robotics

Volume 5, Issue 3, 2016, Article number 18.
DOI:10.3390/robotics5030018

MDPI

Article

Bio-Inspired Vision-Based Leader-Follower Formation Flying in the Presence of Delays

John Oyekan

School of Aerospace, Transport and Manufacturing; Cranfield University, Bedfordshire, Cranfield MK43 0AL, UK; j.o.oyekan@cranfield.ac.uk; Tel.: +44-(0)1234-750-111

Academic Editor: Huosheng Hu
Received: 17 June 2016; Accepted: 11 August 2016; Published: 18 August 2016

Abstract: Flocking starlings at dusk are known for the mesmerizing and intricate shapes they generate, as well as how fluid these shapes change. They seem to do this effortlessly. Real-life vision-based flocking has not been achieved in micro-UAVs (micro Unmanned Aerial Vehicles) to date. Towards this goal, we make three contributions in this paper: (i) we used a computational approach to develop a bio-inspired architecture for vision-based Leader-Follower formation flying on two micro-UAVs. We believe that the minimal computational cost of the resulting algorithm makes it suitable for object detection and tracking during high-speed flocking; (ii) we show that provided delays in the control loop of a micro-UAV are below a critical value, Kalman filter-based estimation algorithms are not required to achieve Leader-Follower formation flying; (iii) unlike previous approaches, we do not use external observers, such as GPS signals or synchronized communication with flock members. These three contributions could be useful in achieving vision-based flocking in GPS-denied environments on computationally-limited agents.

Keywords: bio-inspiration; flocking; vision; delays; communication

1. Introduction

Flocking in micro-UAVs has been attempted in many forms with varying degrees of success. Some attempts have used global observers (e.g., GPS signals or motion capture systems) to obtain positions of flock members for formation flying [1–3]. For example, Hauert et al. used a strategy that involved the combination of communication and GPS signals to achieve outdoor flocking of fixed wing robots [1]. GPS signals were used to obtain the position of each robot. These positions were then communicated to the other members of the flock while Reynolds laws of flocking [4] were used to obtain flock control. Reliance on GPS signals was also used by Vásárhelyi et al. [3] in achieving decentralized flocking of 10 quadrotors in an outdoor environment. However, this approach fails when GPS signals are lost, such as in caves, under heavy forest canopy, or even between tall buildings in urban environments.

Looking to nature, swarms are able to cope with a dynamic environment and keep formation without the need for global position information or explicit communication between members. They are able to do this through the primary use of visual cues derived as a result of the structure of their retinal pathways. It is postulated in this paper that having an overview of the structure of these pathways will enable us to infer the "computations" taking place during their flocking behavior and will guide us to design better vision-based flocking algorithms for micro-UAVs.

Starlings are known to flock at dusk. With the low visibility challenges that come with low light conditions, they are still able to maintain a cohesive formation whilst avoiding collisions with each other. From a computer engineering perspective, they must perform computations that enable them to identify each other, detect the distance to a neighbor, as well as detect the bearing to another individual

within a very short window of time. Outside this critical window, collisions are more likely to happen and flocking behavior will most likely not emerge [5,6].

It is quite likely that, in order to meet this critical window, the computations to achieve flocking in natural systems happen in or close to the retina [7,8]. For example, it has been observed that direction, light, object, texture, distance and motion detection all take place in mammal's retinas and not the visual cortex [6,8–12]. Therefore, it is important to study these retinal "computations" for the purposes of building better, robust, and scalable vision-based flocking algorithms. Most of the research in this area has focused on the subset problem of vision-based Leader-Follower algorithms.

Research in vision-based Leader-Follower algorithms for micro-UAVs often make use of a known mathematical model for the Leader in state estimation algorithms, as well as GPS or visual markers on the Leader [13–16]. State estimation algorithms, such as Kalman filters, are often used as a default to compensate for delays or momentary loss of GPS signals. However, are these state estimation algorithms really necessary to achieve Leader-Follower behavior on micro-UAVs? For example, Cowan et al. argued that if the constraint of maintaining visibility with the Leader is satisfied and as long as the Leader keeps moving, any linear controller, which does not depend on Leader pose estimation, is sufficient to achieve and stabilize Leader-Follower formation flying [5].

Nevertheless, Cowan et al. did not consider the effects of delays in the control loop of the Follower. Delays in the control loop could be caused by factors such as communication delays, visual sensor refresh rates, complexity of vision processing algorithms and delays in the control algorithm approach used. Looking at literature, the effects of communication delays has mostly been investigated during synchronized formation flight [17–19] while state estimation algorithms have been mostly used to deal with noisy and intermittent sensor signals [15,16].

In this work, the following three contributions are made: (i) using a computational methodology [20], the mechanisms discussed in [6] and the references there in were used to derive a high level functional architecture of a vertebrate's visual pathway. This architecture was used to derive: (a) an algorithm capable of extracting the pose of a Leader from the visual field of the Follower and (b) a vision-based Leader-Follower linear controller; (ii) using our developed controller, we show that as long as the delay in the control loop of the Follower is not above a certain critical threshold, we can achieve Leader-Follower formation flying. This is provided that the lighting conditions required to maintain visibility with the Leader is observed and that there are no occlusions between the Leader and the Follower [5,15]; (iii) unlike previous approaches, we do not use external observers, state estimation algorithms, or synchronized communication with flock members,. We only make use of visual information from an onboard camera on a micro-UAV to achieve Leader-Follower formation flying. Our approach is validated with simulations as well as actual physical experiments on micro-UAVs.

The rest of the paper is organized as follows: in Section 2, we present the problem statement followed by Section 3 in which we derive a controller that is robust to delays; Section 4 presents the algorithmic development of the derived controller on a chosen micro-UAV platform while, in Section 5, the results of simulations are presented after which physical experimental results are presented in Section 6; a discussion of our work is presented in Section 7 and conclusions follow in Section 8.

2. Problem Statement

As mentioned in the introduction, delays in the control loop are generated as a result of the following factors: (1) communication delay γ_c between a micro-UAV and a base station; (2) visual sensor refresh rates γ_v; (3) delay caused by the computing time taken to process images γ_i [5]; and (4) delay caused by the computing time taken to perform state estimation and other algorithms γ_a for achieving micro-UAV control [5,21]. This is given by Equation (1):

$$t_{delay} = \gamma_c + \gamma_v + \gamma_i + \gamma_a \tag{1}$$

In order to reduce or eliminate the delays caused by factor (3), we used the approach discussed in Section 4.2. To reduce the delays caused by factor (4), we try to keep computations minimal by not using state estimation algorithms. The values of the delays caused by factor (1) and (2) vary as discussed in [5] and affect the observation of the Leader by the Follower. These two factors are platform and hardware dependent and as a result cannot be controlled.

Nevertheless, in the next section, we derive a Leader-Follower controller that is able to maintain formation flying regardless of the platform dependent delays.

3. Deriving a Delay Robust Leader-Follower Formation Controller

In this section, we derive a delay robust Leader-Follower formation controller and show that, provided the value of the delay t_{delay} is below a certain value τ, our approach to Leader-Follower formation flying will work without the need of a state estimation algorithm.

3.1. Assumptions

Assumption 1. The Leader L is a moving fixed point with forward speed V_L. The Follower F has a speed V_F that can be adjusted to keep up with the Leader.

