Technical University Chemnitz

Faculty of Computer Science

Professorship of Neurorobotics

TECHNISCHE UNIVERSITÄT
CHEMNITZ

Master Thesis

Autonomous Navigation of a mobile robot by using fruit fly brain models for optic flow and path integration

For obtaining the academic degree

Master of Science

Submitted by

Ahmad Hassanien

Matriculation-Nr: 696652

Course of study: Neurorobotics

First supervisor: Prof. Dr. Florian Röhrbein

Second supervisor: S Yousuf Imam, M.Sc.

Contents

Statement of Authorship

I hereby confirm to the Chemnitz University of Technology that this thesis is exclusively my own work and does not use any external materials other than those stated in the text.

This work contains no plagiarism, and all sentences or passages directly quoted from other people's work or including content derived from such work have been specifically credited to the authors and sources.

This paper has neither been submitted in the same or a similar form to any other examiner nor for the award of any other degree, nor has it previously been published.

_____

List of abbreviations

LPTC: Lobula plate tangential cells

HS: Horizontal system

VS: Horizontal system

Mi: Medulla intrinsic neuron

Tm: Trans-medullary neuron

CX: Central Complex

PCB: Proto Cerebral bridge

FB: Fan Shaped Body

SLAM: Simultaneous Localization and Mapping

LIDAR: Light Detection and Ranging

IMU: Inertial Measurement Unit

ANN: Artificial Neural Network

List of Figures

1. Introduction: The next generation of neurorobots

What differentiates a “neurorobot” from more traditional types of robotics is that it must incorporate knowledge from neuroanatomy and neurophysiology, i.e. with a neural network modeled after the architecture of a real biological nervous system [1]. One of the earliest neurorobotic systems is the Valentino Braitenberg vehicle (1986), based on simple photo electric sensors, it was able to exhibit some simple behaviors.

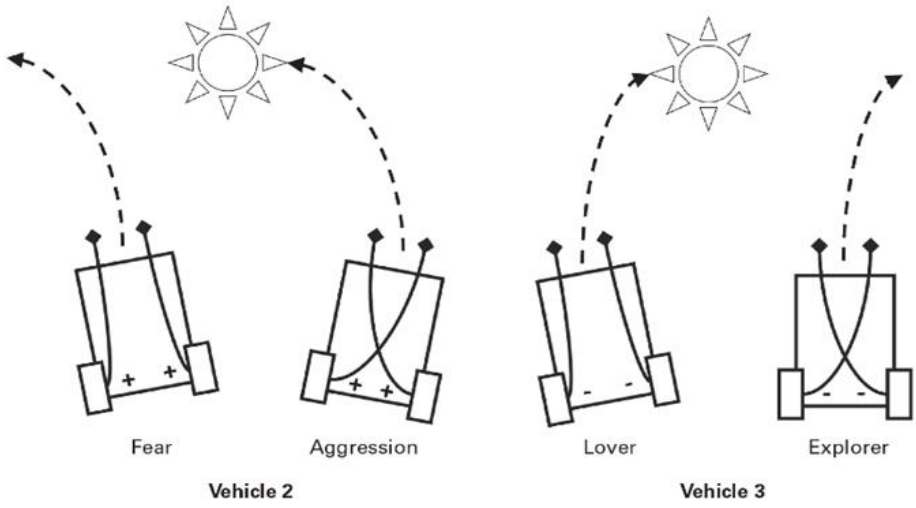

Figure 1.1: The Breitenberg vehicle. It was equipped with photo electric sensors, allowing the robot to exhibit behaviors such as "fear" or "aggression" towards a light source, depending on how the light sensors and motors are wired. That is, if motors and sensors were connected ipsilaterally (on the same side) or contralaterally (on opposite sides). It is interesting to note that even without any implementation for learning and memory, this robot could still exhibit flexible, adaptive behavior [2].

More modern types of neurorobots started including more sophisticated models of neurons from the central nervous systems of animals, such as Central Pattern Generators involved in rhythmic movements in animals such as walking or swimming,

as well as utilizing neuromorphic hardware boards, which can run artificial spiking neural nets for enhanced efficiency. These “artificial brains” mimic biological neural circuitry and possess similar features like parallel processing and efficient energy consumption. Energy efficient computation is one of the hallmarks of biological nervous systems, and especially insects, and finding ways to replicate it in robotics would be of immense benefit to the field.

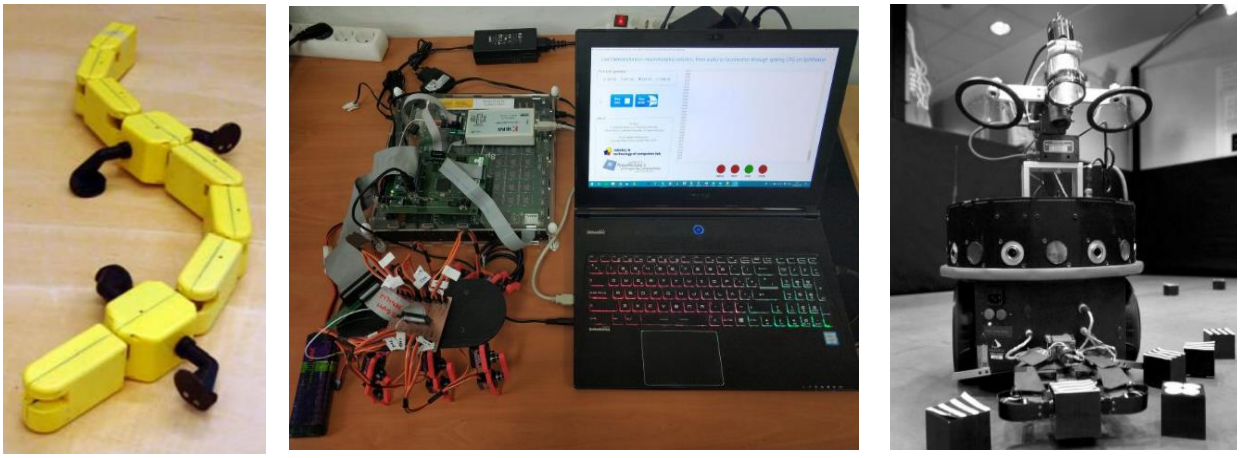

Figure 1.2: Left to right: Modern examples of neuro-robots. Salamander robot [3] is an example of a Neurorobotic system where the movements of the limbs are generated by Central Pattern Generators, which generate rhythmic motions for crawling on land and switching to a different pattern when swimming. Neuropod [4], which used a Spiking Neural Network implemented on the Neuromorphic board Spinnaker 3 to generate the walking movement. Darwin VII [5], the mobile robot with a simulated sense of taste for metals (the “taste” sensor measures the conductivity of the metal blocks). The robot is presented with two kinds of blocks, type 1 has black and white stripes, type 2 has a circular pattern. Additionally, both block types “taste” different (different metal conductivities). Based on the taste, the robot receives negative or positive feedback. The robot was observed for changes in its behavior based on the visual and taste inputs it received during the experiment.

While the design of neurorobots is informed from biological nervous systems, it also goes the other way around: where neurorobots can inform us about how a nervous system works, serving as a form of computational neuroethology [47]. By building a

robot with a biologically comparable brain model, one study showed how neuromodulators affect behavior by introducing artificial lesions in the simulated brain areas and observing the change in behavior [6].

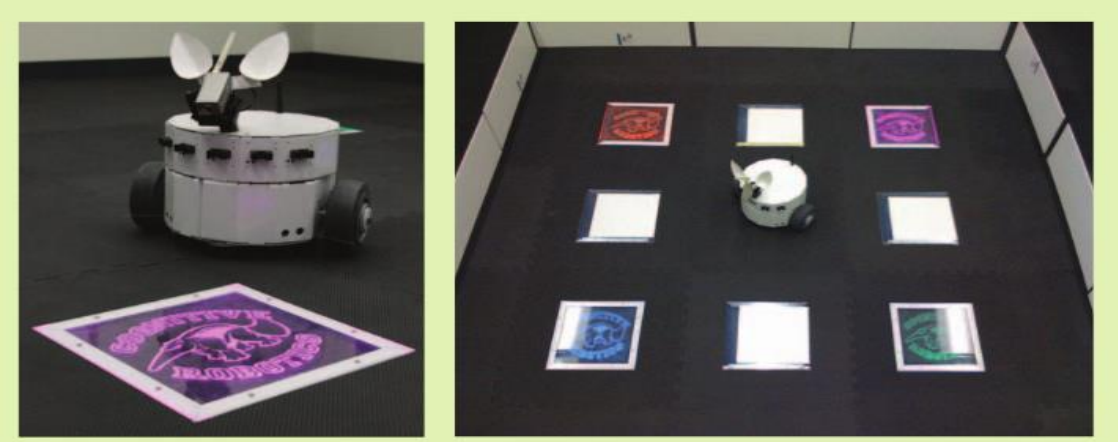

Figure 1.3: In [6], the authors used a mobile robot, CARL-1, to study the effect of neuromodulators on behavior. Specifically, how neuromodulators affect attention, as well as attraction and aversion from positive and negative stimuli. The robot's "brain" consisted of 6700 neurons with roughly 1.3 million synapses. These neurons were split into different regions that mimic brain areas of mammals, namely a Visuomotor area, a neuromodulatory system, action areas and behavior drivers. Three neuromodulators were simulated, Dopamine, Serotonin, and Acetylcholine. By introducing artificial lesions, one could disable the neuromodulatory systems. For example, lesions that break the Dopamine system caused the robot to stop exhibiting the "wanting" behavior (approaching the green tile).

Other studies with neurorobots tried to mimic the visual perception of animals. In [48] the cameras stimulated an artificial visuomotor area in a lamprey robot. Combined with artificial behavior arbitration networks that mimic the basal ganglia of mammals, the robot had the ability to quickly decide the best behavior based on visual information. In [55] the authors presented an artificial system that mimics the behavior of the Arc Reflex in animals. Their system included both a visual pathway and a somatosensory neural pathway. The inputs are encoded into spiking signals that mimic the action potentials which induce the motor system to generate a movement.

Thus the novelty of neurorobots is that they could bring about new, biologically inspired, solutions to ongoing problems in robotics. However, two issues still face the field of neurorobotics:

1- Most neurorobots so far only simulate limited regions of brain areas
2- The path towards an industrial application for neurorobots remains elusive.

However, with recent developments in insect neurobiology, it seems we are inching closer towards tackling the first issue, full scale brain models. Recently [23], a team from the FlyWire project has mapped out the connectome of the fruit fly brain, identifying the excitatory and inhibitory synapses of all 140,000 neurons of the fruit fly, and a study by Schlegel and colleagues [24] has categorized the 140,000 neurons into 8400 cell types. The work of [25] has built on this to develop a connectome-constrained neural network equivalent to the visual system of the fruit fly. And previously [39] had developed an anatomically constrained model for the path integration mechanism of the Central Complex (CX) of the sweet bee, a structure common across insects that enables them to navigate. Thus, we have the basic building blocks for a computational model for visual navigation in insects, but these two models have not been integrated together in a robot so far in the literature. We aim to address this research gap with this thesis and to take one step closer towards a neurorobot with a more complete simulation of a biological brain, by integrating the visual system model with the central complex model in a mobile robot for the task of autonomous navigation, further serving as an example of how neurorobots can be applied on a task with demand in the industry. To assess the performance of our approach, we will be comparing it to the most popular method of visual navigation in robotics: VSLAM

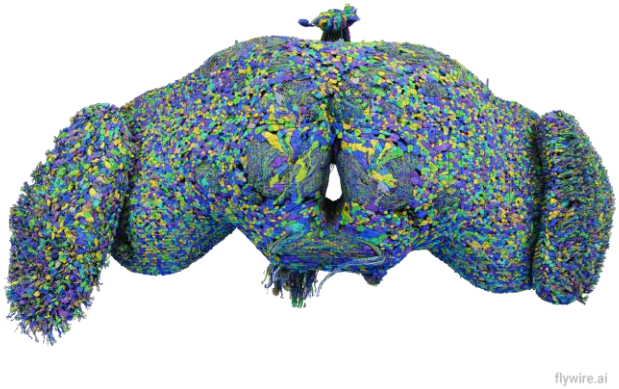

Figure 1.4: The fruit fly brain. Credit: Tyler Sloan for FlyWire, Princeton University, (Dorkenwald et al., 2024) [26]

Thus, the research question we are trying to answer can be formulated as:

How does the optic lobe and central complex navigation method perform compared to Visual SLAM, where both methods are tested in a similar environment?

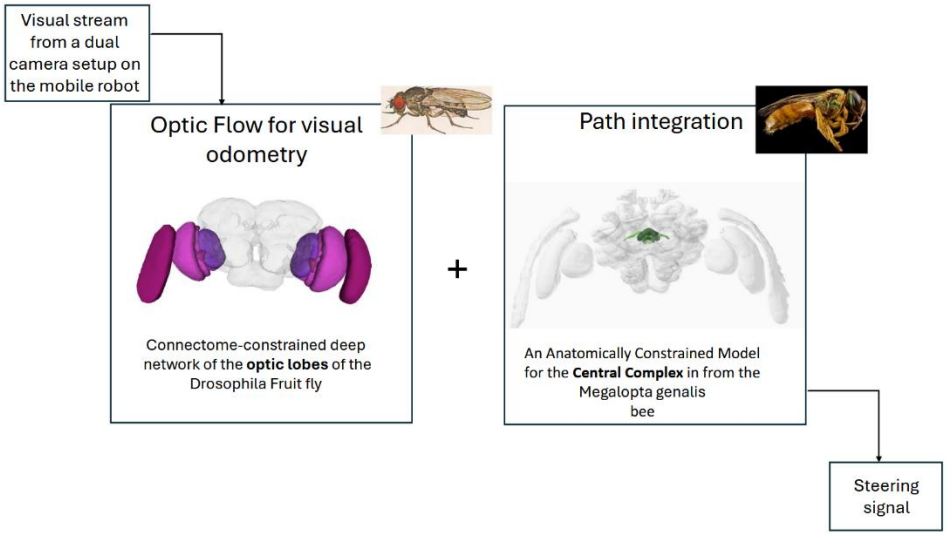

Figure 1.5: Our proposed method: combining the recent connectome constrained optic lobe model from the fruit fly, with the previously developed anatomically constrained model of the central complex of the sweat bee. Graphics from [26] [46] [74] [69].

The structure of this thesis will be as follows:
In the literature review, we will look at V-SLAM and its limitations, as well as deep learning methods and the issues associated with them. Then we will take at the abilities of insect brains and some examples of how they could be of benefit to robotics. We will then look at the fruit fly specifically and the capabilities of its nervous system, followed by a deep dive into the visual system, specifically the motion pathway, and how the artificial model for the optic lobe works. Next, we will take a deep dive into the Central complex of the sweat bee and how its artificial model works.

In the implementation section, we will take a look at an overview of our system, then will observe the behavior of the motion detecting T4 and T5 cells when presented with moving edges, then will build a simulated environment to serve as the first test scenario, then we will take a look at the implementation on the real mobile robot as well as the second test scenario.

In the discussion and conclusion section, we will be comparing our approach to a study that used ORB-SLAM, a method of Visual-only SLAM, as well as discussing the current limitations of our approach and potential areas for improvement for future research directions.

2. Literature Review

2.1 V-SLAM its limitations

The idea of SLAM is for the robot to answer two questions:

1. What does the environment look like?

2. Where is the robot in the environment?

One can use a simple 2D monocular camera to do SLAM, which would be called Visual-only SLAM. To increase robustness, one could add an IMU, which would be called visual-inertial SLAM. One could also use a depth sensor with a 2D camera, which would be called RGB-D SLAM. [7]

However, SLAM algorithms are known for being computationally expensive. This limits real time autonomous navigation to expensive GPU equipped hardware on the robot. Recently, deep learning techniques have been used in SLAM, resulting in an improvement in pose estimation and loop closure. However, this requires multiple data heavy sensors, such as LIDARs and high-resolution video cameras, resulting in large amounts of data, making real-time processing a challenge [8]. Moreover, the demand for memory grows with the size of the environment [9]. With the end of Moore's Law in sight, one can no longer count on improvements in processors to increase efficiency. One work-around would be the use of a multi-agent approach where the heavy lifting is done on a backend server while the agents are equipped with 5G/6G networking devices, but this adds the cost of communication [10].

Alternatively, one could look at end-to-end deep learning methods. Here the approach is to build just one neural network that directly outputs the desired motor signal or desired trajectory [11].

However, end-to-end driving solutions have their own issues:

1. The black-box problem of deep networks means it will be difficult to explain the actions of the end-to-end controller, thus the safety level of the system remains uncertain.
2. Deep neural networks need massive data sets. In the case of visual object identification, this isn't an issue because of the massive amount of data already available on the internet. However, for robotic applications, such data sets are hard to come by since the data sets need to be generated through physical interaction with the real-world environment, which can be expensive and time consuming [56]. Additionally, the inference rates on low-end hardware can make deep networks unsuitable for real time driving.

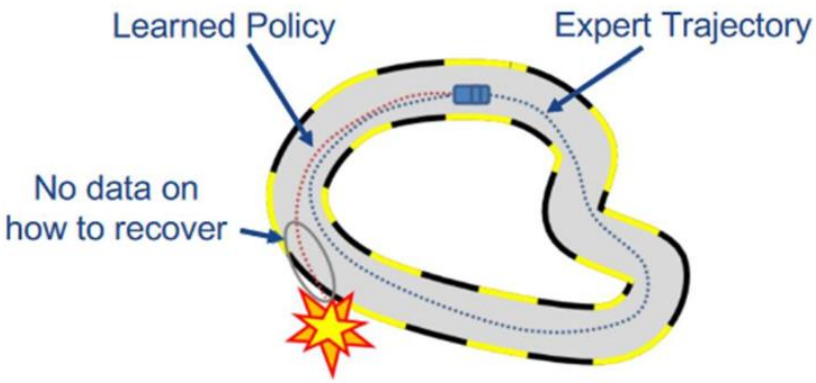

Figure 2.1: Additionally, If the agent's states deviate too much from the learned policy it was trained on and the agent ends up in an unfamiliar environment, it could lead to failures that the agent is unable to recover from [67]

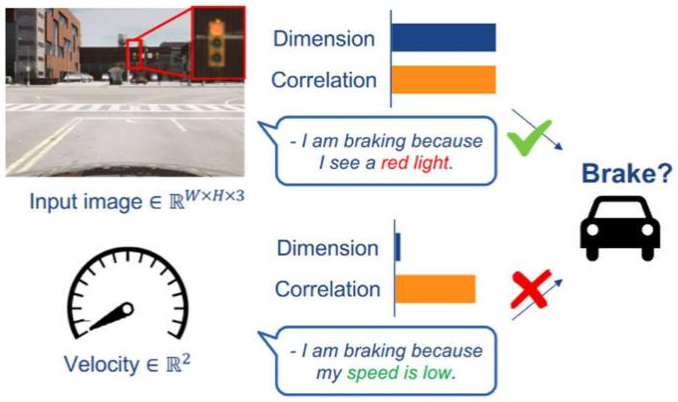

Figure 2.2: End-to-end deep learning methods also suffer from the issue of casual confusion, where the network would correlate an irrelevant dimension to the control output [67]

As an alternative solution to visual navigation, one could turn to insects such as the Cataglyphis desert ant to study how they solve the problem of navigating back to their

nest. These insects are known for their parsimony: i.e. their extreme efficiency with resources. The desert ant is known for going on long meandering paths while foraging, yet it can still return home in a direct route, without the use of chemical trails [13]. Mimicking the navigation of ants was done before in [14], where a mobile robot inspired by ant navigation was able to encode familiar views in the environment with a single ANN layer after a single training run, using only low-resolution vision.

