

Design of an On-Board Vision-Based Obstacle Avoidance System for Flapping-Wing Robots

FU Qiang^{1,2,3}, ZHANG Zuoqing^{1,2,3}, LIU Shengnan^{1,2,3},
WU Xiaoyang^{1,2,3}, HE Wei^{1,2,3*}

1. School of Intelligence Science and Technology, University of Science and Technology Beijing, Beijing 100083, P. R. China;
2. Institute of Artificial Intelligence, University of Science and Technology Beijing, Beijing 100083, P. R. China;
3. Key Laboratory of Intelligent Bionic Unmanned Systems, University of Science and Technology Beijing, Beijing 100083, P. R. China

(Received 14 May 2023; revised 20 December 2023; accepted 24 January 2024)

Abstract: The flapping-wing robot (FWR) is a kind of high-biomimetic aerial vehicle that can perform military reconnaissance and civil monitoring missions. When performing these missions, obstacle avoidance is a necessary function to ensure safety of FWRs. In this paper, an autonomous monocular-vision-based obstacle avoidance system is designed for FWRs, where all the image processing computations are implemented by using an on-board computer. In this system, the weight of the on-board computer is reduced to 48 g so that the FWR could fly properly. The workflow of the system can be divided into the following steps. First, the image acquisition module captures videos of the surrounding environment. Second, the on-board computer calculates the rudder angle and turning direction by processing the optical flow information from the first-person view. Finally, the flight control board receives the calculated results and controls the FWR to avoid obstacles. A ground station performs real-time monitoring of the FWR flight process, and experimental results demonstrate the effectiveness of the obstacle avoidance system designed in this paper.

Key words: flapping-wing robots; vision; obstacle avoidance; on-board computer; optical flow

CLC number: TP242.6 **Document code:** A **Article ID:** 1005-1120(2024)01-0097-10

0 Introduction

Obstacle avoidance is very important for unmanned aerial vehicles(UAVs) to ensure their flight safety^[1-2]. There are various strategies to detect obstacles, such as ultrasonic-based obstacle avoidance^[3], vision-based obstacle avoidance^[4] and laser-based obstacle avoidance^[5], etc. Each of these approaches has its own advantages and disadvantages. Ultrasonic-based obstacle avoidance methods are relatively simple, but are more influenced by the environment. The laser-based obstacle avoidance method has high accuracy and fast feedback, but the measured data is relatively single. Vision-based ob-

stacle avoidance can perceive three-dimensional space, but it is more dependent on the visibility of visible objects.

The on-board computer on the UAVs is equipped with different sensors and algorithms to complete obstacle avoidance experiments. For example, Devos et al.^[6] used a Raspberry Pi as the processor to run adaptive control algorithms with LIDAR sensors to design a rotorcraft autonomous obstacle avoidance system. McGuire et al.^[7] used a camera with an embedded STM32F4 microprocessor to calculate optical flow and stereo parallax to enable the rotorcraft to perform obstacle avoidance. Hu et al.^[8] designed a quadcopter UAV airborne ob-

*Corresponding author, E-mail address: weihe@ieee.org.

How to cite this article: FU Qiang, ZHANG Zuoqing, LIU Shengnan, et al. Design of an on-board vision-based obstacle avoidance system for flapping-wing robots[J]. Transactions of Nanjing University of Aeronautics and Astronautics, 2024, 41(1):97-106.

<http://dx.doi.org/10.16356/j.1005-1120.2024.01.008>

stacle avoidance system, which includes a RealSense camera, a processing board TK1, a flight controller board Pixhawk, and a ground station. In this system, they used image segmentation algorithms to obtain obstacle information.

Now UAVs can be roughly divided into three categories: Fixed-wing aerial vehicles, rotary-wing aerial vehicles and flapping-wing robots^[9]. Application of the on-board obstacle avoidance system in the fixed-wing and rotary-wing aerial vehicles is relatively mature. However, the design of the on-board obstacle avoidance system for FWRs is challenging. Compared to fixed-wing and rotary-wing aerial vehicles, FWRs are weaker in terms of weight-bearing capacity^[10], which makes light-weight vision sensors more suitable for obstacle avoidance of FWRs. A method of binocular obstacle avoidance using a deep learning-based single-shot multibox detector (SSD) for obstacle detection was applied with a Raspberry Pi 3B+ as the processor on a FWR^[11]. Compared with binocular vision, monocular vision is more simple and easy to implement. So Fu et al.^[12] designed a monocular-vision-based obstacle avoidance system for FWRs based on optical flow and fuzzy control. However, they used a ground station to process the obstacle avoidance algorithm of FWRs, which will cause communication delay and degrade the performance of obstacle avoidance.