Assumption 2. There is a system of three virtual springs that connects the Leader L to the Follower F as shown in Figure 1 with a spring AB connecting point A on the Follower to point B on the Leader. The force generated by the spring AB is dependent on the forward distance between the Leader and Follower. Another spring BC connects the points B and C. The force generated by the spring is dependent on the vertical distance between the Follower and Leader while a third spring BD connects points B and D and the force is dependent on the horizontal distance between the Follower and Leader. We describe the entire system using the Hooke's law of spring as in Equation (2) where for simplicity, $\hat{K_S}$ embeds the stiffness constants (k_{AB}, k_{BD}, k_{BC}) of the three springs. H is the force generated by the corresponding springs, $\hat{r}$ is the rest length of the springs while the stretched or compressed lengths are given by $\hat{\rho}$:

$$\hat{H} = \hat{K_S}(\hat{\rho} - \hat{r}) \tag{2}$$

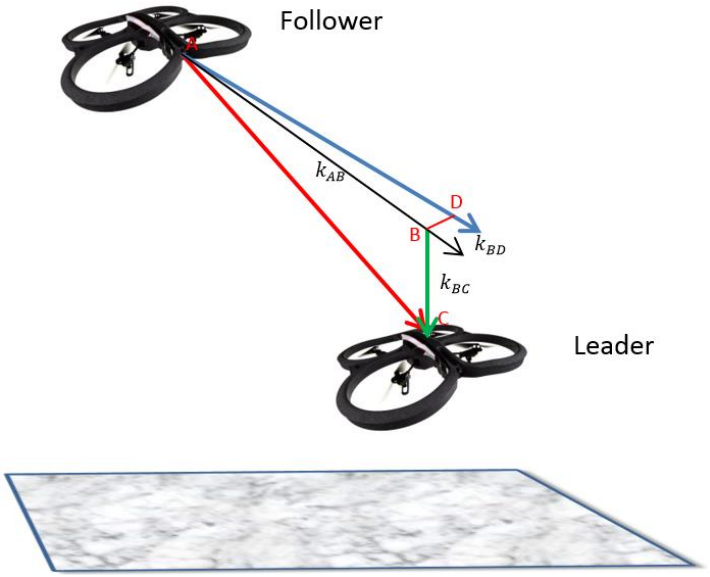


Figure 1. The set of virtual springs (AB, BC, and BD) connecting the Follower to the Leader.

Assumption 3. Using Figure 2, we assume three zones with distance threshold values called *near zone*, *mid-zone* and *far zone*. It is assumed that the value of $\hat{r}$ from Equation (2) is given by the *mid-zone*; while the stretched or compressed length $\hat{\rho}$ is given by the values of the *far zone* or the *near*

zone, respectively. Since $\hat{r}$ and $\hat{\rho}$ are distance terms, we can convert them into velocity components as depicted in Equation (3):

$$H = \hat{K_S}(V_{CS} - V_{mid})\Delta t \tag{3}$$

where V_{CS} is the compressed or stretched velocity as a result of the Follower in the *near* or *far zone*, respectively. V_{mid} is the rest velocity in the *mid-zone*. Equation (3) is rewritten as follows:

$$V_F = \begin{cases} V_{CS} = V_{far} \; if \; follower \; is \; in \; far \; zone \\ V_{mid} \; if \; follower \; is \; in \; mid \; zone \\ V_{CS} = V_{near} \; if \; follower \; is \; in \; near \; zone \end{cases} \tag{4}$$

Assumption 4. Every spring has a breaking point or stretching limit beyond which they will not recover to the rest length and they will be permanently deformed. In the Leader-Follower link, this breaking point is assumed to be the maximum visual range $Visual_{max}$ beyond which the Leader L will not be observable.

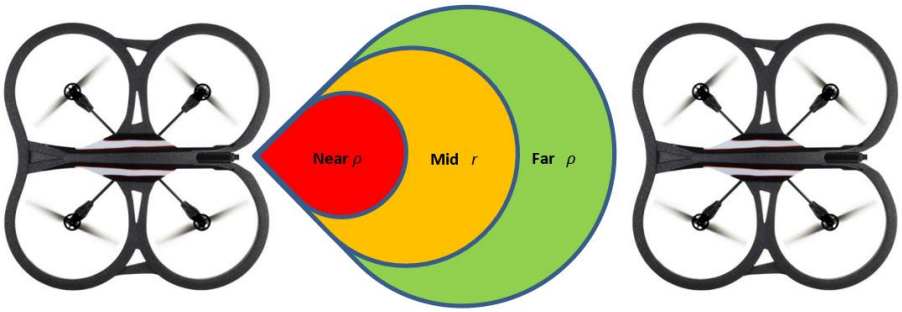


Figure 2. The zones used in Leader-Follower formation flying and their relationship to Equation (2).

3.2. Proof

Assumption 4 leads to the constraint defined in Equation (5) that shows, provided that a distance S_{Break} between the Leader L and Follower F is less than $Visual_{max}$ and greater than 0, the Leader-Follower formation flying will take place. Contradicting the constraint > 0 will lead to a collision or if the value is negative will not lead to flocking:

$$Visual_{max} \geq S_{Break} > 0 \tag{5}$$

$$S_{Break} = \int_{t}^{t+\Delta t} \mu(V_L - V_F)\Delta t_F \tag{6}$$

S_{Break} is given by Equation (6) where V_L is the velocity value of the Leader, V_F is the velocity of the Follower which can be controlled and Δt_F is assumed to be the delay t_{delay} in the Follower. Provided that the delay is below a critical value τ that does not violate the constraint in Equation (5), then Leader-Follower formation flying will take place. μ is a scaling constant.

4. Developing the Delay Robust Leader-Follower Formation Flying Controller

In this section, we discuss the platform and setup used for Leader-Follower formation flying, the derived biologically-inspired architecture, as well as the linear controller developed in order to meet the constraint identified in Equation (5).

4.1. The Micro-UAV Platform

The micro-UAV platform used in this work has four rotors. By adjusting their thrust, it is possible to achieve the position vector control (x, y, z) of the platform in the inertial reference frame according to Equation (7) [5,21,22]:

$$
\begin{aligned}
m\ddot{x} &= -u(\cos(\theta)sin(\varphi)cos(\varnothing) + sin(\theta)\sin(\varnothing)) \\
m\ddot{y} &= -u(\sin(\theta)sin(\varphi)cos(\varnothing) - cos(\theta)sin(\varnothing)) \\
m\ddot{z} &= -u(\cos(\varphi)cos(\varnothing)) + mg \\
u &= \sum_{i=1}^{4} f_i
\end{aligned} \tag{7}
$$

where f_i is the force generated by motor i, $\varnothing$ is the roll, φ is the pitch, and θ is the yaw. The values for roll, pitch, and yaw were obtained from the gyroscope on board the platform. The unmanned aerial vehicle platform also has an ultrasonic sensor for altitude measurements.

It is equipped with two monocular cameras, one forward facing and the other downward facing. As a result of the single forward facing camera, only monocular vision could be achieved. Images obtained from the front facing camera had a resolution of 320 × 240 pixels and a field of view of 73.5° × 58.5°. The images from the downward facing camera had a resolution of 176 × 144 with a field of view of 47.5° × 36.5°.

Due to the platform's frontal monocular vision, a bio-inspired lateral visual system was investigated. This was developed in software using Java and the OpenCV 2.3.1 library. The software was developed on a laptop tethered via WiFi to the platform. The laptop was running windows 7 on a dual core processor with 3 Gb of RAM. Each platform used in our experiments had its own tethered laptop. The images and data from the onboard sensors were streamed via WiFi to the laptop.

4.2. Bio-Inspired Visual Processing and Object Detection (Figure 3B,C))

Detecting flockmates is the first necessary step in flocking, as this would be used to determine the estimated time to collision, as well as other flocking parameters. The mechanism of object detection in biological systems is a non-trivial task and researchers are still in the process of decoding the neurological visual processes involved.