2.2 Why insect brains?

Let's first establish why we picked the brain of the fruit fly. The fruit fly is not the first animal who's connectome has been mapped out, indeed this was done much earlier in 1986 with a far simpler organism : the C. Elegans worm [72], mapping out all of its 302 neurons. However, the worm lacks a highly developed visual and navigational system. On the other end of the complexity spectrum, the rat has a much larger nervous system with 71 million neurons, with a well-studied mechanism for navigation. In the hippocampus of the rat brain are place cells, which fire when the rat is at determined landmarks, utilizing multiple sensory modalities like vision, olfactory sensing and vestibular stimuli [73]. A model for these place cells was even implemented in [74] as a bio inspired SLAM method. Additionally, the rat has a much better visual acuity compared to the fruit fly. However, the full connectome of the rat brain is yet to be

completed, presenting a hurdle our first goal of developing a neurorobot with a complete brain model.

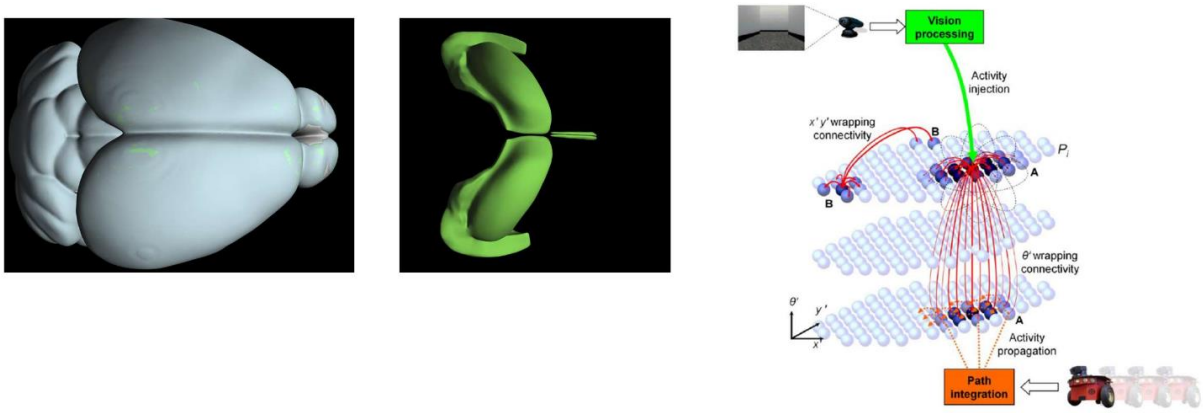

Figure 2.3*: Left to right: The rat brain, the hippocampus, the simulated place cells previously used in RatSLAM.*

The fruit fly thus strikes a balance between the two organisms, bringing in a nervous system with just enough complexity to be applicable in autonomous navigation, while still being on the low end of the complexity spectrum with just 140,000 neurons and

with a recently fully developed connectome.

Figure 2.4: Brains in the animal kingdom, from the simplest, to the more complex animal with larger nervous systems. Starting from the left: C. Elegans, only 302 neurons. Fruit flies, 140 thousand neurons. Ants, 250 thousand. Honeybee, 1 million. Geckos, around 4 million. Frogs, around 16 million. Rats, around 71 million.[71]

Insect brains present an especially ideal candidate to study for robotics due to their ability to perform actions in the short and long term (from the extremely fast escape reflex of the fruit fly to long distance migration), all retained in an exceedingly small package. For example, honeybees exhibit a variety of complex behaviors, such communicating and coordinating with other bees in the hive with waggle dances [68], yet they only have one million neurons. The Megaphragma mymaripenne wasp has an even smaller brain with less than 10,000 neurons [27]. In comparison, the human brain has 86 billion neurons, which presents a massive challenge to fully modeling it

Figure 2.5*: The common fruit fly. Interesting to note that, like the Breitenberg vehicle, the fruit fly also displays similar a behavior as the "aggression" mode towards the light source, called the phototactic response, which causes the animal to turn towards a source of light.*

Insects also have much shorter life cycles, the Drosophila only has a life cycle of around 40 days [69], vastly shorter than most mammals, which greatly lowers the barrier to studying them. Additionally, many of the brain structures are conserved across species of insects, especially flies, where different species such as the hover fly, blow fly, and tsetse fly, all share similar brain areas [28], which further simplifies the process of studying them.

Figure 2.6: Cataglyphis nigra, also known as the black desert ant. [57]

Multiple techniques for saving computational resources are seen in insects, such as neural reuse, where the same circuit is used for different tasks [15]. Despite their relative simplicity, insects can still perform complex tasks such as counting and learning different concepts such as sameness and difference. The honeybee, Apis mellifera, even demonstrated learning by Pavlovian conditioning [13], a learning behaviour first observed in dogs. It can also distinguish between a new stimulus and a familiar one. Additionally, brains in insects are sparsely connected, which also reduces the energy requirements [2].

Sensory-motor coordination is another technique used in insects to avoid cognitive load. For example, ants have been observed to rotate on the spot till they find a viewing direction with a familiar view, instead of using more cognitive resources to try to find the familiar view mentally. Another instance is active vision, which entails physically moving the eye to simplify visual processing. For example, insects use their neck muscles to tilt the head, such that it stays at a constant orientation during flight,

instead of mentally transforming the image to stabilize it. So, this allows the compound eyes to physically filter out some of the rotational flow in the image [61].

Another method for reducing the required cognitive load is morphological computation [2]. This can be seen in stabilizing mechanisms in the fruit fly during flight, or in bipedal walking in humans. Essentially, offloading the computation so it's performed by physical properties of the body.

There are 3 major structures in the insect brain involved in navigation in insects:

1- The Optic lobes, providing movement perception

2- The Central Complex, performing path integration

3- The Mushroom bodies, for storing and retrieving memories and learning landmarks in the environment

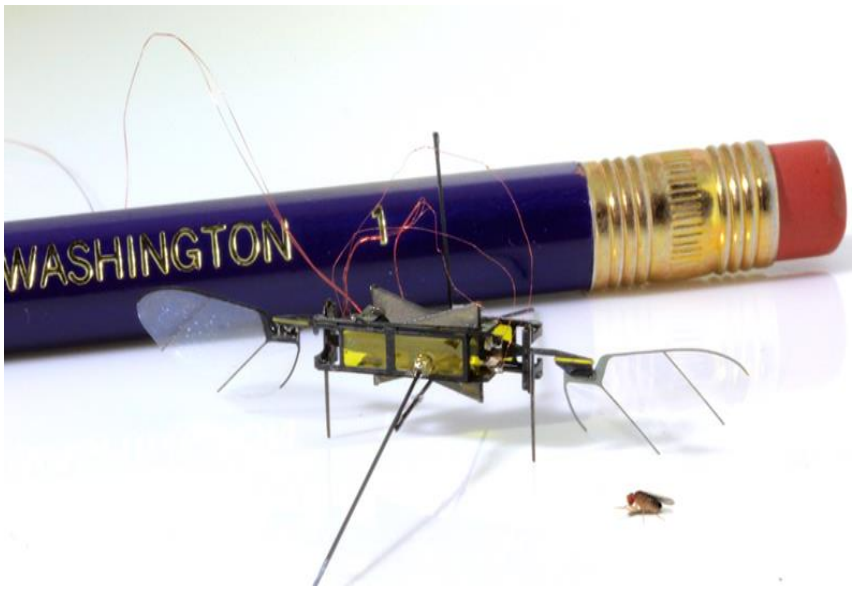

Figure 2.7: In a recent paper, the authors used optic flow in a tiny flying robot to maintain the flight path in a corridor. Rather than using state estimation derived from a stored map (SLAM). They rely on optic flow, a measure of the velocity of motion of visual scenery as the robot or animal moves through it. Optic flow requires less computation and provides a crude measure of distances to obstacles [18].

Insects like bees rely on optic flow to navigate the environment, which is the motion seen across the visual field. Bees were found to compare the optic flow from the left and right visual fields to avoid hitting the walls when flying in a narrow tunnel [2]. More recently this technique of translational optic flow was used in a mobile robot to maintain a safe path in cluttered environments, achieving 97% collision free travel [19].

Path integration has been used successfully in [16], allowing a mobile robot to travel outdoor paths of hundreds of meters while only requiring 3MB/Km. In contrast, a SLAM implementation would require hundreds of Megabytes. Interesting to note, path integration is multimodal; insects can still return home even in the absence of light [2]. In addition to path integration, insects rely on spatial navigation maps that contain memories of actions performed and how various landmarks in the environment are related to each other spatially and temporally [17].

Recently, two simulated mushroom bodies were used in a mobile robot relying only on low resolution visual input and allowing efficient heading correction based on visual familiarity. This technique does not rely on feature detection or deep learning, each Mushroom body is modelled as a shallow neural network of only two layers. [21] [13]

2.3 The Fruit fly brain: connectome, reflexes and abilities

An aspect of the insect nervous system that could be of immense use for robots acting on time critical tasks is the ability to handle short term, fast reflexes with simple pathways. The LPTCs in the fruit fly provide the quick reflexes needed for in-flight stabilization [30]. Another reflex that has been well documented in the fruit fly is the jumping response, triggered by looming stimuli (from an approaching predator). One of the neurons that trigger this is called the Foma-1 neuron, which sends its signal to the giant fiber escape pathway. Interestingly, when this neuron was externally triggered in experimentation, even in a blind fly, the fly would still jump [31]. In addition to quick reflexes, they are also capable of long duration tasks like foraging and migration, which is a more complex behaviour, combining visual homing mechanism (using the Mushroom Bodies), with a path integration mechanism (using the Central Complex) when the insect is in an unfamiliar area [32].

The fruit fly is very well studied in terms of its behaviour and physiology, due to the ease with which one can introduce mutations to specific genes [31]. However, their visual acuity is much worse than mammals [28], and experiments suggest that they do not possess object recognition. Still, they do possess sensory modalities that humans don't have, like detecting the polarization of sky light, which they utilize for navigation [31]. Despite its relative simplicity, the visual system of flies is in some areas on par with that of more advanced animals: They were shown to be able to perceive second order motion, an ability that was previously thought to be only possible in animals with

advanced cortex. They also, like humans, are tricked by the reverse-phi visual illusion of motion. Additionally, fruit flies exhibit parallax motion perception: when crossing a gap in the floor while walking, the fly was able to estimate if the gap is too large to cross and thus avoided attempts at crossing [36].

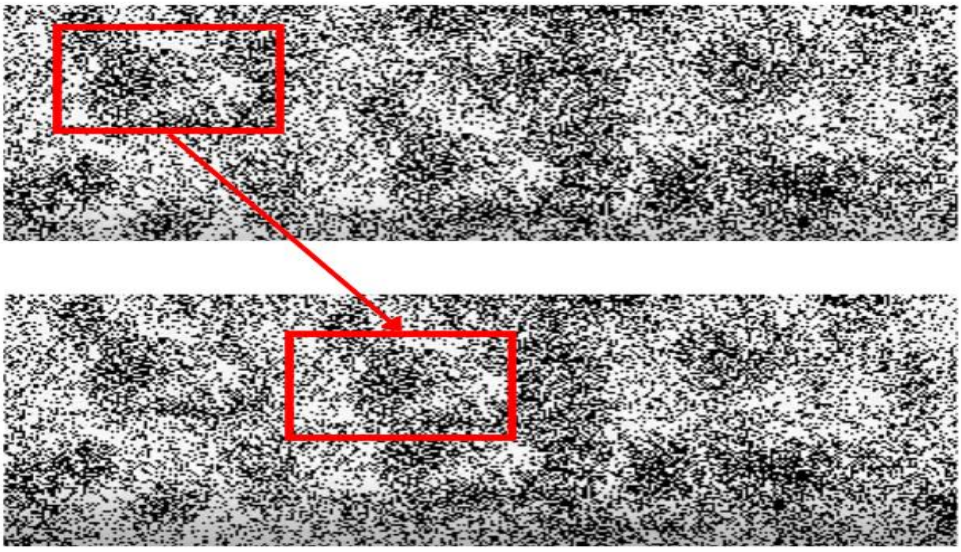

Figure 2.8*: Example of second order motion perception. The patch marked with the red box (upper frame) has moved to the right (lower frame). The patch is only visibly distinct while the patch is moving, since the patch has the same random texture as the background. It is a motion only defined by change in texture without change in luminance. [62]*

Additionally, visual perception in fruit flies does not rely on the visual input alone, they also make use of self-generated motion to aid perception. They exhibit motor efference [36], an ability also found in the extrastriate cortex of non-human primates, combining information from vestibular motion and eye movement to be fed back into the visual system. This is necessary so that the organism can tell the difference between visual stimuli caused by something moving in the environment, and a visual stimulus caused by the organism turning its own head or body. The HS cells in the LPTCs, which are involved in processing optic flow, were found to be actively inhibited during flight. In voluntary walking, the HS cells were found to exhibit an inhibited response, compared

to if the fly was still and environment was moving around it instead. Thus, suggesting that non-visual information is used in conjunction with visual stimuli for visual perception [2] [30].

2.4 The Optic Lobes of the fruit fly

As in other insects, the eyes of the fruit fly are compound, consisting of 750 facets called Ommatidia. Each ommatidia contains 8 types of photoreceptor cell. Following a light signal as it hits the eye, we pass by the 5 major neuropils in the visual system, before reaching the central brain, where the CX will integrate the homing vector. These 5 major areas, in order, are the retina, the lamina, the medulla, the lobula and lobula plate.

Figure 2.9: Optic lobe of the fruit fly, showing all the major neuropils [35]

The retinal cells scale the incoming light signal logarithmically to match the dynamic range of the neurons downstream. There are 8 types of retinal cells, R1 to R8. The last two, R7 and R8, can evaluate the spectral characteristics of light to perceive color vision (from UV to green), while R1 to R6 are involved in the motion pathway [36] [28] [31].

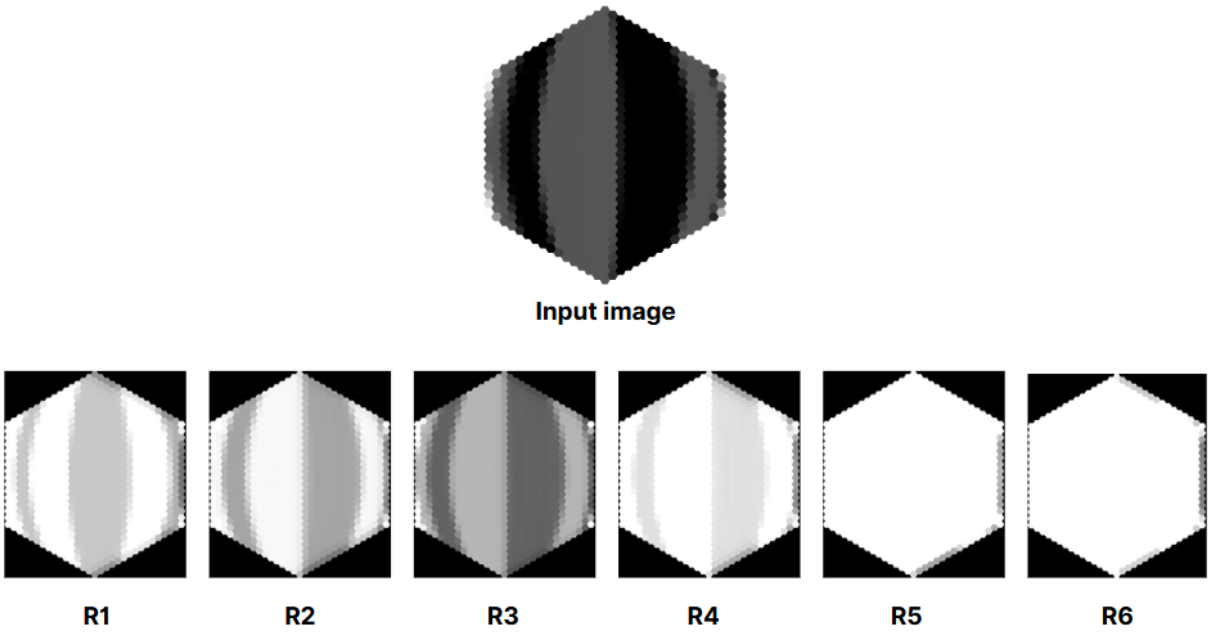

Figure 2.10: Response from the artificial optic lobe model retinal cells R1 to R6 when presented with the input image above

From the retina to the lobulas, the neurons are arranged in separate columns behind each facet, and the signal remains in a topographic form till it leaves the lobula plate (i.e. the spatial relationships are preserved). The signal from one photo receptor is not enough to detect motion, so there needs to be some mechanism for comparing the signals from neighboring photoreceptors as well. Indeed, there is some cross communication between these visual columns, similar to the elementary motion detector model described by Hassenstein and Reichardt [36].

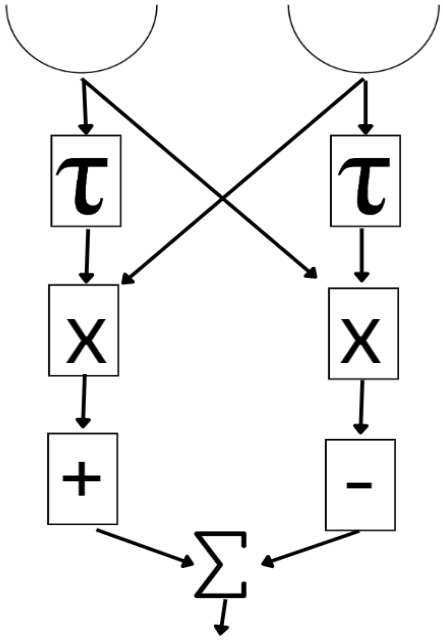

Figure 2.11: This mechanism of detecting motion was proposed by Hassenstein and Reichardt after analyzing the motion of the Cholorophanus viridis beetle, tending to turn toward the direction of motion to maintain visual stability [31]. The detector functions by comparing the temporal difference between the two visual units. The two semi circles represent two simple photo receptors, and τ is a time delay. The delayed signal is multiplied with the signal from the other photo receptor, then both signals are subtracted from each other. The sign of the resulting summation would tell us if the motion was perceived from right to left or vice versa [37].

