Arthropod Structure & Development xxx (2017) 1–14

Contents lists available at ScienceDirect

Arthropod Structure & Development

journal homepage: www.elsevier.com/locate/asd

A lightweight, inexpensive robotic system for insect vision

Chelsea Sabo*, Robert Chisholm, Adam Petterson, Alex Cope**

University of Sheffield, Sheffield, S10 2TN, UK

ARTICLE INFO

Article history:
Received 20 June 2016
Received in revised form
5 August 2017
Accepted 10 August 2017
Available online xxx

Keywords:
Embodiment
Insect vision
Honeybees
Robotics
Computational modelling

ABSTRACT

Designing hardware for miniaturized robotics which mimics the capabilities of flying insects is of interest, because they share similar constraints (i.e. small size, low weight, and low energy consumption). Research in this area aims to enable robots with similarly efficient flight and cognitive abilities. Visual processing is important to flying insects' impressive flight capabilities, but currently, embodiment of insect-like visual systems is limited by the hardware systems available. Suitable hardware is either prohibitively expensive, difficult to reproduce, cannot accurately simulate insect vision characteristics, and/or is too heavy for small robotic platforms. These limitations hamper the development of platforms for embodiment which in turn hampers the progress on understanding of how biological systems fundamentally work. To address this gap, this paper proposes an inexpensive, lightweight robotic system for modelling insect vision. The system is mounted and tested on a robotic platform for mobile applications, and then the camera and insect vision models are evaluated. We analyse the potential of the system for use in embodiment of higher-level visual processes (i.e. motion detection) and also for development of navigation based on vision for robotics in general. Optic flow from sample camera data is calculated and compared to a perfect, simulated bee world showing an excellent resemblance.

1. Introduction

Recent improvements in sensors, processing, and batteries have made new technologies low-weight, low-power, and low-cost. This has allowed robots, particularly sUAVs (small Unmanned Aerial Vehicles), to be more accessible and users to broaden their applications. As robots decrease in size, they are subject to different constraints (limited power, size, GPS capabilities, etc.), more complex missions (such as searching buildings or other confined areas), and bigger disturbances due to small changes in the environment (i.e. wind or moving objects). Flying insects are of interest to the design of small robotic platforms, because they represent a complete working solution which is capable of the behaviours required for performing in challenging environments.

Flying insects are capable of sophisticated odometry, including estimating flight duration, integrating their course over time to generate a direct home vector ('path integration'), and regulating their flight speed (Srinivasan et al., 1996, 2015). They can perform smooth landing on unfamiliar targets, and optimise routing around a set of target locations in a few flights (Lihoreau et al., 2012). All these capabilities are performed using primarily visual inputs and processing (Srinivasan et al., 2000; Esch et al., 2001; Barron and Srinivasan, 2006; Srinivasan, 2011), and thus, understanding insect vision is critical to understanding how these complex behaviours arise.

To understand how insects' neural systems perform their complex behaviours, it is important to create models of these behaviours and then embed them in physical systems, like robots. Many studies primarily focus on only simulating behaviours and cognitive processes within computing platforms (e.g. 'Blue Brain' Project (Markram, 2006)) while ignoring the implications of embodiment. In reality, neural processes are just a part of the computational loop where sensing and action play an equally critical role. Behaviour is not solely the result of a system's internal make up, and in fact, the body helps to shape the brain both physically and functionally. The environment in which a system senses and interacts in addition to the physical make-up of the system affects its behaviour (Pfeifer et al., 2007). For example, two different physical systems (e.g. one with a nose versus one with eyes and wheels versus limbs) will each have very unique experiences with different environments, which will produce very distinctive behaviours. In turn, this results in unique ways in which systems perceive the world and process

* Corresponding author.
** Corresponding author.
E-mail addresses: c.sabo@sheffield.ac.uk (C. Sabo), a.cope@sheffield.ac.uk (A. Cope).

http://dx.doi.org/10.1016/j.asd.2017.08.001

information (Webb, 2013; Lungarella and Sporns, 2006). Robots can help to understand the underlying sensing, processing, and behaviour. Similarly, the sensory systems used to provide input data to cognitive and behavioural models should be as accurate to the biological source as possible. Otherwise, even a perfect model might not be able to replicate the performance of a biological agent.

In order to improve lightweight robotics and better understand flying insects, we try to replicate their behaviours, and therefore their sensory and cognitive processes, in robotic systems. Visual processing is the focus in this paper due to its significant role in flight behaviour. Expensive, bespoke cameras are impractical to use in everyday applications. For this reason, we propose a lightweight and inexpensive design for modelling insect vision for embodied mobile robotics. The methodology presented here allows for the flexible replication of a variety of insect visual systems on any hardware that meets the design requirements. This work goes on to outline a single implementation of this methodology as a proof-of-concept. The performance is evaluated against the camera and insect models. Additionally, the implementation is based on the suitability of the system for further embodiment of behaviour and cognition; that is, we calculated optic flow and compare it to a simulated bee world. In the conclusions and future work, we highlight advantages that could be gained through alterations to this proof-of-concept implementation. Ultimately, this design paradigm will help to improve both robotic capabilities and the understanding of insect behaviour.

2. Background information

The vision system in insects is linked to complex cognitive behaviours which are not currently understood (Srinivasan, 2011). In particular, there has been extensive research in recent years into honeybee vision and flight navigation, as bees are known to have impressive capabilities. For example, honeybees will seek out food over miles and directly return to their hive, provide navigational instructions to each other, use landmarks for location identification, distinguish colours to identify good sources of food, navigate in corridors and other, complex environments, and more. It has been shown that bees use their vision to regulate their velocity in flight, control their course, estimate distance travelled through path integration, avoid obstacles, and land smoothly (Srinivasan, 2011). Bees are able to accomplish these tasks through the estimation of angular velocity or optic flow of the visual world (Ibbotson, 2001; Thurrowgood et al., 2014). Current computational research (e.g. Green Brain Project and “Brains on Board” Project) is trying to model and embody these behaviours to show how bee's physiology is able to accomplish these impressive tasks with such efficient coding and processing of information (Cope et al., 2013).

This section discusses the important physical aspects of insect vision and the state-of-the-art in embodiment of insect vision. While the methodology presented in this work can be applied to any insect vision model, the focus of implementation is on replicating honeybee vision as this is an ideal candidate for further study of navigation and cognition. As such, most of the details of the visual system discussed in this section relate to honeybees.

2.1. Insect vision

Insect vision can be characterised in three major dimensions, spatial organisation, temporal response, and chromatic response. We will now describe each dimension in turn, with special reference to the honeybee.

In terms of the spatial organisation of their vision, insects can have an extremely wide and deep Field-of-View (FoV) consisting of individual lensed units called ommatidia. Each ommatidium detects light coming from a specific direction. Ommatidial numbers and density very considerably across species: from almost 30,000 placements in some dragonflies (Zufferey, 2005) to approximately 800 or fewer placements in the fruit fly (Power, 1943). The honeybee FoV is almost panoramic and has a total of ~5500 ommatidia per compound eye (Seidl and Kaiser, 1981a). The fruit fly is similar in FoV despite less than 800 ommatidia (Power, 1943) leading to a much larger angular spacing between neighbouring ommatidia and thus, poorer spatial resolution. In many insect species including the honeybee, each ommatidium accepts light from an angle similar to the spacing angle between neighbouring ommatidia, and thus can be thought of as a single pixel element. In addition, the photoreceptors in each ommatidium lie along a single vertical axis below the lens. As such, the ommatidia form an array of single pixel elements with very little inhomogeneity in spacing compared to that found in mammalian vision but of much larger angular extent.

The spacing and acceptance angle of the ommatidia results in the large field of view and the spatial resolution of insect eyes. It has been found that the ommatidia are packed more densely near the centre of the eyes than at the edges. The central ommatidia have a visual angle of about 1°, whereas those furthest from the centre can be up to 3°. Additionally, the honeybee's eye has greater resolution in the ventral to dorsal direction than in the anterior to posterior direction (Hecht and Wolf, 1929). It is these two parameters (spacing and acceptance angles) that produce the bee's spatial resolution. While these exact parameters vary across insects, the pattern is comparable.

The two compound eyes have fixed focus as they cannot move with respect to each other and can only move as the insect head moves with respect to the body. There is, however, a region of overlap in the fields of view of the compound eyes which provides a fixed convergent zone, and this may be used for some insect species for specific purposes. For example, it is used for prey capture and pursuit in dragonflies (Olberg et al., 2000).

The temporal characteristics of the insect visual system are described by two main parameters, the speed of response and the latency of response. The response speed of the insect visual system varies largely between species depending on the requirements for detecting motion. For example, honeybees can reach flight speeds of up to 0.7 m/s in a 0.2 m wide corridor, and as such require higher temporal resolution than the slow flying fruit fly (Srinivasan et al., 1996, 2011). Studies have been done to try to determine how fast bees can actually see rapidly changing images. It has been behaviourally established that bees can only make decisions on stimuli up to frequencies of 165–300 Hz implying that they resolve images up to a maximum of 300 Hz (Autrum and Stoecker, 1950). In comparison, humans have a temporal solution in the range of 20–70 Hz meaning that bee vision is roughly five to six times faster (Rabin, 2010).

The insect nervous system consists of several pathways from visual input to motor output, which determine the latency of behavioural responses. The shortest of these runs through the optic neuropils, then directly to the rear of the insect brain where the dendrites of neurons descending to the motor ganglion are found. This pathway therefore determines the minimum processing delay from sensory stimulation to motor response, which is an important factor for stable flight control. In the dragonfly, behavioural responses to environmental changes have been found to have latency as low as 30 ms (Olberg et al., 2007).

Finally insects are trichromats, possessing photoreceptors responding to three evenly spaced sections of the electromagnetic spectrum and covering a larger section overall than mammalians, notably including the ultraviolet part of the spectrum (Dyer and Chittka, 2004).

2.2. Visual systems for robotic applications

While there are many types of visual systems for robotic applications, the lightweight requirement considerably narrows down the possibilities. Discussed here are the current robotic solutions for lightweight visual systems and also those already used for modelling insect vision.

There are currently many lightweight FPV (First-Person View) sensors units available at low cost, because demand for mobile computing has increased. The two types of FPV cameras are CCD (Charged Coupling Device) and CMOS (Complementary Metal Oxide Semiconductor), but CCD are generally more suited for robotics applications since they have better performance under varied lighting conditions and are less susceptible to rolling shutter artefacts that can cause image wobble during motion (Gunn, 2013). These cameras come in many shapes and sizes, but of greatest interest is that they are the most common cameras used for applications which require extremely lightweight real-time visual information (i.e. in sUAV visual navigation applications). An example of their use on a small quadcopter is the Parrot AR Drone 2.0 (Engel et al., 2012). An example of a commonly used CMOS camera used for lightweight robotics applications is the Go-Pro Hero 4 (GoPro Website, 2016). It is commonly used on the DJI Phantom and other sUAVs (Smashing Drones, 2016).