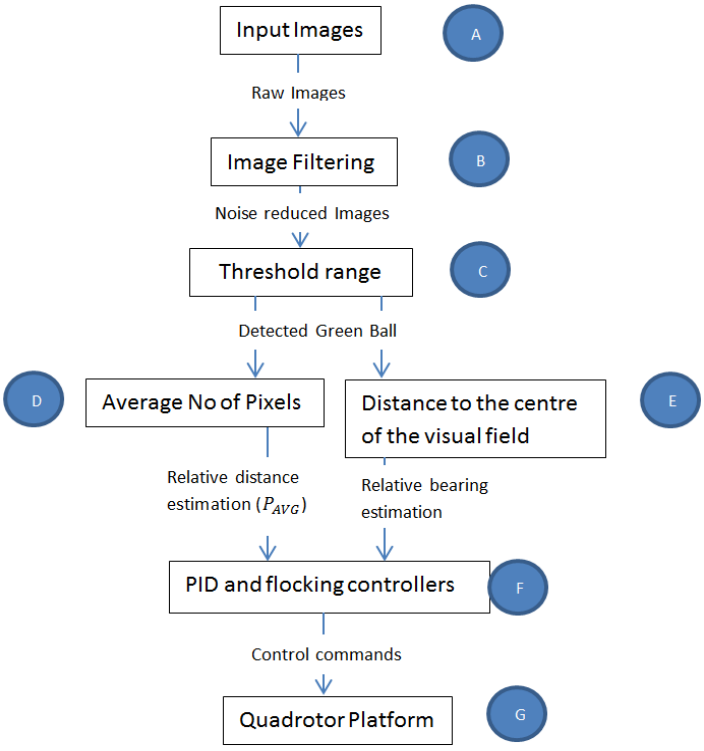


Figure 3. Visuomotor architecture to track an object to achieve Leader-Follower formation flight.

Nevertheless, in [6,7,23], Baccus, Eizaburo, and Gollisch et al. respectively argue that computations necessary for detecting object motion, distance, texture, and light are carried out in higher vertebrates' retinas before being passed to the brain for further processing. They argued that these functions are achieved by the photo-receptors using a combination of temporal filtering, threshold values and summing. Temporal filtering is used to filter out noise while thresholds are used to discard any signal value below a particular value. The filtered threshold values are then summed. Gollisch and Meister mentioned that it is essential that temporal filtering and the use of threshold is carried out before summing in order to avoid amplifying noise in each individual rod [6].

Taking inspiration from [6,23], a visuomotor architecture as shown in Figure 3 was developed. The term visuomotor is used here because it is often used in literature to describe how vision is transformed into motor actions. It was assumed that a pixel in the camera represented a biological rod and produced input images (Figure 3A) to the rest of the vision-motion architecture.

Temporal filtering was achieved through Equation (8), where I_t and I_{t-1} are the processed image output at t and $t-1$, respectively. R_t is the raw image input obtained at Figure 3A at time t with α being a gain value for choosing how much of I_{t-1} to include in I_t, where $0 < \alpha < 1$.

$$I_t = \propto I_{t-1} + (1 - \propto) R_t \tag{8}$$

In order to benchmark the use of a temporal filtering technique as suggested by [6,23], a spatial filtering technique was also developed and compared with the temporal filtering technique. Spatial filtering of an image is often done in image processing to reduce noise. This, however, leads to a blurry image with a loss of some information [24]. In this work, spatial filtering was achieved by convoluting an image with a Gaussian kernel where each pixel's value $I_t(x, y)$ in an input image R_t is changed by the Gaussian function G of the intensities of pixels in the neighborhood of (x, y). Equation (9) shows how an input image R_t is convoluted with a Gaussian kernel, where i runs from 1 to $Y - m + 1$ and j runs from 1 to $X - n + 1$ for an image with Y rows, X columns and a kernel with m rows, n columns. Both filtering techniques were used interchangeably in Figure 3B.

$$I_t(x, y) = \sum_{k=1}^{m} \sum_{l=1}^{n} R_t\ (x + k - 1,\ j + 1 - 1)\ * G(k,\ l) \tag{9}$$

After the filtering stage, a threshold range (Figure 3C) of D_L to D_U was used to achieve the detection of a unique object. The unique object of interest in this work was assumed to be a green ball. The ball was chosen to be a unique point on the Leader and was separated from surrounding imagery using *HSV* (Hue, Saturation, Value) values to define a range between values of D_L to D_U as in Equation (10):

$$I(x, y) = \begin{cases} I(x, y)\ if\ D_L < I(x, y) < D_U \\ 0\ else \end{cases} \tag{10}$$

4.3. Relative Bearing Estimation and Tracking by the Follower (Figure 3E)

The relative bearing of the ball to the Follower micro-UAV's attitude and altitude was estimated using the relative distance between the center point of the visual frame (C_x, C_y) and the position of the ball (B_x, B_y) on the visual field as shown in Figure 4 and according to Equation (11). Where θ_τ and $\aleph_\tau$ are the relative yaw and pitch angles respectively. The constants k_x and k_y embed the spring constants connecting the Follower to the Leader in these axes as well as scaling the pixel values to the angle values. The values of the constants were empirically determined. Using Equation (11), the Follower can adjust its relative bearing to the Leader:

$$\begin{bmatrix} \theta_\tau \\ \aleph_\tau \end{bmatrix} = \begin{bmatrix} C_x - B_x \\ C_y - B_y \end{bmatrix} \begin{bmatrix} k_x \\ k_y \end{bmatrix} \tag{11}$$

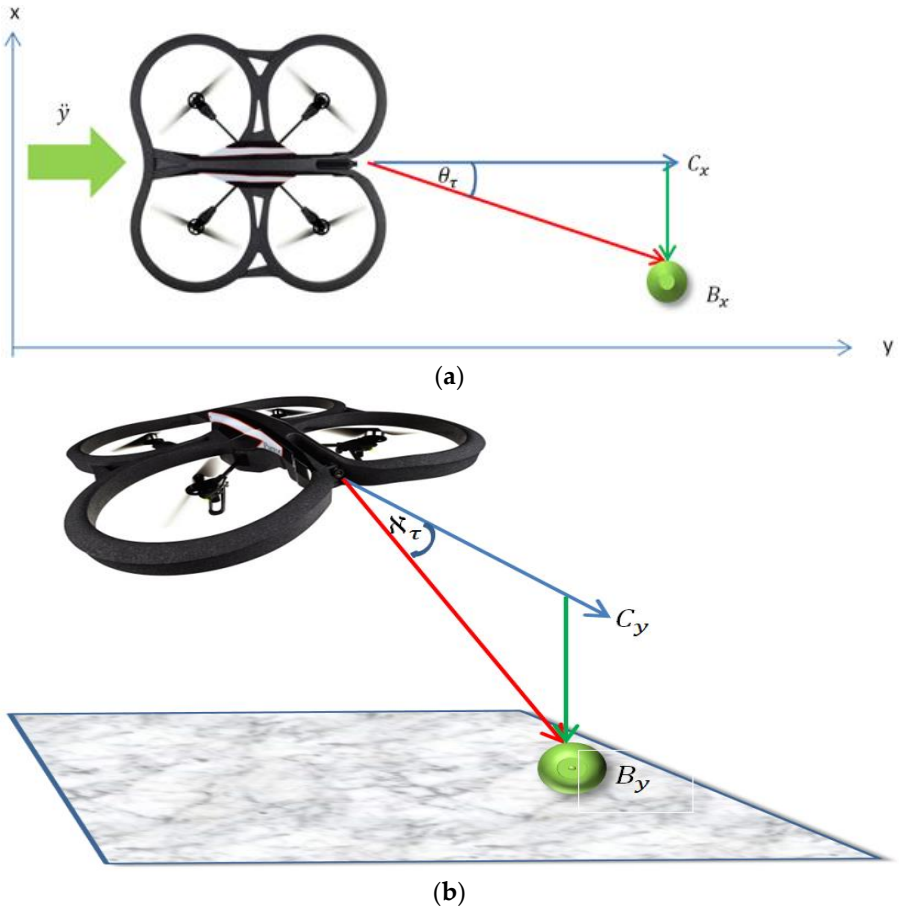


Figure 4. Relative pitch calculation using the distance between the center point of the visual frame and the position of the ball in the visual field of frame. (**a**) Relative bearing calculation; and (**b**) Relative pitch calculation.