This paper focuses on the design of an obstacle avoidance system based on an on-board monocular camera for our self-developed FWR named USTB-Hawk. The system has the following innovations: (1) The system is completely autonomous using an on-board computer with high arithmetic power to process the optical flow algorithm; (2) the weight-saving design is implemented for the on-board computer to ensure that USTB-Hawk can carry it; (3) this autonomous operation algorithm reduces the communication delay and improves the obstacle avoidance efficiency compared to the ground station processing algorithm.

The structural composition of this paper is described as follows. Section 1 describes the main hardware components of the system. Section 2 introduces the simple principle of the algorithm, the de-

ployment algorithm process, and some hardware communication protocols. Section 3 describes the design and results of the experiments. Finally, Section 4 discusses and analyzes the experimental results.

1 Hardware Components and Platforms

1.1 Flapping-wing robot

The FWR flies like a bird^[13], which can be applied in various fields^[14-15]. The obstacle avoidance system designed in this paper needs to work with a FWR that has comparatively strong load capacity. Therefore, we used a self-developed large-wing-span FWR named USTB-Hawk, which is shown in Fig.1. USTB-Hawk has a wingspan of 1.78 m and a bare weight of 985 g^[16] (The specific parameters are shown in Table 1). Its load weight is up to 300 g, while the weight of the obstacle avoidance system is less than 200 g. So USTB-Hawk meets our experimental requirements.



Fig.1 USTB-Hawk

Table 1 USTB-Hawk main specifications

Parameter	Value
Mass m / kg	0.985
Wingspan b / m	1.78
Wing surface S / m ²	0.52
Tail surface S_t / m ²	0.185
Flapping amplitude β / (°)	52
Flapping frequency f / Hz	2—5.5
Speed V / (m · s ⁻¹)	5—10
Endurance T / min	60
Battery capacity C / (mA·h)	7 000

The flight speed of USTB-Hawk in the experiment is 6 m/s. The empirical relationship between the turning radius and rudder angle of the FWR in a windless or breezy environment and at that speed

has been measured after extensive experiments. It is as

$$a = 120.73r^{-0.93} \quad (1)$$

where a is the sum of the upper and lower reverse angles of the left and right rudder surfaces; and r the turning radius of USTB-Hawk. In this formula, r should be greater than 1.5 m.

1.2 Image acquisition module

We use a USB industrial camera, which is shown in Fig.2. This camera can shoot images of $1\,920\text{ pixel} \times 1\,080\text{ pixel}$. And the maximum frame rate of this camera can reach 30 frame/s. The light weight and easy installation make it suitable for the system designed in this paper.



Fig.2 USB camera

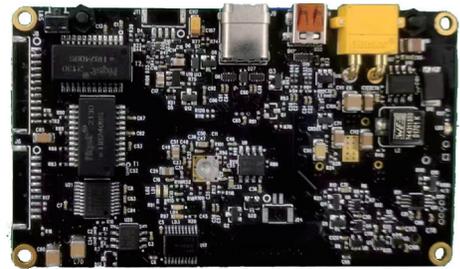
1.3 On-board processing module

We chose Jetson Nano as the on-board computer. The core board is shown in Fig.3(a). The most important reason why we use Jetson nano is that it includes a 128-core Maxwell architecture GPU. Thanks to this GPU, Jetson Nano can deploy and execute relatively simple deep-learning-based vision algorithms. In addition, the core board of Jetson Nano is removable, and it has the characteristics of small size and low power consumption. It is exactly the right fit for FWRs with a low load capacity.

The original interface board is large in size and mass, so we design a lightweight interface board as shown in Fig.3(b). The weight of this board is only 20 g after some useless interfaces are removed. The core board and the interface board make up the on-board processing module, and their total weight is 48 g.



(a) Core board



(b) Interface board

Fig.3 On-board processing module

1.4 Flight control module and ground station

USTB-Hawk is equipped with a mini-pix flight control board. The flight control module is responsible for obtaining the attitude information, controlling the movement pattern of the prototype and returning the data to the ground station. The ArduPilot Mega firmware is burned on the flight control board. The ground station uses Mission Planner to receive the information from USTB-Hawk.

2 Algorithm Deployment and Hardware Connectivity

2.1 Optical flow algorithm

Objects are in motion all the time, and this motion is observed in the image as the movement of pixel points. The velocity vector of the pixel points as they move forms the optical flow^[17]. The structure of the optical flow field is a useful piece of information that can be used for motion detection and target segmentation^[18]. Here we use the Farneback optical flow.