A limiting factor on both of these types of cameras are the field-of-view and framerate. Typical applications do not have the same requirements as needed to model insects which means they only take in about 90° (varies by about ±10°) in the front of the visual field. While this is suitable for some applications, this narrow FoV is insufficient to model the complexity of insect visual navigation as insects rely on comprehensive situational awareness.

2.3. Systems for modelling insect vision

Other visual systems have attempted to model biologically realistic insect vision. There have been models developed of insect vision in simulated virtual environments (Giger, 1996; Song et al., 2013). In the simulated world, they are able to reproduce the fine details of insect vision. Titus R. Neumann for example created a system using 3D rendering techniques to create a bee's field of vision from a textured cube (Neumann, 2002), and Andrew Giger (1996) produced an online applet allowing simulation from a simple bitmap image. However, these are missing a key component relative to our current purpose, which is to provide a robotic platform with this insect vision model.

To mimic the large FOV of the compound eyes, panoramic and catadioptric imaging systems are often used. These systems use both reflection and refraction to gain an incredibly large visual field. An example of this is a catadioptric system by Stürzl et al. (2010) which was capable of mounting on a flying robot. The use of a single optical system to provide vision for both eyes reduces the overall weight. Stürzl's system was well-designed to account for the additional distortion and blur generated from the approach, but it eliminates the binocular characteristic of the eyes, still creates a lot of distortion (relatively), and comprises multiple optical elements making it more susceptible to performance degradation from vibration when mounted on an aerial platform. Additionally, the bespoke nature of such systems inflates the cost and makes reproduction more difficult. Research has also been done to develop sensor and lens arrays to mimic the distribution of ommatidia found in compound eyes (Song et al., 2013; Leitel et al., 2014). These cameras are more biologically realistic, but at a fixed low-resolution they have limited flexibility. Additionally, these are also solely bespoke designs which makes them difficult and costly to implement in practice.

3. Problem definition

Despite recent advances, there is a lack of suitable robotic systems that are capable of mimicking insect visual capabilities on lightweight aerial vehicles. This research addresses this gap. The problem is then defined as the selection of a visual hardware system for deployment on a lightweight robot to be used for visual processing and navigation. Hence, the system must collect real-time images of the world and then accurately transform and map the incoming image to an insect-centric view with the correct ommatidia placement.

3.1. Design requirements/metrics

In order to realistically emulate insect vision, several design requirements need to be met as a result of the problem formulation. These design requirements also define the performance metrics and are as follows:

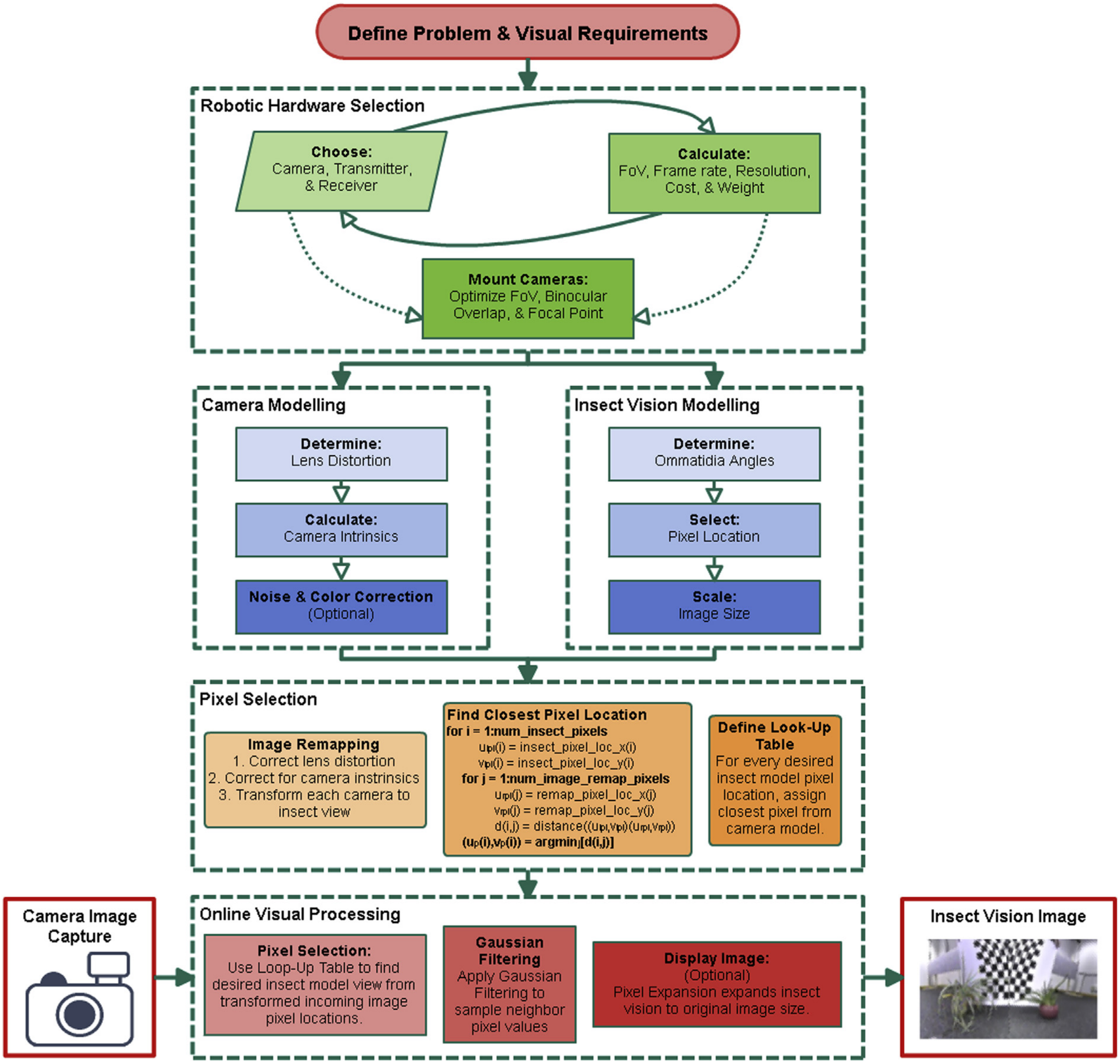
- Large Field-of-View (minimum of 180° horizontal and 90° vertical)
- Fast Frame Rate (minimum of 30 Hz)
- Sufficient Resolution (minimum of 75 × 75 pixels)
- Lightweight (less than 250 g)
- Inexpensive System (less than $500)

Most of the requirements are imposed to match insect biology, but several (lightweight and inexpensive) are enforced to ensure usability and reproducibility of the system. Arguably, the most important design metric is based on the large field-of-view of the insect which is essential to their navigational capabilities. It is desirable to have a horizontal view greater than 180° to capture artefacts in the periphery and a vertical view more than 90° to capture both the view ahead and in the ventral foreground. The framerate is also a key design component, because a minimum of 30 Hz is necessary to ensure reliable control of the robot. This is especially true of aerial robots (Gunn, 2013). Further, there is evidence that honeybees can see up to 300 Hz (Autrum and Stoeker, 1950). Finally, the system also has to meet the minimum resolution necessary for mapping each ommatidium from the insect onto the image with minimal error.

4. Methodology

This section gives an overview of the methodology used in designing the robotic visual system presented in this research. While the implementation shown later gives good results as per the design parameters, the general methodology described here, is sufficiently flexible to be applied across a variety of hardware setups and insect model parameters while retaining similar performance. This workflow is depicted in Fig. 1.

Initially, the visual requirements are laid out in the problem definition which in turn defines the design performance metrics (as described previously in Section 3). A robotic hardware system is first selected to optimize these performance metrics. One or more cameras, if weight constraints allow, for lightweight robotics and insect embodiment should be considered alongside the field-of-view requirement. Since each ommatidium in flying insects essentially functions as a pixel in an image, there is therefore a wide selection of cameras that will provide the necessary acuity. For example, honeybee vision works out to be roughly 75 × 75 pixels which even low-resolution cameras possess. Better resolution provides more flexibility for applications including modelling insects, where each ommatidium has a wide acceptance angle or there are very large numbers of ommatidia. The minimum


Define Problem & Visual Requirements
Robotic Hardware Selection
Choose: Camera, Transmitter, & Receiver
Calculate: FoV, Frame rate, Resolution, Cost, & Weight
Mount Cameras: Optimize FoV, Binocular Overlap, & Focal Point
Camera Modelling
Determine: Lens Distortion
Calculate: Camera Intrinsics
Noise & Color Correction (Optional)
Insect Vision Modelling
Determine: Ommatidia Angles
Select: Pixel Location
Scale: Image Size
Pixel Selection
Image Remapping
1. Correct lens distortion
2. Correct for camera instrinsics
3. Transform each camera to insect view
Find Closest Pixel Location
Define Look-Up Table
For every desired insect model pixel location, assign closest pixel from camera model.
Camera Image Capture
Online Visual Processing
Pixel Selection: Use Loop-Up Table to find desired insect model view from transformed incoming image pixel locations.
Gaussian Filtering
Apply Gaussian Filtering to sample neighbor pixel values
Display Image: (Optional) Pixel Expansion expands insect vision to original image size.
Insect Vision Image

```
for i = 1:num_insect_pixels
    $u_{ipi}(i)$ = insect_pixel_loc_x(i)
    $v_{ipi}(i)$ = insect_pixel_loc_y(i)
  for j = 1:num_image_remap_pixels
      $u_{rpi}(j)$ = remap_pixel_loc_x(j)
      $v_{rpi}(j)$ = remap_pixel_loc_y(j)
      d(i,j) = distance(($u_{ipi}$,$v_{ipi}$)($u_{rpi}$,$v_{rpi}$))
  ($u_p(i)$,$v_p(i)$) = $\text{argmin}_j$[d(i,j)]
```

Fig. 1. Methodology for modelling insect vision with robotics. The workflow can be applied across a variety of hardware and insect model parameters. The visual hardware is selected first, and the resultant camera properties can be used to model both the camera and insect vision. Pixel selection is done in pre-processing to generate a look-up table. The table can be used for fast online processing of camera images to look up the values at each desired pixel and sample the neighbouring pixels. Lastly, image display is completed (with optional pixel expansion).

framerate of 30 Hz and real-time data streaming requirements can be used to narrow down camera selection from there. The limiting factors on camera choice ultimately come down to the field-of-view, weight, and cost, and the final hardware is chosen when these metrics are appropriately optimized according to the problem definition. The lightweight design requirement and computational power needed for visual processing limits the on-board computing capabilities of any robotic platform. Therefore, a camera with a transmitter (Tx)/receiver (Rx) pair allows for quick, off-board processing of visual information and is recommended for mobile robotics. This is utilized in the design implementation presented in the next section (Section 5). This Tx/Rx combination is reflected in the methodology in Fig. 1, and while this setup is not necessary, there does not currently exist a good, lightweight alternative to off-board computation in many situations. Even further, solutions with a Tx/Rx do not meet the cost constraint and limit the forms of neural modelling that can be performed within the time constraints of real-time robotic control. Yet, this does open up the possibility for future improvements on the design implementation.

