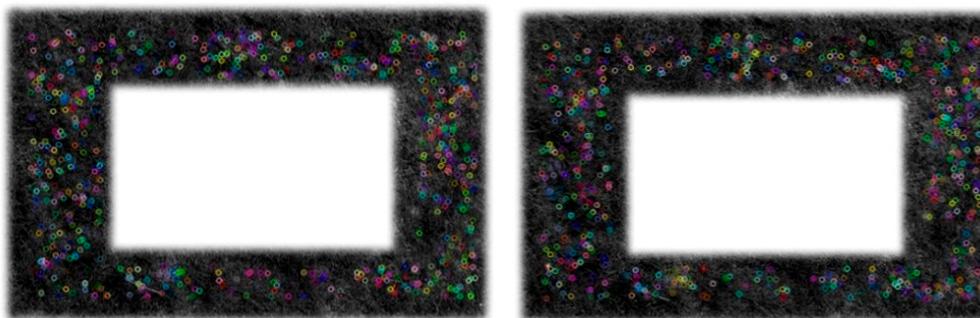


Figure 4. Feature extraction results of the reference frame (**Left**) and one of sequence frames (**Right**).

2.2.2. Feature Matching

After feature extraction, the corresponding points between frames should be matched. Conventional techniques include the correlation coefficient method, the Hausdorff distance method, measurements of similarity, *etc.* We can obtain a multitude of information, such as position, scale, principal orientation and feature vector, about the points when the feature extraction is executed. The feature vectors contain the neighborhood information of the feature points, so we can use the nearest neighbor method to find out the potential matching points to achieve a faster processing speed. The calculation formula can be expressed as follows:

$$D = \left\{ \min_{j=1,2,\dots,N_2} \sqrt{\sum_{i=1}^k (f_{ik} - f_{jk})^2} \mid i = 1, 2, \dots, N_1 \right\} \quad (6)$$

where N_1 and N_2 are the number of feature points extracted from two frames, respectively. In order to get a robust matching result, we pick out two minimal candidates denoted as d_{ij} and d'_{ij} . If $d_{ij} \leq \alpha \cdot d'_{ij}$ (α is set as 0.7 in the experiment), then d_{ij} is considered as the optimal matching point, and the wrong matching can be rejected effectively. The matching results are shown in Figure 5.

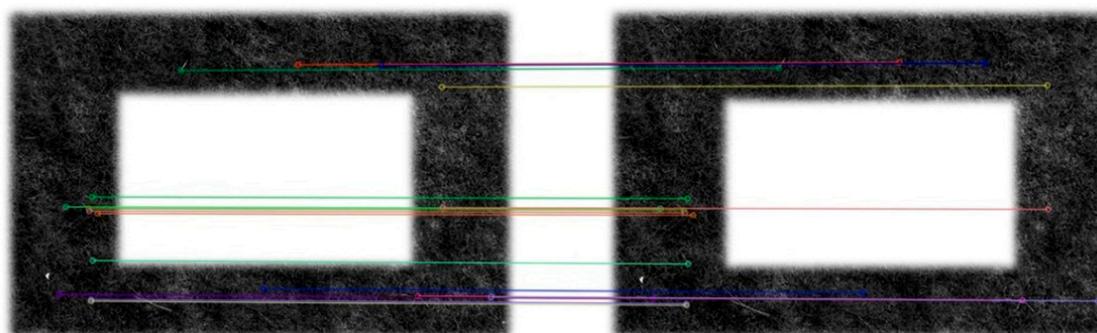


Figure 5. The matching result of two frames. The lines connect the match points between two frames.

2.2.3. Distance of Frames

In order to reduce measurement errors and get accurate data, we use the statistical method to calculate the distance between frames. Assume that we have found N pairs of matching points, and then, we can get the height and x , y variations between two frames from:

$$\begin{cases} \Delta x = \frac{\sum_{i=1}^n x_i}{n} - \frac{\sum_{i=1}^n x'_i}{n} \\ \Delta y = \frac{\sum_{i=1}^n y_i}{n} - \frac{\sum_{i=1}^n y'_i}{n} \\ \frac{h}{h'} = \frac{\sum_{i=1}^n a'b'_i}{\sum_{i=1}^n ab_i} \end{cases} \quad (7)$$

where (x_i, y_i) is the coordinate of the feature on the image plane at the target height and (x'_i, y'_i) is the coordinate at other moments. $a'b'_i$ and ab_i denote the Euclidean distances of each of the two features on the image plane of two moments. Then, we can get the variable $(\Delta x, \Delta y)$ in the x and y directions and variable $\frac{h}{h'}$ in the z direction.

From Equation (7), we can figure out that the position estimation of the MFB depends on the feature position on the image plane, but is unrelated to the intrinsic parameters of the camera. This is because the position parameters we consider here are relatively variable. The intrinsic parameters of the camera have been eliminated during the calculation. Thus, our method can avoid the measurement error introduced by the calibration error of the camera.