The estimation of the displacement in the Farneback optical flow algorithm utilizes a polynomial to model the image. The modeling relationship equation is as follows^[19]

$$f(x) \sim x^T Ax + b^T x + c \quad (2)$$

where \boldsymbol{x} is a matrix representation of image coordinates; $\boldsymbol{f}(\boldsymbol{x})$ a signal in the same dimension as \boldsymbol{x} , \boldsymbol{A} a symmetric matrix, and it is reversible; \boldsymbol{b} a column vector and c a scalar.

The modeling approach from Eq.(2) assumes that the pixel neighborhood in a two-dimensional plane is represented as

$$\boldsymbol{f}_1(\boldsymbol{x}) = \boldsymbol{x}^T \boldsymbol{A}_1 \boldsymbol{x} + \boldsymbol{b}_1^T \boldsymbol{x} + c_1 \quad (3)$$

The neighborhood after the introduction of a displacement \boldsymbol{d} is

$$\boldsymbol{f}_2(\boldsymbol{x}) = \boldsymbol{x}^T \boldsymbol{A}_2 \boldsymbol{x} + \boldsymbol{b}_2^T \boldsymbol{x} + c_2 \quad (4)$$

Based on Eqs.(3,4), \boldsymbol{A}_2 , \boldsymbol{b}_2 and c_2 are related as follows

$$\boldsymbol{A}_2 = \boldsymbol{A}_1 \quad (5)$$

$$\boldsymbol{b}_2 = \boldsymbol{b}_1 - 2\boldsymbol{A}_1 \boldsymbol{d} \quad (6)$$

$$c_2 = \boldsymbol{d}^T \boldsymbol{A}_1 \boldsymbol{d} - \boldsymbol{b}_1^T \boldsymbol{d} + c_1 \quad (7)$$

Displacement \boldsymbol{d} can be derived

$$\boldsymbol{d} = -\frac{1}{2} \boldsymbol{A}_1^{-1} (\boldsymbol{b}_2 - \boldsymbol{b}_1) \quad (8)$$

This process holds for signals of any dimension. The `cv2.calcOpticalFlowFarneback()` function is a python implementation of the Gunnar Farneback algorithm for calculating dense optical flows.

We deploy the dense optical flow algorithm on Jetson Nano and try to accelerate it. OpenCV is a library that contains many machine vision algorithms. CUDA is a GPU acceleration tool for NVIDIA graphics cards. Many image processing running in OpenCV use the GPU by default, so CUDA is used in this paper for OpenCV to accelerate the image processing speed.

Once the system environment variables are configured on Jetson Nano, the program is ready to run. A sample image of the testing results is shown in Fig.4. From Fig.4, it is found that trees, curb-

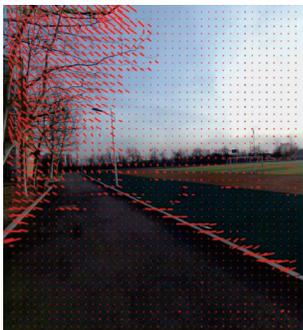


Fig.4 Sample image of the testing results

stones and street light poles can be detected by the optical flow algorithm. The algorithm is successfully deployed.

2.2 Control strategy

In this paper, there is a lack of depth information in vision-based obstacle avoidance, and it can be said that there is no direct measurement of obstacle distance. The optical flow difference is used to indirectly measure the distance information under certain conditions, so as to map the magnitude of the obstacle avoidance action. The premise of the proposed obstacle avoidance method designed in this paper is to control the FWR flight at a fixed altitude of 20 m and the surrounding wind speed is assumed to be less than 5.4 m/s (breeze weather).

Our experimental configurations are: Known flight speed of 6 m/s, known image resolution of the camera of 1 920 pixel \times 1 080 pixel, and a uniformly set optical flow calculation interval.

The optical flow matrix calculated in the algorithm is the displacement of the pixel points in the optical flow calculation area, and contains two quantities: The horizontal displacement \boldsymbol{x} and the vertical displacement \boldsymbol{y} . In this paper, we calculate \boldsymbol{x} in the left half view (X_L) and \boldsymbol{x} in the right half view (X_R), respectively. In the image, it is specified that from left to right is \boldsymbol{x} forward, and bottom up is \boldsymbol{y} forward. The optical flow points towards both sides of the image in the forward process, so $X_L < 0$, $X_R > 0$.