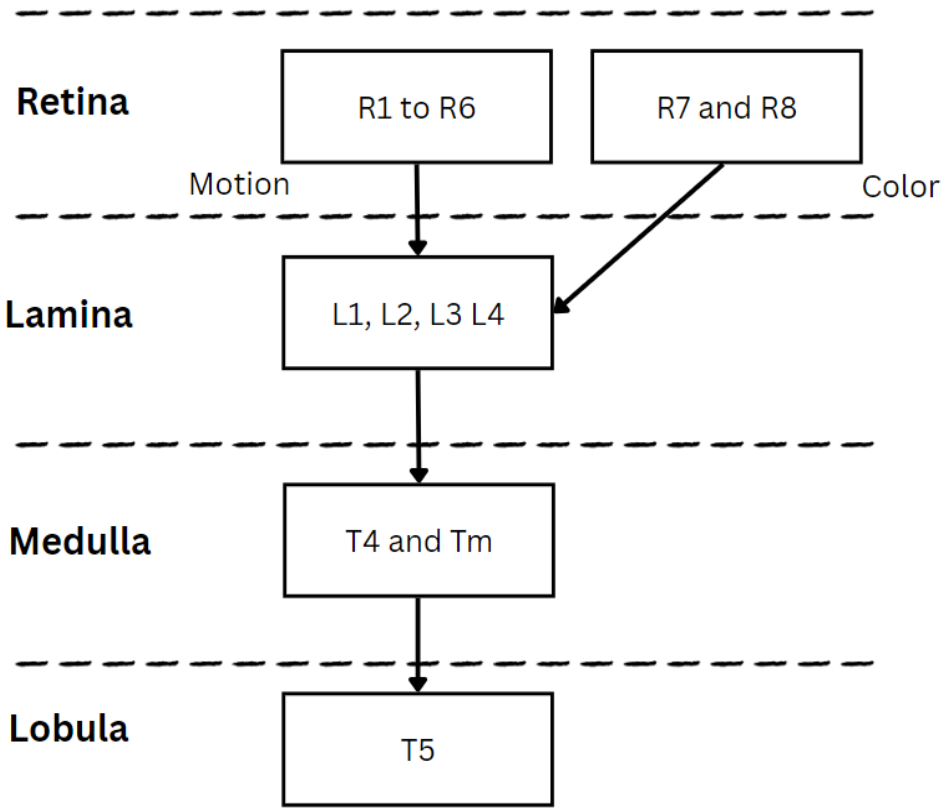

Figure 2.12: Neurons of interest in the visual columns across major visual neuropils. The R1 to R6 retinal cells are mainly involved in motion perception, and this pathway is separated from the colour (R7 and R8) pathway from the start.

We can observe motion perception in action in a common reflex seen in all insects and the fruit fly where they tend to follow objects that move in their field of vision, this is called the optomotor response [40]. The L1 and L2 neurons in the visual system play an important role in this reflex. Indeed, blocking the Lamina neurons produced flies that do not respond to motion stimuli. However, these flies were not fully blind either, just motion blind. They still exhibited the phototactic response, a response seen across species of insect where they tend to follow a moving source of light, but no optomotor response [36].

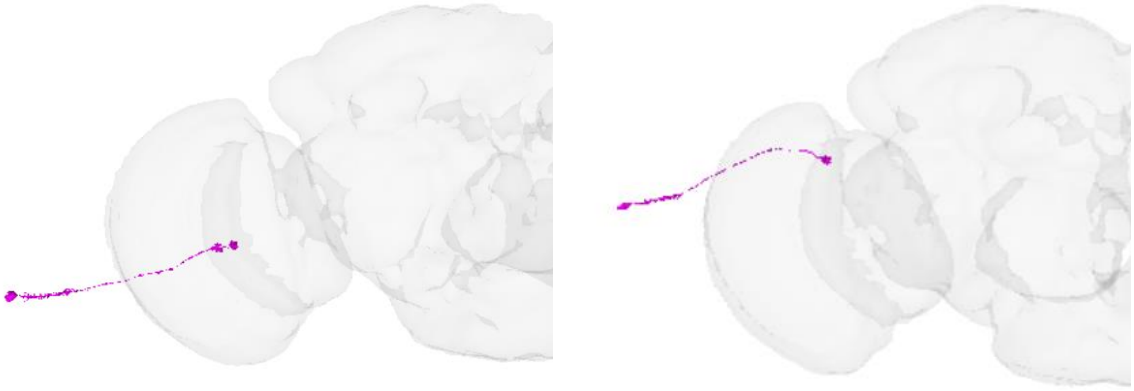

Figure 2.13: The L1 and L2 neurons respectively from the left hemisphere [35]

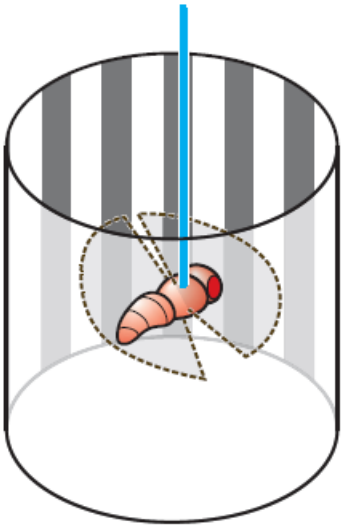

Figure 2.14: The optomotor response: in experiments where the fruit fly is placed in a rotating drum with a striped pattern, the fly tends to follow the movement of the cylinder [38]

The L1 neurons in the lamina were found to be selective for dark-to-bright transitions (ON pathway), specifically responding to increments in brightness. While L2 neurons favour bright-to-dark transitions (OFF pathway), responding to decrements of brightness. Indeed, these two pathways behave as two Reichardt-Hassenstein detectors. The L1 and L2 neurons eventually lead to our main point of interest namely the bushy T cells (T4 and T5 cells respectively), which have four subtypes (a,b,c, and d)

that respond selectively to motion in the four cardinal directions (up, down, left and right). Like the L1 cells, T4 cells favour dark-to-bright edges (ON Pathway), while T5 cells favour bright-to-dark edges (OFF pathway). Once the signal reaches the lobual plate, it goes to the central brain via the Visual Projection Neurons (VPNs) [36] [37].

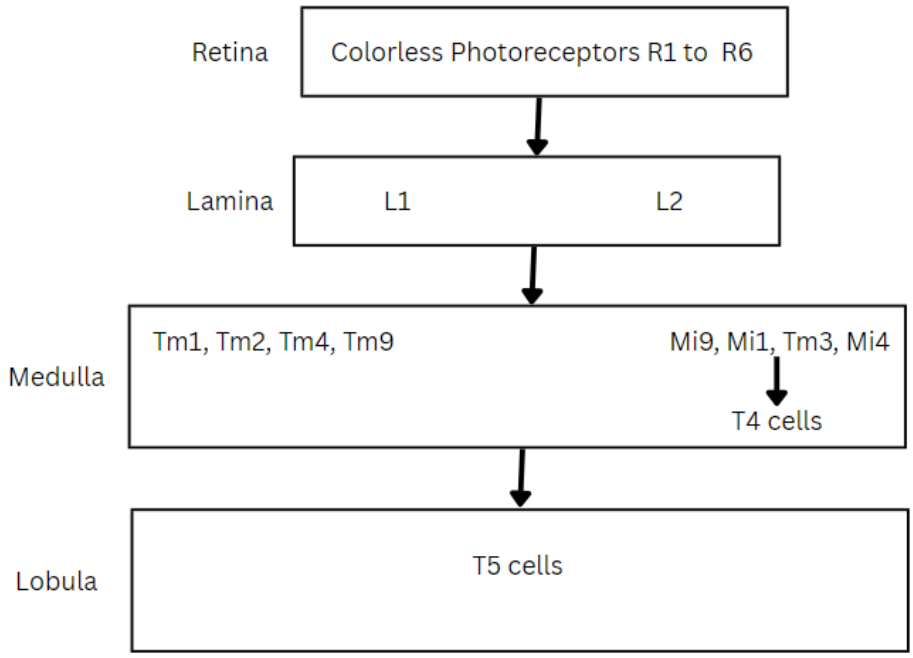

Figure 2.15: Showing the locations of T5, T4, and Tm cells in motion pathway of the visual system

But what differentiates the subtypes of T4 and T5 cells, causing them to respond differently to different directions of motion? It all depends on how delay neurons are connected to which part of the dendrite.

T4 cells get their input from the following neurons:

Mi1 , Tm3 , Mi4, Mi9, and CT1

While T5 cells get their input from:

Tm1, Tm2, Tm4, Tm9, and CT1

Similar to the delay element in the Reichardt-Hassenstein model, the CT1 and Mi9 act as delay elements (slow response neurons). While Mi1 and Tm3 act as fast response neurons. Where these neurons connect to the dendrites of the T4 or T5 cell sets which stimuli direction will have an amplified response, and which stimuli direction will have a null response, i.e. it sets the direction selectivity for the T4 or T5 cell. Thus, the difference between the subtypes of T4 and T5 cells lies in how their dendrites are connected: For example, T4c (responding to upward motion) receives input from Mi1, Tm3 (fast neurons) in the central part of the dendrite, while Mi9 and CT1 (slow neurons) on the ventral part of the dendrite. Similarly, T4d (responding to downward motion), has the inputs from the Mi9 and CT1 (delay) neurons on the dorsal side of the dendrite, while the (fast) Mi1 and Tm3 on the central part of the dendrite. [37] [28]

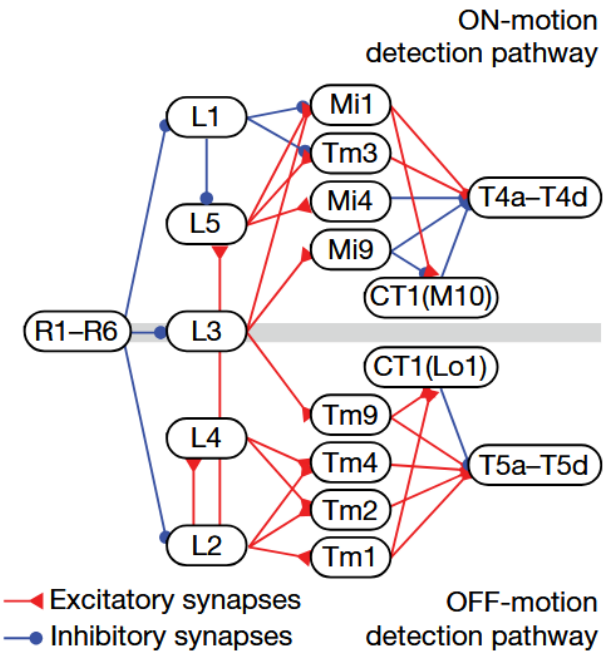

Figure 2.16: The motion pathway, starting from the color blind retinal cells to the L cells in the Lamina, to the Mi and Tm cells in the medulla, to the T4 and T5 cells in the Lobula [25]

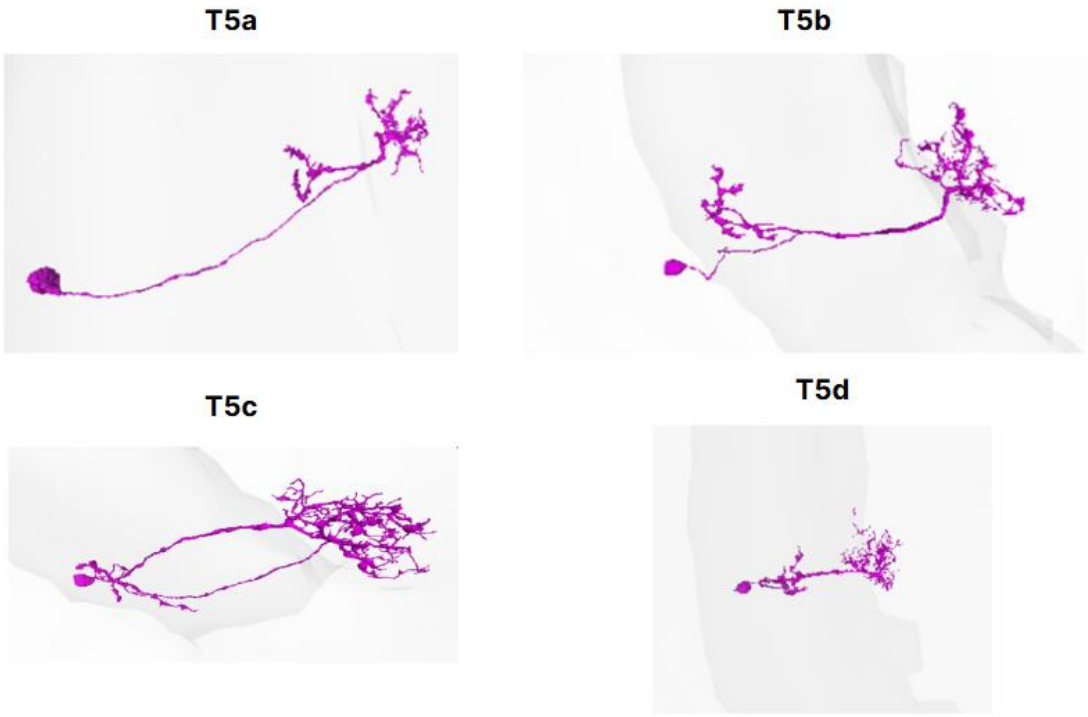

Figure 2.17: The four types of the T5 neuron from the left hemisphere [35]

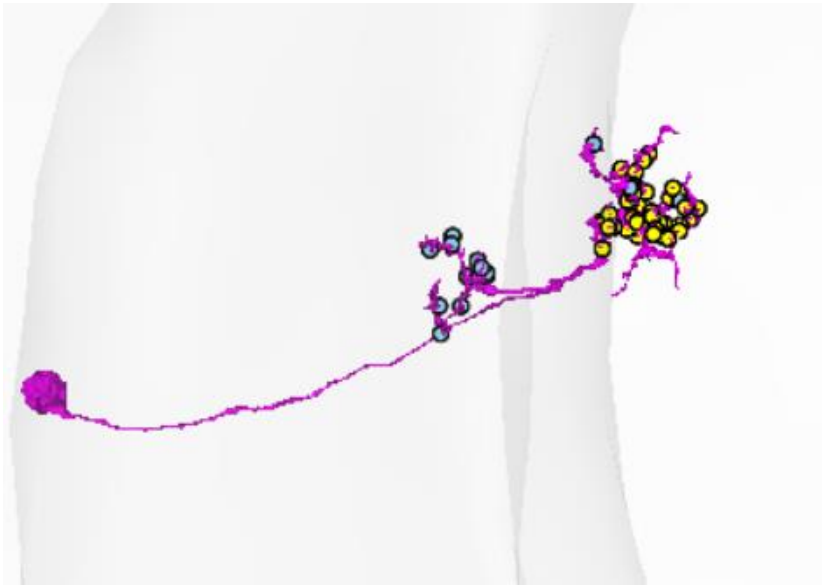

Figure 2.18: The T5a neuron with the input (yellow dots) and output (blue dots) synapses overlayed [35]

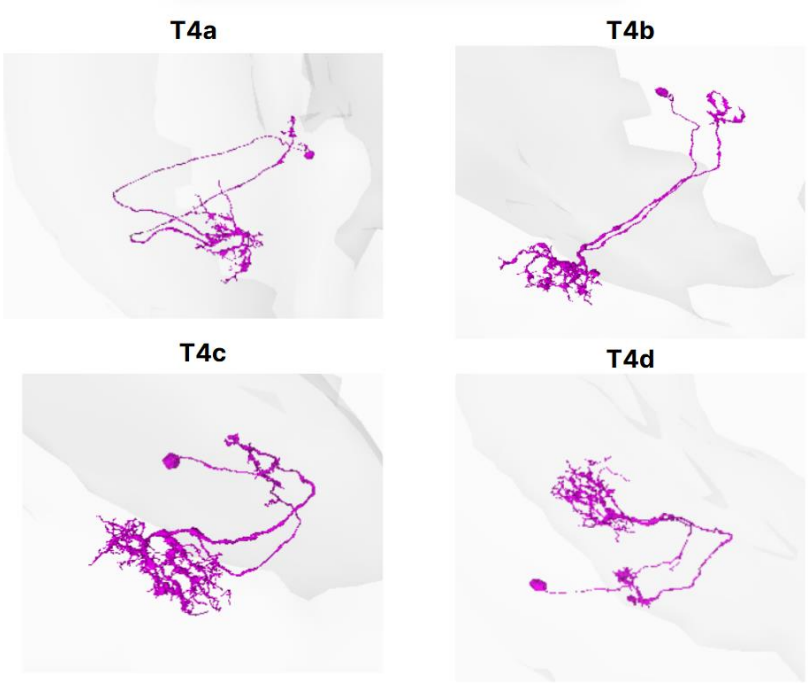

Figure 2.19: The four types of the T4 neuron from the left hemisphere [35]

2.5 The connectome constrained optic lobe model

So far, we have a connectome diagram that tells us how every neuron in the visual system is connected to its neighboring neurons. However, on its own, the connectome is not sufficient to simulate the response of the neurons in the visual system, one still needs to describe the dynamics of each neuron. The authors of [25] have recently set out to tackle this and developed a recurrent deep network of the optic lobe modeling all 65 visual cell types in the optic lobe and matching connections from the connectome, where each neuron was modeled with a point neuron model:

$$\tau_{t_i}\dot{V}_i = -V_i + \sum_j s_{ij} + V_{t_i}^{\text{rest}} + e_i$$

Where:

τ_{t_i} describes the time constant for cell number i of type t, which tells us how fast the neuron would respond

V_i is the post synaptic voltage of that cell

$V_{t_i}^{\text{rest}}$ is it's resting voltage potential

e_i is the external input current (this will only be given to the retinal cells, R1 to R8)

$\sum_j s_{ij}$ is the total input voltage coming into neuron i neurons j above it in the visual column, modulated by the synaptic strength w_{ij}

$$s_{ij} = w_{ij}\,(V_j)$$

Where the strength of each connection is a product of how many discrete synapses there are between two neuron types, $N_{t_i t_j}$, times the strength of an individual synapse between them, $\alpha_{t_i t_j}$:

$$w_{ij} = \alpha_{t_i t_j} N_{t_i t_j}$$

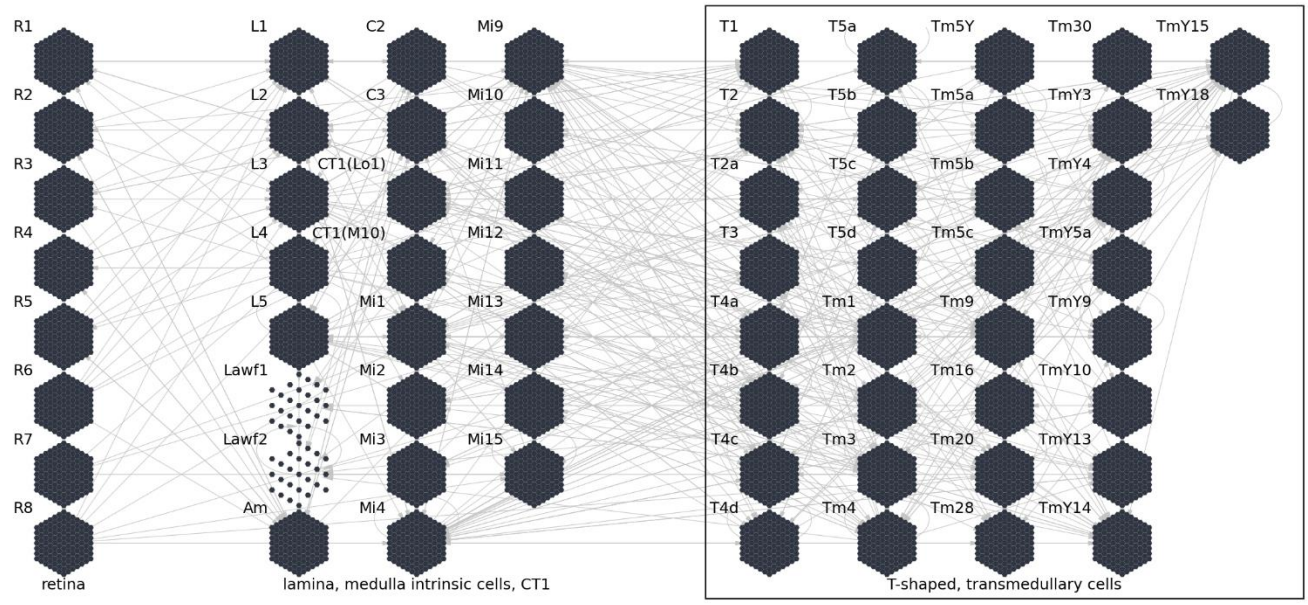

Figure 2.20: The connectome constrained recurrent network of the visual system developed by [25]. The specific models of each neuron are simplified, each neuron is modeled as non-spiking with a single electrical compartment (most neurons in the optic lobe in the fly are non-spiking).