2.3. Kalman Filter

A Kalman filter can utilize the basic physics model of the MFB to estimate a dot from its previous position when it is not sensed correctly by the vision system. The Kalman filter will smooth the sensing results and prevent drastic changes in the control commands, which will cause the MFB to become unstable.

In the implementation, when a datum is considered as an outlier, the Kalman filter will respond by modifying the measurement update covariance matrix in order to reduce the confidence in that data. The state vector (T_j) of the Kalman filter is composed of h and the x and y position variations of the frames, which are being filtered (Equation (8)). The measurement input (Z_j) of the filter is also h and the x and y position variations of each frame, which make the H matrix of the update equations in the basic Kalman filter equations the identity matrix.

$$T_j = \begin{bmatrix} h \\ x \\ y \end{bmatrix} \quad (8)$$

Three variations are independent of each other, so the equations can be even further transformed, as shown in Equation (10). The equations are then simplified by breaking all matrix operations into single element operations.

$$\begin{aligned}
 \hat{u}(k)_j &= z(k)_j - \hat{t}(k)_{j|j+1} \\
 \hat{s}(k)_j &= p(k, k)_{j|j-1} + r(k)_j \\
 k(k)_j &= p(k, k)_{j|j-1} s(k)_j^{-1} \\
 \hat{t}_{j|j} &= \hat{t}(k)_{j|j-1} + k(k)_j \hat{u}(k)_j \\
 p(k, k)_{j|j} &= p(k, k)_{j|j-1} - k(k)_j p(k, k)_{j|j-1} \\
 k &= 1, 2, 3.
 \end{aligned} \tag{9}$$

2.4. Controller Design

In order to achieve the desired position, three variables h , x and y of the MFB should be controlled with a specific controller. Couplings between the three channels are ignored for simplification. The h (height) and the x and y variations are controlled by three off-board PID controllers independently.

For the position x/y of the MFB, the two controllers are similar. P gain is limited to ensure the stability due to the latency of the system. The I term is used to reduce the steady-state errors. The vision data are used to calculate the D terms, since there are no other sensors to measure the speed. Because the field of view (FOV) of the camera is limited, it is necessary to set a threshold value as the output of the PID controller. Otherwise, the corresponding feature points will be out of the FOV, and then, the MFB will be uncontrollable.

For the h -controller, the processing method is basically the same as with x/y . The main difference is that the I term is adjusted to provide enough thrust for the hovering MFB according to different payloads and changing battery status.

3. Materials

To validate the proposed hovering control strategy, we ran a series of experiments in a real flying platform developed in our laboratory (Figure 1). In this section, we describe the hardware and software architecture that we used.

3.1. Hardware Description

The whole system (Figure 6) consists of two parts: the MFB aircraft, consisting of coaxial contra rotating twin rotors, control surfaces, an onboard controller and sensors, such as the camera and IMUs; and the ground station, which receives and processes sensor data, computes the pose of the aircraft, performs the controller strategy and sends the control commands back to the aircraft.

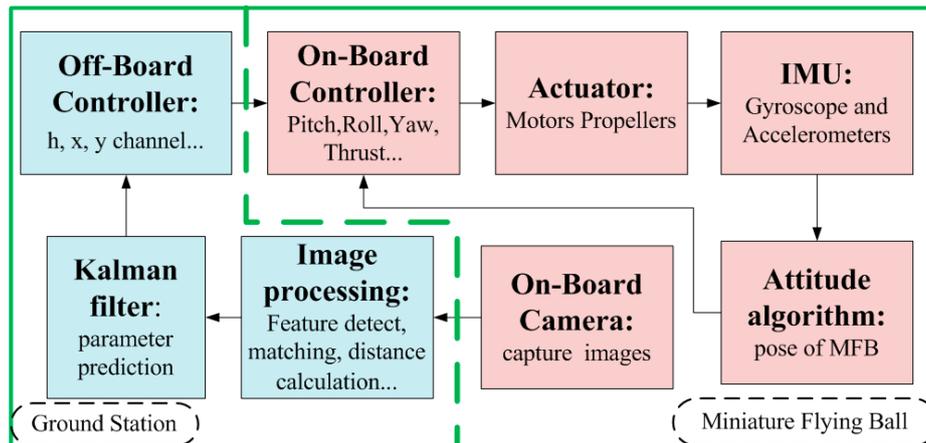


Figure 6. System overview.