After extensive experiments, it is proved that the value of Eq.(9) can be used as a basis for the mapping distance under the set conditions.

$$S_{RL} = X_R - X_L \quad (9)$$

where S_{RL} is the sum of the absolute values of the lateral displacement of the optical flow in the left and right fields of view. The minimum turning radius of USTB-Hawk is 1.5 m, so in the experiment we set the safety distance to 4 m. At this point, $S_{RL} \approx 6\ 330$. We also set the minimum distance to start the execution of obstacle avoidance mode to 10 m and $S_{RL} \approx 2\ 000$. Logarithmic regression of the previously accumulated experimental data yields the relationship between S_{RL} and the distance (D_{IS}) be-

tween the obstacle and the USTB-Hawk, which is

$$D_{IS} = 51.1 - 5.38 \ln S_{RL} \quad (10)$$

We ensure that the distance is greater than the turning radius. The relationship between the rudder angle and S_{RL} can be derived by Eqs.(1, 10)

$$a = \begin{cases} 0 & S_{RL} < 2000 \\ 120.73 \times (51.1 - 5.38 \ln S_{RL})^{-0.93} & 2000 \leq S_{RL} \leq 6330 \\ 90 & S_{RL} > 6330 \end{cases} \quad (11)$$

2.3 Communication between major hardware

The communication between the on-board computer and the flight control board is achieved by the MAVLink protocol. MAVLink is a lightweight communication protocol between ground stations and UAVs. It defines a set of information links between the UAV and the ground station, and this information exchange is bidirectional^[20]. During the serial communication between the on-board computer and the flight control board, the on-board computer sends the output of the algorithm to the flight control board for the next action.

Two Xbee modules are used for wireless communication between the ground station and the flight control board. The ground station is used as a monitoring and control platform to send and receive data in real time. These data include the route, altitude, speed and flight mode, etc. Once the Xbee module is configured, we can use the serial port to communicate with the flight control board and the ground station.

2.4 Overall workflow

The internal communication of the overall on-board obstacle avoidance system is shown in Fig.5. The working sequence and operation logic of each component is shown in Fig.6.

First, USTB-Hawk flies according to the route set by the ground station. After USTB-Hawk takes off, USB camera captures a first-view video and transmits it to the on-board computer. Second, the on-board computer runs the optical flow obstacle avoidance algorithm to process the incoming video in real time. In this process, X_R , X_L and S_{RL} are calculated, compared and judged. If S_{RL} is less

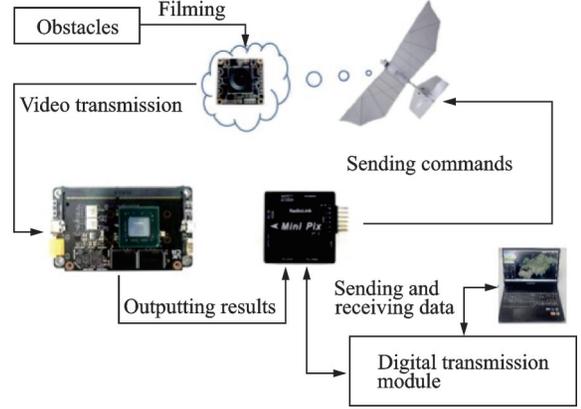


Fig.5 System communication diagram

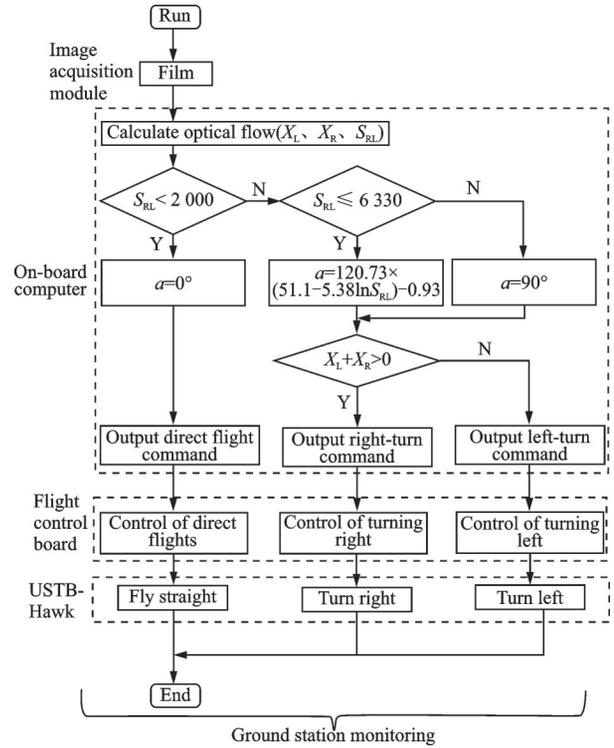


Fig.6 System logic diagram

than 2000, it is assumed that no obstacle is ahead. Otherwise, the corresponding rudder angle and turn-direction are calculated based on the value of S_{RL} . Third, the on-board computer outputs the commands to the flight control board. Finally, USTB-Hawk completes the action according to the commands.