So far, the connectome only tells us the discrete number of synapses $N_{t_i t_j}$, but not the individual strength of each synapse.

So, in total there are three unknowns that need to be optimized for each neuron type:

1- The resting potential of the cell $V_{t_i}^{\text{rest}}$

2- The time constant of the cell τ_{t_i}

3- The strength of the synapse between each pair $\alpha_{t_i t_j}$

Using the connectome the authors narrowed down the number of synapse types $\alpha_{t_i t_j}$ to 604. Thus, since there are only 65 neuron types, there remain only 734 free parameters to optimize for: 65 resting potentials, 65 time-constants, and 604 synapse strength factors. Without the connectome, the number of unknown parameters would have been around two million.

Since it is known that the function of the visual system is largely motion perception, the authors trained their network in an end-to-end fashion to predict optic flow, with a simple two-layer artificial network attached at the end to of the optic lobe network to decode its output. After training on the Sintel dataset [70], the neurons of the artificial optic lobe were found to have matching response with the real optic lobe neurons acquired from experimental recordings.

2.6 The Central Complex

The Central Complex (CX) is 400-million-year-old structure that is highly conserved across different species of insects and has been recently identified as the potential seat of consciousness in insects [13][43]. Containing largely the same substructures across multiple insect species such as the Fan Shaped Body (FB), the Proto Cerebral bridge (PCB), and noduli neurons, the CX is mainly involved in path integration and connecting landmarks in the environment. When going out to forage for food, the CX integrates the vectors of motion, to keep track of a homing vector. The CX is also multi modal, integrating information from the optic flow from the eyes, mechanosensory inputs from the antennae and wings, as well light polarization and geomagnetic field in some species.

Depending on the species, the sensory input used by the CX is also different (relying on step count in ants, while relying on optic flow in bees) [33]. In one study it was found that directly stimulating regions in the CX with electrodes in the Balaberus discoidalis cockroach resulted in it turning its direction in 16 out of 27 subjects. Additionally, the CX plays a role in the optomotor response, shown by deficits in the response when the CX is disabled with an anesthetic. It also plays a role in visual learning in Drosophila [34].

Figure 2.21: The Central Complex of the fruit fly brain, highlighting the major neuropils within such as the Noduli, Protocerebal bridge, Ellipsoid body, and Fan-shaped Body [35]

In [39] the authors presented a model for the Central Complex (CX) from the Sweat bee (Megalopta genalis). The reason the sweat bee was chosen was due their relatively simple path integration mechanism compared to other insects: they only rely on polarized light for as a heading signal, meanwhile other insects like fruit flies also integrate their self-generated motion signals for estimating the heading. An advantage of including self-generated motion can be seen in that it allows the cockroach to update its heading direction even the absence of vision [41]. In cockroaches, it has been noted that the activity in the central complex is a reliable predictor for the intended movement of the roach [44]. The use of this path integration mechanism to return home has been noted in ants when far away from the nest, only switching to odour-based guidance once near the nest [45]. Thus, CX models from other species could offer additional capabilities like multi-modal path integration, but for the sake of simplifying implementation, we will use the sweat bee cx model in this project.

Figure 2.22: The Megalopta genalis bee [74]

There are two main inputs to the CX are the optic flow from the optic lobe (in our case, this will come from the artificial optic lobe model developed in [25]), and a reference compass signal. The bee relies on the polarization of sunlight as a signal for the reference compass, where the TL (tangential) cells in the CX respond to their preferred orientation. In our case with the mobile robot, we will use a 3-axis magnetometer for the reference compass signal.

2.7 The anatomically constrained CX model

The following neurons in the CX are our main points of interest. For each neuron we first present the behaviour of the real neuron according to [46], followed by the anatomically constrained model from [39]:

1- The TL Neurons, belonging to the tangential CX neuron class, representing the current heading. This signal is later flipped by the CL1 neurons before going to the TB1 ring neurons.

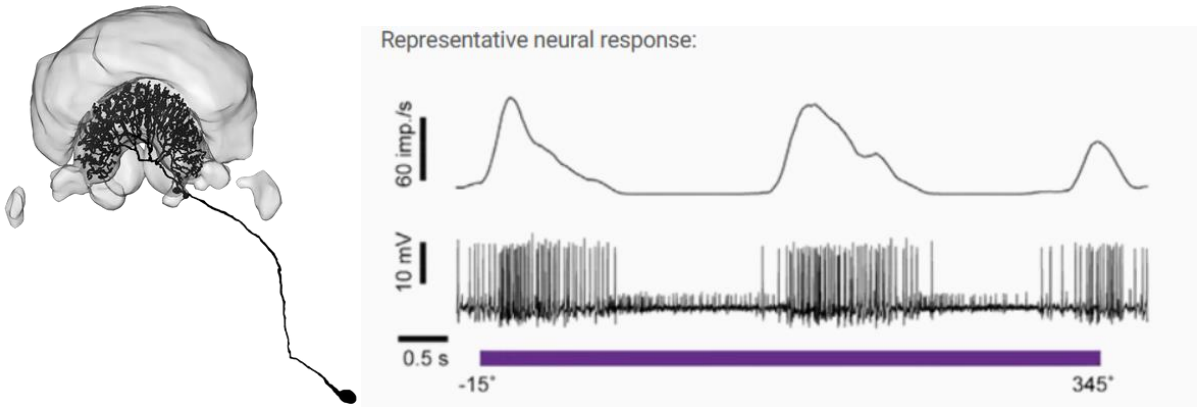

Figure 2.23: The TL (left side) cell in the central complex of the sweat bee (Megalopta Genalis), and its response showing a strong preference in response when the presented stimulus of polarized light is at their favorite angle. [46]

The model we will be using from [39] has 16 TL neurons, representing 8 possible heading directions (two neurons per direction) around the insect. Each of the 8 pairs has a preferred heading:

$$\theta_{\mathrm{TL}} \in \{0, \pi/4, \pi/2, 3\pi/4, \pi, 5\pi/4, 3\pi/2, 7\pi/4\}$$

Thus, the pair with the preferred angle that most closely matches the current heading θ_{h} should have the strongest signal:

$$I_{\mathrm{TL}} = \cos\left(\theta_{\mathrm{TL}} - \theta_{\mathrm{h}}\right)$$

and CL1 neurons merely flip the signal from the TL neurons before proceeding to the TB1 neurons:

$$I_{\mathrm{CL1}} = -I_{\mathrm{TL}}$$

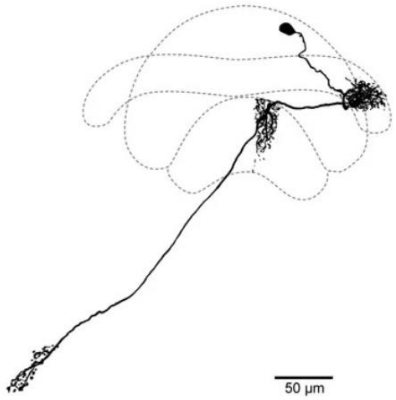

Figure 2.24: CL1 neuron in the CX. It only outputs the negative of the TL neurons and is only included in the model for completeness [46].

2- The TN1 neurons, also belonging to the tangential CX neuron class, which respond with excitation to backward motion. These neurons, together with the TN2 neurons, are noduli neurons, which are the main points of entry into the CX from the optic flow estimation from the visual lobes. [33]

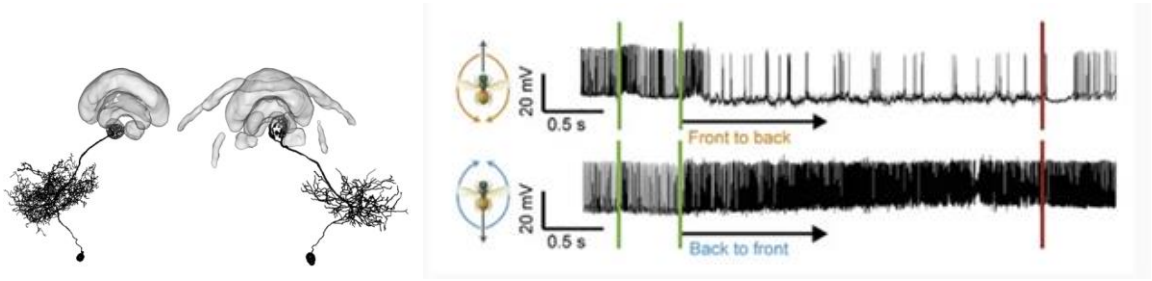

Figure 2.25: The left and right TN1 neurons of the sweat bee. These neurons respond strongly to translational optic flow, while showing no response to rotational optic flow. Forward motion causes them to respond with inhibition while backward motion causes excitation. [46]

3- The TN2 neurons, belonging to the tangential CX neuron class, which respond with linear excitation to increasing forward velocity.

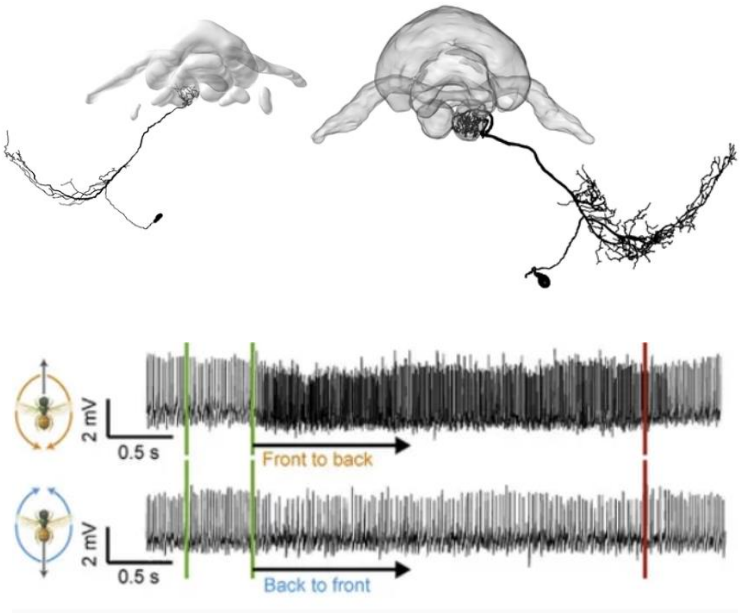

Figure 2.26: The left and right TN2 neurons of the sweat bee. Forward motion causes increased excitation; however, they are not inhibited due to backward motion, unlike TN1. Their excitation response to increased forward speed is almost linear with the stimulus, indicating that TN2 encode forward flight speed. Similarly to TN1, they do not respond to rotation. [46]

The response for TN1 and TN2 cells is then modeled as:

$$\mathrm{Tn}_1 = \min\left(\max\left(\frac{(1 - \text{ flow })}{2}, 0\right), 1\right)$$

$$Tn_2 = \min(\max(\text{ flow }, 0), 1)$$

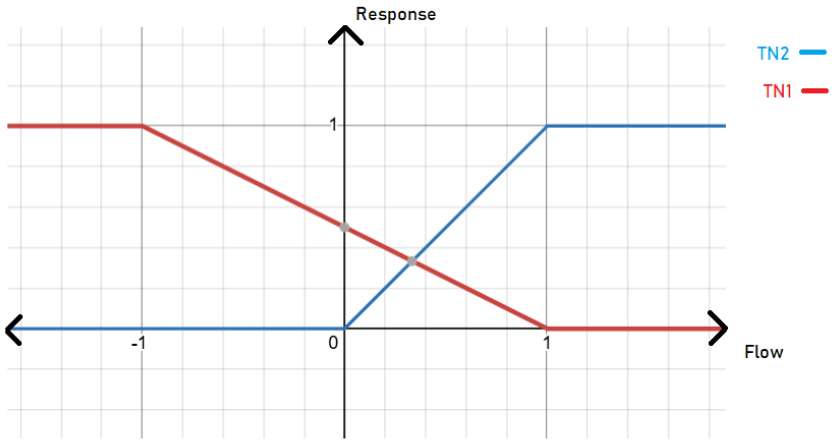

Figure 2.27: Visualizing the response of TN1 and TN2 against flow

4- The TB1 neurons (which act as a ring attractor to stabilize the signal from the CL1 neurons). TB1 neurons are in the PCB, and the activity of these 8 neurons represents the current heading direction.

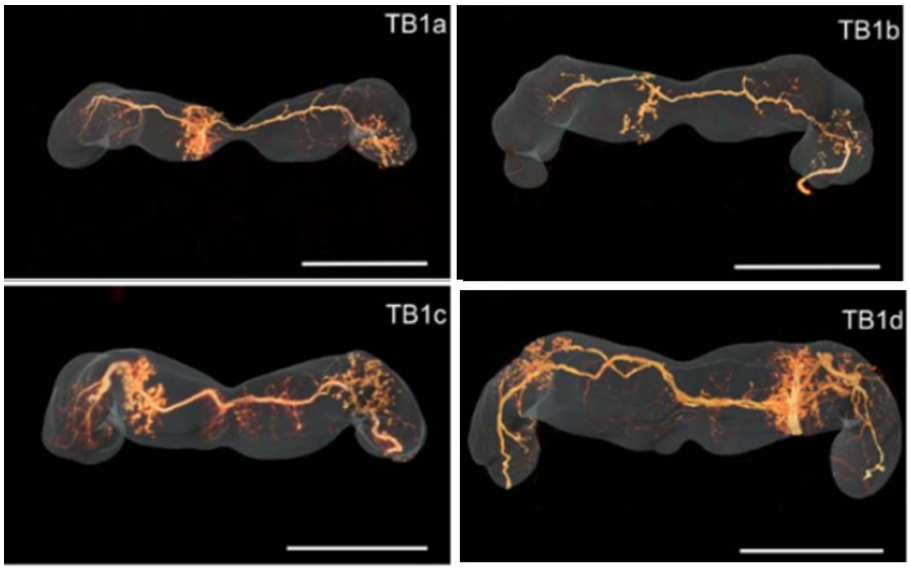

Figure 2.28: The TB1 (compass)neurons, their main input is from the CL1 neurons. They have inhibitory self-connections and act as a ring attractor, arriving at a stable heading encoding over time. Essentially filtering out the noise coming from the CL1 neurons. [46]

The update function for TB1 neurons from the [39] model is thus represented as:

$$TB1 = \alpha_{CL1} \times (CL1 \cdot W_{CL1 \rightarrow TB1}) - \alpha_{TB1} \times (TB1 \cdot W_{TB1 \rightarrow TB1})$$

where both proportions α_{CL1} and α_{TB1} add up to 1. The positive section of the equation represents the new heading from the CL1 neurons, and the negative section represents the inhibitory self-connections of the TB1 neurons. The weight of the CL1 to TB1 connections are:

$$\mathrm{W_{CL1->TB1}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

And the weights of the self-connections are:

$$\mathrm{W_{TB1\,->\,TB1}} = \begin{bmatrix} 0 & 0.14 & 0.5 & 0.85 & 1. & 0.85 & 0.5 & 0.14 \\ 0.14 & 0 & 0.14 & 0.5 & 0.85 & 1. & 0.85 & 0.5 \\ 0.5 & 0.14 & 0 & 0.14 & 0.5 & 0.85 & 1. & 0.85 \\ 0.85 & 0.5 & 0.14 & 0. & 0.14 & 0.5 & 0.85 & 1. \\ 1. & 0.85 & 0.5 & 0.14 & 0. & 0.14 & 0.5 & 0.85 \\ 0.85 & 1. & 0.85 & 0.5 & 0.14 & 0. & 0.14 & 0.5 \\ 0.5 & 0.85 & 1. & 0.85 & 0.5 & 0.14 & 0. & 0.14 \\ 0.14 & 0.5 & 0.85 & 1. & 0.85 & 0.5 & 0.14 & 0 \end{bmatrix}$$

Thus, each neuron most strongly inhibits the neurons furthest it (in terms of preferred heading).