4.4. Relative Distance Estimation (Figure 3D)

Detecting the distance to another flock mate using monocular vision, can be achieved in a variety of ways including kinetic depth perception and relative size. In kinetic depth perception, the motion of the flockmate is used to estimate distance. A flock mate receding into distance gets smaller, and vice versa. In the relative size method, the perceived size of a flockmate in relation to others is used. If the flockmate gets larger in comparison to other flockmates, then it means that it is closer, and vice versa.

In this work, the relative size approach was used. The image patch taken up by the green ball in the photoreceptors of the camera was used to encode the relative distance of the Follower to the Leader. In software, this was achieved through the use of the average of pixel intensity values on the visual field as shown in Equation (12):

$$P_{AVG} = \frac{1}{X * Y} \sum_{x=0}^{X} \sum_{y=0}^{Y} (I(x, y)) \tag{12}$$

The average value P_{AVG} was used instead of the direct sum of pixel values as suggested in [20] because the value of the sum is susceptible to noise.

The P_{AVG} value obtained is used to empirically determine the relationship between the relative distance from the green ball on the Leader and the zones as described in Equation (13) and shown in Figure 5.

$$\text{distance} = \begin{cases} \text{far } if\ P_{AVG} < 5 \\ \text{mid } if\ 5 < P_{AVG} < 30 \\ \text{near } if\ P_{AVG} > 30 \end{cases} \quad (13)$$

In the experiments conducted, the far zone corresponded to distances above 1000 mm, the mid-zone corresponded to distances between 300 mm and 1000 mm and the near zone was less than 300 mm. The values of B_X, B_Y, P_{AVG}, and distance were passed to the low-level controller, which will be discussed next.

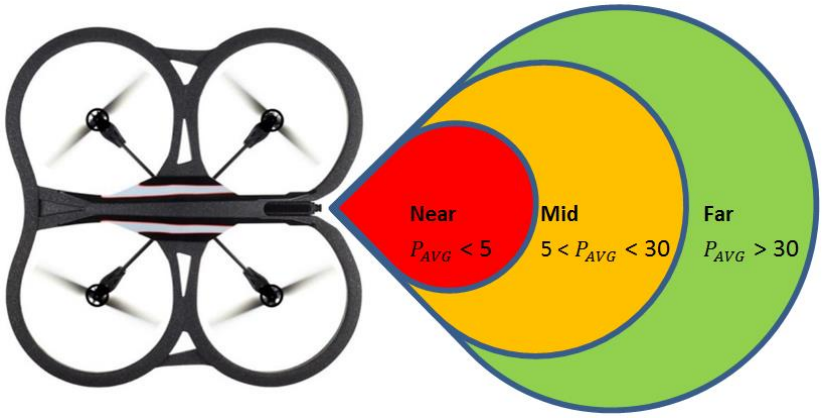


Figure 5. Relationship between flocking zones and detected pixel values.

4.5. Closed-Loop Control of the Micro-UAV Platform (Figure 3F)

Fifty hertz closed-loop low-level controllers were developed on a tethered laptop for the real time control of the unmanned aerial vehicle's attitude (That is its roll $\varnothing$, pitch φ, yaw θ) and altitude $\aleph$. The controllers were written in Java programming language. Movement on the x-axis was achieved by adjusting the roll $\varnothing$ value, while adjusting the pitch φ value would result in movement in the y-axis (Figure 4a). The yaw θ value is used to control the direction the platform is facing.

The attitude of the unmanned aerial vehicle platform during flight was obtained via the onboard gyroscope and accelerometer sensors on it, while the altitude was obtained via readings from the onboard ultrasonic sensor. Separate classical proportional-derivative controllers, whose gains were obtained through experimental trial and error, were used for the altitude, as well as the roll $\varnothing$ and pitch φ axes. A proportional controller was used for the yaw θ controller.

During the take-off states and hover states, the values: $(\varnothing_{\text{desired}}, \varphi_{\text{desired}}, \theta_{\text{desired}}) = (0^\circ, 0^\circ, 0^\circ)$ were used for the attitude of the micro-UAV platform as well as a desired altitude of $\aleph_{\text{desired}} = 500$ mm. For the tracking state, the relative bearing of the Leader $(\theta_\tau, \aleph_\tau)$ in respect to the Follower was obtained as discussed in Section 4.3 and passed to the respective controllers according to Equations (14)–(17) where ω is given by Equation 18 and embeds the velocities according to Equation (4):

$$u \propto U^{\aleph} = k_p^{\aleph} \aleph_\tau(t) + k_d^{\aleph} \Delta\aleph_\tau(t) \quad (14)$$

$$\ddot{\theta} \propto U^{\theta} = k_p^{\theta}\, \theta_\tau(t) \quad (15)$$

$$\ddot{\varnothing} \propto U^{\varnothing} = -k_p^{\varnothing} \varnothing(t) + k_d^{\varnothing}\, \Delta\varnothing(t) \quad (16)$$

$$\ddot{\varphi} \propto U^{\varphi} = -k_p^{\omega} \omega + k_d^{\varphi}\, \Delta\omega(t) \quad (17)$$

$$\omega = k_\omega \begin{cases} 2^\circ\ if\ distance = far \\ 0^\circ\ if\ distance = mid \\ -2^\circ\ if\ distance = near \end{cases} \quad (18)$$

The constants K_p^i and K_d^i are all proportional and derivative gains for the respective controllers obtained via the classical PID tuning approach. Equation (17) causes the Follower to follow the Leader in the y-axis at a velocity $\ddot{y}$ govern by the system of equations in Equation (7) and provides repulsion when the Leader is too close, or attraction when the Leader is far away, according to Equation (18), where k_ω corresponds to the spring constant and was empirically determined. The bearing of the Follower is controlled via Equation (15) and the altitude by Equation (14). The platform was commanded to maintain a roll $\varnothing$ value of 0° via Equation (16). The outputs of the respective controllers are directly proportional to the forces required to change the altitude, $\aleph$, and attitude ($\varnothing$, φ, θ) of the Follower to maintain formation with the Leader.

5. Simulation

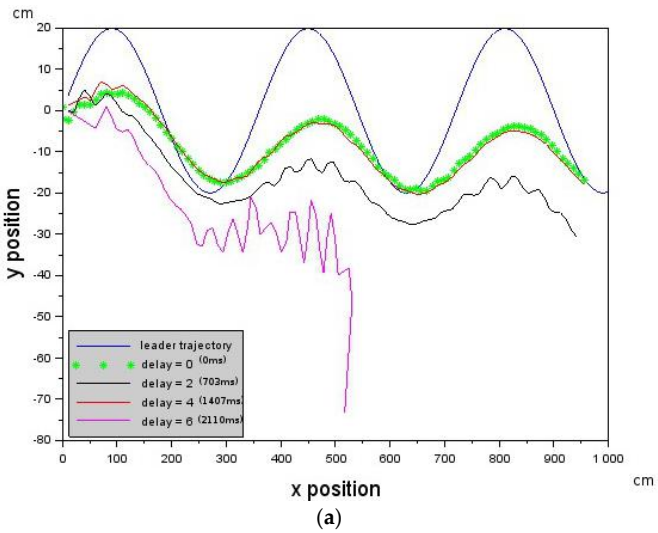
Simulations were conducted in a synthetic environment developed in a game engine called Panda3D (Walt Disney Imagineering, Carnegie Mellon University, Pittsburgh, PA, USA) as used in [25]. The simulation environment contained a Leader-Follower micro-UAV pair. Their dynamics were restricted to a point mass system with the Leader commanded to generate a cosine-like trajectory.

Simulated cameras that generated synthetic images were installed on the simulated Follower micro-UAV. These synthetic images were processed in order to extract the red color of a ball installed on the Leader micro-UAV. Using the position of the ball on the visual field, the Follower micro-UAV was able to follow the Leader. The visual max in the simulated environment was 160 cm.