Note that during the whole operation of the obstacle avoidance system, the ground station does not involve in any calculations and only plays a monitoring role. All the image processing computations are implemented by using the on-board computer.

3 Experiment

3.1 Experimental design

The obstacles in the experiment are four buildings and two connecting corridors. They are named Building A, Building B, Building C, Corridor D, Corridor E, and Building F. Their locations are distributed as shown in Fig.7. Next to these buildings there is a playground we can use to set routes.

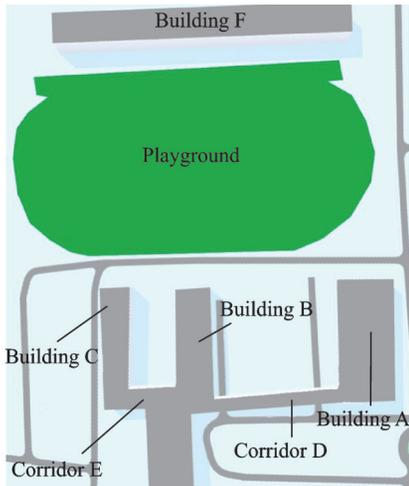


Fig.7 Topographic map

We set four different routes for USTB-Hawk on the Mission Planner ground station and set the flight altitude of USTB-Hawk to 20 m, as shown in Fig.8. Among them, the red, the yellow, and the blue are the simpler experimental routes, while the green routes are more complex. USTB-Hawk takes off from the playground and flies along the route. When flying to point 2 of the route setting, USTB-Hawk turns on the obstacle avoidance mode. If the obstacle avoidance is successful, and after manually closing the obstacle avoidance mode, USTB-Hawk flies to the next route to prepare for the next obstacle avoidance.

We also design a comparative experiment by using a laser ranging module. The distance measuring module we choose is “TFmini-S”, which can measure up to 20 m in the actual test. We set the obstacle avoidance distance to 4—10 m, when using the laser ranging module for obstacle avoidance. In our comparison experiment, USTB-Hawk

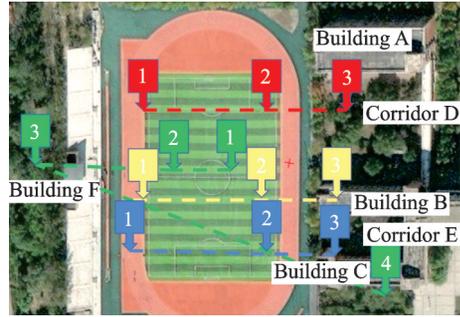


Fig.8 Route setting

equipped with this sensor flies directly to Building B, and we observe its obstacle avoidance response.

Finally, we test the latency of using ground station processing in order to highlight the benefits of on-board processing. In this experiment, it is possible to ensure that the other experimental conditions are the same. We use the DJI FPV digital video transmission system as the hardware for image transmission, XBee as the hardware for data transmission, and Jetson nano as the image processing system.

3.2 Experimental results

Through flight experiments, we recorded the first view and third view image data of USTB-Hawk. In the actual test, USTB-Hawk performed four obstacle avoidance actions and successfully avoided all obstacles.

USTB-Hawk flew according to the yellow route shown in Fig.8 until the obstacle avoidance conditions were met. The first obstacle avoidance action of USTB-Hawk was to execute the command to turn left and avoid Building A, Building B and Corridor D. At this point ($X_L = -1\ 986$, $X_R = 2\ 461$), USTB-Hawk turned left to avoid the obstacles. A sample image of the first view of USTB-Hawk on the first route is shown in Fig.9. The third view of USTB-Hawk’s obstacle avoidance behavior is shown in Fig.10. The blue arrow represents its obstacle avoidance trajectory and route.

When it flew according to the red route in Fig.8, Building A and Corridor D were also detected in the second obstacle avoidance action of USTB-Hawk.



Fig.9 Sample image of the first view of the first obstacle avoidance



Fig.10 Sample image of the third view of the first obstacle avoidance

At this point ($X_L = -1\ 253$, $X_R = 1\ 727$), USTB-Hawk turned left to avoid the obstacles, and changed the original straight route to a left-turn route. A sample image of the third view of USTB-Hawk's second obstacle avoidance behavior is shown in Fig.11.