3.1.1. MFB Aircraft

The MFB can perform at 200-Hz control frequency to drive the high-torque DC brushless motors, which enables rapid reaction to changes in the environment. A pair of coaxial rotors spin clockwise and anticlockwise, respectively, which counteracts the rotors' torque and provide thrust. This is different from similar works, such as the flying robot in [20] and the gimball in [21]. For them, the control surfaces are needed to balance the rotors torque, and this will cause the aircrafts to tilt toward the ground. Our aircraft just needs the thrust of the coaxial contra rotating twin rotors, and it will always keep straight down to the ground during hovering.

The IMU on the MFB aircraft consists of three gyroscopes sensing the angular velocity of each rotation axis and three accelerometers sensing the acceleration of each translation axis. It is used to provide the reference pose of the MFB. Limited by the space and payloads of the MFB, a miniature camera is used in the vision system. The specification of the camera will be discussed in more detail later in the article.

3.1.2. Ground Station

To avoid putting a heavy and powerful computer on the aircraft, images are wirelessly transmitted to a personal computer (PC), through our custom software. An off-board PC (Intel i5, 2 GB RAM) is used for data processing, pose estimation and performing the controller strategy in our system. The aircraft is equipped with the wireless image transmission subsystem and the command transmission subsystem modules, which can transmit image data and control commands at a rate of 20 Hz wirelessly.

3.2. Software Architecture

The main software on the ground station consists of two major parts: vision-based position estimation and a controller. Besides, there are two minor blocks in the entire software structure: the GUI block and the communication block. As shown in Figure 7, the GUI block can provide the interface between the entire system and the user for sensory data displaying, parameter controlling and parameter tuning. The communication block is used to integrate the hardware and software between the aircraft and the ground station.

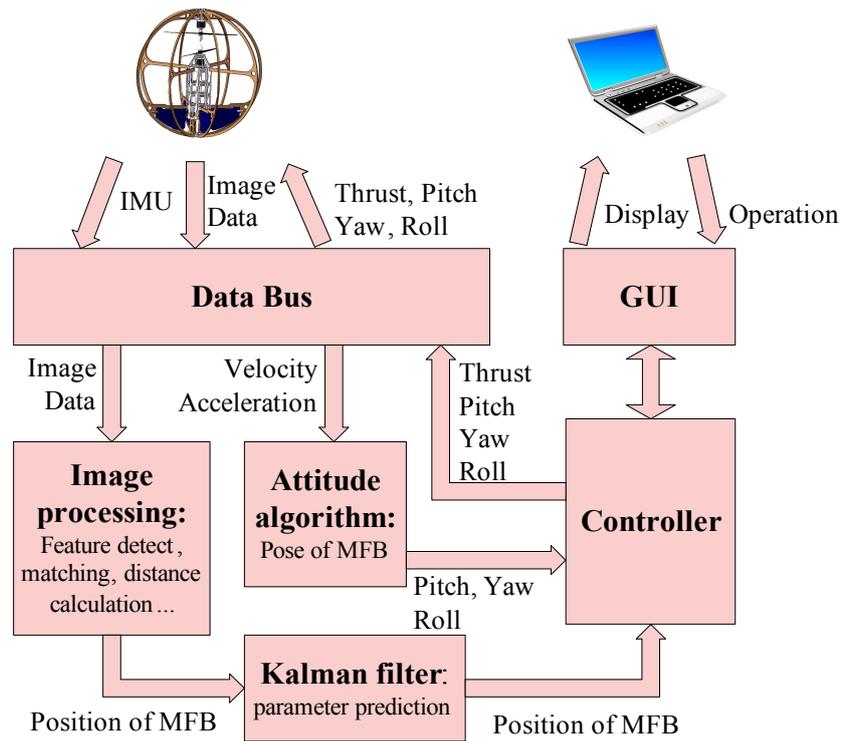


Figure 7. Software architecture.

4. Experiments

The performance of the vision-based hovering method is demonstrated using the MFB described in the previous sections. The diameter of the platform is 42 cm, and the total mass is 625 g. The parameters of the selected camera are as follows: the focus length is $8 \times 10^{-3} m$; the resolution of CCD is 640×480 ; the size of the CCD is $1/3''$; all algorithm development is in C++, which utilizes the OpenCV library for feature extraction. The experiment environment is a football field shown in Figure 8. The entire system was evaluated in four parts: (1) evaluating the solution of the system; (2) evaluating the static measurement precision of the system; (3) testing the dynamic hovering precision of the MFB; and (4) testing the robustness of the system.



Figure 8. The flying test on the football field.

4.1. Evaluating the Solution of the System

Through the derivation of Equation (4) with respect to the length on the image plane, we can get:

$$\Delta h = -\frac{f \cdot l}{x^2} \cdot \Delta x = -\frac{h^2}{f \cdot l} \cdot \Delta x \quad (10)$$

where Δh represents the height resolution according to the resolution of the CCD Δx . This means that we can sense the minimum change Δh with a specific CCD. Here, assume the height of MFB is set as 3 m. The parameters are substituted into Equation (10), and we can get the relation of the resolution (Δh) with the distance (l) from the center. As can be seen from Figure 9, the resolution near the center (less than 0.8 m) is too coarse, and it will become stable after a certain distance (large than 1.4 m). We can choose the appropriate distance and resolution as we need.