5.1. Deterministic Delay Times

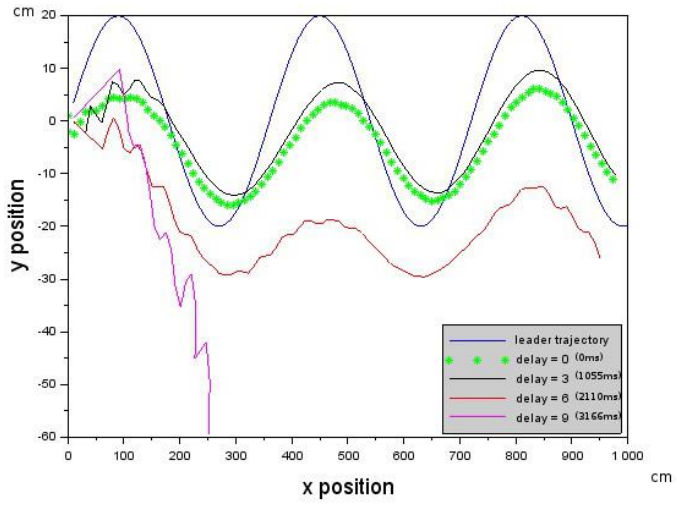
The Leader was commanded to generate the cosine trajectory with a forward velocity of 10 cm/s while the Follower was commanded to have the following velocity values $[V_{near}, V_{mid}, V_{far}]$ = [−10.0, 0.0, 10.0] cm/s. Deterministic delays were introduced on the follower. Delay values that increase from 0 (0 ms) to 6 (2110 ms) were first introduced into the control loop. Figure 6a shows that the Follower is able to track the trajectory of the Leader (blue curve) up to a delay of 4 (1407 ms) after which it loses the Leader.

However, this can be remedied by increasing the velocity of the Follower or reducing the velocity of the Leader according to Equation (6). We chose to increase the velocity of the Follower according to $[V_{near}, V_{mid}, V_{far}]$ = [−10.0, 0.0, 10.2] cm/s. Figure 6b shows that the Leader-Follower formation was maintained for longer up to a delay of 6 (2110 ms). The above results show that Leader-Follower formation flying is possible in the presence of delays provided that the constraints according to Equation (5), as well as Equation (6), are not violated.



(**a**)

Figure 6. *Cont.*

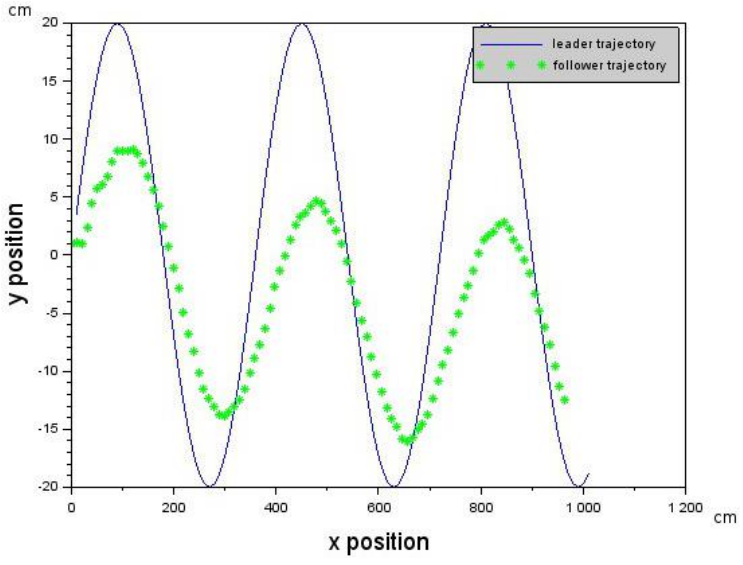


(**b**)

Figure 6. Showing how introduced deterministic delays affect Leader-Follower formation. Each curve is an experiment with an associated delay value. (**a**) The Follower loses track of the Leader at delay = 6 (2110 ms). The velocity of the Follower was 10 cm/s, while the velocity of the Leader was 10 cm/s as well. Each curve is an experiment with an associated delay value; and (**b**) due to an increase in the Follower's velocity to 10.2 cm/s, it is able to keep track of the Leader up to a delay of 9 (3166 ms). The Leader's velocity was 10 cm/s.

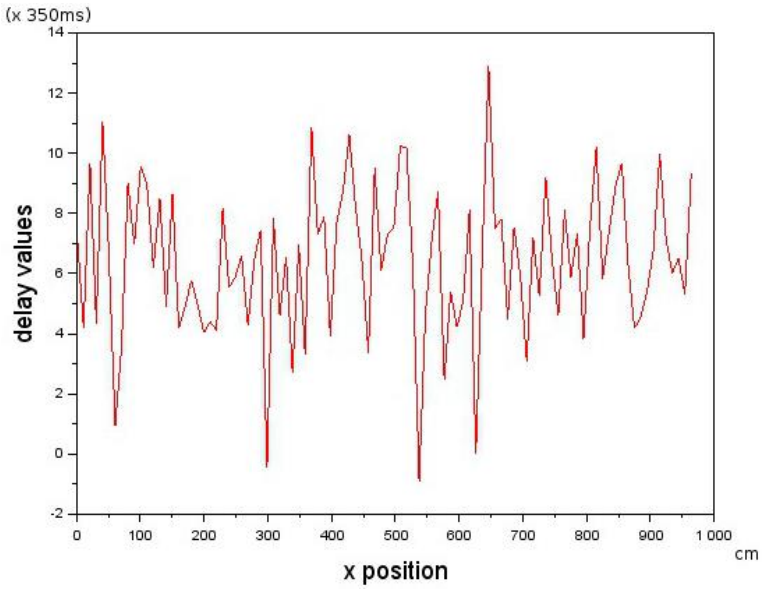
5.2. Stochastic Delay Times

In order to determine the effect of stochastic delay values on Leader-Follower formation flying, stochastic delay times were simulated with a mean of 6 (2110 ms) and standard deviation of 2.5 (879 ms). The velocity of the Follower was according to $[V_{near}, V_{mid}, V_{far}] = [-10.0, 0.0, 10.2]$ cm/s with a Leader velocity of 10 cm/s. Figure 7a shows that the Follower is able to track the Leader in the presence of the stochastic delays shown in Figure 7b. The results show that Leader-Follower formation flying is possible on physical systems with stochastic delay values provided that the constraint of Equations (5) and (6) are not violated. This was used to inform physical experiments on two micro-UAVs which are discussed in the next section.



(**a**)

Figure 7. *Cont.*



(**b**)

Figure 7. Showing how the Follower is able to track the Leader in the presence of introduced stochastic delays. (**a**) The trajectory of the Follower following the Leader in the presence of stochastic delays seen in (**b**); and (**b**) stochastic delays introduced into the control loop. Whenever the delay is below 0, a control command is sent immediately to the Follower.

6. Physical Experiments and Results

In this section, the proposed minimal cost vision algorithms (spatial and temporal filtering) were first compared (Section 6.1). The experiments were conducted in an office space environment that had little or no changes in artificial illumination. The environment had window blinds which could be opened to allow in natural lighting or closed to exclude it. A green ball with a diameter of 63 mm was used in the experiments. Other green objects that could lead to false detections were removed from the environment.

Secondly, the best performing minimal cost vision algorithm was used in the Leader-Follower algorithm which was tested by commanding a micro-UAV to follow a green ball held by a human (Section 6.2). The human's motions were random and not predictable. This enabled us to test the robustness of the Leader-Follower algorithm for formation flying. Finally, two micro-UAVs were used in Leader-Follower formation flying with the Leader commanded to fly at a forward velocity and the Follower commanded to maintain formation with the Leader (Section 6.3).