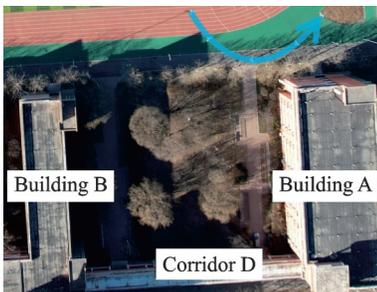


Fig.11 Sample image of the third view of the second obstacle avoidance

USTB-Hawk avoided Building B, Building C and Corridor E during the blue route in Fig. 8. At the start of obstacle avoidance ($X_L = -1\ 856$, $X_R = 1\ 435$), it executed a right turn command. A sample image of the third view of USTB-Hawk's third obstacle avoidance behavior is shown in Fig.12.

In the last complex obstacle avoidance process, USTB-Hawk first avoided Building F, then flew according to the route, and finally avoided Building C, as shown in Fig.13.



Fig.12 Sample image of the third view of the third obstacle avoidance



Fig.13 Sample image of the third view of the fourth obstacle avoidance

In addition, we extracted a portion of the optical flow frames from takeoff to obstacle avoidance. We extracted the S_{RL} values and plotted Figs.14—17.

The two lines in the diagrams represent the case where $S_{RL} = 2\ 000$ and $S_{RL} = 6\ 330$, respectively. These two values are the thresholds for ob-

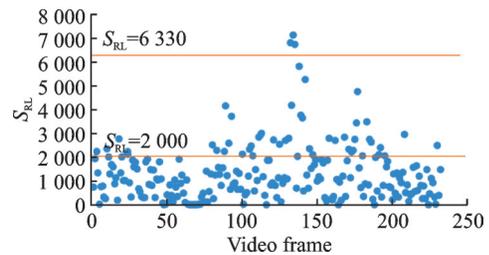


Fig.14 S_{RL} scatter plot for the first experiment

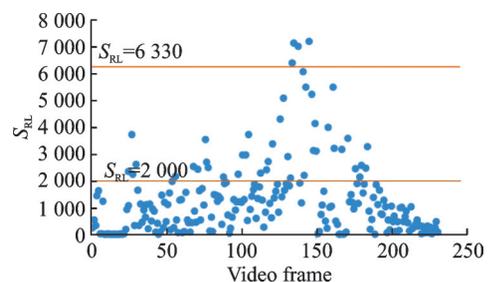
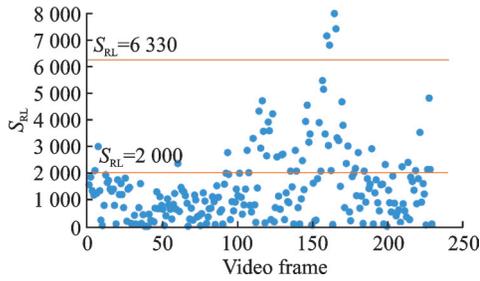
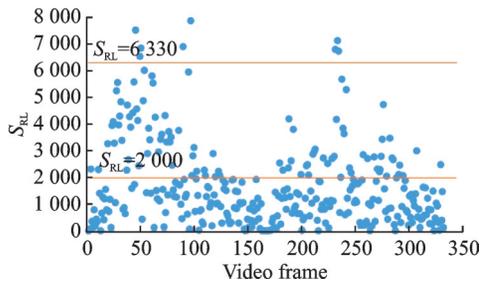


Fig.15 S_{RL} scatter plot for the second experiment

Fig.16 S_{RL} scatter plot for the third experimentFig.17 S_{RL} scatter plot for the fourth experiment

stacle avoidance actions mentioned above. A peak can be observed in the data of a single obstacle avoidance experiment (Figs.14—16), which is the maximum value of the S_{RL} . Two peaks can be seen in the fourth experimental result (Fig.17), which is the result of continuous obstacle avoidance. These can verify that the logic of our algorithm is feasible.

In the process of using the laser ranging sensor, we found that the distance between USTB-Hawk and the obstacle can be obtained directly. However, in this case we cannot get additional information to determine whether to avoid the obstacle by turning left or right, which results in the randomness of obstacle avoidance. In contrast, optical flow obstacle avoidance method can obtain rich visual information that is sufficient to determine what kind of obstacle avoidance action to perform. In the experiments with the laser ranging module, two left-turn obstacle avoidance experiments failed, only one right-turn successfully avoided the obstacle. Fig.18 records the distance measured by the laser ranging module over three experiments as a function of time.