Once the hardware has been selected, it is mounted onto a robotic platform in order to optimize the field-of-view, overlap in the foreground, and focal point of each camera (or "eye").

Offline computations are done upfront in order to maximize online processing speed. The resultant camera properties from hardware selection can be used to model both the camera and insect vision for later handling. Camera modelling requires determining lens distortion and camera intrinsic values based on a set of calibration images. Insect modelling requires determining ommatidium angles, selecting the corresponding pixels in a 2D plane, and then scaling the result to the desired frame size (e.g. camera frame size). Offline pixel selection is then done to generate a look-up table which can in turn be used for quick online processing of camera images. First, the incoming camera image needs to be remapped according to the camera model and then transformed via a rotation to get an insect-centric view. For each desired insect pixel location, the closest pixel in the camera image can be found and stored in the look-up table.

Finally, the table is used to find the corresponding image pixel location and value for each desired pixel location. Gaussian sampling is performed on neighbouring pixels to produce the final matrix of values which constitute insect vision. Lastly, displaying the image is completed where pixel expansion to match the original image size is optional.

5. Implementation

The design is based on satisfying the visual constraints for a range of insect species along with the practicalities of low cost and the ease of sourcing components. A further consideration arises from the nature of embodying models of neural systems. While the goal of such investigations is to generate completely autonomous agents, investigations may require detailed simulation of neurons and synaptic connectivity which is too computationally demanding for mobile processing. This consideration makes it desirable to allow for off-board computations of sensory processing and calculation of motor commands which are then transmitted wirelessly to the agent. We therefore present and investigate the design paradigm of the more demanding off-board case as depicted in Fig. 2.

The implementation and design requirements were optimized to match the competencies of honeybee vision and navigational skills. The implementation shown here for the honeybee has been made open-source and available online for easy replication and use.[1]

5.1. Robotic hardware selection

Due to the widespread availability, inexpensive cost, and appropriateness for robotics applications, CCD cameras were chosen for this implementation. As previously stated, the minimum resolution requirement is largely not an issue as most units (specifically CCD here) come with higher than NTSC/PAL resolution which is sufficient for modelling honeybee vision. The NTSC/PAL sensors permit analog output in fields, and this allows either 60/48 fields per second (consisting of every other horizontal scanline per field) or 30/24 full frames per second (constructed by combining field pairs). Ultimately, 2 mini FPV (First-Person View) cameras were selected with the following specifications: Turnigy Micro FPV, 600TVL, 768 × 494 resolution, 30 fps, and a 2.1 mm diameter lens with a 150° viewing angle.

Since a majority of these cameras have limited field-of-view, a balance between the weight and the number of cameras needed to maximize the field-of-view had to be found. The weight was minimized by selecting 2 cameras and equipping them with wide-angle optics that can meet the constraint of the field-of-view possessed by many insects. Each camera replicates one of the two insect compound eyes by using standard fish-eye type lenses with 150° each (similar to the bee eye).

For the local data transmission required for on-board computation, there are no additional design requirements. As such, each camera was streamed over a 5.8 GHz 20mw FPV transmitter to an 8CH Diversity A/V receiver in-line with generic USB capture cards with the appropriate format and resolution (as shown in Fig. 2). In this application, signals were sent over short distances (<25 m), and as a result, a low-power transmitter was suitable.

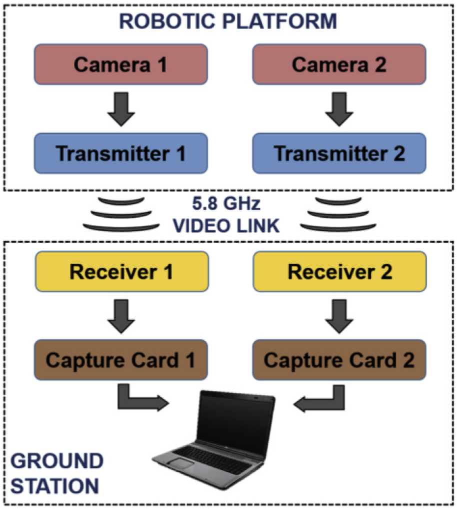


Fig. 2. Robot ←→ ground control communication. Visual data from each camera is sent via a transmitter/receiver pair to an off-board ground station for processing.

After hardware selection has been completed, the cameras need to be fixed to a robot. A quadcopter sUAV (<2 kg total) named BeeBot (shown later in Fig. 10) was utilized in this implementation (Sabo et al., 2017). This platform has the same 6 degrees-of-freedom as a flying insect and enables the fundamental capabilities of honeybee flight. This includes hovering and vertical take-off/landing while allowing for a reasonable payload (necessary to equip a quadcopter with honeybee senses). The robot moves by spinning the four propellers at different speeds to gain roll, pitch, yaw, and thrust control.

The honeybee has a near-panoramic field-of-view with a significant binocular overlap (~30° in the front and similar in the dorsal and ventral regions) (Seidl and Kaiser, 1981a). The only occlusion is from where their body obstructs their vision and creates a blind spot in the back. To imitate this, these 2 cameras are mounted symmetrically off the front of the quadcopter in a similar orientation to the honeybee eyes (see Figs. 3,4). The binocular overlap of the BeeBot is close to 60° as shown in Fig. 4 as opposed to ~30° for a bee. This is to minimize loss of information in the front of the quadcopter's field-of-view due to distortion from the lenses. In total, this results in BeeBot's field-of-view to be ~240° horizontally and ~150° vertically (very similar to the bee's ~280° by ~150°).

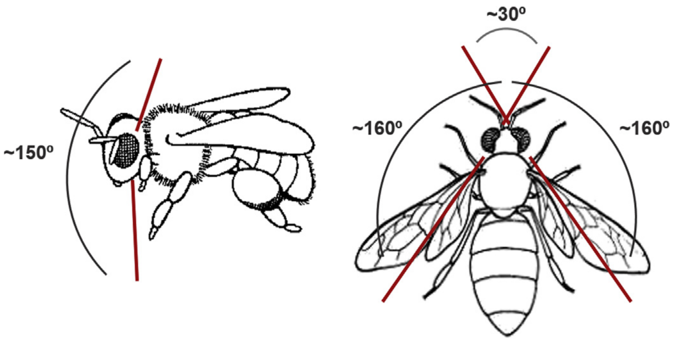


Fig. 3. Honeybee field-of-view. This figure displays the approximate vertical (left) and horizontal (right) fields-of-view, including binocular overlap, of the honeybee.

[1] The code for this implementation is open-source and available. Details can be found at http://brainsonboard.co.uk/2017/08/05/a-lightweight-inexpensive-robotic-system-for-insect-vision/.

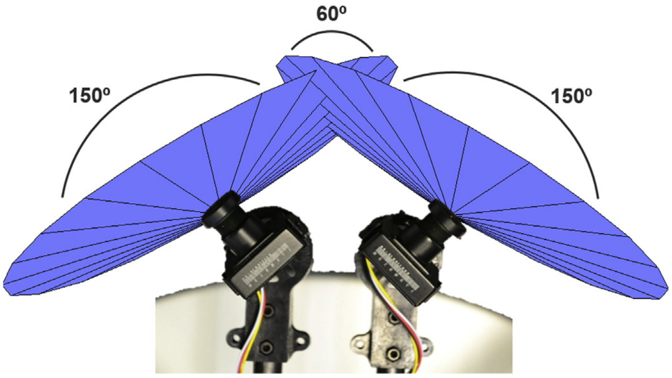


Fig. 4. Insect vision horizontal field-of-view. This figure displays the mounting configuration for optimal balance of field-of-view, overlap, and focal point on a robot.

5.2. Camera modelling

After the image has been captured by the sensor, the image is processed to compensate for the sensor properties, image noise, colour, etc. It was imperative to model the impact of radial and tangential distortion as this severely impacts the actual location of the desired pixel. Further, camera intrinsic parameters should be modelled in order to transform the image from the camera's view into the insect's view. The use of wide-angle lenses in this implementation also requires correction for lens distortion because they cause large pixel errors. Radial distortion is largely due to the wide-angle optics and causes the images to appear curved as the distance from the centre of the image increases. Tangential distortion is a result of the lens not being perfectly parallel to the imaging plane and can cause objects to appear closer than they are.

Bouguet (2015) describes how to transform any standard distorted image into one that you would get with a perfect pin-hole camera or vice versa. The camera matrix helps to transform the real world into pixel coordinates, and its parameters (the focal length (f_x, f_y), skew factor (s), and image centre (c_x, c_y)) are referred to as the camera intrinsic values. Extrinsic parameters are the other component of the transformation and correspond to the rotation and translation vectors which translate the coordinates of a 3D point in the world to a body-fixed coordinate system before the body-frame points are normalized.

Optimization is completed to find the best-fit camera model using Zhengyou Zhang (2000) popular calibration method implemented in the MATLAB camera calibration toolbox by Jean-Yves Bouguet (2015). These parameter values can be optimized using a set of known calibration images. To find the extrinsic and intrinsic parameter values, a series of images with well-defined patterns are taken with the camera at various perspectives and used in the optimizer. The images presented in this case were of a checkerboard pattern with known square size. As the coordinates are now known both in the real world and the image, a mapping between the two can occur. Zhang's method uses points extracted from the calibration images and their corresponding relationship to each other when they are at different orientations and positions. A nonlinear refinement method is then used based on the maximum likelihood criterion.

5.3. Insect vision model

To produce the resolution found in the compound eye of the honeybee, the implementation uses the established ommatidial placement model created by Andrew Giger (1996). Similar to Stürzl et al. (2010), the robotic system provides an extension of Giger's model to cover the complete FoV of the bee as measured by Seidl and Kaiser (1981a). The remapping of ommatidia assumes the bee head is a ball. While it is not, this is a reasonable approximation, and there is some uncertainty to the individual ommatidial position anyways due to small variations in each insect. The Giger model is used here, but other models could be used for ommatidia placement.

The Giger model consists of three quadratic equations: (1) describing the interommatidial spacing changing with elevation (El), (2) describing the interommatidial spacing changing with positive azimuth (Az_{pos}), and (3) describing the interommatidial change with negative azimuth (Az_{neg}). These equations form an iterative solution which can be used to enumerate a set of ommatidia placements over the desired angular field of view:

$$El += \left(0.000734 \cdot \left(El^2\right)\right) - (0.1042253 \cdot El) + 4.9 \tag{1}$$

$$Az_{neg} -= \left(0.00037 \cdot \left(Az_{neg}^2\right)\right) - (0.04462 \cdot Az_{neg}) + 3.438 \tag{2}$$

$$Az_{pos} += \left(0.00069 \cdot \left(Az_{pos}^2\right)\right) - (0.08333 \cdot Az_{pos}) + 4.6 \tag{3}$$

To create a hexagonal arrangement of placements, every other ommatidial row is offset by half a placement, and to position the maximum ommatidial density the placements are enumerated from a central angular position of 60° azimuthally from the forward position. Fig. 5 shows the ommatidial placements in green with the placements mapped onto the left and right cameras' fields-of-view coloured blue and red respectively. As the ommatidial placements also correspond to a row and column index during the enumeration process, these row-column locations are stored so that a compact representation of the bee field of view can be created.