In Equation (10), the $-\frac{h \cdot \Delta x}{f \cdot l}$ is constant for the system. Therefore, the height resolution Δh is linearly related to the height h . This means that the higher the aircraft hover, the lower the detection precision will be. Therefore, our method is better applicable within a proper altitude according to the sensitivity of the detector.

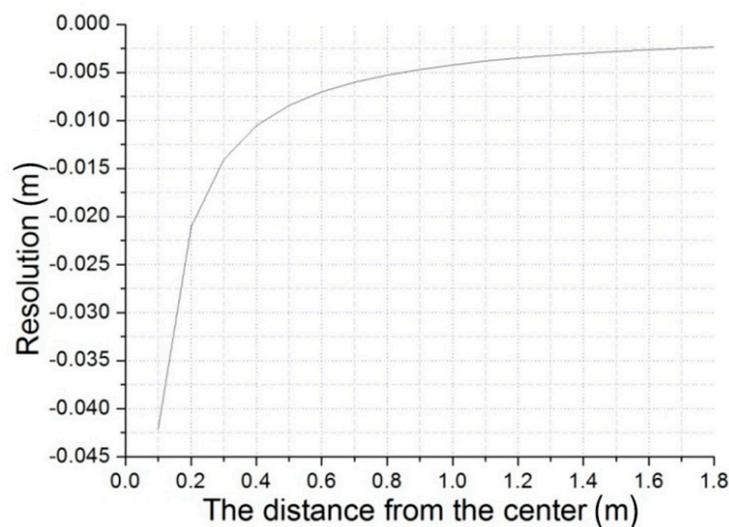


Figure 9. The relation of the resolution with the distance from the center.

4.2. Static Measurement Precision of the Vision System of MFB

In this part, we designed experiments to show the validity of our measurement method. The MFB was hung up by a rope, and its height can be accurately measured by measuring tape. The initial height of the MFB was set at 2 m, and the height could be moved up and down freely. The camera on the MFB looked straight down to the ground, and it took several pictures every time when the height of the MFB changed as planned. According to the schematic diagram presented in Figure 4 and the basic Equation (7), we can get the relationship of $\frac{h}{h'}$ and the actual distance variation (Figure 10).

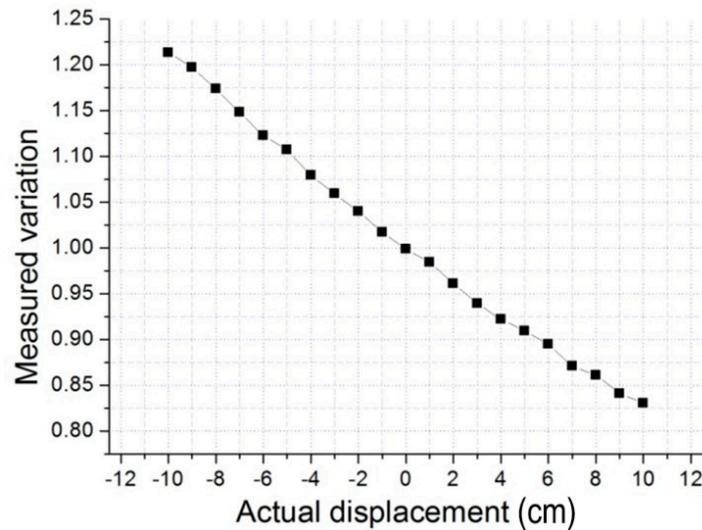


Figure 10. The relation of actual displacement and the measured variation.

The graph's horizontal axis shows the actual distance variation from the hovering position, and the vertical axis shows the $\frac{h}{h'}$ calculated by Equation (7). Therefore, in the actual flight, we can figure out the actual distance variation from the hovering position by $\frac{h}{h'}$, which we get from monocular vision. For example, when the $\frac{h}{h'}$ equal to 1, this means that the actual position is at the hovering position. When the $\frac{h}{h'}$ is less than 1, this shows that the actual position is above the hovering position. Otherwise, the actual position is beneath the hovering position. Therefore, we can maintain the value of $\frac{h}{h'}$ equal to 1 by controlling the motion of the MFB to achieve stable hovering.

4.3. Dynamic Hovering Precision of MFB

When hovering, the quadrotor is very stable. The 6 DOF of the MFB can be traced by an optical tracking system from NDI Polaris, which uses precise marks to track. The 6 DOF pose of the system obtained by NDI Polaris is used as the standard to inspect the hovering accuracy of the MFB. It can work at a speed of 60 frames per second and recover the position with a precision of around 0.3 mm RMS.