Finally, we tested the latency of ground station processing compared to on-board processing. The delay consists of two parts: Delay of image transmission and delay of data transmission. The average

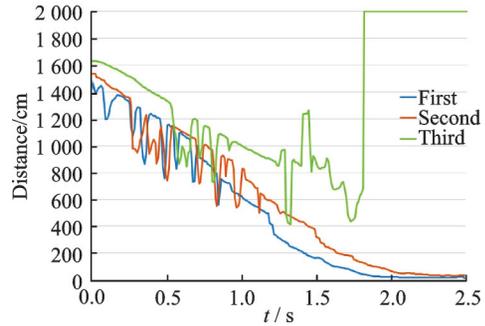


Fig.18 Distance measured by the laser ranging module in three different experiments

delay of the digital video transmission system within 100 m is 30 ms. The average data transmission delay is 100 ms. But on-board processing does not have the average latency of 130 ms compared to the ground station processing. Although the delays are in the millisecond range, our FWR flies fast, and these delays still have an adverse impact.

4 Conclusions

An obstacle avoidance system based on a monocular camera and an on-board computer is established. The system is lightweight designed to meet the requirements of the FWR with low load capacity. The on-board computer processes the first view of the optical flow information in real time and calculates the corresponding turn direction and rudder angle. This processing mode avoids the delay caused by ground station processing. The communication between the flight control board and the ground station allows us to monitor the flight status of USTB-Hawk at any time. The success of the experiments prove that the obstacle avoidance system is effective.

In future work, we will add an electronic stabilization algorithm to further improve the obstacle avoidance performance.

References

- [1] CHEN M, TAO G. Adaptive fault-tolerant control of uncertain nonlinear large-scale systems with unknown dead zone[J]. IEEE Transactions on Cybernetics, 2016, 46(8): 1851-1862.
- [2] REN Y, CHEN M, LIU J Y. Bilateral coordinate boundary adaptive control for a helicopter lifting system with backlash-like hysteresis[J]. Science China In-

- formation Sciences, 2020, 63: 1-3.
- [3] ZHAO X X, WANG H X, LU X. On obstacle avoidance of multiple ultrasonic sensors based on Aloha robot[J]. Applied Mechanics and Materials, 2013, 336/337/338: 1059-1062.
- [4] GAYA J O, GONCALVES L T, DUARTE A C, et al. Vision-based obstacle avoidance using deep learning[C]//Proceedings of 2016 XIII Latin American Robotics Symposium and IV Brazilian Robotics Symposium (LARS/SBR). Recife, Brazil: IEEE, 2016: 7-12.
- [5] DUMITRASCU B, FILIPESCU A, PETREA G, et al. Laser-based obstacle avoidance algorithm for four driving/steering wheels autonomous vehicle [C]//Proceedings of System Theory, Control & Computing. Sinaia, Romania: IEEE, 2013: 187-192.
- [6] DEVOS A, EBEID E, MANOONPONG P. Development of autonomous drones for adaptive obstacle avoidance in real world environments[C]//Proceedings of the 21st Euromicro Conference on Digital System Design (DSD). Prague, Czech Republic: IEEE, 2018: 707-710.
- [7] MCGUIRE K, DE C G, De WAGTER C, et al. Efficient optical flow and stereo vision for velocity estimation and obstacle avoidance on an autonomous pocket drone[J]. IEEE Robotics & Automation Letters, 2017, 2(2): 1070-1076.
- [8] HU J, NIU Y F, WANG Z C. Obstacle avoidance methods for rotor UAVs using RealSense camera[C]// Proceedings of 2017 Chinese Automation Congress (CAC). Jinan, China: IEEE, 2017: 7151-7155.
- [9] HE W, MU X X, ZHANG L, et al. Modeling and trajectory tracking control of bird-like flapping-wing micro air vehicles[J]. IEEE/CAA Journal of Automatica Sinica, 2021, 8(1): 148-156.
- [10] FU Q, CHEN X Y, ZHENG Z L, et al. Research progress on visual perception system of bionic flapping-wing aerial vehicles[J]. Chinese Journal of Engineering, 2019, 41(12): 1512-1519.
- [11] FU Q, CHEN X Y, SHANG Y L. Vision-based obstacle avoidance for flapping-wing aerial vehicles[J]. Science China Information Sciences, 2020, 63(7): 115-117.
- [12] FU Q, WANG J, GONG L, et al. Obstacle avoidance of flapping-wing air vehicles based on optical flow and fuzzy control[J]. Transactions of Nanjing University of Aeronautics and Astronautics, 2021, 38(2): 206-215.
- [13] YIN Z, HE W, ZOU Y, et al. Efficient formation of flapping-wing aerial vehicles based on wild geese queue effect[J]. Acta Automatica Sinica, 2020, 46: 1-13.
- [14] YUAN C S, LI Y Z, TAN J. Investigation in flight control system of flapping-wing micro air vehicles[J]. Computer Measurement & Control, 2011, 19(7): 1527-1529.
- [15] PIETRUHA J, SIBILSKI K, LASEK M, et al. Analogies between rotary and flapping wings from control theory point of view[C]//Proceedings of AIAA Atmospheric Flight Mechanics Conference and Exhibit. Montreal, Canada: AIAA, 2001: 4002.
- [16] WU X Y, HE W, WANG Q, et al. A long-endurance flapping-wing robot based on mass distribution and energy consumption method[J]. IEEE Transactions on Industrial Electronics, 2022, 70(8): 8215-8224.
- [17] EGELHAAF M, KERN R. Vision in flying insects [J]. Current Opinion in Neurobiology, 2002, 12(6): 699-706.
- [18] BEAUCHEMIN S S, BARRON J L. The computation of optical flow[J]. ACM Computing Surveys (CSUR), 1995, 27(3): 433-466.
- [19] FARNEBACK G. Two-frame motion estimation based on polynomial expansion[C]//Proceedings of the 13th Scandinavian Conference on Image Analysis (SCIA 2003). Berlin, Germany: Springer, 2003: 363-370.
- [20] ALLOUCH A, CHEIKHROUHOU O, KOUBAA A, et al. MAVSec: Securing the MAVLink protocol for ardupilot/PX4 unmanned aerial systems[C]//Proceedings of the 15th International Wireless Communications & Mobile Computing Conference (IWCMC). Tangier, Morocco: IEEE, 2019: 621-628.