5.4. Pixel selection and online processing

Pixel selection is done in offline processing. The camera model described in Section 5.2 is used to remap the ommatidial placements from azimuth and elevation coordinates to pixel locations on the camera CCD. These mappings are used to generate a look-up table for quick online processing between the desired destination row-column indices of each ommatidial placement and the source pixel locations in the camera image. This mapping is then used with the OpenCV cvRemap function (Bradski and Kaehler, 2000) to efficiently transfer the pixels from the camera image to an insect-centric image with interpolation if required. Fig. 6 shows the camera image pixel locations for the red ommatidial placements shown in Fig. 5.

5.5. Other considerations

Other considerations for image processing typically include a two-camera calibration, noise reduction, colour correction, and synchronisation between cameras. Like bees, we do not use the stereoscopic part of the visual field (though, this is possible with our system but left to future work), and so we do not show calibration between the two cameras here. Since some noise is desirable in embodiment and accurately represents the real-world, this is measured for evaluation but the correction in pre-processing is left for further evaluation in future work. Additionally, the colour of an image can be adjusted to compensate for varied light sources and other desired special effects. Again, this was left for future implementation as the implications of these corrections on modelling and embodiment are unknown at this stage.

Please cite this article in press as: Sabo, C., et al., A lightweight, inexpensive robotic system for insect vision, Arthropod Structure & Development (2017), http://dx.doi.org/10.1016/j.asd.2017.08.001

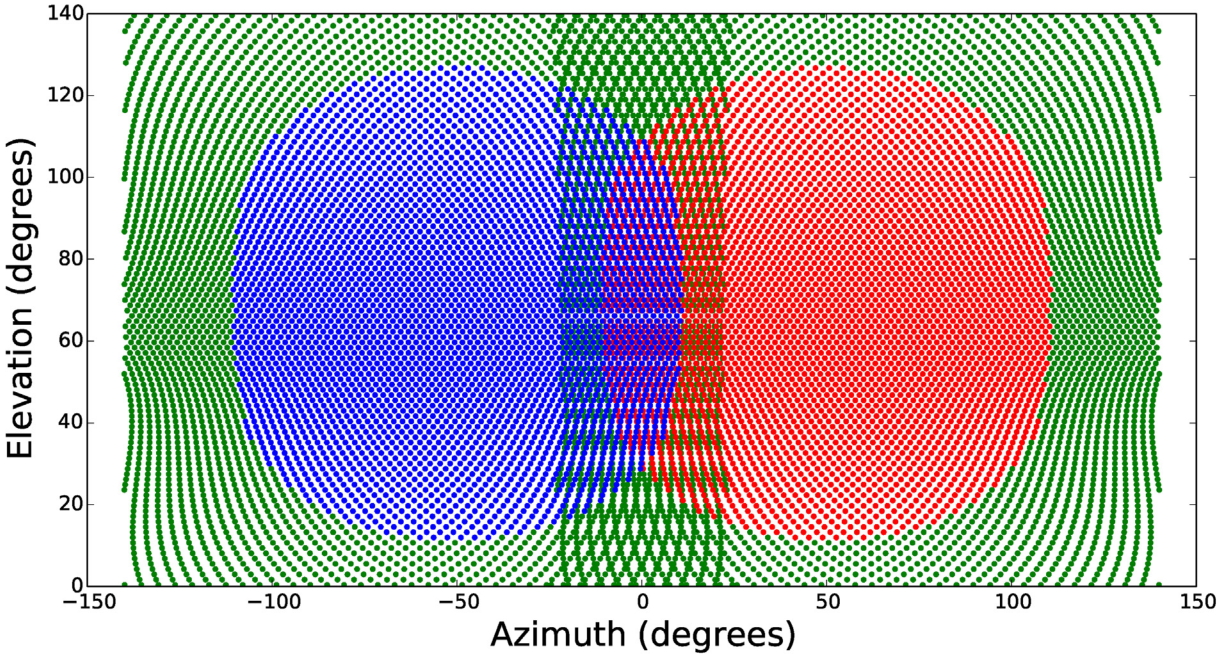


Fig. 5. Ommatidial placements in the field-of-view. This figure depicts the mapping of the ommatidia placements from the insect model (green) as well as those of which are in the field-of-view of each of the left (blue) and right (red) cameras.

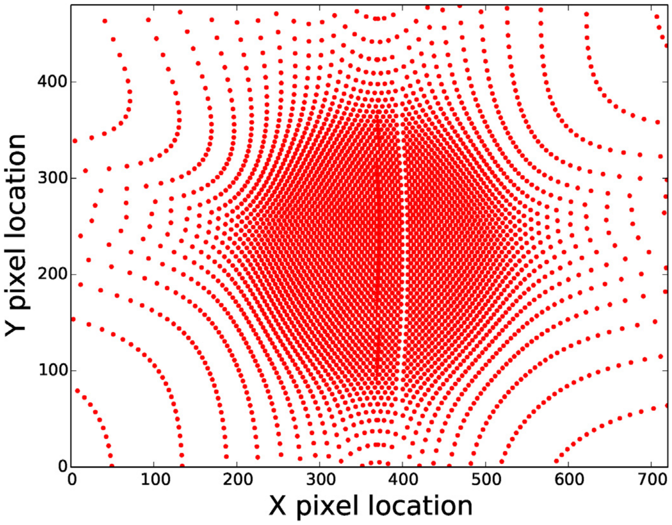


Fig. 6. Ommatidial placements remapped on camera image. After modelling of the cameras, the ommatidia placements get remapped and transferred to a new pixel location on the camera image (depicted here for the right (red) camera from Fig. 5). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article).

Further, one consideration for multiple cameras is synchronisation. This can be a significant problem if the processing of frames is undertaken across both cameras simultaneously. However, our system design reproduces each insect eye using a single camera and therefore processing for each camera is separate. This allows the camera processing to be undertaken in threads, avoiding synchronisation issues, but a different approach might be considered for visual tasks that require synchronized outputs.

6. Results and discussion

The visual system is evaluated on its ability to simulate insect visual characteristics and its suitability for further development. The evaluation of this design can then be completed by assessing the following criteria:

- **Evaluation:** Overall Design Performance
  Testing: Measure data transmission and final field-of-view, framerate, weight, cost, and resolution
- **Evaluation:** Camera Quality of Service
  Testing: Compute and assess the camera latency
- **Evaluation:** Image Remapping Model
  Testing: Evaluate pixel error and variance
- **Evaluation:** Robotic Embodiment
  Testing: Assess noise and motion detection in benchmark hallway navigation experiment

6.1. Overall design performance

The first test of the visual system was a visual inspection of the system output. The robotics embodiment testing involved taking a series of images and videos of well-defined images (a checkerboard pattern) along a hallway navigation task. Resulting images are shown here for each camera (noted left and right from now on) to capture the original image from the camera and the remapped model view. Also as part of testing, the data retrieved from the robotic system were compared to that calculated in a simulated bee world. The results for each are shown below in Figs. 7–9. As shown, the model output appears as expected and can be verified against Giger's model (Giger, 1996).

The final design of the visual system was also evaluated against the performance metrics (field-of-view, weight, cost, framerate, and resolution). As the objective is to realistically model honeybee visual inputs, the results should be compared against biology as well as several state-of-the-art lightweight robotic visual systems. These include a Parrot AR Drone 2.0 (Engel et al., 2012) as it is a good representation of a typical small quadcopter UAV, a Go-Pro Hero 4 (https://gopro.com) as they are commonly used with a wide range of other sUAVs (Drones, 2016), the catadioptric system by Stürzl et al. (2010), and the curved artificial compound eye

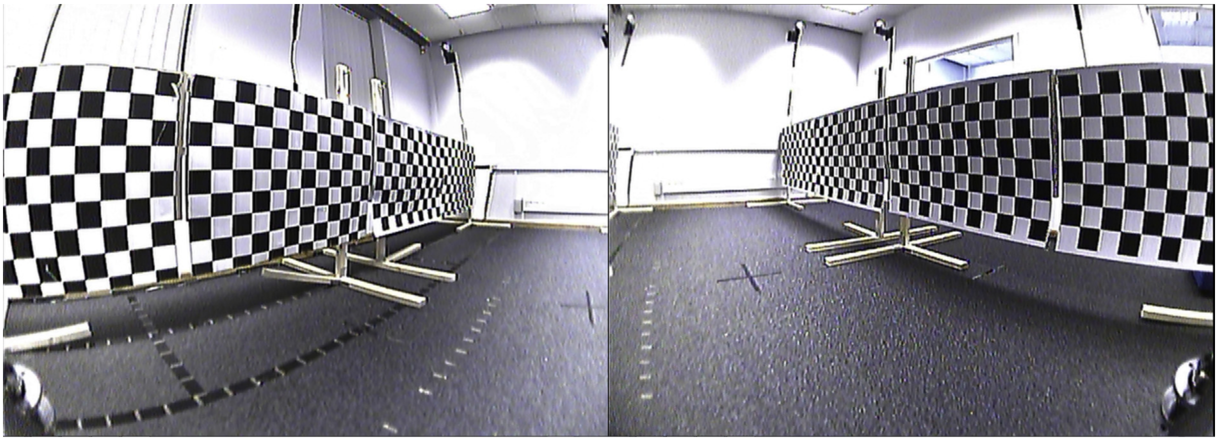


Fig. 7. Image from the camera output – original view. Depicts the camera view without any image processing or pixel selection.

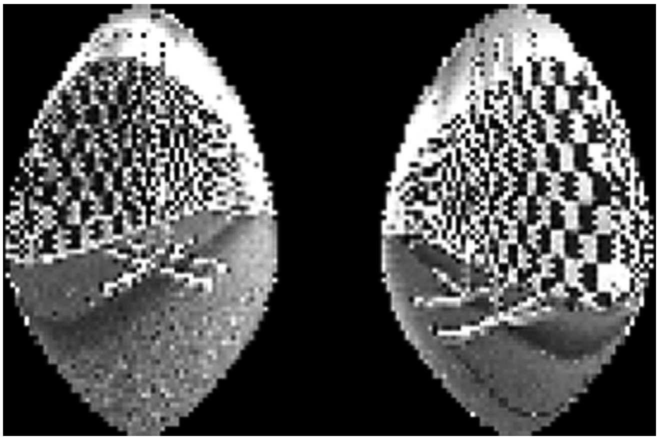


Fig. 8. Image from the camera output – insect view. Depicts the transformed and remapped insect view captured from the original camera view.

(Curvace) which has been utilized for autonomous navigation (Leitel et al., 2014).