5- The CPU4 neurons, belonging to the columnar CX neuron class in the Fan Shaped Body (FB), which receive their input from the TB1 neurons and the TN2 neurons. These are the neurons responsible for motion vector integration, in the form of optic flow, in proportion to the current heading (from the TB1), acting as a set of directionally locked odometers. The CPU4 neurons are the integrator

cells, which eventually point to the desired (home) vector. They are inhibited by the TB1 neurons. Thus, the least inhibited cells in the CPU4 population would accumulate the optic flow information [33]

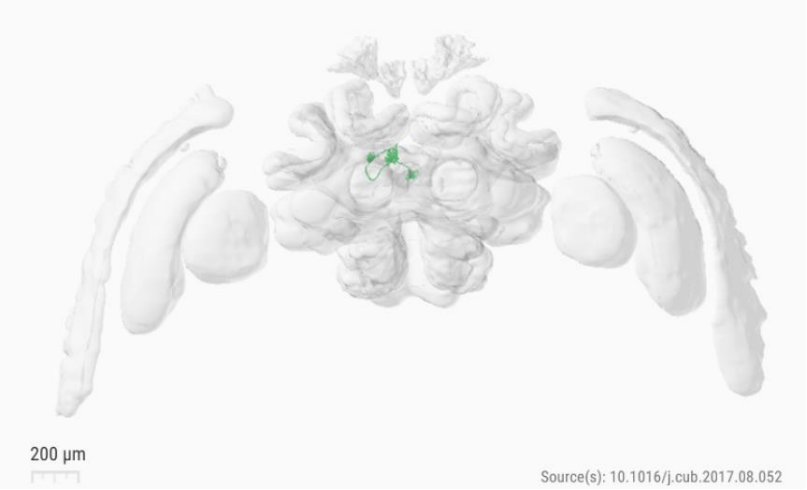

Figure 2.29: The CPU4 neurons in an adult female bee brain [46]

The update function for the CPU4 neurons is modeled as:

$$CPU4_{(t)} = CPU4_{(t-1)} + (u - d)$$

Where is the update component pointing to the opposite of the current heading, modulated by the response of tn1:

$$u = (0.5 - tn_1) \cdot (1 - tb_1)$$

And d is the decay component:

$$d = k \cdot (0.5 - tn_1)$$

Where K is the decay gain.

Intuitively, since TB1 is pointing to the current heading (assuming we are moving forward), then the opposite direction is pointing to the home, so we want to accumulate the opposite of TB1 onto CPU4. Depending on how fast we were

moving away from the nest, the larger the accumulation on CPU4 will be, hence the modulation with tn1.

If the insect was moving backwards, then tn1 would be less than 0.5, thus CPU4 would grow in the same direction as TB1.

6- CPU1 neurons, also belonging to the columnar CX neurons, are the final stop where we will get the steering signal. These neurons compare the current heading from the TB1 neurons with the desired heading from the CPU4 neurons to generate the steering signal. The difference in the activities of the TB1 and CPU4 neurons will cause an uneven activation between the left and right side of the CPU1 neurons, triggering steering [33].

There are two subtypes of CPU1 cells, CPU1a, of which there are 7 cells per hemisphere, and CPU1b, of which are only two cells, one per hemisphere. The main steering correction comes from the CPU1a cells, CPU1b are only needed for the wrapping case (around a heading of 0 degrees) [58]. The update function for the CPU1 neurons on both brain hemispheres will be defined as:

$$CPU1a = (DesiredHeading) \cdot (W_{DH\rightarrow \mathrm{CPU1a}}) \times (1 - CurrentHeading) \cdot (W_{CurrentHeading->CPU1a})$$

$$CPU1b = (DesiredHeading) \cdot (W_{DH\rightarrow \mathrm{CPU1b}}) \times (1 - CurrentHeading) \cdot (W_{CurrentHeading->CPU1b})$$

Where the current heading (TB1) connections to the CPU1a and CPU1b neurons are as such:

$$W_{(CurrentHeading->CPU1a)} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$W_{\text{CurrentHeading ->CPU1b}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Similarly, $W_{DH\rightarrow \text{CPU1a}}$ defines the connection weights from the CPU4 (desired heading) neurons to CPU1a and $W_{DH\rightarrow \text{CPU1b}}$ to CPU1b respectively, matching the same connections in the CX anatomically:

$$W_{\text{Desired Heading -> CPU1a}} = \begin{bmatrix} 0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0 \\ 0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0 \end{bmatrix}$$

$$W_{\text{Desired Heading -> CPU1b}} = \begin{bmatrix} 1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1 \end{bmatrix}$$

Note that the CPU4 connections to the CPU1b are ipsilateral (on the same side), meanwhile the connections to CPU1a are contralateral:

$$W_{\text{Desired Heading} \rightarrow \text{CPU1a}} = \begin{bmatrix} 0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0 \\ 0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0 \end{bmatrix}$$

CPU1a(Left hemisphere)

CPU1a(Right hemisphere)

CPU4(Left hemisphere)

CPU4(Right hemisphere)

We also note that the wrapping case is not handled by the CPU1a, hence there are no connections from the CPU4 in this case:

$$W_{\text{Desired Heading} \rightarrow \text{CPU1a}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Going to the motor neurons, CPU1a neurons connect ipsilaterally (on the same side as the motor neuron), while the CPU1b neurons are connecting contralaterally (opposite side):

$$\mathrm{W}_{\text{CPU1a -> motor}} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$\mathrm{W}_{\text{CPU1b ->motor}} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Finally, the motor signal is a scalar which is the summation of both types

$$\text{motor} = \text{CPU1a} \cdot W_{\text{Cpu1a} \rightarrow \text{motor}} + \text{CPU1b} \cdot W_{\text{CPU1b} \rightarrow \text{motor}}$$

3. Implementation and results

3.1 System overview

Bringing the visual network and the CX model together, our system will look as follows, with cameras as visual input and a 3-axis magnetometer for compass input:

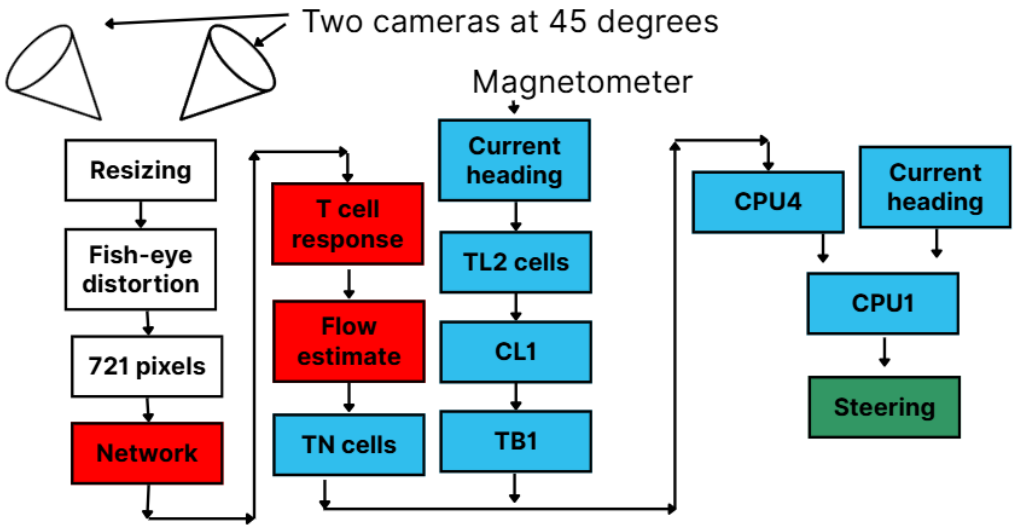

Figure 3.1: System overview showing the flow signal from the optic lobe model and the steering signal from the CX model. For the optic lobe model, we used the library from [49] and for the CX model we used the implementation from [20]. The two input cameras are pointing at 45 degrees from the midline to match the preference angle of the tn neurons in the CX.

3.2 Recording responses to moving edges

We first examined the responses from the central T cells when presented when presented with moving edges:

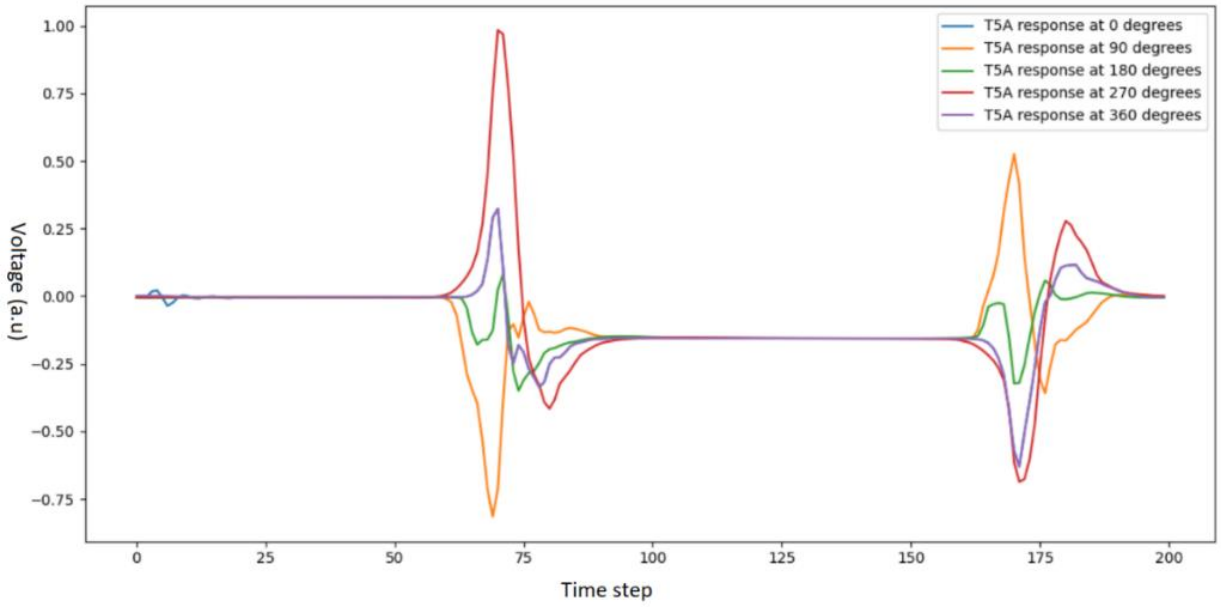

Figure 3.2: Response of the central T5A cell to a dark moving edge over a gray background. At 0 degrees, the edge is moving vertically (up) and the response is recorded. Then the stimulus edge is rotated clockwise and presented to the neuron again. Responses were also recorded when the neuron was presented with a moving edge at different orientations (right, left, up and down). A clear preference to the edge moving at 270 degrees (horizontally, left) is seen.

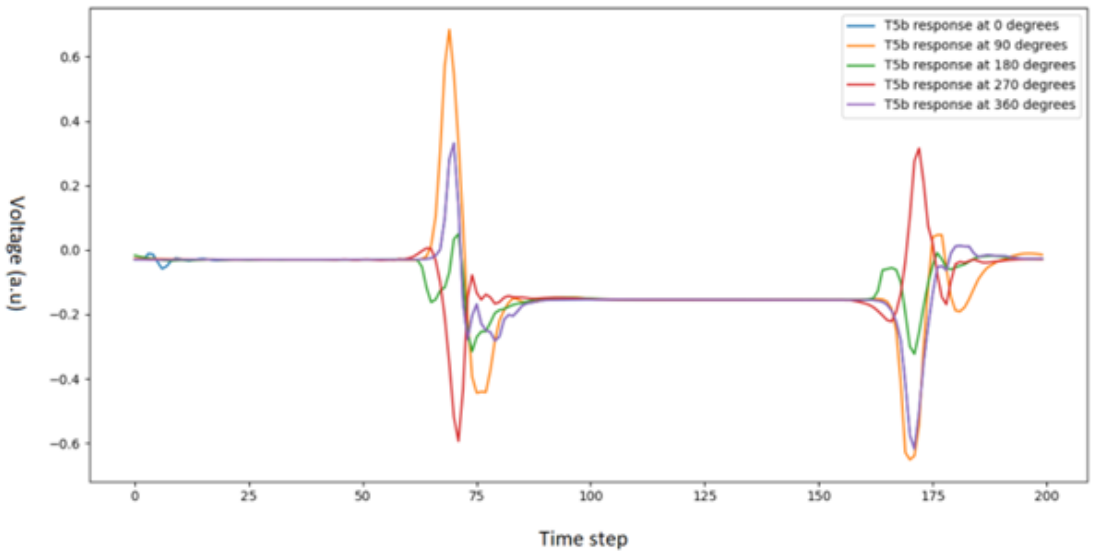

Figure 3.3: Response from the central T5B cell to a dark moving edge over a gray background. As expected, this cell responds most strongly when the edge is oriented at 90 degrees (edge moving to the right)

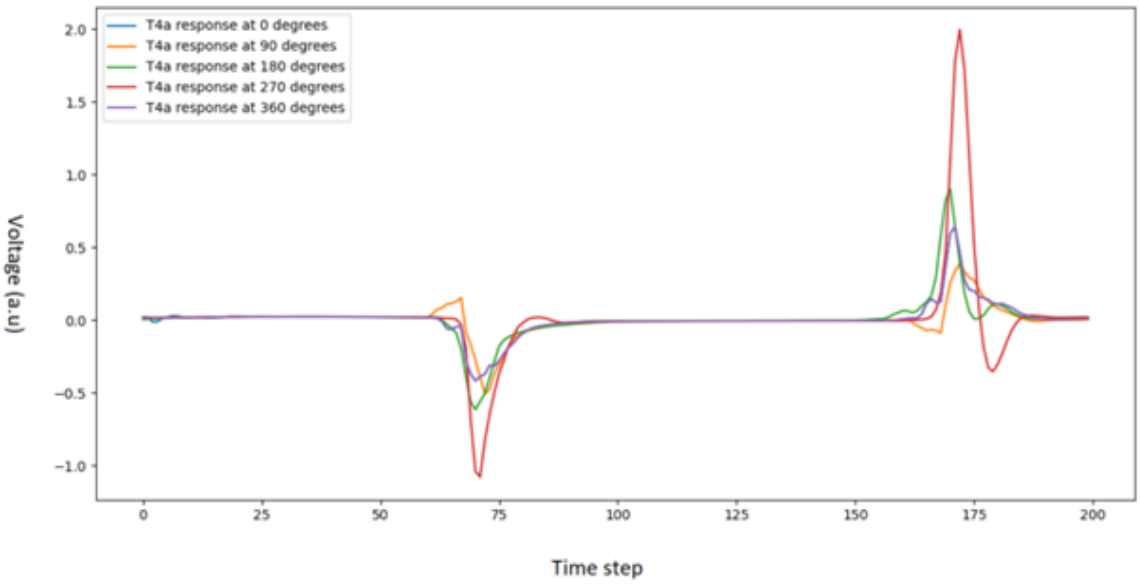

Figure 3.4: Response from the central T4A cell to a light moving edge moving over a grey background. As expected, this is the ON pathway cell that complements the T5A, and it shows preferred response to light edge moving to the left (stimulus edge rotated clockwise 270 degrees)

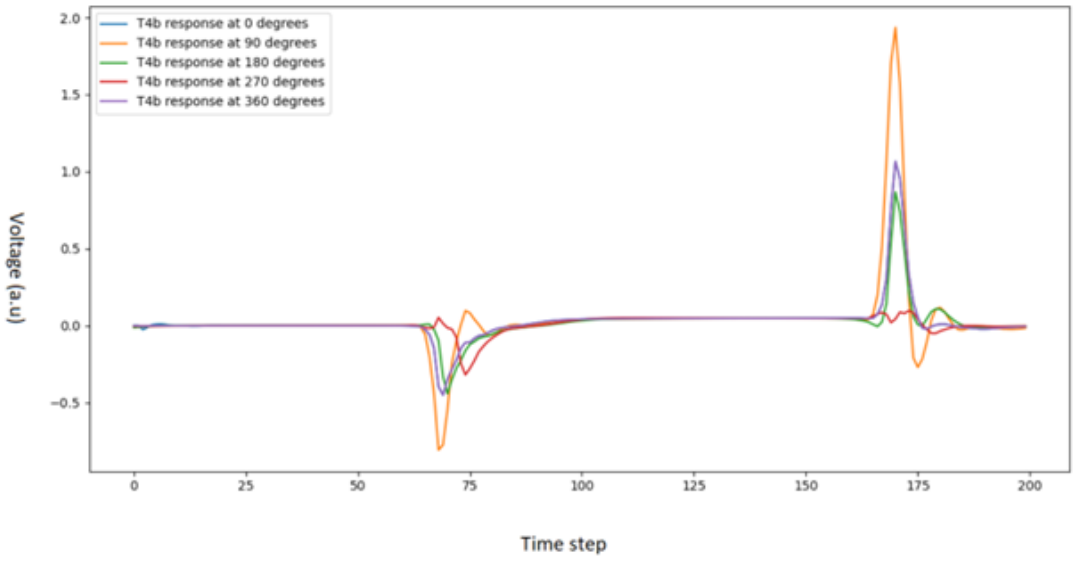

Figure 3.5: Response of the T4B cell to a light moving edge over a gray background, showing a clear preference for an edge moving to the right (90 degrees).

Next, we recorded the response from the central T cells in the case where the simulated agent is moving forward

3.3 Testing in a simulated square hall environment

A simple three-dimensional environment was built based on the project from [60], which was originally for developing simple 3D maze games in python using raytracing.

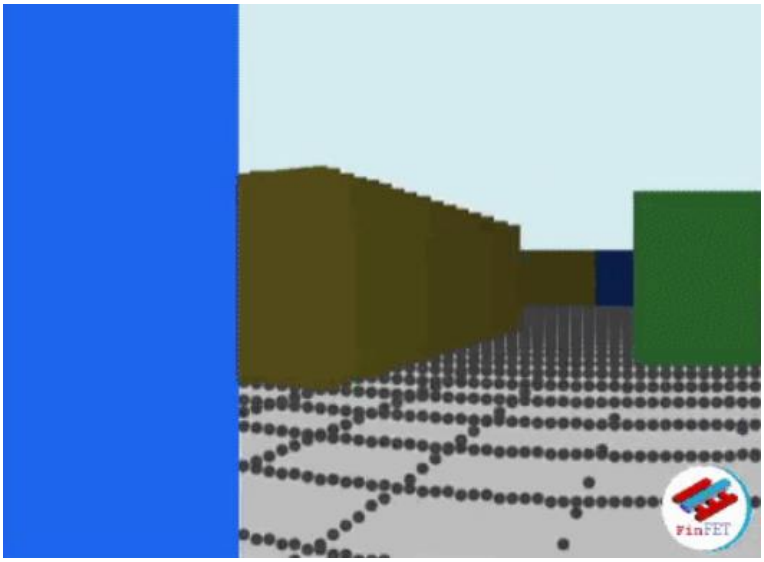

Figure 3.6: FinFet, a 3D maze game library with raytracing, utilizing only a minimal set of additional libraries, where each frame is visualized with Matplotlib

We designed our environment as a square arena with high contrast walls. The simulated agent would start in the centre of the arena. The inputs to the agent are stereo images from the left and right camera views (going into the optic lobe model), as well as the current heading (going into the compass neurons in the CX). Then the outputs from the motor neuron determine if the steering will be to the right or to the left.