The main experimental processes are as follows: Firstly, the MFB is controlled by the remote control handle to an arbitrary height. Then enable the proposed hovering method is enabled to keep the MFB stable. During the hovering control, NDI Polaris is used to record the trajectory of the MFB. In Figure 11, the height was set at 3 m, and the flight path for a 60-s hover can be recorded and displayed. Overall, the position error during 60 s of hovering has an RMS value of 1.54 cm in the X position, 1.04 cm in the Y position and 0.92 cm in Z position, which yields an absolute RMS error value of 1.36 cm, as shown in Figure 12.

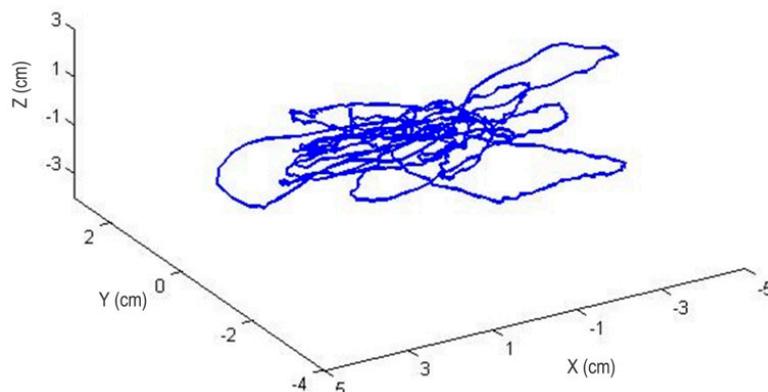


Figure 11. Position error while hovering during 60 s. The RMS value of the position error is 1.54 cm in X , 1.04 cm in Y and 0.92 cm in Z .

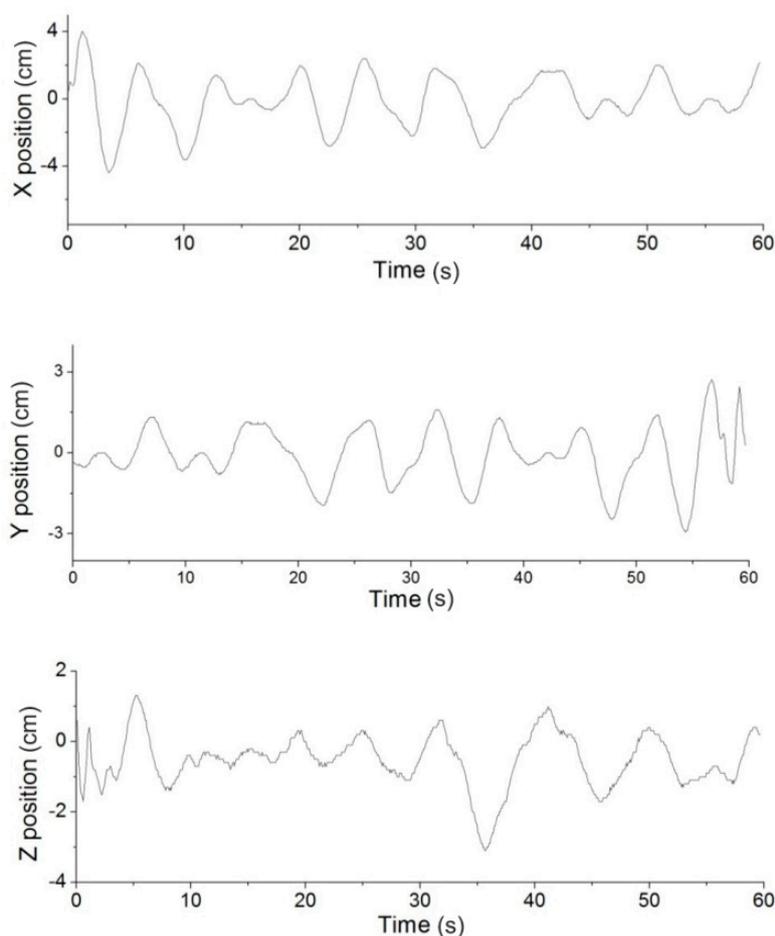


Figure 12. Position error in the X , Y and Z positions. The value remains ± 4 cm. The Z position is more accurate than the X and Y positions.

4.4. Testing the Robustness of the System

We conduct this experiment to demonstrate the method's feasibility when the system is affected by a strong disturbance; that is to demonstrate if the MFB can go back to the original hovering altitude when it suffers a sudden change in height, so as to verify the robustness of the system. The

experimental setup is the same as that of the previous experiments in Section 4.3. The only difference here is that a sudden height change is employed. To be specific, after the MFB is hovering for a while, we pull it down to certain height (24 cm in this trial) and observe the MFB's position variation.