BeeBot's visual system provides nearly the same visual field-of-view as the honeybee (240° × 150° versus 290° × 160° for the bee). The catadioptric system is the only comparable system which gives 280° × 180° whereas the Curvace artificial eye only provides 180° × 60°. Also, typical quadcopter cameras like the AR Drone have around 92° diagonal or 94° × 123° max with a Go-Pro. Further, BeeBot has comparable framerate, resolution, and cost to other lightweight robotics systems. The other large benefit over Go-Pro is the ability to transmit the data and process it off-board in real-time. BeeBot's camera system does fall short to the Curvace artificial eye in framerate (30 Hz versus 300 Hz) but makes up for it in cost and accessibility. The catadioptric system represents many unknowns, because the capabilities will depend upon both the choice of camera and the quality of the optic surface and lens used. This will have a strong bearing on the overall cost of the system.

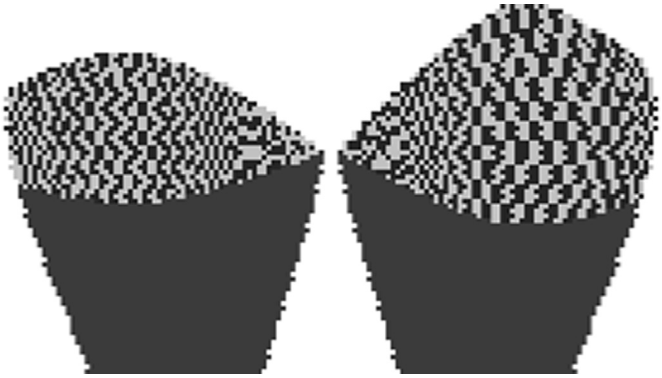


Fig. 9. Image from the virtual bee view. Depicts the matching view of the virtual bee in the simulated world.

The visual system is also extremely lightweight at 110 g which includes both cameras, transmitters, and cables. While this would need to be mounted on a robot larger than the AR Drone 2.0, this can undoubtedly be mounted on a sUAV as demonstrated with BeeBot. Further, the visual system weighs less than the Go-Pro showing its suitability for sUAVS. Weight could be reduced by increasing cost slightly to purchase newer, smaller cameras and/or moving computing on-board to eliminate the transmitter/receiver combination (though the additional processing power would come with a minor weight penalty).

6.2. *Camera quality of service*

Several metrics determine the suitability of the transmission protocol: noise, mean latency, and Quality of Service (QoS: a measure of the consistency of the latency) where noise and latency are typically a trade-off. Analog transmission provides the best QoS and latency for transfer as data are transferred with no confirmation of receipt. The analog data do, however, have to be converted into digital for input into computational models, and this stage can add latency and noise. The absence of confirmation of receipt also increases the chance of corrupted or missing data due to interference (Chebrolu et al., 2006). Wi-Fi is an example of a robust digital signal that guarantees the information arrival but it comes at the cost of latency and QoS. When data receipt fails, data transmission slows considerably to resend the data.

For this testing, video was streamed to the off-board computer for evaluation of the data transmission. The average latency was measured by outputting a single frame white flash signal through an Oculus DK1 screen and then measuring the latency from the time until a frame is received showing the flash, less the pixel switching time of the DK1. As a result, the averaged latency was measured to be:

Average Latency: 146 ms

While this might seem high, the current system must convert the analog data to digital data for use. The extra latency arises from the capture cards used, as well as the processing performed on the data. Given that the robotic platform and environment is a scaled version of an insect's world, the speed (and perceived speed) of the robot must also be scaled. This lowers the angular speeds perceived by cameras and, therefore, we must also lower the speed of the neural models to account for this disparity. Currently, these scaling factors will be in excess of 10×, and thus, the latency we measure is equivalent to less than 14.6 ms when scaled accordingly. Therefore, overall, the latency is acceptable given the time scale scaling of the system relative to insect vision. Dragonflies, which require extremely low latencies for prey capture, have only been recorded as having latencies of 30 ms from sensory input to wing motor response (Olberg et al., 2007). As a result, this latency is acceptable service expected for most applications and given the time scale scaling of the system relative to insect vision. This could be improved further by the use of low-latency video capture hardware and/or by moving processing on-board.

6.3. *Image remapping model*

The performance of the image remapping determines the correctness of the pixel selection method. As Giger's ommatidia placement model has been thoroughly verified against biology (Giger, 1996), the primary error in the pixel selection comes from the distortion from the camera/lens system and the corresponding model of the system found in Section 5.2. Therefore, the effectiveness of the camera model was evaluated by pixel error. The results from the model produce the desired outcome with minimal distortion, but it is not possible to eliminate all errors. Some distortion artefacts could still be observed, which motivated a further analysis of the pixel error. In this analysis, the efficiency of the camera model was measured using the difference between a set of known pixel locations from the model output and from the undistorted image. Using the images of the checkerboard pattern (shown in Figs. 7–9), the average pixel error and corresponding variance in the 2D image plane were found to be:

Average Pixel Error: [0.642 0.475]

Pixel Error Variance: [0.214 0.158]

It can be seen that both the average and the variance of the pixel error is small with the maximum error being approximately 3 pixels. This is found to be suitably small, as it is well beyond insect vision resolution. The implication of this is that the remapped image used in combination with the insect ommatidia pixel selection method will be an accurate representation of what an insect would see in the real world.

6.4. *Robotic embodiment*

The system was tested for use with embodied visual processing models by utilizing BeeBot (Fig. 10), the robotic platform on which the visual hardware was mounted and previously described, in a scaled laboratory environment of a standard bee benchmark test. Bee navigation is often investigated by examining their capability to traverse corridors (Kirchner and Srinivasan, 1989). This behaviour is largely explained by three simple rules: (1) maintain lateral position by balancing the angular velocity in the left and right eye, (2) hold forward velocity constant by regulating the total angular velocity against an empirical setpoint, and (3) adjust altitude by balancing the ventral angular velocity against an empirical setpoint (Kirchner and Srinivasan, 1989; Srinivasan et al., 1991). In this research, the lateral degree-of-freedom was isolated for study and so the left and right fields-of-view were of interest. Accordingly, the lab was configured with two walls along the length (fitted with the checkerboard pattern along the walls) and a motion tracking

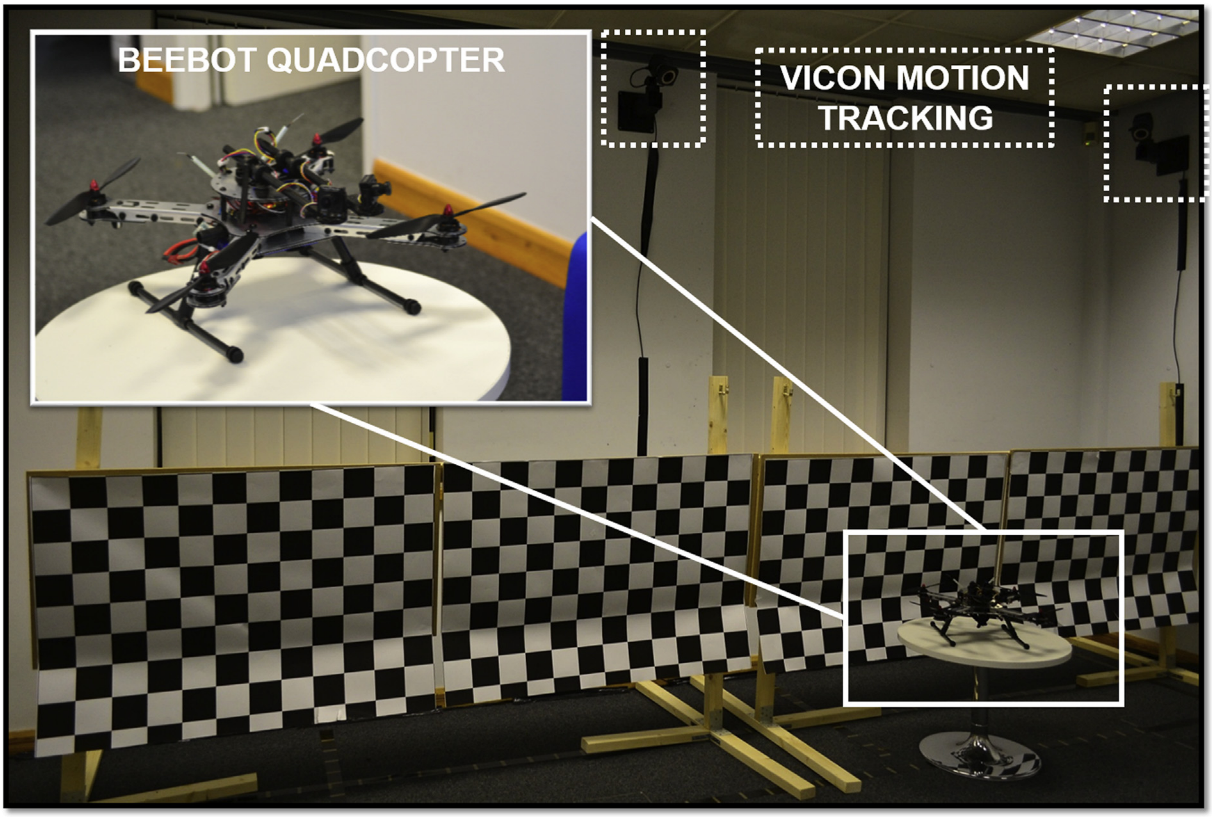


Fig. 10. Scaled laboratory benchmark testing scenario. Experiments with honeybees use corridors to evaluate reactive flight control. This was replicated in this scaled laboratory environment. The vision system was mounted on BeeBot quadcopter and used for mobile testing.

system providing ground truth data (Fig. 10). An off-board workstation processes the camera data as previously described.

In each trial, BeeBot starts at one end of the hallway and then travels to the other end at a fixed velocity for a number of trials as can be seen in Figs. 11,12. At the end of the hallway, the robot decelerated quickly as it approached a wall. A total of 12 trials was completed for approximately the same forward velocity and path. Incoming visual data were collected from the cameras along with the exact time, position, and velocity from the tracking system. The addition of tracking data helps to determine the impact of position and velocity on the ensuing camera frames and motion detection. The resultant paths (for each case and the mean) from these trials are shown in Fig. 11. Additionally, the velocity is shown in Fig. 12. It can be seen that the distance from the wall does not vary more than ±5 cm and that the velocity has some variations throughout each trial. Part of the source of variance in the velocity is noise, but a further look into this needs to be completed, as small changes in velocity can cause large changes in angular velocity. So it is important to know when the velocity changes are genuine.