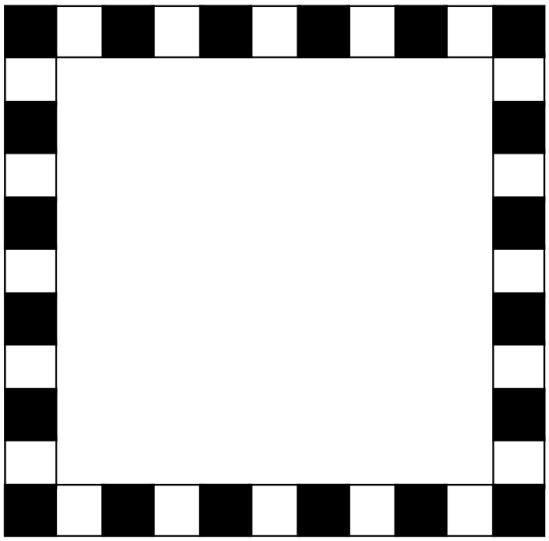

Figure 3.7: Top view of the square arena, 11 by 11 units, with high contrast walls. The simulated agent is placed in the center of the arena for the starting position.

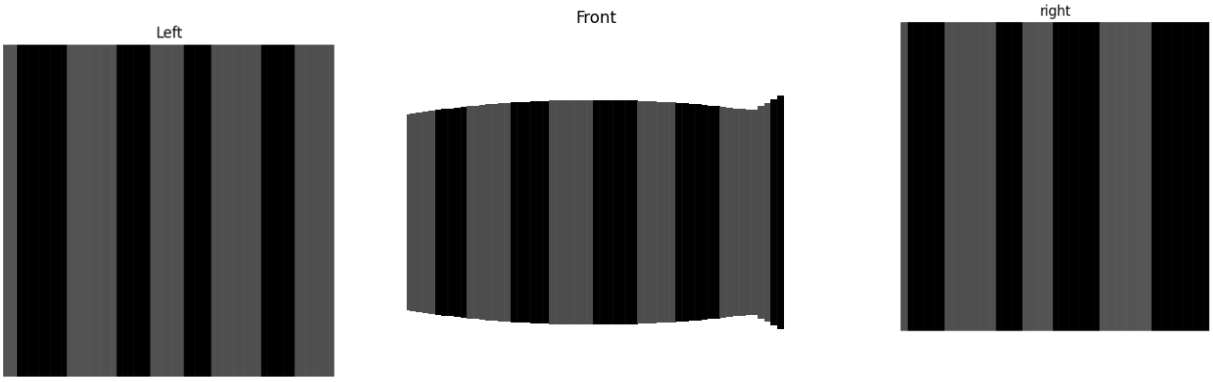

Figure 3.8: What the agent sees: Left view, front view and right view

Next, we conducted a simulation in which the agent was moved forward in a straight line (at a heading of 90 degrees) for 300 milliseconds, then moving backwards (with the same heading) for another 300 milliseconds, and recorded the response from the T cells of the right eye. During this, we recorded the responses from the central T4/5 cells from the right eye. As expected, the T4b and T5b showed the strongest response when moving forward, since from their view of the right eye the flow is positive (it will

be seeing edges moving to the right in the visual field). While going backward we see the strongest response from the T4A and T5A cells (here the eye was seeing the edges move to the left). Note how the response alternates between the type 5 and type 4 cell, as a dark then bright edge moves over the field of view respectively.

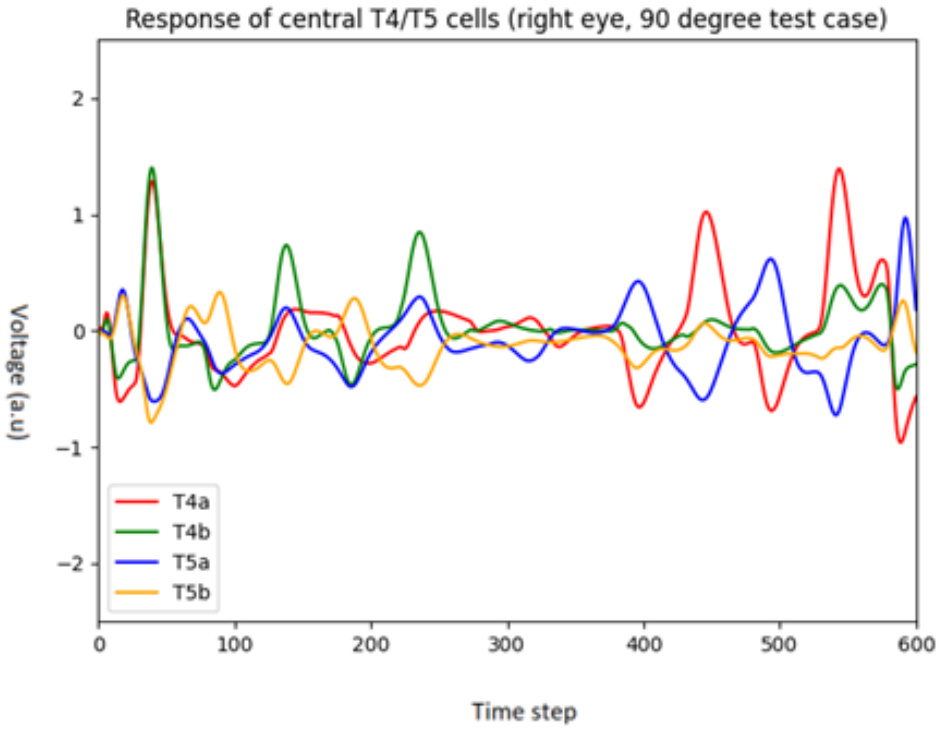

Figure 3.9: Responses from the right eye T cells when moving forward versus moving backward.

We then set a threshold of a response around 1 volt to be considered as a detected motion event for each cell:

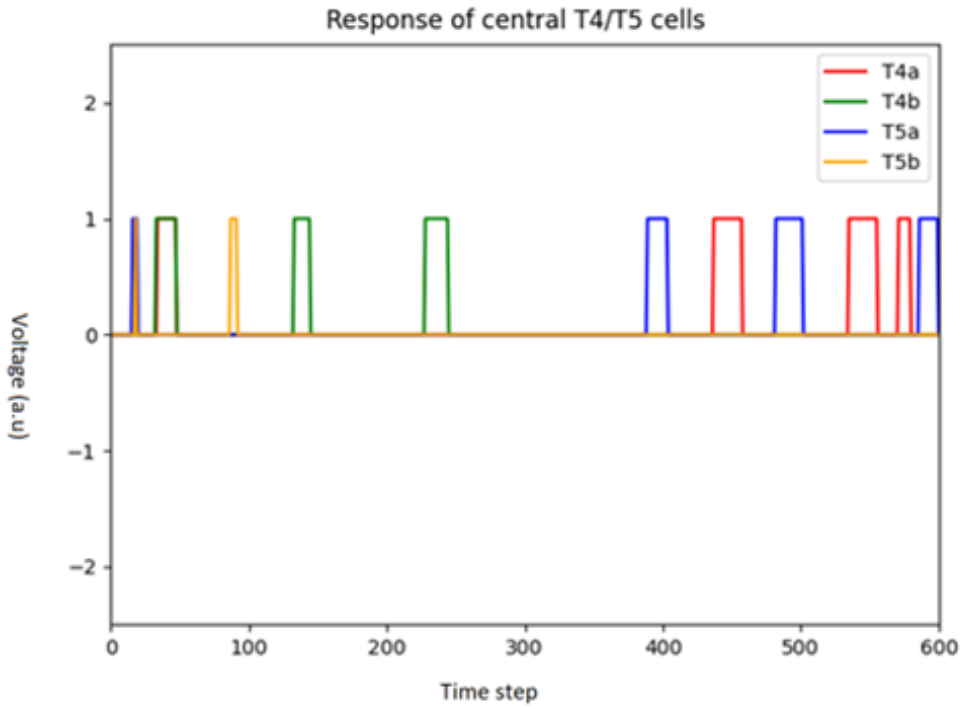

Figure 3.10: Responses after applying the threshold, when moving forward for 300ms then backward for 300ms.

Finally, we can visualize the responses from the entire population from both eyes:

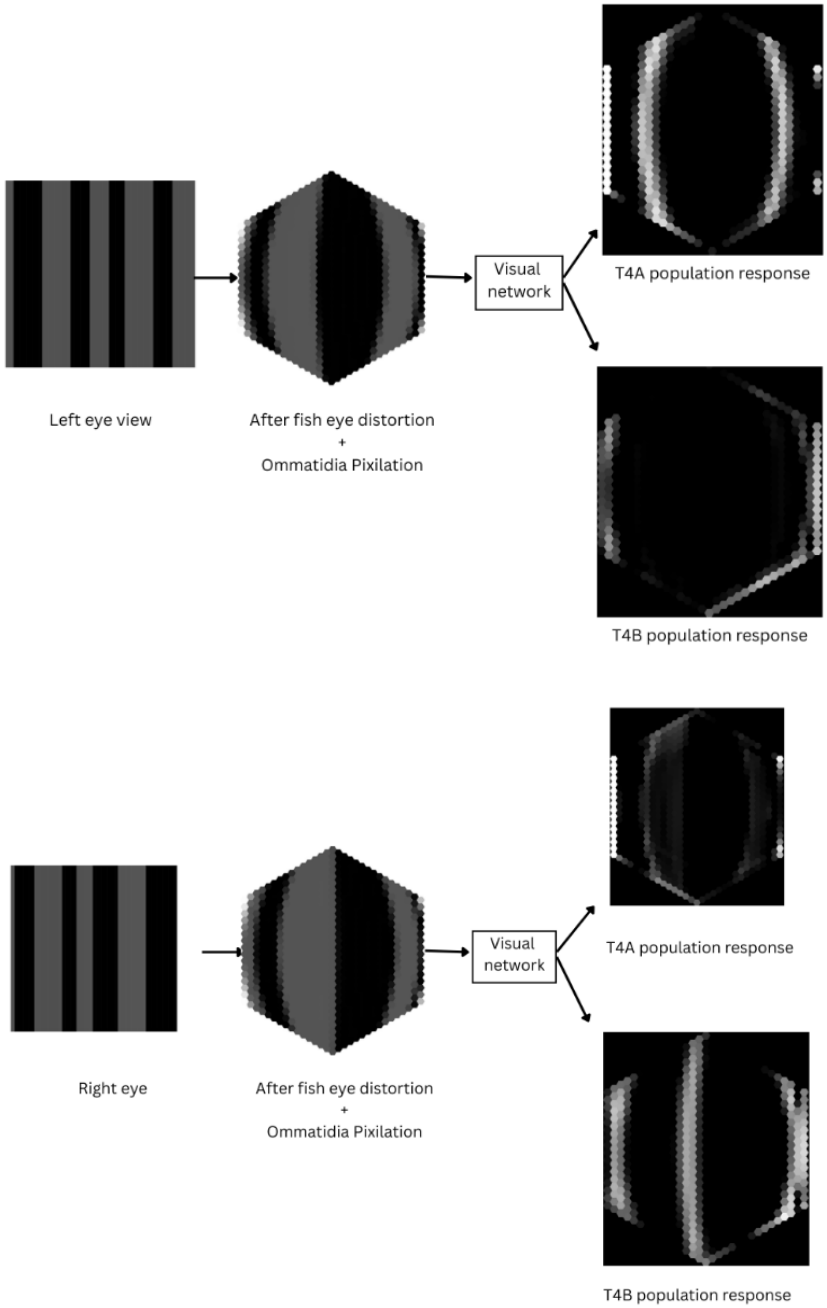

Figure 3.11: Visualizing the input image after fish-eye distortion and pixelation, which replicates the wide-angle, low-resolution image that the fruit fly sees, as well as the responses from the T4A/B populations in the right and left eye, recorded during forward motion. Bright pixels indicate a strong response to the moving edge. As expected during forward motion, the B cell responses are stronger in the right eye while the A cell responses are stronger in the left eye

If we were to look only at the response of the central cells, we would miss the movements detected in the remaining field of view. So, to consider the events from the whole population, we count the number of cells with a response above the threshold. If more than 50 cells show a response that exceeds the threshold of one volt, then the output flow would have the sign of that specific subtype.

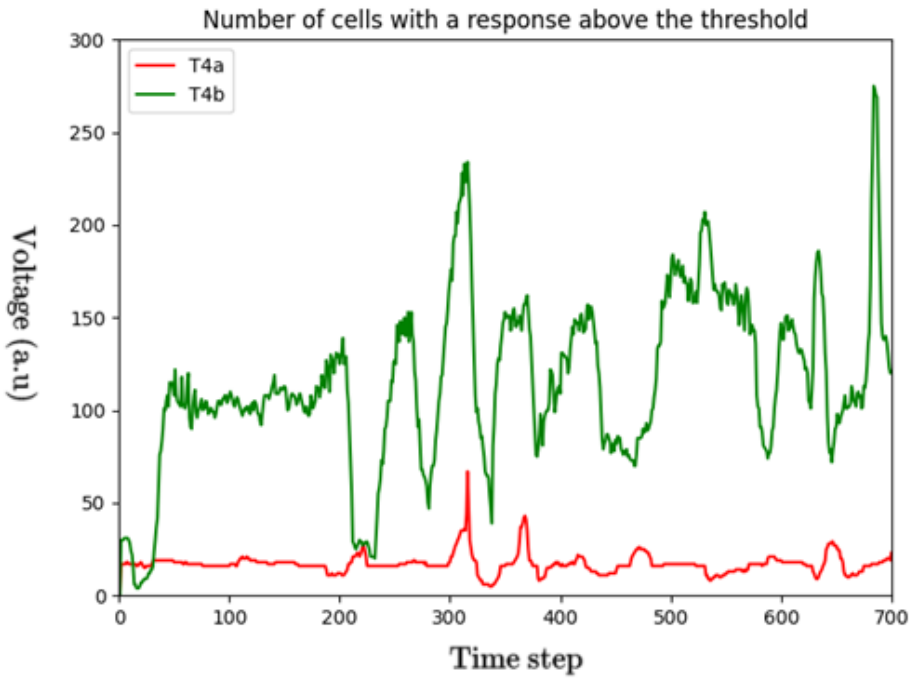

Figure 3.12: The number of cells with a response above the threshold, when moving forward for 700ms. Since the B type is the clear winner here, the flow estimation for this (right) eye will be negative (left direction).

Finally, we can move to a full navigation test. For the first phase, the outbound journey, the agent is driven forward on a straight line with an external control signal, away from the home location in the centre. In the second phase, the inbound journey, the agent takes control of the steering.

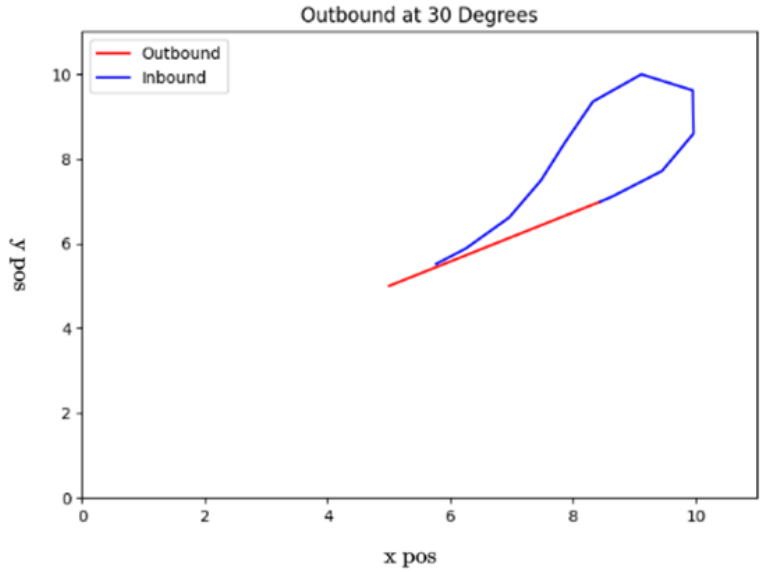

Figure 3.13: The first test scenario: outbound at a heading of 30 degrees (red path), then returning to the center (blue path)

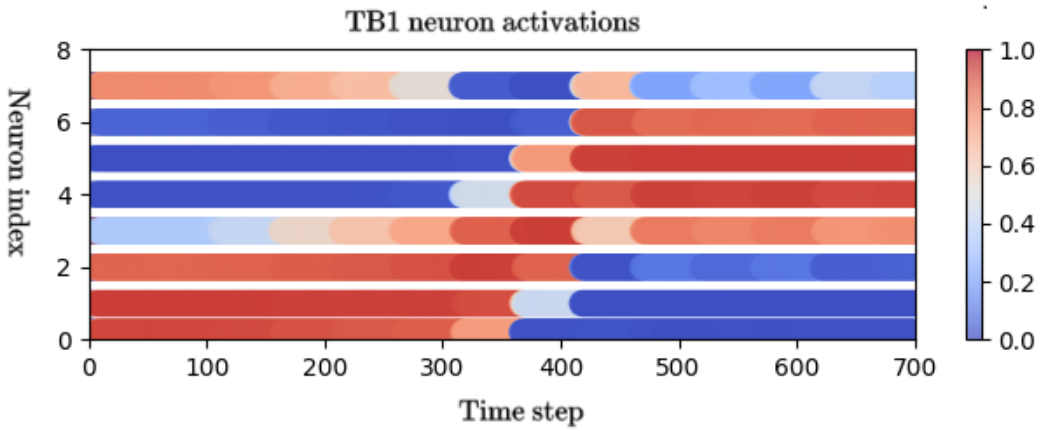

Figure 3.14: Activity of the TB1 (current heading) neurons

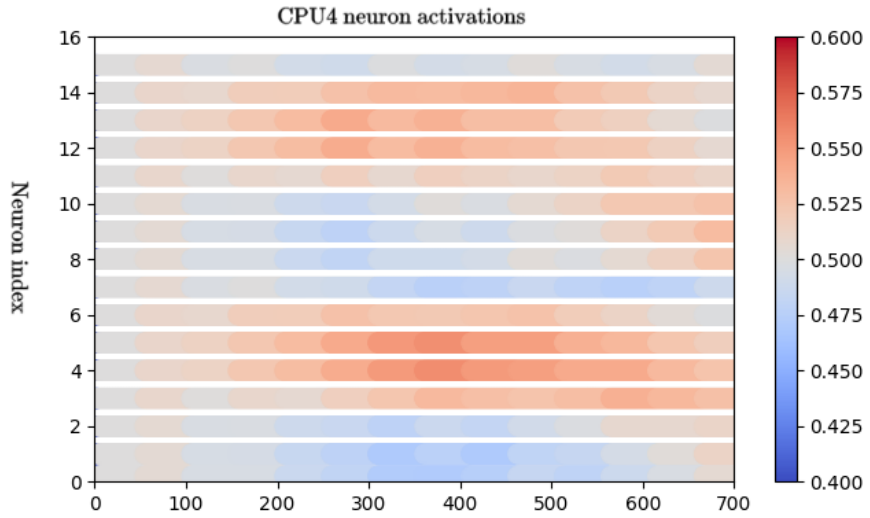

Figure 3.15: Activity of the 16 CPU4 (desired heading) neurons. Note that there are two sets of CPU4 neurons, one per hemisphere

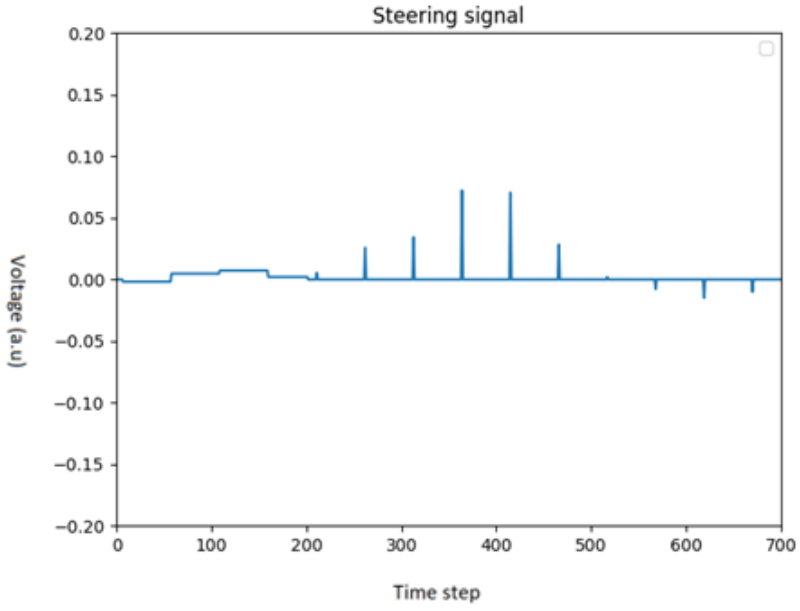

Figure 3.16: Motor steering signal (from the CPU1). Positive values are for counterclockwise rotation and negative values are for clockwise rotation.