The recorded data are shown in Figure 13. The blue line shows the height of the MFB; the red line is the desired height; and the black bars present the corresponding output of the vision system with the proposed method. Note that the outputs are too small to display, so they have been magnified 100-times here. In the experiment, during the first 37 s, the MFB was kept hovering at the expected position with a small deviation. Right at 38 s, it was pulled 24 cm down promptly and released soon. The output of the vision system reached the maximum at this moment. Note that a threshold is set to prevent drastic changes of the output, which will cause the MFB to become unstable. Then, during 38 to 48 s, the MFB returned back to the original height, as expected. In this process, the output decreases with the rising height of the MFB. After 48 s, the ball was hovering around the original height again. The graph above shows that our proposed vision system can always detect the MFB's orientation and distance and generate negative feedback to control the MFB hovering at the required height.

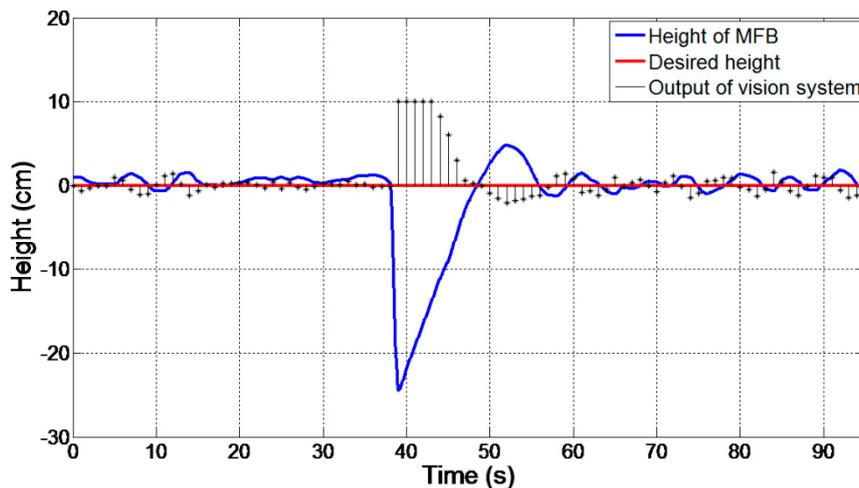


Figure 13. The response to the sudden height change.

4.5. Discussion

The experiments are based on the assumption that the MFB can always keep straight down to the ground, so that the image planes are completely parallel to each other. Of course, it is impossible to keep the image plane completely parallel without error during the flight. Here, we discuss the error caused by the vibration of the MFB.

Figure 14 shows the altitude variation recorded by the IMU of the MFB during one flight with coaxial rotors operating only. At the stage of preparing for taking off, the MFB was kept straight downward, and three Euler angles were initialized to 0. During the flight, the pitch and roll in the horizontal direction varied slightly and kept within a range of less than 1.5° . The yaw in the vertical direction deflected with small angles. When landed, the MFB rolled and bounced to slow down until stopping. This caused the three Euler angles to change drastically. After stopping, the three Euler angles kept steady.

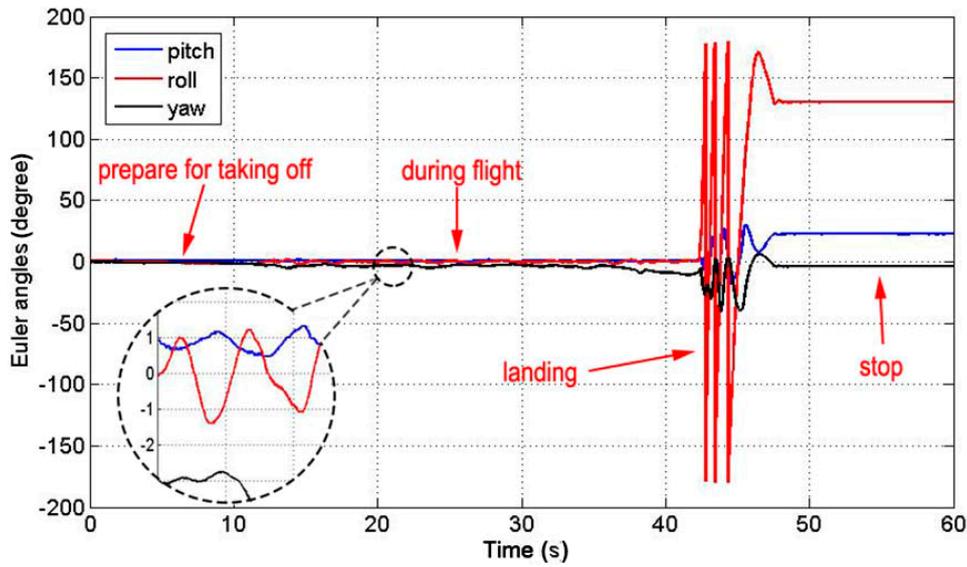


Figure 14. IMU data during one flight.