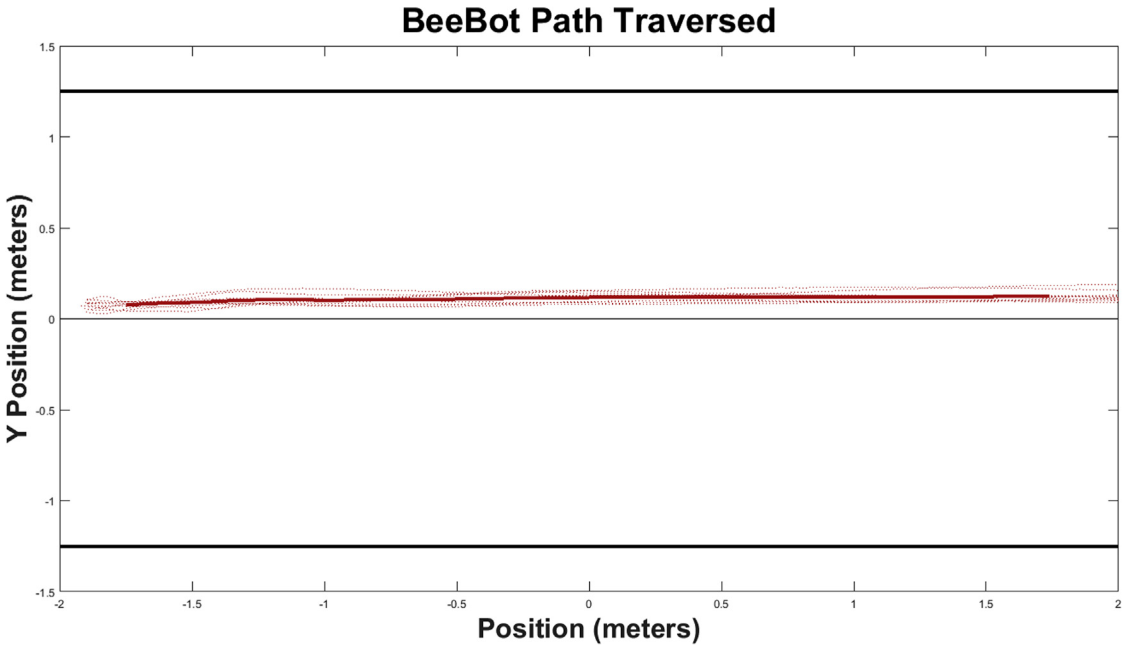


Fig. 11. BeeBot path traced in hallway test. The path traversed by BeeBot between the 2 walls for each trial is plotted with a dotted line and with the mean in bold.

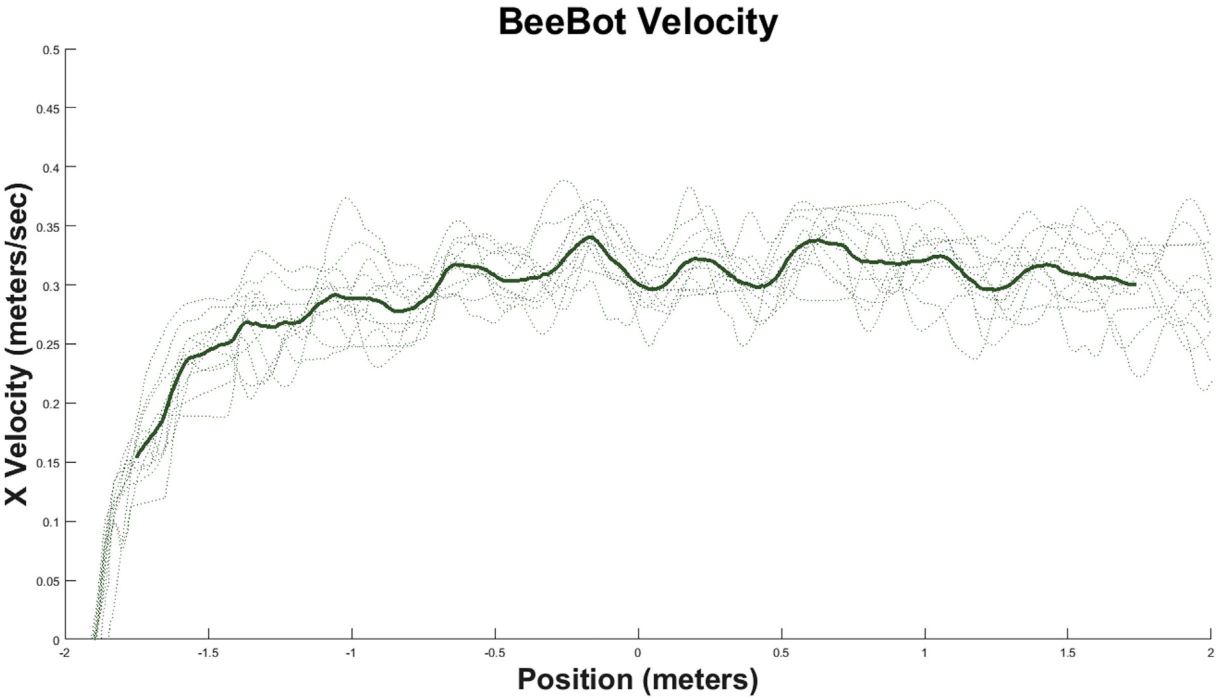


Fig. 12. BeeBot velocity in hallway test. The velocity of BeeBot for each trial is plotted with a dotted line and with the mean in bold.

The video data were then processed for comparison against position and velocity. As stated earlier, noise can be accounted for in the camera modelling and is an important measure of the quality of the data transmission method. The noise in the system (mainly due to interference or dropouts/transmission error) is evaluated further here, since it was not cancelled out in the calibration step. Each frame in the raw video was treated as a separate image and filtered using a linear Wiener filter. The filter adapts itself to the local image variance: if it's large, little smoothing is completed and vice versa if the variance is small. While it requires more computational time than comparable methods, it often produces better results, particularly at preserving edges and other high-frequency parts of an image (Lim, 1990). The noise for each trial and the corresponding average and variance is plotted in Fig. 13. It can be seen that there are some large spikes in noise which could cause unidentified artifacts in the images at these points (which in turn can affect visual processing).

Finally, the system was evaluated for its potential for embodiment of computational models based on flying insect vision. Because insect behaviour relies heavily on the detection of optic flow, we investigate and assess biological models of optic flow. The performance of our camera-based implementation is compared to a perfect, simulated bee world. For the latter, we use the robot telemetry data obtained by the motion tracking system to drive the trajectories of a virtual bee in a simulated ray-traced environment, BeeWorld (Cope et al., 2016). Both instances use the same number of ommatidia and ommatidial placements.

The insect model visual system output was used to feed into biologically-based models of angular velocity, or optic flow, estimation. Specifically, we employ two different methods: the Reichardt-Hassenstein Detector (Egelhaaf et al., 1989) and the Angular Velocity Detector Unit (AVDU) model by Cope et al. (2016). The Reichardt Detector is based on Elementary Motion Detectors (EMDs) which sense brightness changes across the eye. The AVDU is founded on the Reichardt Detector and essentially calculates angular velocity by taking the ratio of outputs from two neighbouring detector units. For each method, the models are scaled to the dimensions of the insect model. The AVDU model shows a log-linear response to increasing angular velocity and exhibits invariance of response to changes in the environment's contrast or spatial frequency unlike the Reichardt Detector and typical computer vision motion detection algorithms.

The optic flow over each "eye" for each method is summed, normalized according to the forward velocity, and used as the final output for comparison. We divide by the velocity for better analysis, since small changes in velocity can cause large changes in optic flow. Shown in the subsequent figures are the resultant motion from the AVDU model in BeeWorld (Fig. 14) and the AVDU model (Fig. 15) in the scaled, lab world. Additionally, we calculate the Reichardt Detector on the camera data (Fig. 16). In this assessment, we calculate the output of the model in each "eye" (or camera), left and right, individually. The mean and variance is plotted across the 12 trials. The simulated world does not contain the noise and detail of the real environment. It consists only of the two patterned walls (see Fig. 11), but since the AVDU model is contrast and spatial frequency invariant, the base values of angular velocity should be equivalent across tests. The only differences should arise from any noise in the real environment to which the AVDU model is sensitive.

The results show good agreement across the virtual and real-world trials for the AVDU model with almost identical means and overlapping confidence intervals. The trend shows even better matching especially after the initial onset of motion. As previously stated and expected, noise is much larger in the robotic system than in the BeeWorld model. The left is consistently larger than the right which is consistent with expectations as BeeBot was slightly closer to the left wall. The main differences appear right after the onset and at the end. Both are to be expected since the largest changes in velocity and distance from the wall occur in this region. Also, we approach a blank wall at the end of the hallway in the laboratory

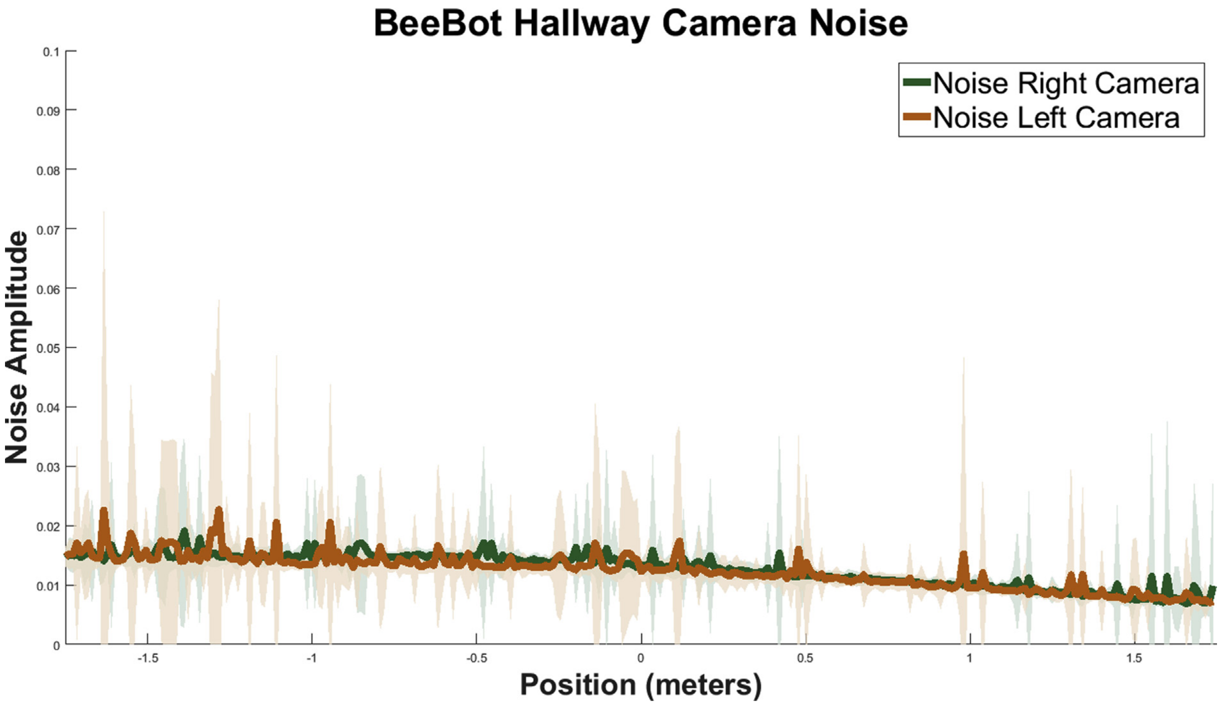


Fig. 13. Image noise during hallway test. The image noise was calculated and filtered. The large spikes seen can cause unidentified artifacts in the images. The spikes are noted for later study during visual processing.