6.1. Comparing Spatial and Temporal Filtering

Experiments comparing the spatial and temporal approaches to image filtering were conducted in both artificial and natural lighting. The results in Figure 8 show that the spatial filtering technique has a higher pixel value than the temporal filtering technique at close distances. At larger distances, both have similar pixel values, but with the spatial filtering technique it is higher than that of the temporal filter technique. The reason for this result is shown in Figure 9 where it can be observed that the spatial filtering technique captures more of an object's presence but less of an object's detail and vice versa for the temporal filtering technique. As a result, the output of a temporal filtering technique will be more suited for object recognition tasks that are carried out in later stages of the visual cortex [6].

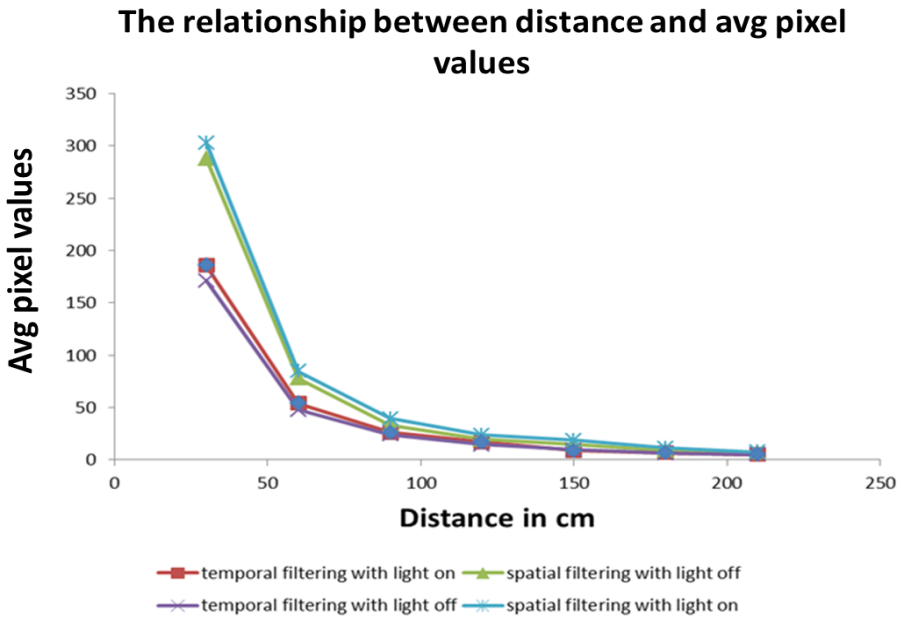
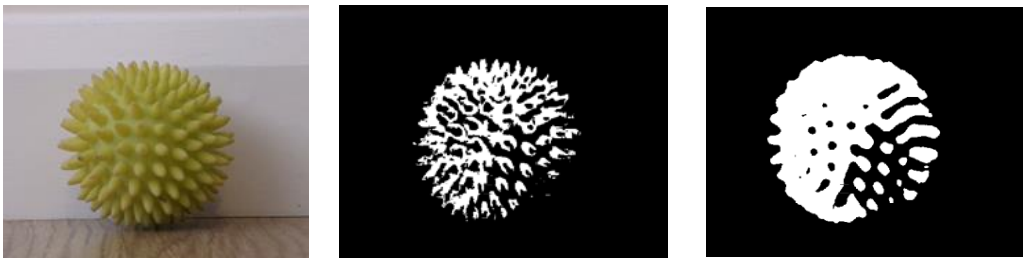


Figure 8. The effects of spatial and temporal vision filtering under different lighting conditions. The spatial filtering technique has a higher number of average pixels value P_{AVG} than the temporal filtering technique at closer distances but similar values at larger distances. For both approaches, the maximum visual distance $Visual_{max}$ is about 200 cm.



(**a**) Green ball (**b**) Temporal filtering (**c**) Spatial filtering

Figure 9. The green ball seen from 30 cm and filtering results. The spatial filtering results in more of object presence detection but less object detail while the temporal filtering results in more object detail. More object detail may help in object recognition tasks further up the visual cortex.

6.2. Following a Human Holding a Ball

From the experiments carried out in Section 6.1 above, it was decided that using the spatial filtering technique was the best approach to ensure Leader-Follower formation flying between two micro-UAVs due to the greater number of pixels detected. An experiment in which a micro-UAV was following a ball was carried out. The ball was held by a moving human. Results in Figure 10 show that the micro-UAV successfully tracked the human [26]. However, it should be noted that the presence of colors that fall into the range of HSV values used as feature descriptors sometimes affects the Leader-Follower formation flying. For example, Leader-Follower formation flying experiments conducted in three different indoor environments achieved a success rate of 80% in two of the environments and 0% in the last one. A 0% success rate was obtained because of the micro-UAV falsely identifying an object as a feature. Nevertheless, biological sensors are not always foolproof [27].

Figure 10. The Micro-UAV following a ball.

6.3. Leader-Follower Micro-UAV Formation Flying

Two experiments using two micro-UAVs in a Leader-Follower configuration were carried out. Each micro-UAV was tethered wirelessly to its own dedicated laptop and the green ball attached to the Leader micro-UAV. In the first experiment, the Leader was commanded to maintain a hover. As seen in Figure 11, the Follower was able to detect and fly towards the Leader platform [28]. However, due to the use of a small mid-zone range and image noise, this results in collision as in Figure 11.

In order to deal with this, the strategy proposed by [3] was used. This resulted in another experiment in which the Leader micro-UAV was commanded to move with a forward velocity. Results in Figure 12 show that Leader-Follower formation flying of two vehicles was then achieved [29].

Figure 11. The first autonomous aerial flocking attempt with a stationary Leader micro-UAV and moving Follower micro-UAV did not result in continuous flocking of two vehicles.

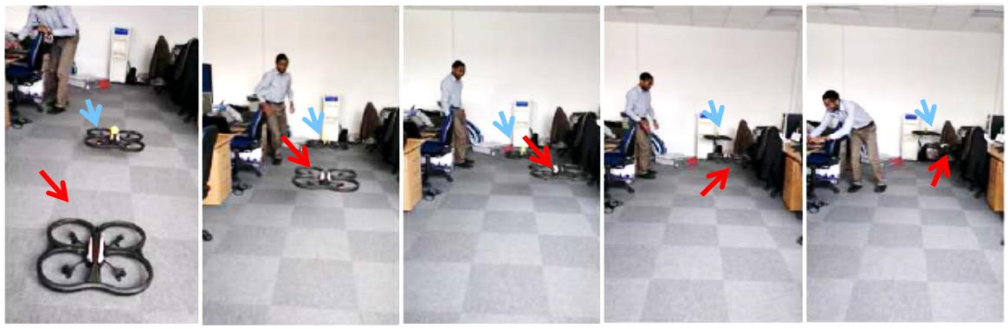


Figure 12. The second autonomous aerial flocking attempt with a Leader micro-UAV (Blue arrow) moving forward and a Follower micro-UAV (Red arrow) results in continuous flocking of two vehicles.

7. Discussion

In this section, we discuss how our approach and results are relevant to biology (Sections 7.1 and 7.2) and how the concepts as well as lessons learnt can be applied to robotics (Section 7.3).