We further tested our approach in multiple experiments going out at different starting headings and observed the agent returning close to the starting point :

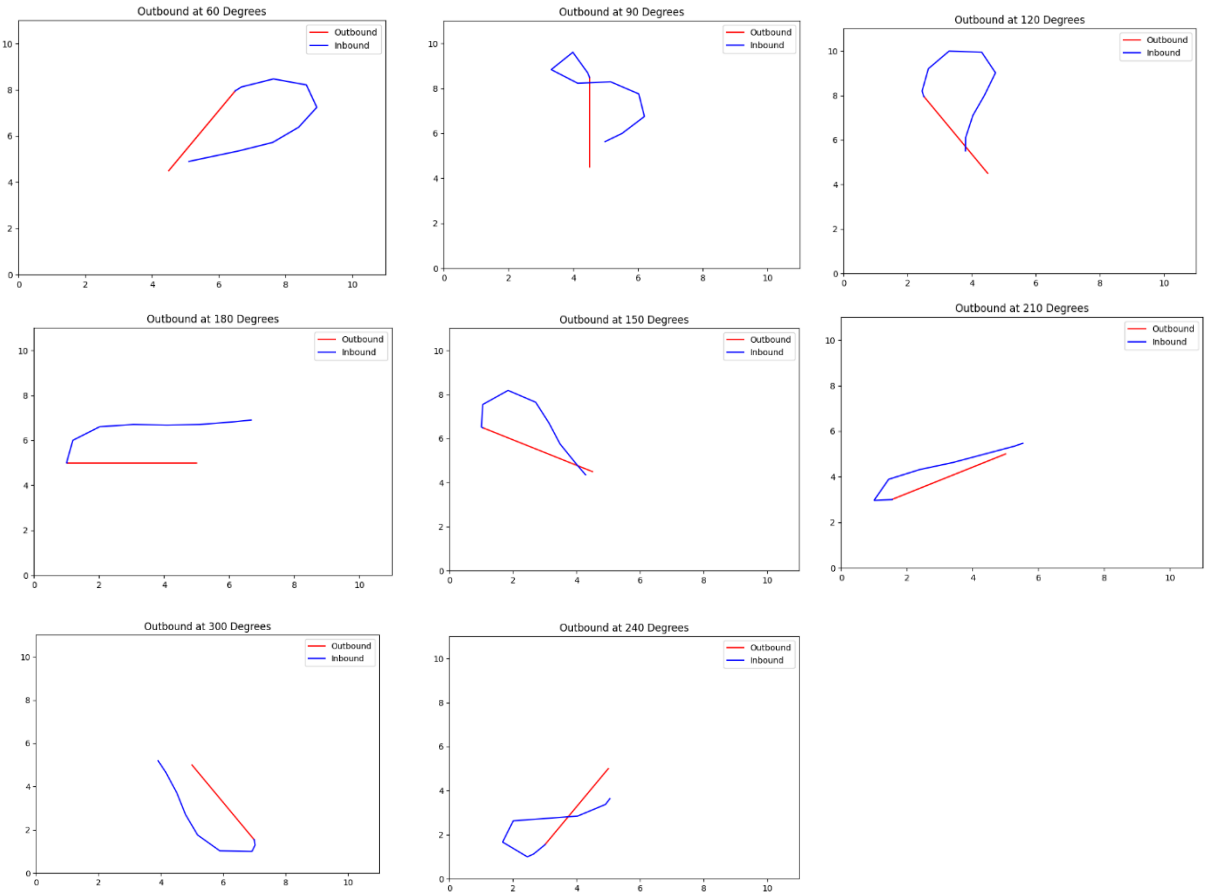

Figure 3.17: Position logs for navigation test cases, starting at outbound journeys at a heading 30 degrees, 60 degrees, 90 degrees 120 degrees, 180 degrees, 150 degrees, 300 degrees, 210 degrees, 240 degrees

3.4 Implementation in the real

For the second part of the implementation, we built a differential mobile robot with similar actuators and sensors as the simulated agent, using the same laptop for running the optic lobe/cx algorithm, and tested in a corridor environment.

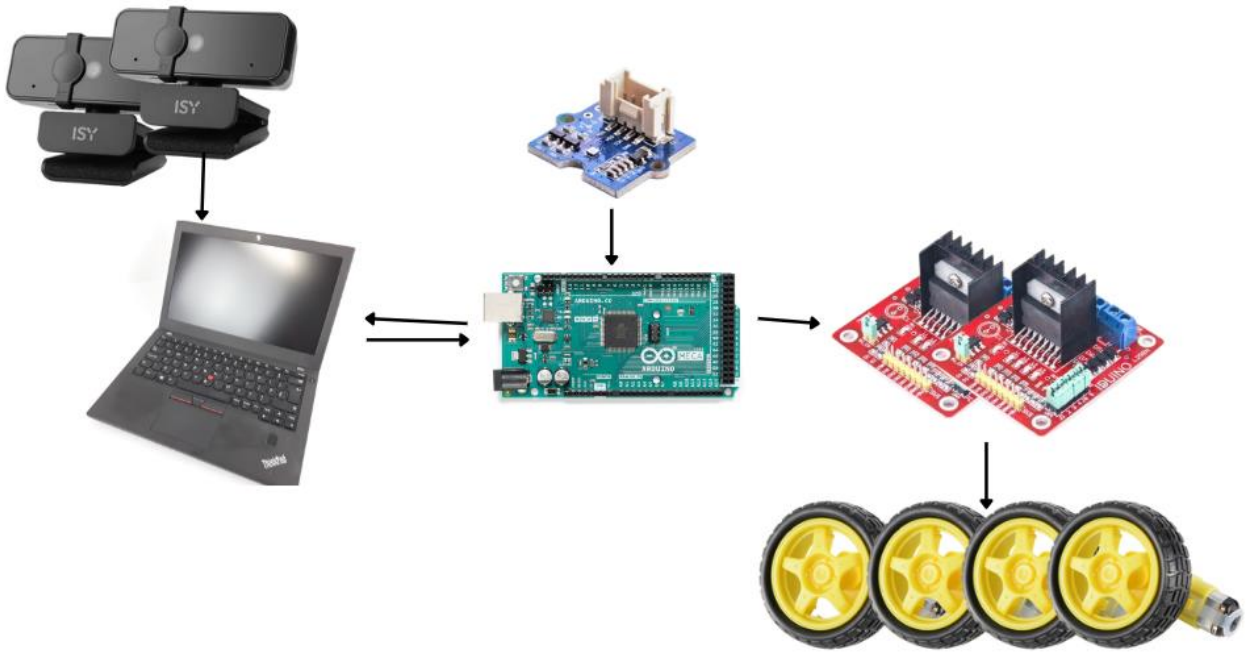

Figure 3.18: Hardware implementation overview. The two cameras were mounted on the wooden frame of the robot pointed at 45 degrees from the midline, while the 3 axis magnetometer from Grove was connected to the Arduino Mega 2560 board. The PWM signal from the Arduino for setting the motor speeds goes to the two L298 h-bridges, which then provided power to the 4 motors. A ThinkPad X270 laptop would take the images from the cameras and run them through the insect brain model and communicate with the Arduino using PySerial.

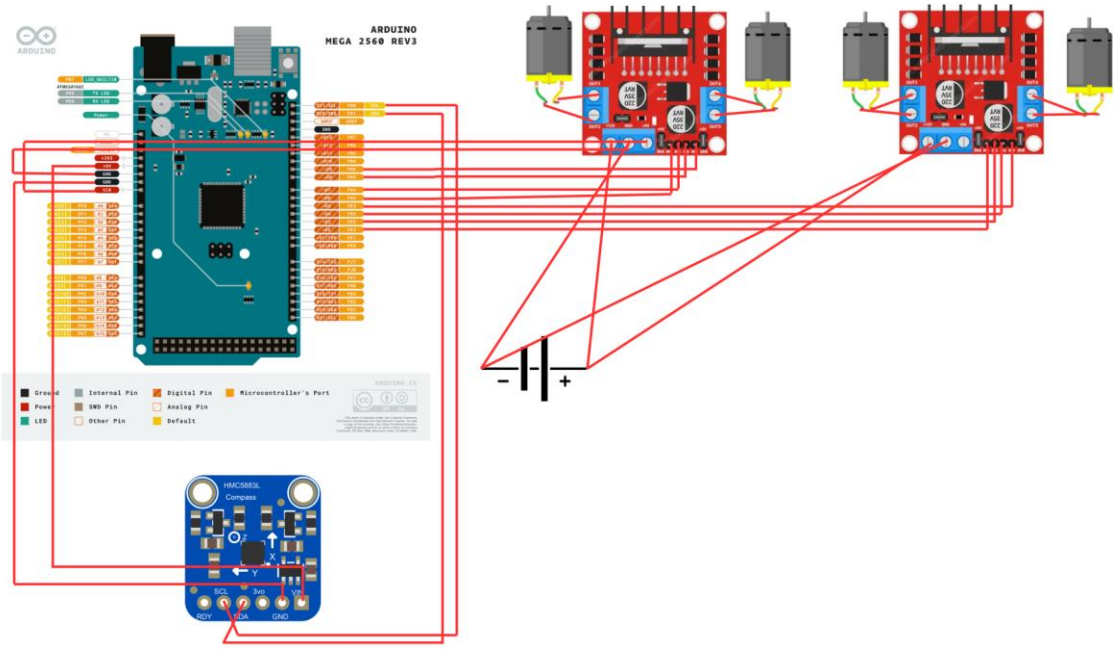

Figure 3.19: Wiring diagram of the Arduino, the two l298 h-bridges, the compass and the 12v source battery. Component diagrams from [63][64][65][66]

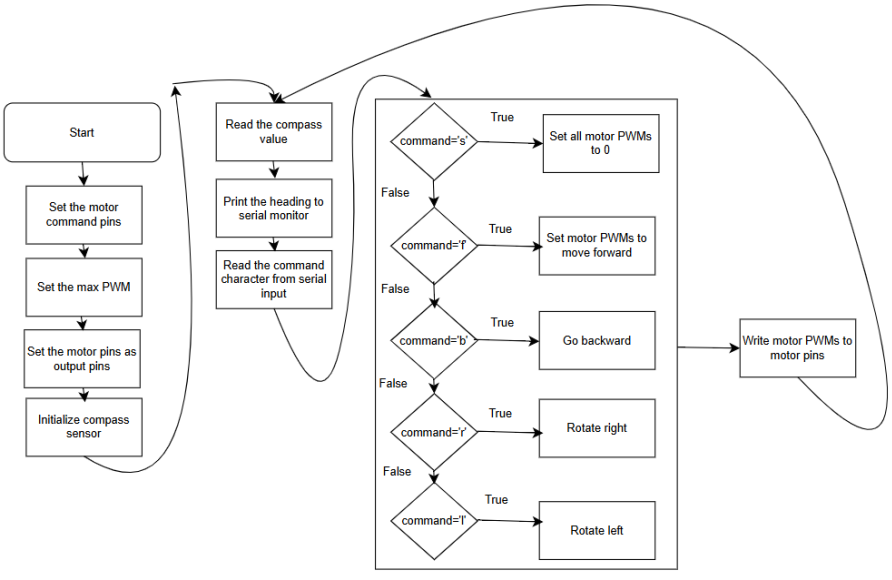

Figure 3.20: Control loop on the Arduino

Then we placed the robot in the following corridor environment. For the first phase , the robot was driven forward for a pre-defined duration, during this it should accumulate a homing vector to steer it back to the starting point. In this case the robot was not able to return to the starting home position. Compared to the square arena environment in simulation, the corridor environment had relatively sparse edges with lower contrast, causing erroneous optic flow estimates from the optic lobe network.

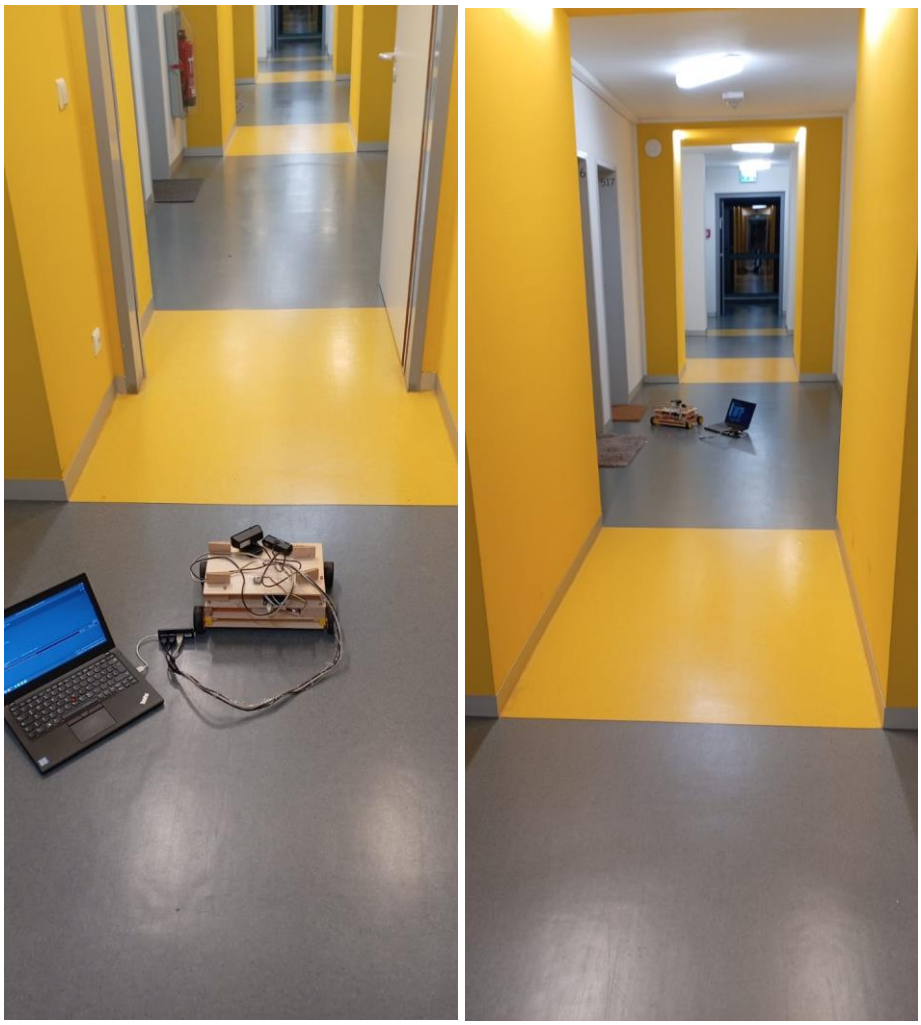

Figure 3.21: The mobile differential steering robot in the second test environment next to the laptop running our algorithm

4. Discussion

4.1 Limitations and comparison with visual SLAM

We finally revisit the question we set out to answer at the start of this project: how does the CX/Optic Lobe approach perform compared a visual SLAM approach? To answer this, we looked at experiments using visual slam in a similar hall-like environment as the one used in our simulation. The work in [59] has looked at the use of ORB- SLAM in a wheeled indoor humanoid robot named Pepper and was tested in two different hall environments. The sensory inputs were a monocular, 640x480 forehead camera and wheel odometers. The robot was also given a map for both test cases, which were mapped by remotely controlling the robot to build the 3D map. A Google Tango Tablet mounted on the robot was used for ground truth position reference.

The first environment was a feature rich rectangular lab. In this case, the robot was able to navigate correctly. While the second environment was a large and relatively featureless environment, in this case the robot was not able to navigate correctly due to the large distance between the robot and the landmarks used by ORB-SLAM. However, in both cases the algorithm was robust to changes in the environment like furniture positions.

Figure 4.1: Pepper robot in the two rectangular environments, the mechatronics lab which is a 10x9 meters area (left) and the university hall which is 16x27.5 meters (right)

It thus appears that both the visual slam and our approach are both viable solutions for navigation in feature rich environments, while struggling with featureless environments like the university hall above and low-contrast environment like the dorm hallway in our second test. Similarly, the size of the environment also seems to present a difficulty to both approaches. In our case, the depth of the moving edge in the field of view of the camera greatly affects the apparent speed of motion, which can lead to erroneous inputs to the CX.

Another limitation of our approach compared to SLAM is the inability to go around corners, since the CPU4 neurons only provide a homing vector and there is no mechanism to represent obstacles in the environment. That said, the CX/Optic lobe approach has the advantage of not needing the mapping step, which in the case of Pepper robot had to be done manually with remote control, and the features had to be captured from multiple angles due to the narrow field of view of the camera.

4.2 Future work

So far, we have arrived at a neurorobotic approach that could serve as an alternative to V-SLAM in the limited case of a high contrast environment that doesn't have obstacles. There are two aspects of our approach that could be further explored in the future to arrive at a more complete solution:

1- The inclusion of models for the remaining areas of the brain, specifically the Mushroom Bodies, since they play a critical role in memory formation and plume navigation, which is a major component of navigation in insects besides visual navigation.