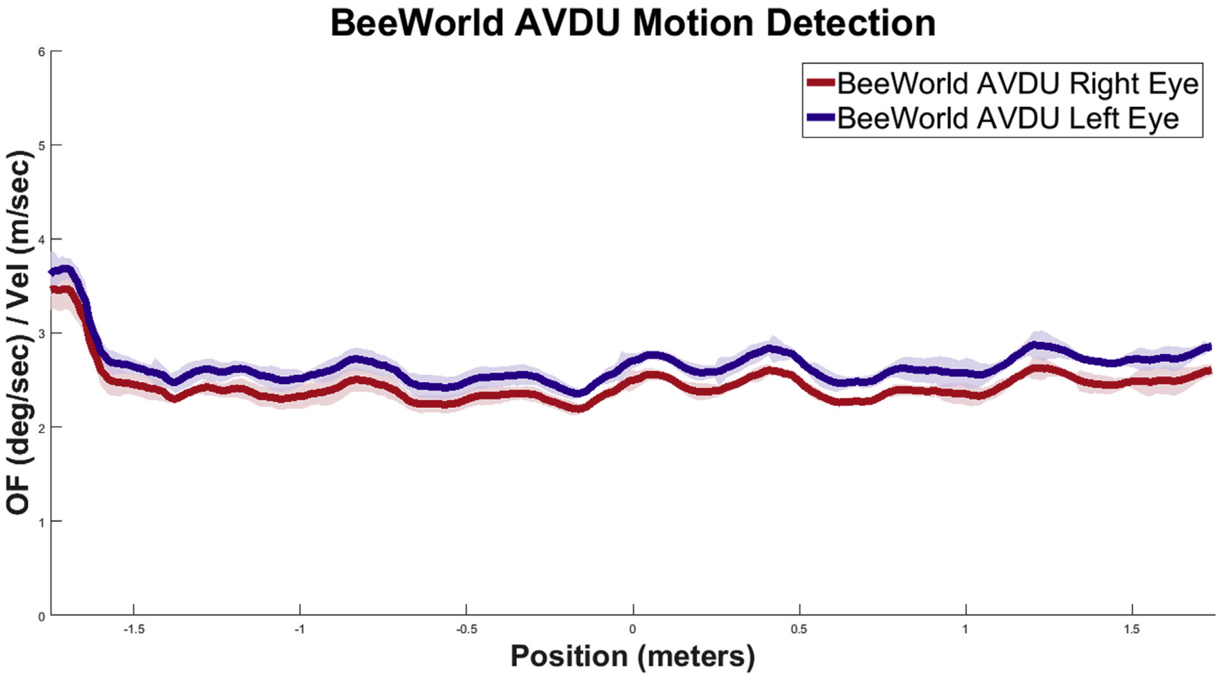


Fig. 14. BeeWorld AVDU model optic flow. The AVDU model calculates optic flow in BeeWorld which is summed over each the left (blue) and right (red) "eye". The average over the trials is depicted (dark solid line) along with the variance (shaded region).

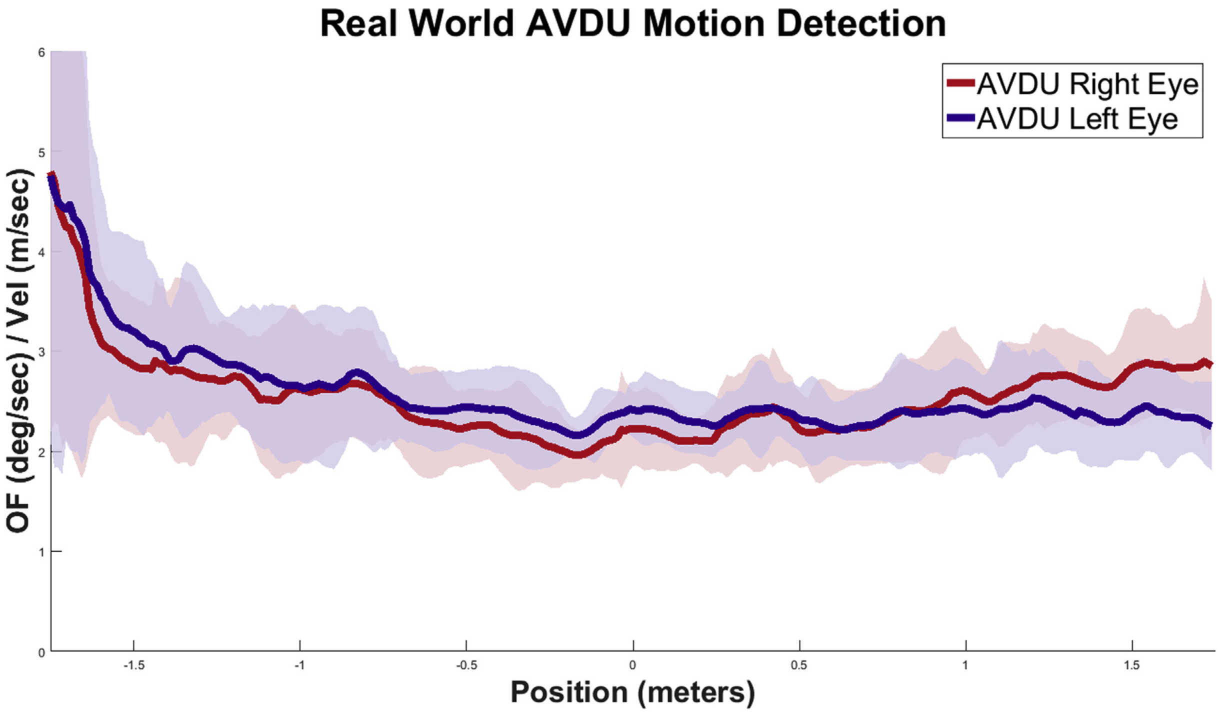


Fig. 15. Real world BeeBot AVDU model optic flow. The AVDU model calculates optic flow on the system which is summed over each the left (blue) and right (red) camera. The average over the trials is depicted (dark solid line) along with the variance (shaded region).

environment whereas BeeWorld doesn't have an end. The AVDU model trend even has a similar pattern of deviations across simulation and real-world in the responses. The variances in the virtual world arise from minute changes in position and velocity since the simulation is deterministic. As a result, it can be assumed that variations of any higher magnitude in the real world come from noise and additional artifacts in that environment and can be used for evaluation of further embodiment.

The camera system was further verified by applying the Reichardt-Hassenstein Detector method on both eyes (Fig. 16). The Reichardt Detector is much more susceptible to noise which can be seen in the frequent spikes and larger variations than the AVDU

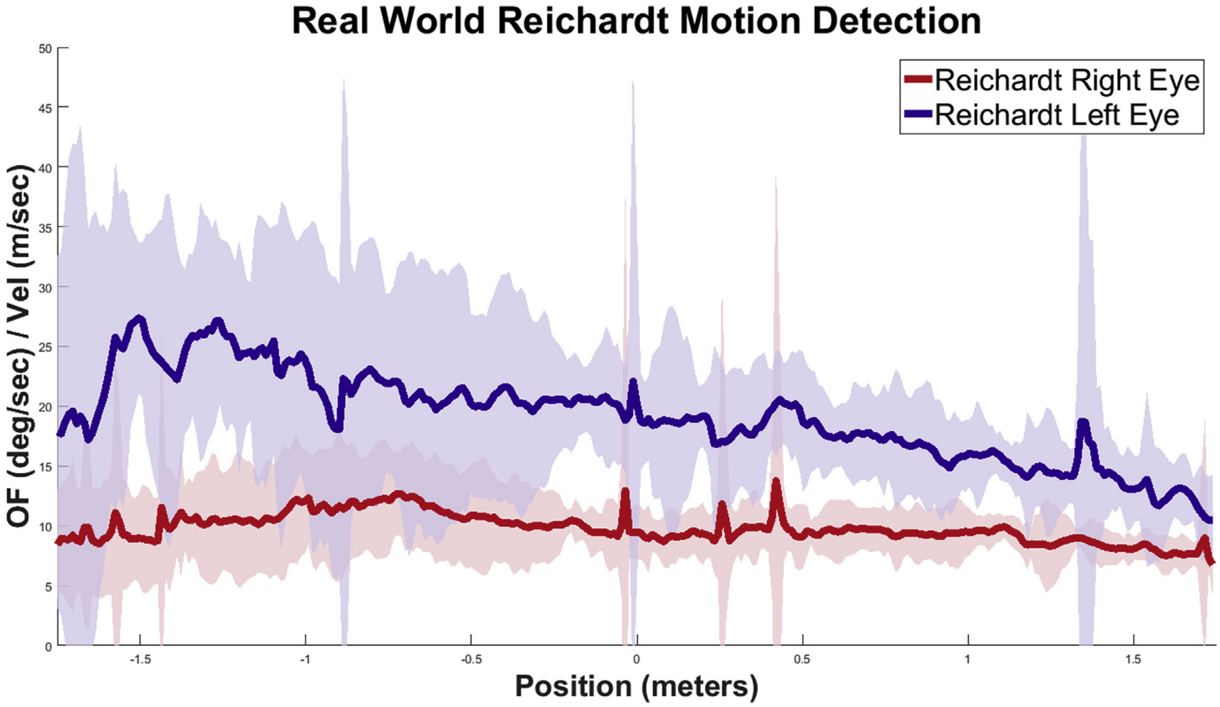


Fig. 16. Real world BeeBot Reichardt detector optic flow. The Reichardt Detector calculates optic flow on the system which is summed over each the left (blue) and right (red) camera. The average over the trials is depicted (dark solid line) along with the variance (shaded region).

method. This can be verified against the trends in the camera system noise that are shown in Fig. 13. Otherwise, noise levels are relatively low and within expectations. A further look into specific examples would help to determine where noise is impacting optic flow calculations. This even further motivates moving computations on-board and removing the transmission/receiver in the system in order to eliminate drop-outs and interference.

7. Conclusions

A methodology for designing and implementing an insect-inspired robotic visual system for low-cost, lightweight robotics was presented. The comprehensive approach gives a method for selecting the robotics hardware, modelling the camera system, modelling ommatidia placements, and doing pixel selection for optimal online processing. BeeBot, the sUAV used in this research, is a proof-of-concept platform demonstrating support of necessary payload to replicate the sensing capabilities which are vital to bees' flight navigation including their wide visual field-of-view. The implementation employed here based on honeybee vision is open-source and can be found online.

The visual system was tested and analysed based on its ability to simulate insect visual characteristics and its suitability for further development of insect navigation and cognitive models. Results show good accuracy of the camera model used and excellent comparison of optic flow to a perfect, simulated world. Furthermore, this was done with inexpensive ($350 total), lightweight (1850 g) off-the-shelf components, which are advantageous for research development.