7.1. Using Fixed Action Pattern Sequence for Leader-Follower Formation Flying

In this work, we started out by taking inspiration from the visual pathways of higher level vertebrates (mammals and primates). Their visual pathway for feature detection and the beginning of an appropriate response (such as eye saccade movements) occurs in, or very close to, the retinal circuits. In contrast, this function takes place in the midbrain of lower vertebrates, such as fish, birds, and amphibians. The visual pathways of fish, birds, and amphibians bear striking similarities in the way they are structured. This could be because they all share the same evolutionary origin. In all of these organisms the optic tectum forms the midbrain and sits in between the forebrain and cerebellum. Prey, predator, flockmate, and other detectors are found in the optic tectum in fish, birds, and toads, and also initiates or triggers the required response needed for the feature detected [30–32]. The trigger is passed to the deeper layers of the tectum responsible for motor functions in order to guide eye and body movement to salient environmental stimuli without the need for cortical processing [30]. Eventually, these neural signals enter the cerebellum, which seems to activate the limbs necessary to carry out the responses. Elman suggested that the detection of a prey feature in toads invariably sets off a sequence of prey-catching events that are very predictable: (1) orienting towards prey; (2) stalking up to prey; (3) binocular fixation; (4) snapping; (5) swallowing; and (6) mouth-wiping with forelimb [32]. The completion of each event in the sequence is used to trigger the next event in the sequence. This is called a Fixed Action Pattern sequence. The FAP sequence often involves very short to medium length sequence of actions triggered by a detected feature. The sequence is often hardwired into organisms as instinct and as such, they tend to carry out the sequence without variation when presented with the right trigger. In other words, they do not need to learn the stimulus-action sequence. In this work, the Fixed Action Pattern sequence needed to start the Leader-Follower formation flying is initiated at the visual detection of the ball.

In toads, the type of triggered FAP (flee or approach) depends on the configuration and size of the black shape. In measuring the size of the black shape, the toad uses angular size (i.e., the amount of degrees on the visual angle) and absolute size (i.e., the distance between the toad and the object) [32]. We used a similar technique in this work with the detected size of the ball used to decide whether to move closer, move away, or remain (Equation (13)). As toads, fish, and birds share similar brain structures, we hypothesize that similar Leader-Follower FAP sequence is initiated in starlings when a flockmate is detected.

7.2. Application of an Artificial Neural Network as Feature Detectors

The motivation to understand in the way organisms acquire and process signals from their environment, as well as convert those signals into actions, has resulted from the artificial neural network community. Artificial neural networks enable us to understand how organisms process signals, as well as develop algorithms that mimic their capabilities. The neural networks in organisms are a combination of arrangements which exhibit plasticity for learning from the environment and rigidity for encoding instinctive knowledge, such as Fixed Action Pattern sequences. The "rigid" or instinctive neural networks could also embed feature detectors that are subsequently used as triggers for FAP sequences.

In this work, spatial and temporal filtering were used to extract and detect a relevant visual feature from the environment. The values for the Gaussian kernels and $\propto$ used were hardwired and ad hoc to our application and, as such, embed feature detectors that were used for triggering the Leader-Follower FAP sequence. From the perspective of artificial neural networks, a closer study of Figure 3 and the spatial filtering scheme used in this work reveal that the scheme has a mathematical

structure close to that of a feedforward convolutional neural network. The pixels from the visual field form the inputs to this network and the detected feature as an output. The use of a Gaussian kernel in Equation (9) results in a Gaussian weighted sharing scheme among the layer of hidden neurons.

Equations (11) and (13) could resemble the function of the deeper layers of the optic tectum towards decoding of the configuration of the green ball in order to get the relative bearing estimation, tracking and relative distance estimation as outputs. These outputs are then passed to the PID controllers.

Furthermore, a simple recurrent neural network with a unit delay z^{-1} could be used to explain the temporal filtering scheme used in this work [33]. The weights ($\propto$) of the network were such that $|w| < 1$ in order to make the output signal exponentially convergent and the system stable. It also means that the system has an infinite memory capability.

However, temporal filtering (recurrent networks) could be seen as more expensive than spatial filtering (feedforward convolutional neural network) because it requires more memory, at least an extra memory element for image storage compared to spatial filtering. However, it results in more detail of an object being preserved and this is necessary for other object recognition tasks further up the visual circuitry.

In the experiments conducted, spatial filtering led to blurriness and the loss of the intricate details of an object being observed. Nevertheless, the result is sufficient for detecting an object's presence in the environment. For example, even though blurry vision results in loss of information, it is sufficient to trigger an escape behavior in frogs when a rapidly approaching object is detected in the visual field [34,35]. A similar technique is used in jumping spiders to detect prey and mates [36]. It is possible that the evolutionary mechanism favored this minimal cost approach, thereby resulting in the organisms' survival over millions of years. Furthermore, as identified in our experiments, it should be noted that the object detection range of spatial filtering is higher than that of the temporal filtering.

As a result of the above, we hypothesize that during high speed flocking in starlings, for example, the minimal cost spatial filtering might be used for the following reasons: (1) to reduce sensory and biological computation overload as well as maintain flock cohesion; (2) the spatial filtering technique has the capability to observe an object's presence at greater distances than the temporal filtering technique; (3) in starlings, flocking takes place at dusk in low light conditions where object recognition might be reduced. In this situation, the object's presence detection might be more appropriate than object recognition. Nevertheless, we also hypothesize that it is possible that both circuits are present in the optic tectum with each providing support to the other circuitry. The level of provided support could be dependent on the task being performed at the time and the amount of visual representation required for surviving in a particular vertebrate's environment [37].

7.3. Application to Robotics

In robotics, hidden Markov models could be used to represent FAPs. The developed FAP could be inspired by a biological organism or dependent on the application. The triggers of the FAP could be a feature detector that is extracted from vision, sound, touch, or other types of signals. In designing the feature detector, neural networks could be used. The weights of these neural networks could be obtained through manual tweaking (i.e., embedding prior knowledge of the environment into the network) or automated training. However, it should be noted that, if performing automated training, the quantity of free parameters needs to be controlled or it could lead to the need for a long training time.

The type of neural network to use really depends on the application, characteristics of the signal, and the environment from which the signal is being extracted. If the system could benefit from a memory of past measurements, than a recurrent neural network is perhaps a good choice. However, this should be used carefully due to the memory elements required by the network and the possibility of oscillations and instability due to feedback loops.

8. Conclusions

In this work, we have made use of visual information from an onboard camera on a micro-UAV to achieve Leader-Follower formation flying. Unlike other previous work, our approach did not make use of state estimation algorithms, GPS, or synchronized communication, and was validated with simulations, as well as actual physical experiments on micro-UAVs.

This was achieved through a minimal cost vision processing algorithm that was inspired by the structure of vertebrate retinas. We believe that understanding the retina's structure could lead to better design of vision-based autonomous flocking algorithms. Towards this, we used a computational approach, that focuses on "tasks organisms must face and then explore the minimal mechanisms that extract signals of relevance to these tasks" [20], to develop an architecture representative of the structure of a vertebrate's retina using mechanisms discussed in [6] (Section 4.2). We then used the developed architecture to achieve Leader-Follower formation flying which is a subset of the vision-based flocking problem.

Using our approach, we were able to achieve a Leader-Follower formation flying that is potentially robust to delays caused by communication delays and visual sensor refresh rates. In past Leader-Follower formation flying experiments such as in [15,16], Kalman filtering or other estimation algorithms are often used to estimate the path of the Leader. However, in this work, we have shown that provided that the delays are below a critical value τ, Leader-Follower formation flying is achievable using a zones scheme proposed in this work. This critical value τ is constrained by the maximum visual distance observable by the Follower.