Acknowledgements This work was supported by the National Natural Science Foundation of China (Nos.62225304, 61933001, 62173031, U20A20225), the Interdisciplinary Research Project for Young Teachers of Fundamental Research Funds for the Central Universities (No.FRF-IDRY-22-029), and the Beijing Top Discipline for Artificial Intelligent Science and Engineering, University of Science and Technology Beijing.

Authors Dr. FU Qiang received the B.S. degree in thermal energy and power engineering from Beijing Jiaotong University, Beijing, China, in 2009, and the Ph.D. degree in control science and engineering from Beihang University (formerly Beijing University of Aeronautics and Astronautics),

Beijing, in 2016. He is currently an associate professor with the School of Intelligence Science and Technology and Institute of Artificial Intelligence, University of Science and Technology Beijing. His main research interests include vision-based navigation, visual servoing and flapping-wing robots.

Prof. **HE Wei** received his B.Eng. and his M.Eng. degrees from College of Automation Science and Engineering, South China University of Technology (SCUT), China, in 2006 and 2008, respectively, and his Ph.D. degree from Department of Electrical & Computer Engineering, the National University of Singapore (NUS), Singapore, in 2011. He is currently working as a full professor in School of Intelligence Science and Technology and Institute of Artificial Intelli-

gence, University of Science and Technology Beijing, Beijing, China. His current research interests include robotics and intelligent control systems.

Author contributions Prof. **FU Qiang** directed the research and revised the manuscript. Mr. **ZHANG Zuoqing** designed the study, programmed, deployed it on the experimental platform, analyzed the data and wrote the manuscript. Dr. **LIU Shengnan** and Dr. **WU Xiaoyang** helped with the experiment. Prof. **HE Wei** revised the manuscript. All authors commented on the manuscript draft and approved the submission.

Competing interests The authors declare no competing interests.

(Production Editor: ZHANG Bei)

扑翼机器人机载视觉避障系统设计

付 强^{1,2,3}, 张祚卿^{1,2,3}, 刘胜南^{1,2,3}, 吴晓阳^{1,2,3}, 贺 威^{1,2,3}

(1. 北京科技大学智能科学与技术学院, 北京 100083, 中国; 2. 北京科技大学人工智能研究院, 北京 100083, 中国; 3. 北京科技大学智能仿生无人系统教育部重点实验室, 北京 100083, 中国)

摘要:扑翼机器人是一种高仿生度的飞行器,它可以执行军事侦察和民用监测任务。在执行这些任务时,避障是一种确保扑翼机器人安全的必要功能。本文设计了一种基于单目视觉的扑翼机器人自主避障系统,其中所有图像的处理计算均使用机载处理器实现。在这个系统中,机载处理器的质量被减至48 g,以便于扑翼机器人可以携带它稳定飞行。该系统的工作流程可以分为以下几步:首先,图像采集模块获取周围环境的视频图像;然后,机载处理器通过处理扑翼机器人第一视角的光流信息来计算方向舵角度和转弯方向;最后,飞控板接收计算结果并控制扑翼机器人避开障碍物。地面站对扑翼机器人飞行过程进行实时监测,实验结果验证了本文所设计的避障系统的有效性。

关键词:扑翼机器人;视觉;避障;机载处理器;光流