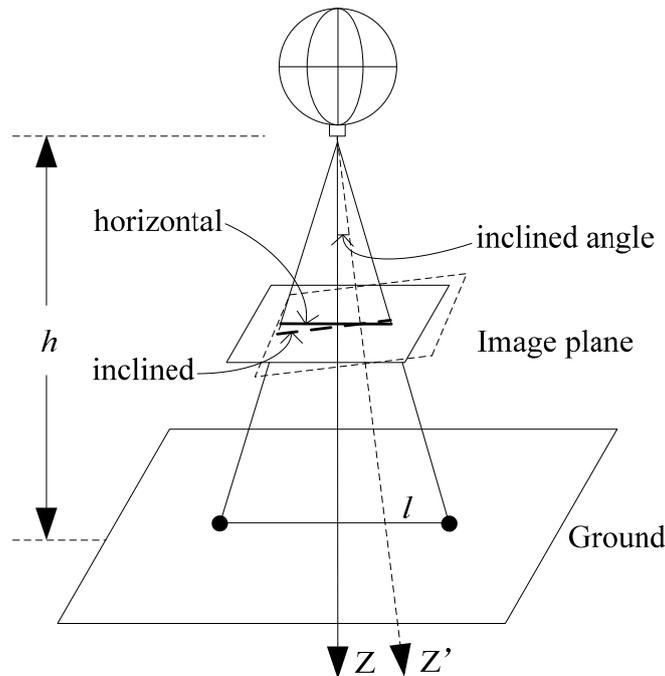


Figure 15. Inclined model of the MFB.

The schematic of the inclined image plane is shown in Figure 15. The relative error caused by the inclined angle to the measurement result can be expressed as:

$$\delta = \frac{\left(\tan(\arctan(l/h) + \Delta\alpha) + \tan(\arctan(l/h) - \Delta\alpha) \right) - 2l/h}{2l/h} \times 100\% \quad (11)$$

where l is the distance of the feature to the center on the ground, h is the height of the MFB, $\Delta\alpha$ is the inclined angle and δ is the relative error. Plugging in the conventional value ($l = 1 \text{ m}, h = 3 \text{ m}$), then we can get the relationship of the relative error δ and the inclined angle $\Delta\alpha$, as shown in Figure 16.

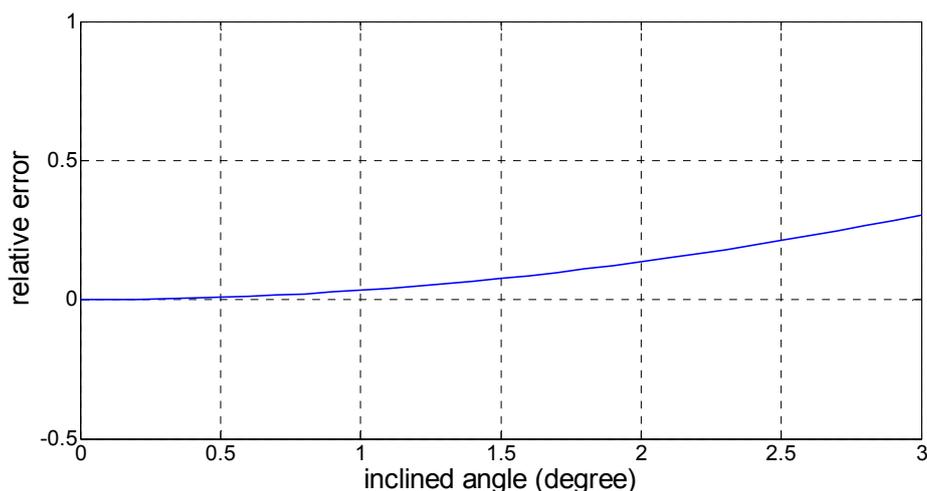


Figure 16. The relationship of the inclined angle with the relative error.

In Figure 16, we find that the pitch and roll in the horizontal direction keep within a range of less than 1.5° during normal flight. Then, the relative error caused by the inclined angle will not exceed 0.076%. Therefore, the effect of the inclined angle caused by the tilt of the MFB during flight has little impact on the measurement results. In some other cases, the MFB may be unstable with a great tilt angle caused by the wind or other unpredictable events.

5. Conclusions and Future Work

An onboard camera is utilized to measure the position of an MFB to maintain a stable hover. The position variation between the subsequent frames and the reference image can be acquired from analyzing their correspondence points. Then, the measurement is processed by the Kalman filter and sent to the PID controller to control the MFB. The experimental results show that the visual method is able to estimate the aircraft's position in unknown environments and to guide the MFB to achieve stable hovering.