This observed constraint might shed a light on flocking formation maintenance in different lighting conditions and from a biological perspective, it could affect the structures of natural flocks. Furthermore, the results of this work could inform more robust and computationally minimal cost algorithms that can be used to develop vision-based high-speed autonomous flocking systems for GPS-denied environments.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hauert, S.; Leven, S.; Varga, M.; Ruini, F.; Cangelosi, A.; Zufferey, J.-C.; Floreano, D. Reynolds flocking in reality with fixed-wing robots: communication range vs. maximum turning rate. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), San Francisco, CA, USA, 25–30 September 2011; pp. 5015–5020.
2. Kushleyev, A.; Mellinger, D.; Powers, C.; Kumar, V. Towards a swarm of agile micro quadrotors. *Auton. Robots* **2013**, *35*, 287–300. [CrossRef]
3. Vásárhelyi, G.; Virágh, C.; Somorjai, G.; Tarcai, N.; Szorenyi, T.; Nepusz, T.; Vicsek, T. Outdoor Flocking and Formation Flight with Autonomous Aerial Robots. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Chicago, IL, USA, 14–18 September 2014; pp. 3866–3873.
4. Reynolds, C.W. Flocks, herds and schools: A distributed behavioral model. In Proceedings of the ACM SIGGRAPH 87: 14th Annual Conference on Computer Graphics and Interactive Techniques, Anaheim, CA, USA, 27–31 July 1987; Volume 21, pp. 25–34.
5. Cowan, N.; Shakerina, O.; Vidal, R.; Sastry, S. Vision-based follow-the-Leader. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 27–31 October 2003; Volume 2, pp. 1796–1801.
6. Gollisch, T.; Meister, M. Eye smarter than scientists believed: neural computations in circuits of the retina. *Neuron* **2010**, *65*, 150–164. [CrossRef] [PubMed]
7. Doi, E.; Gauthier, J.L.; Field, G.D.; Shlens, J.; Sher, A.; Greschner, M.; Machado, T.A.; Jepson, L.H.; Mathieson, K.; Gunning, D.E.; et al. Efficient coding of spatial information in the primate retina. *J. Neurosci.* **2012**, *32*, 16256–16264. [CrossRef] [PubMed]
8. Hecht, S.; Shlaer, S.; Pirenne, M.H. Energy at the threshold of vision. *Science* **1941**, *93*, 585–587. [CrossRef] [PubMed]

9. Demb, J.B. Cellular mechanisms for direction selectivity in the retina. *Neuron* **2007**, *55*, 179–186. [CrossRef] [PubMed]
10. Demb, J.B.; Haarsma, L.; Freed, M.A.; Sterling, P. Functional circuitry of the retinal ganglion cell's nonlinear receptive field. *J. Neurosci.* **1999**, *19*, 9756–9767. [PubMed]
11. Enroth-Cugell, C.; Freeman, A.W. The receptive-field spatial structure of cat retinal Y cells. *J. Physiol.* **1987**, *384*, 49–79. [CrossRef] [PubMed]
12. Enroth-Cugell, C.; Robson, J.G. The contrast sensitivity of retinal ganglion cells of the cat. *J. Physiol.* **1966**, *187*, 517–552. [CrossRef] [PubMed]
13. Johnson, E.N.; Calise, A.J.; Watanabe, Y.; Ha, J.; Neidhoefer, J.C. Real-time vision-based relative aircraft navigation. *J. Aerosp. Comput. Inf. Commun.* **2007**, *4*, 707–738. [CrossRef]
14. Oh, S.-M.; Johnson, E.N. Relative motion estimation for vision-based formation flight using unscented Kalman filter. In Proceedings of the AIAA Guidance, Navigation, and Control Conference, Hilton Head, CO, USA, 20–23 August 2007; pp. 1–17.
15. Panagou, D.; Kumar, V. Cooperative Visibility Maintenance for Leader—Follower Formations in Obstacle Environments. *IEEE Trans. Robot.* **2014**, *30*, 831–844. [CrossRef]
16. Wilson, D.B.; Goktogan, A.H.; Sukkarieh, S. A vision based relative navigation framework for formation flight. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 4988–4995.
17. He, W.; Chen, G.; Han, Q.-L.; Qian, F. Network-based leader-following consensus of nonlinear multi-agent systems via distributed impulsive control. *Inf. Sci.* **2015**. [CrossRef]
18. Rezaee, H.; Abdollahi, F.; Talebi, H.A. Based Motion Synchronization in Formation Flight with Delayed Communications. *IEEE Trans. Ind. Electron.* **2014**, *61*, 6175–6182. [CrossRef]
19. Zhang, X.; Jiang, X.; Han, Q.-L. An improved stability criterion of networked control systems. In Proceedings of the American Control Conference (ACC), Baltimore, MD, USA, 30 June–2 July 2010; pp. 586–589.
20. Bialek, W.; Owen, W.G. Temporal filtering in retinal bipolar cells: Elements of an optimal computation. *J. Biophys. Soc.* **1990**, *58*, 1227–1233. [CrossRef]
21. Engel, J.; Sturm, J.; Cremers, D. Camera-Based Navigation of a Low-Cost Quadrocopter. In Proceedings of the International Conference on Intelligent Robots and Systems, Vilamoura, Portugal, 7–12 October 2012; pp. 2815–2821.
22. Bristeau, P.-J.; Callou, F.; Vissiere, D.; Petit, N. The Navigation and Control technology inside the AR. Drone micro UAV. In Proceedings of the 18th IFAC World Congress, Milano, Italy, 28 August–2 September 2011; pp. 1477–1484.
23. Baccus, S.A. Timing and computation in inner retinal circuitry. *Annu. Rev. Physiol.* **2007**, *69*, 271–290. [CrossRef] [PubMed]
24. Vespie, A.W. Image Processing and Filtering Techniques of Static and Dynamic Images: A Technologist s Review. *Int. J. Adv. Res. Comput. Sci. Softw. Eng.* **2013**, *3*, 198–202.
25. Oyekan, J. A Vision-based terrain morphology estimation model inspired by the avian hippocampus. *Digit. Commun. Netw.* **2015**, *1*, 134–140. [CrossRef]
26. Oyekan, J. Follow Ball Behaviour. Available online: https://www.youtube.com/watch?v=h8bzzK7DXSk. (accessed on 15 August 2016).
27. Perry, S.F.; Rivero-Lopez, L.; McNeill, B.; Wilson, J. Fooling a freshwater fish: How dietary salt transforms the rainbow trout gill into a seawater gill phenotype. *J. Exp. Biol.* **2006**, *209*, 4591–4596. [CrossRef] [PubMed]
28. Oyekan, J. Preliminary Flocking. Available online: https://www.youtube.com/watch?v=CNaM3Cbp0fI. (accessed on 15 August 2016).
29. Oyekan, J. Flocking Two. Available online: https://www.youtube.com/watch?v=NWu0xxBisT8. (accessed on 15 August 2016).
30. Vincze, O.; Vágási, C.I.; Pap, P.L.; Osváth, G.; Møller, A.P. Brain regions associated with visual cues are important for bird migration. *Biol. Lett.* **2015**, *11*. [CrossRef] [PubMed]
31. Northmore, D. Optic tectum. In *Encyclopedia of Fish Physiology: From Genome to Environment*; Elsevier: Cambridge, MA, USA, 2011; pp. 131–142.
32. Ewert, J.-P. *The Neural Basis of Visually Guided Behavior*; Freeman: London, UK, 1974.
33. Elman, J. Finding structure in time. *Cognit. Sci.* **1990**, *14*, 179–211. [CrossRef]

34. Ishikane, H.; Gangi, M.; Honda, S.; Tachibana, M. Synchronized retinal oscillations encode essential information for escape behavior in frogs. *Nat. Neurosci.* **2005**, *8*, 1087–1095. [CrossRef] [PubMed]
35. Lettvin, J.Y.; Maturana, H.R.; McCulloch, W.S.; Pitts, W.H. What the Frog's Eye Tells the Frog's Brain. *Proc. IRE* **1959**, *47*, 1940–1951. [CrossRef]
36. Nagata, T.; Koyanagi, M.; Tsukamoto, H.; Saeki, S.; Isono, K.; Shichida, Y.; Tokunaga, F.; Kinoshita, M.; Arikawa, K.; Terakita, A. Depth perception from image defocus in a jumping spider. *Science* **2012**, *335*, 469–471. [CrossRef] [PubMed]
37. Land, M.F.; Nilsson, D.-E. *Animal Eyes*; Oxford University Press: Oxford, UK, 2002.

2016-08-18

Bio-inspired vision-based leader-follower formation flying in the presence of delays

Oyekan, John

M D P I AG

Oyekan, J. Bio-Inspired Vision-Based Leader-Follower Formation Flying in the Presence of Delays. Robotics, Volume 5, Issue 3, 2016, Article number 18
http://dx.doi.org/10.3390/robotics5030018