2- We have used traditional CMOS cameras for visual input. Replacing this with event-based cameras could add the benefit of lower energy consumption, lower latency and higher temporal resolution [53]. Event based cameras work by capturing the changes in each pixel intensity independently, instead of waiting for a full frame to be captured like in traditional CMOS cameras.

3- Many neurons in the early visual system of the fruit fly are not spiking and were modeled as such in the optic lobe model. However, replacing the non-spiking neuron dynamics model used in the optic lobe with a Leaky Integrate and Fire neuron (LIF) model, i.e. implementing the optic lobe model as a Spiking neural net, would enable us to exploit the advantages of running the model on neuromorphic boards [50][51]. Neuromorphic boards, such as the IBM NS1e board, contain asynchronously connected processing units which run in parallel. One advantage of such boards is that they offer a reduction in weight, size and power requirements. An example of this being used in mobile robots can be found in [51], where a mobile robot with IBM's TrueNorth chip was able to achieve self-driving in a rugged environment. However, one difficulty with training SNNs is that they are non-differentiable at spike time, which means that

traditional back propagation mechanisms cannot be used to update SNNs. Instead, the Reward-Modulated-Spike-Timing-Dependent-Plasticity learning rule could be used to train the SNN [52].

4.3 Summary

This project set out to bring together the recently developed connectome constrained model for the Optic Lobe of the fruit fly with the previously developed anatomically constrained Central Complex model of the sweat bee. The goal was then to investigate if the insect brain model approach could serve as an alternative to the currently popular method of V-SLAM. A comprehensive literature review was then conducted to understand motion perception in flies and the path integration technique in the central complex. Both models were then integrated together, and a simulation environment was built with a final test scenario to test if our approach could navigate back to a starting position. Then a differential mobile robot was built and tested in a real environment. Finally, we compared our approach to a study using ORB Slam, noting similarities in the limitations of both approaches.

References

[1] *Neurorobotics*. (n.d.). Wikipedia. https://en.wikipedia.org/wiki/Neurorobotics

[2] Hwu, T. J., & Krichmar, J. L. (2022). *Neurorobotics: Connecting the Brain, Body, and Environment*. MIT Press.

[3] Ijspeert, A. J., Crespi, A., Ryczko, D., & Cabelguen, J. (2007). From Swimming to Walking with a Salamander Robot Driven by a Spinal Cord Model. *Science*, *315*(5817), 1416–1420. https://doi.org/10.1126/science.1138353

[4] Gutierrez-Galan, D., Dominguez-Morales, J. P., Perez-Peña, F., Jimenez-Fernandez, A., & Linares-Barranco, A. (2019). Neuropod: A real-time neuromorphic spiking CPG applied to robotics. *Neurocomputing*, *381*, 10–19. https://doi.org/10.1016/j.neucom.2019.11.007

[5] Krichmar, J. L. (2002). Machine Psychology: autonomous behavior, perceptual categorization and conditioning in a brain-based device. *Cerebral Cortex*, *12*(8), 818–830. https://doi.org/10.1093/cercor/12.8.818

[6] Cox, B., & Krichmar, J. (2009). Neuromodulation as a robot controller. *IEEE Robotics & Automation Magazine*, *16*(3), 72–80. https://doi.org/10.1109/mra.2009.933628

[7] Barros, A. M., Michel, M., Moline, Y., Corre, G., & Carrel, F. (2022). A comprehensive survey of visual SLAM algorithms. *Robotics*, *11*(1), 24. https://doi.org/10.3390/robotics11010024

[8] Muhammad, K., Ullah, A., Lloret, J., Del Ser, J., & De Albuquerque, V. H. C. (2020). Deep learning for safe autonomous driving: current challenges and future directions. *IEEE Transactions on Intelligent Transportation Systems*, *22*(7), 4316–4336. https://doi.org/10.1109/tits.2020.3032227

[9] Kazerouni, I. A., Fitzgerald, L., Dooly, G., & Toal, D. (2022). A survey of state-of-the-art on visual SLAM. *Expert Systems With Applications*, *205*, 117734. https://doi.org/10.1016/j.eswa.2022.117734

[10] Cheng, Jun & Zhang, Liyan & Chen, Qihong & Hu, Xinrong & Cai, Jingcao. (2022). A review of visual SLAM methods for autonomous driving vehicles. Engineering Applications of Artificial Intelligence. 114. 104992. 10.1016/j.engappai.2022.104992.

[11] Yurtsever, E., Lambert, J., Carballo, A., & Takeda, K. (2020). A survey of Autonomous Driving: Common practices and Emerging technologies. *IEEE Access*, *8*, 58443–58469. https://doi.org/10.1109/access.2020.2983149

[13] Behera, R. K., Pradhan, S. K., Sahoo, S., & Negi, N. (2022). Small brain, big smarts: An insight into learning and cognitive abilities of insects. *www.thepharmajournal.com*. https://www.thepharmajournal.com/special-issue?year=2022&vol=11&issue=11S&ArticleId=16709

[14] Knight, J. C., Sakhapov, D., Domcsek, N., Dewar, A. D., Graham, P., Nowotny, T., & Philippides, A. (2019). Insect-Inspired Visual Navigation On-Board an autonomous robot: Real-World routes encoded in a single layer network. *The 2019 Conference on Artificial Life*, 60–67. https://doi.org/10.1162/isal_a_00141

[15] P. Arena, A. Li Noce, L. Patanè and S. Taffara, "Insect-Inspired Spiking Neural Controllers for Adaptive Behaviors in Bio-Robots," in IEEE Instrumentation & Measurement Magazine, vol. 25, no. 9, pp. 19-27, December 2022, doi: 10.1109/MIM.2022.9955469.

[16] Stelzer, A., Vayugundla, M., Mair, E., Suppa, M., & Burgard, W. (2018). Towards efficient and scalable visual homing. *The International Journal of Robotics Research*, *37*(2–3), 225–248. https://doi.org/10.1177/0278364918761115

[17] Menzel, R., Greggers, U. The memory structure of navigation in honeybees. *J Comp Physiol A* 201, 547–561 (2015). https://doi.org/10.1007/s00359-015-0987-6

[18] Yu, Z., Zardini, G., Censi, A., & Fuller, S. (2022). Visual Confined-Space navigation using an efficient learned bilinear optic flow approximation for insect-scale robots. *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 4250–4256. https://doi.org/10.1109/iros47612.2022.9981585

[19] Schoepe, T., Janotte, E., Milde, M. B., Bertrand, O. J. N., Egelhaaf, M., & Chicca, E. (2024). Finding the gap: neuromorphic motion-vision in dense environments. *Nature Communications*, *15*(1). https://doi.org/10.1038/s41467-024-45063-y

[20] Sun, X., Mangan, M., Peng, J., & Yue, S. (2025). I2Bot: an open-source tool for multi-modal and embodied simulation of insect navigation. *Journal of the Royal Society Interface*, *22*(222). https://doi.org/10.1098/rsif.2024.0586

[21] Yihe, Lu & Cen, Jiahao & Alkhoury Maroun, Rana & Webb, Barbara. (2024). Embodied visual route following by an insect-inspired robot. 10.21203/rs.3.rs-4222706/v1.

[23] Devineni, A. V. (2024). A complete wiring diagram of the fruit-fly brain. *Nature*, *634*(8032), 35–36. https://doi.org/10.1038/d41586-024-03029-6

[24] Schlegel P, Yin Y, Bates AS, Dorkenwald S, Eichler K, Brooks P, Han DS, Gkantia M, Dos Santos M, Munnelly EJ, Badalamente G, Serratosa Capdevila L, Sane VA, Fragniere AMC, Kiassat L, Pleijzier MW, Stürner T, Tamimi IFM, Dunne CR, Salgarella I, Javier A, Fang S, Perlman E, Kazimiers T, Jagannathan SR, Matsliah A, Sterling AR, Yu SC, McKellar CE; FlyWire Consortium; Costa M, Seung HS, Murthy M, Hartenstein V, Bock DD, Jefferis GSXE. Whole-brain annotation and multi-connectome cell typing of Drosophila. Nature. 2024 Oct;634(8032):139-152. doi: 10.1038/s41586-024-07686-5. Epub 2024 Oct 2. PMID: 39358521; PMCID: PMC11446831.

[25] Lappalainen, J.K., Tschopp, F.D., Prakhya, S. *et al.* Connectome-constrained networks predict neural activity across the fly visual system. *Nature* 634, 1132–1140 (2024). https://doi.org/10.1038/s41586-024-07939-3

[26] *FlyWire*. (n.d.). https://flywire.ai/for_media

[27] Savage, N. (2022). Insects offer inspiration for robot advances. *Nature*, *610*(7931), S18. https://doi.org/10.1038/d41586-022-03213-6

[28] Borst, A., Drews, M., & Meier, M. (2020). The neural network behind the eyes of a fly. *Current Opinion in Physiology*, *16*, 33–42. https://doi.org/10.1016/j.cophys.2020.05.004

[30] Fujiwara, T., Cruz, T. L., Bohnslav, J. P., & Chiappe, M. E. (2016). A faithful internal representation of walking movements in the Drosophila visual system. *Nature Neuroscience*, *20*(1), 72–81. https://doi.org/10.1038/nn.4435

[31] Paulk A, Millard SS, van Swinderen B. Vision in Drosophila: seeing the world through a model's eyes. Annu Rev Entomol. 2013;58:313-32. doi: 10.1146/annurev-ento-120811-153715. Epub 2012 Sep 27. PMID: 23020621.

[32] Sun, X., Yue, S., & Mangan, M. (2020). A decentralised neural model explaining optimal integration of navigational strategies in insects. *eLife*, *9*. https://doi.org/10.7554/elife.54026

[33] Honkanen, A., Adden, A., Da Silva Freitas, J., & Heinze, S. (2019). The insect central complex and the neural basis of navigational strategies. *Journal of Experimental Biology*, *222*(Suppl_1). https://doi.org/10.1242/jeb.188854

[34] Martin, J. P., Guo, P., Mu, L., Harley, C. M., & Ritzmann, R. E. (2015). Central-Complex control of movement in the freely walking cockroach. *Current Biology*, *25*(21), 2795–2803. https://doi.org/10.1016/j.cub.2015.09.044

[35] *FlyWire*. (n.d.-b). https://flywire.ai/

[36] Zhu, Y. (2013). The Drosophila visual system. *Cell Adhesion & Migration*, *7*(4), 333–344. https://doi.org/10.4161/cam.25521

[37] Borst, A., Haag, J., & Mauss, A. S. (2019). How fly neurons compute the direction of visual motion. *Journal of Comparative Physiology A*, *206*(2), 109–124. https://doi.org/10.1007/s00359-019-01375-9

[38] Borst, A., Haag, J., & Reiff, D. F. (2010). Fly Motion Vision. *Annual Review of Neuroscience*, *33*(1), 49–70. https://doi.org/10.1146/annurev-neuro-060909-153155

[39] Stone, T., Webb, B., Adden, A., Weddig, N. B., Honkanen, A., Templin, R., Wcislo, W., Scimeca, L., Warrant, E., & Heinze, S. (2017). An anatomically constrained model for path integration in the bee brain. *Current Biology*, *27*(20), 3069-3085.e11. https://doi.org/10.1016/j.cub.2017.08.052

[40] Götz, K.G. Flight control in *Drosophila* by visual perception of motion. *Kybernetik* 4, 199–208 (1968). https://doi.org/10.1007/BF00272517

[41] Seelig, J. D., & Jayaraman, V. (2015). Neural dynamics for landmark orientation and angular path integration. *Nature*, *521*(7551), 186–191. https://doi.org/10.1038/nature14446

[43] Sun, X., Yue, S., & Mangan, M. (2021). How the insect central complex could coordinate multimodal navigation. *eLife*, *10*. https://doi.org/10.7554/elife.73077

[44] Martin, J. P., Guo, P., Mu, L., Harley, C. M., & Ritzmann, R. E. (2015b). Central-Complex control of movement in the freely walking cockroach. *Current Biology*, *25*(21), 2795–2803. https://doi.org/10.1016/j.cub.2015.09.044

[45] Buehlmann, C., Hansson, B. S., & Knaden, M. (2012). Path integration controls Nest-Plume following in desert ants. *Current Biology*, *22*(7), 645–649. https://doi.org/10.1016/j.cub.2012.02.029

[46] *Insect Brain Database*. (n.d.). https://www.insectbraindb.org/app/search/neuron/schematic

[47] Chen K, Hwu T, Kashyap HJ, Krichmar JL, Stewart K, Xing J, Zou X. Neurorobots as a Means Toward Neuroethology and Explainable AI. Front Neurorobot. 2020 Oct 19;14:570308. doi: 10.3389/fnbot.2020.570308. PMID: 33192435; PMCID: PMC7604467.

[48] Youssef, I., Mutlu, M., Bayat, B., Crespi, A., Hauser, S., Conradt, J., Bernardino, A., & Ijspeert, A. (2020). A Neuro-Inspired computational model for a visually guided robotic lamprey using frame and event based cameras. *IEEE Robotics and Automation Letters*, *5*(2), 2395–2402. https://doi.org/10.1109/lra.2020.2972839

[49] Wang-Chen, S., Stimpfling, V. A., Lam, T. K. C., Özdil, P. G., Genoud, L., Hurtak, F., & Ramdya, P. (2024). NeuroMechFly v2: simulating embodied sensorimotor control in adult Drosophila. *Nature Methods*, *21*(12), 2353–2362. https://doi.org/10.1038/s41592-024-02497-y

[50] Bing, Z., Meschede, C., Röhrbein, F., Huang, K., & Knoll, A. C. (2018). A survey of robotics control based on Learning-Inspired spiking neural networks. *Frontiers in Neurorobotics*, *12*. https://doi.org/10.3389/fnbot.2018.00035

[51] Hwu, T., Isbell, J., Oros, N., & Krichmar, J. (2016, November 4). *A Self-Driving robot using deep convolutional neural networks on neuromorphic hardware*. arXiv.org. https://arxiv.org/abs/1611.01235

[52] Bing, Z., Meschede, C., Huang, K., Chen, G., ŘOhrbein, F., Akl, M., & Knoll, A. (2018). End to End Learning of Spiking Neural Network based on R-STDP for a Lane Keeping Vehicle, to be appear. *International Conference on Robotics and Automation*. https://mediatum.ub.tum.de/node?id=1429575

[53] G. Chen, H. Cao, J. Conradt, H. Tang, F. Rohrbein and A. Knoll, "Event-Based Neuromorphic Vision for Autonomous Driving: A Paradigm Shift for Bio-Inspired Visual Sensing and Perception," in IEEE Signal Processing Magazine, vol. 37, no. 4, pp. 34-49, July 2020, doi: 10.1109/MSP.2020.2985815.

[55] Sun L, Du Y, Yu H, Wei H, Xu W, Xu W. An Artificial Reflex Arc That Perceives Afferent Visual and Tactile Information and Controls Efferent Muscular Actions. Research (Wash D C). 2022 Feb 11;2022:9851843. doi: 10.34133/2022/9851843. PMID: 35252874; PMCID: PMC8858381.

[56] Cruz, F., Solis, M. A., & Navarro-Guerrero, N. (2023). Editorial: Cognitive inspired aspects of robot learning. *Frontiers in Neurorobotics*, *17*. https://doi.org/10.3389/fnbot.2023.1256788

[57] Wikipedia contributors. (2023, December 28). *Cataglyphis nigra*. Wikipedia. https://en.wikipedia.org/wiki/Cataglyphis_nigra

[58] https://archiv.ub.uni-heidelberg.de/volltextserver/29324/1/accelerated_neuromorphic_cybernetics.pdf

[59] Gómez, C., Mattamala, M., Resink, T., & Ruiz-Del-Solar, J. (2019). Visual SLAM-Based localization and Navigation for service robots: the Pepper Case. In *Lecture notes in computer science* (pp. 32–44). https://doi.org/10.1007/978-3-030-27544-0_3

[60] FinFetChannel. (n.d.). GitHub - FinFetChannel/RayCastingPythonMaze: A very simple 3D game made in Python. GitHub. https://github.com/FinFetChannel/RayCastingPythonMaze

[61] De Croon, G. C. H. E., Dupeyroux, J. J. G., Fuller, S. B., & Marshall, J. a. R. (2022). Insect-inspired AI for autonomous robots. *Science Robotics*, *7*(67). https://doi.org/10.1126/scirobotics.abl6334

[62] Wikipedia contributors. (2025, April 8). *Motion perception*. Wikipedia. https://en.wikipedia.org/wiki/Motion_perception

[63] *Controlling a DC motor with Arduino*. (n.d.). Controlling a DC Motor With Arduino. https://www.robotique.tech/robotics/controlling-a-dc-motor-with-arduino/

[64] Cirkit Design. (n.d.). *Cirkit Designer Tutorials*. https://docs.cirkitdesigner.com/component/44529a08-68c3-5f13-2ff5-c7f69de74de3/adafruit-hmc5883l-triple-axis-magnetometer

[65] LME Editorial Staff. (2022, October 16). *Interface L298N DC Motor Driver Module with Arduino*. Last Minute Engineers. https://lastminuteengineers.com/l298n-dc-stepper-driver-arduino-tutorial/

[66] *Arduino - home*. (n.d.). https://www.arduino.cc/

[67] Chen, L., Wu, P., Chitta, K., Jaeger, B., Geiger, A., & Li, H. (2024). End-to-end autonomous driving: challenges and frontiers. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1–20. https://doi.org/10.1109/tpami.2024.3435937

[68] Wikipedia contributors. (2024, December 15). *Bee learning and communication*. Wikipedia. https://en.wikipedia.org/wiki/Bee_learning_and_communication

[69] Wikipedia contributors. (2025a, March 30). *Drosophila*. Wikipedia. https://en.wikipedia.org/wiki/Drosophila

[70] Williams, J. (n.d.). *MPI Sintel Flow*. Perceiving Systems | Max Planck Institute for Intelligent Systems. https://is.mpg.de/ps/research_projects/mpi-sintel-flow

[71] Wikipedia contributors. (2025c, April 18). *List of animals by number of neurons*. Wikipedia. https://en.wikipedia.org/wiki/List_of_animals_by_number_of_neurons

[72] White, J. G., Southgate, E., Thomson, J. N., & Brenner, S. (1986). The structure of the nervous system of the nematodeCaenorhabditis elegans. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, *314*(1165), 1–340. https://doi.org/10.1098/rstb.1986.0056

[73] Wikipedia contributors. (2025d, April 28). *Place cell*. Wikipedia. https://en.wikipedia.org/wiki/Place_cell

[74] *OpenSLAM.org*. (n.d.). https://openslam-org.github.io/openratslam.html

[75] Wikipedia contributors. (2024a, August 18). *Megalopta genalis*. Wikipedia. https://en.wikipedia.org/wiki/Megalopta_genalis