The system could be improved with minor modifications and the use of new technological advances. While the insect view appears pixelated in the results, it is accurate in relation to the interommatidial distance and could be improved at the expense of additional computational cost. Further work needs to be done to analyse the sources of noise and isolate those variations that can be expected from data transmission and those that can be expected from the environment. This could be addressed by moving processing on-board. This solution would reduce noise and latency from lengthy data transmissions through multiple devices. It also improves quality of service considerably while dramatically increasing the autonomous capabilities of the robot. This could be realized with the use of improved mobile computing technology like NVIDIA GPUs (http://www.nvidia.co.uk/). Additionally, because the physics and aerodynamics of flight do not scale linearly, more work and analysis needs to be completed to study the dynamic response of both flying insects and small quadcopters like BeeBot.

Acknowledgments

The research was supported by the EPSRC Green Brain Project (grant number EP/J019690/1) and EPSRC Brains on Board (grant number EP/P006094/1). The authors would also like to acknowledge J. A. R. Marshall, E. Vasilaki, and K. Gurney for their support and guidance in this research.

References

Autrum, H., Stoecker, M., 1950. Die Verschmelzungsfrequenzen des Bienenauges. Z. für Naturforsch. B 5 (1), 38–43. http://dx.doi.org/10.1515/znb-1950-0107.

Barron, A., Srinivasan, M.V., 2006. Visual regulation of ground speed and headwind compensation in freely flying honey bees (*Apis mellifera* L.). J. Exp. Biol. 209 (Pt 5), 978–984. http://dx.doi.org/10.1242/jeb.02085.

Bouguet, J.-Y., 2015. Camera Calibration Toolbox for MATLAB. Retrieved March 1, 2016 from: http://www.vision.caltech.edu/bouguetj/calib_doc/.

Bradski, G., Kaehler, A., 2000. OpenCV Library. Dr. Dobb's Journal of Software Tools for the Professional. Retrieved June 1, 2014 from: http://opencv.org/.

Chebrolu, K., Raman, B., Sen, S., 2006. Long-distance 802.11b links. In: Proceedings of the 12th Annual International Conference on Mobile Computing and Networking – MobiCom'06. ACM Press, New York, New York, USA, p. 74. http://dx.doi.org/10.1145/1161089.1161099.

Cope, A., Sabo, C., Yavuz, E., Gurney, K., Marshall, J.A.R., Nowotny, T., Vasilaki, E., 2013. The Green Brain Project – Developing a Neuromimetic Robotic Honeybee. Living Machines Conference, London, July, 2013.

Cope, A., Sabo, C., Gurney, K., Vasilaki, E., Marshall, J.A.R., 2016. A model for an angular velocity-tuned motion detector accounting for deviations in the corridor-centering response of the bee. PLOS Comput. Biol. 12 (5) http://dx.doi.org/10.1371/journal.pcbi.1004887.s001.

Drones, S., 2016. Best Drones for GoPro Cameras. Retrieved June 18, 2016, from: http://smashingdrones.com/gopro-camera-best-drones-for-gopro/.

Dyer, A.G., Chittka, L., 2004. Bumblebees (*Bombus terrestris*) sacrifice foraging speed to solve difficult colour discrimination tasks. J. Comp. Physiol. A Neuroethol. Sens. Neural Behav. Physiol. 190 (9), 759–763. http://dx.doi.org/10.1007/s00359-004-0547-y.

Egelhaaf, M., Borst, A., Reichardt, W., 1989. Computational structure of a biological motion-detection system as revealed by local detector analysis in the fly's nervous system. J. Opt. Soc. Am. 6 (7), 1070. http://dx.doi.org/10.1364/JOSAA.6.001070.

Engel, J., Sturm, J.J., Cremers, D., 2012. Camera-based navigation of a low-cost Quadrocopter. In: Intelligent Robots and Systems (IROS), 320, pp. 2815–2821. http://dx.doi.org/10.1109/IROS.2012.6385458.

Esch, H.E., Zhang, S., Srinivasan, M.V., Tautz, J., 2001. Honeybee dances communicate distances measured by optic flow. Nature 411 (6837), 581–583. http://dx.doi.org/10.1038/35079072.

Giger, A.D., 1996. Honeybee vision: analysis of pattern orientation. J. Exp. Biol. 200 (8), 1271–1280. http://hdl.handle.net/1885/10813.

Gunn, T., 2013. Vibrations and Jello Effect, Causes and Cures. Retrieved June 18, 2016, from: http://www.flitetest.com/articles/vibrations-and-jello-effect-causes-and-cures.

Hecht, S., Wolf, E., 1929. The visual acuity of the honeybee. J. Gen. Physiol. 12 (6), 727–760. Retrieved from: http://www.ncbi.nlm.nih.gov/pubmed/19872494.

Ibbotson, M.R., 2001. Evidence for velocity-tuned motion-sensitive descending neurons in the honeybee. Proc. Biol. Sci.R. Soc. 268 (1482), 2195–2201. http://dx.doi.org/10.1098/rspb.2001.1770.

Kirchner, W.H., Srinivasan, M.V., 1989. Freely flying honeybees use image motion to estimate object distance. Naturwissenschaften 76 (6), 281–282. http://dx.doi.org/10.1007/BF00368643.

Leitel, R., Brückner, A., Buß, W., Viollet, S., Pericet-Camara, R., Mallot, H., Bräuer, A., 2014. Curved artificial compound-eyes for autonomous navigation. In: Thienpont, H., Mohr, J., Zappe, H., Nakajima, H. (Eds.), International Society for Optics and Photonics. http://dx.doi.org/10.1117/12.2052710.

Lihoreau, M., Chittka, L., Le Comber, S.C., Raine, N.E., Sternberg, R.J., Pretz, J.E., Burr, A., 2012. Bees do not use nearest-neighbour rules for optimization of multi-location routes. Biol. Lett. 8 (1), 13–16. http://dx.doi.org/10.1098/rsbl.2011.0661.

Lim, J.S., 1990. Two-dimensional Signal and Image Processing. Prentice Hall.

Lungarella, M., Sporns, O., 2006. Mapping information flow in sensorimotor networks. PLoS Comput. Biol. 2 (10), e144. http://dx.doi.org/10.1371/journal.pcbi.0020144.

Markram, H., 2006. The blue brain project. Nat. Rev. Neurosci. 7 (2), 153–160. http://dx.doi.org/10.1038/nrn1848.

Neumann, T.R., 2002. Modeling insect compound eyes: space-variant spherical vision. LNCS 2525, 360–367. http://dx.doi.org/10.1007/3-540-36181-2_36.

Olberg, R.M., Seaman, R.C., Coats, M.I., Henry, A.F., 2007. Eye movements and target fixation during dragonfly prey-interception flights. J. Comp. Physiol. A Neuroethol. Sens. Neural Behav. Physiol. 193 (7), 685–693. http://dx.doi.org/10.1007/s00359-007-0223-0.

Olberg, R.M., Worthington, A.H., Venator, K.R., 2000. Prey pursuit and interception in dragonflies. J. Comp. Physiol. A Sens. Neural Behav. Physiol. 186 (2), 155–162. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/10707313.

Pfeifer, R., Bongard, J., Grand, S., 2007. How the Body Shapes the Way We Think a New View of Intelligence. MIT Press. Retrieved from: https://mitpress.mit.edu/books/how-body-shapes-way-we-think.

Power, M.E., 1943. The effect of reduction in numbers of ommatidia upon the brain of *Drosophila melanogaster*. J. Exp. Zool. 94 (1), 33–71. http://dx.doi.org/10.1002/jez.1400940103.

Rabin, J.C., 2010. Visual perception: a clinical orientation (4th ed.). Optometry Vis. Sci. 87 (8), 615–616. http://dx.doi.org/10.1097/OPX.0b013e3181eee2d1.

Sabo, C., Yavuz, E., Cope, A., Gurney, K., Vasilaki, E., Nowotny, T., Marshall, J.A.R., 2017. An inexpensive flying robot design for embodied robotics research. In: 2017 International Joint Conference on Neural Networks, Anchorage, May 15–18, 2017.

Seidl, R., Kaiser, W., 1981a. Visual field size, binocular domain and the ommatidial array of the compound eyes in worker honey bees. J. Comp. Physiol. 143 (1), 17–26. http://dx.doi.org/10.1007/BF00606065.

Song, Y.M., Xie, Y., Malyarchuk, V., Xiao, J., Jung, I., Choi, K.-J., Rogers, J.A., 2013. Digital cameras with designs inspired by the arthropod eye. Nature 497 (7447), 95–99. http://dx.doi.org/10.1038/nature12083.

Srinivasan, M., Zhang, S., Lehrer, M., Collett, T., 1996. Honeybee navigation en route to the goal: visual flight control and odometry. J. Exp. Biol. 199 (Pt 1), 237–244. Retrieved from: http://www.ncbi.nlm.nih.gov/pubmed/9317712.

Srinivasan, M.V., 2011. Honeybees as a model for the study of visually guided flight, navigation, and biologically inspired robotics. Physiol. Rev. 91 (2), 413–460. http://dx.doi.org/10.1152/physrev.00005.2010.

Srinivasan, M.V., 2015. Where paths meet and cross: navigation by path integration in the desert ant and the honeybee. J. Comp. Physiol. A Neuroethol. Sens. Neural Behav. Physiol. 201 (6), 533–546. http://dx.doi.org/10.1007/s00359-015-1000-0.

Srinivasan, M.V., Zhang, S.W., Chahl, J.S., Barth, E., Venkatesh, S., 2000. How honeybees make grazing landings on flat surfaces. Biol. Cybern. 83 (3), 171–183. http://dx.doi.org/10.1007/s004220000162.

Srinivasan, M.V., Lehrer, M., Kirchner, W.H., Zhang, S.W., 1991. Range perception through apparent image speed in freely flying honeybees. Vis. Neurosci. 6 (05), 519. http://dx.doi.org/10.1017/S095252380000136X.

Stürzl, W., Boeddeker, N., Dittmar, L., Egelhaaf, M., 2010. Mimicking honeybee eyes with a 280 degrees field of view catadioptric imaging system. Bioinspir. Biomim. 5 (3) http://dx.doi.org/10.1088/1748-3182/5/3/036002.

Thurrowgood, S., Moore, R.J.D., Soccol, D., Knight, M., Srinivasan, M.V., 2014. A biologically inspired, vision-based guidance system for automatic landing of a fixed-wing aircraft. J. Field Robot. 31, 699–727. http://dx.doi.org/10.1002/rob.21527.

Webb, B., 2013. Issues in invertebrate learning raised by robot models. In: Menzel, R., Benjamin, P. (Eds.), Invertebrate Learning and Memory, pp. 81–88.

Zhang, Z., 2000. A flexible new technique for camera calibration. IEEE Trans. Pattern Anal. Mach. Intell. 22 (11), 1330–1334. http://dx.doi.org/10.1109/34.888718.

Zufferey, J.-C., 2005. Bio-Inspired Vision-Based Flying Robots (Doctoral Dissertation). Retrieved from: https://infoscience.epfl.ch/record/33663.