Several improvements will be made in the future work. First, we will develop a robust control algorithm, which allows the aircraft to adapt itself to the uncertainty state estimate. Secondly, our work in this paper lacks consideration of bad weather, especially wind, so we plan to expand the functionality of the system by visual techniques in order to adapt to more applications. Thirdly, from Figure 14, we can figure out that the maximum rotation of the MFB is about 10.6° during the 30-s flight. Although this has no effect on our hovering control method, we can derive the rotation of the system from Equation (7) and apply it in the precise control of the MFB in future work.

Author Contributions

Junqin Lin and Baoling Han conceived of and designed the experiments. Junqin Lin and Qingsheng Luo performed the experiments and analyzed the data. Junqin Lin wrote the paper.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Garratt, M.A.; Lambert, A.J.; Teimoori, H. Design of a 3D snapshot based visual flight control system using a single camera in hover. *Auton. Robot.* **2013**, *34*, 19–34.
2. Amidi, O.; Kanade, T.; Fujiita, K. A visual odometer for autonomous helicopter flight. *Rob. Auton. Syst.* **1999**, *28*, 185–193.
3. Oertel, C.-H. Machine vision-based sensing for helicopter flight control. *Robotica* **2000**, *18*, 299–303.
4. Mejias, L.; Saripalli, S.; Campoy, P.; Sukhatme, G.S. Visual Servoing of an Autonomous Helicopter in Urban Areas Using Feature Tracking. *IFAC Proc. Vol.* **2007**, *7*, 81–86.
5. Bonak, M.; Matko, D.; Blai, S. Quadcopter control using an on-board video system with off-board processing. *Rob. Auton. Syst.* **2012**, *60*, 657–667.
6. Gomez-Balderas, J.E.; Salazar, S.; Guerrero, J.A.; Lozano, R. Vision-based autonomous hovering for a miniature quad-rotor. *Robotica* **2014**, *32*, 43–61.
7. Cherian, A.; Andersh, J.; Morellas, V.; Papanikolopoulos, N.; Mettler, B. Autonomous altitude estimation of a UAV using a single onboard camera. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, MO, USA, 10–15 October **2009**; pp. 3900–3905.
8. Garratt, M.A.; Chahl, J.S. Vision-Based Terrain Following for an Unmanned Rotorcraft. *IFAC Proc. Vol.* **2007**, *7*, 81–86.
9. Beyeler, A.; Dario, J.Z. Vision-based control of near-obstacle flight. **2009**, *27*, 201–219.
10. Zingg, S.; Scaramuzza, D.; Weiss, S.; Siegwart, R. MAV navigation through indoor corridors using optical flow. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 3–7 May 2010, Anchorage, AK, USA; pp.3361–3368.
11. Beyeler, A.; Zufferey, J.C.; Floreano, D. 3D vision-based navigation for indoor microflyers. In Proceedings of the IEEE International Conference on Robotics and Automation, Roma, Italy, 10–14 April 2007; pp. 1336–1341.
12. Chowdhary, G.; Johnson, E.N.; Magree, D.; Wu, A.; Shein, A. GPS-denied Indoor and Outdoor Monocular Vision Aided Navigation and Control of Unmanned Aircraft. *J. F. Robot.* **2013**, *30*, 415–438.
13. Shen, S.; Mulgaonkar, Y.; Michael, N.; Kumar, V. Vision-Based State Estimation and Trajectory Control Towards High-Speed Flight with A Quadrotor. Available online: <http://www.kumarrobotics.org/wp-content/uploads/2014/01/2013-vision-based.pdf> (accessed on 5 June 2015).
14. Zhang, T.; Li, W.; Kühnlenz, K.; Buss, M. Multi-Sensory Motion Estimation and Control of an Autonomous Quadrotor. *Adv. Robot.* **2011**, *25*, 1493–1514.
15. Cartwright, B.A.; Collett, T.S. How honey bees use landmarks to guide their return to a food source. *Nature* **1982**, *295*, 560–564.
16. Cartwright, B.A.; Collett, T.S. Landmark learning in bees. *J. Comp. Physiol. Psychol.* **1983**, *151*, 521–543.

17. Wu, A.D.; Johnson, E.N. Methods for localization and mapping using vision and inertial sensors. In Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit, Honolulu, HI, USA, 18–21 August 2008.
18. Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110.
19. Bay, H.; Ess, A.; Tuytelaars, T.; Van Gool, L. Speeded-Up Robust Features (SURF). *Comput. Vis. Image Underst.* **2008**, *110*, 346–359.
20. Klaptocz, A. Design of Flying Robots for Collision Absorption and Self-Recovery. Ph.D Thesis, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, 2012.
21. Briod, A.; Kornatowski, P.; Jean-Christophe, Z.; Dario, F. A Collision-resilient Flying Robot. *IFAC Proc. Vol.* **2007**, *7*, 81–86.

© 2015 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/4.0/>).