Insect-Inspired Visual Guidance

are current familiarity-based models ready for long-ranged navigation?

June 2022

J.K.N. Verheyen

Delft University of Technology

TU Delft

Insect-Inspired Visual Guidance

are current familiarity-based models ready for long-ranged navigation?

Master of Science Thesis

by

J.K.N. Verheyen

For obtaining the degree of Master of Science in Aerospace Engineering
at the Delft University of Technology

Student number:	4458192
Project Duration:	March 2020 – June 2022
Supervisors:	Prof. G.C.H.E. de Croon
	Dr. J. Dupeyroux
	Ir. F. Paredes-Valles

Cover Image: Megachile fortis, U, Face, Jackon County, S. Dakota 2013. Photo courtesy of Sam Droege.
Retrieved from https://flic.kr/p/dQKHwR

TU Delft

Acknowledgements

First, I would like to thank my supervisors, Julien, Guido, and Federico, for their guidance and support throughout this thesis. This work would not have been possible without you.

Another word of appreciation goes out to Casper, Eoghan, and Reuben, you and Anna Beijerstraat 29 have always been a place that felt like home in Delft.

To Kilian and the boys of the GSC, thank you for the fun times over the past years, and years to come.

Lotte, thank you for being by my side all the way. Your unconditional support and love kept me motivated to succeed.

And finally to my family, I appreciate your interest, care, and support you have given me throughout this thesis tremendously.

J.K.N. Verheyen

Delft, June 2022

Abstract

Developing navigational algorithms for Micro Air Vehicles (MAVs) poses a constant challenging trade-off between performance, reliability, adaptability and efficiency. To tackle these challenges, one might want to look at and take inspiration from insects. Insects are skillful navigators which can guide themselves reliably through cluttered environments over long distances, while using very little energy. This thesis looks at the current state of insect-inspired local visual guidance methods and evaluates to what extent they are applicable for long-ranged visual guidance onboard MAVs. For this purpose, a novel dataset containing omnidirectional event vision, frame-based vision, depth frames, Inertial Measurement Unit (IMU) readings, and centimeter-accurate Global Navigation Satellite Systems (GNSS) positioning over kilometer long stretches in and around the TUDelft campus was collected. The analysis demonstrates that current scene familiarity models are not suited for long-ranged navigation, at least not in their current state.

Contents

List of Symbols

Greek Symbols

α	scaling factor
η	learning rate
ϕ	post-spike quantile of the amount of released neurotransmitter
σ	standard deviation
τ_a	time constant of decay of variable a
$\zeta \sim N(0, \sigma)$	Gaussian white noise current

Latin Symbols

A_+, A_-	magnitude of synaptic change due to long-term potentiation or depression
a, b, C, k	Izhikevich neuron model parameters
C	contrast sensitivity
c	synaptic tag (eligibility trace)
c, d	Izhikevich neuron reset potential and current increment after spiking
d	extracellular concentration of biogenic amine
Δw	change in weight/synaptic efficacy
E	energy
e_k	event
g	synaptic conductance
h_i	novelty unit input (Infomax neural network)
$I(t)$	neural input current
$I(x, y, t)$	image intensity function
L	log photocurrent
N, M	amount of ...
p	event polarity
S	amount of active neurotransmitter
s_j^l	postsynaptic spike at neuron j in layer l
t	time
u	neuron postsynaptic recovery current
u, v	optical flow components

v	neuron membrane potential
$\boldsymbol{v}$	velocity vector
$v_{\text{rest}}, v_t, v_{\text{rev}}$	neuron membrane rest/threshold/reversal potential
$w, \boldsymbol{w}$	weight or synaptic efficacy (vector)
x, y	position on the image plane/in the pixel array
y_i	novelty unit output (Infomax neural network)

Math Symbols

$\approx$	approximately equal
$\arg\min_a, \arg\max_a f(a)$	argument a of the minimum/maximum value of function $f(a)$
Δ	difference operator
δ	Dirac delta function
$\dot{a}$	the time derivative of a
$\doteq$	equality that is true by definition
$\in$	is an element of
$\int$	integral operator
∇	nabla operator
$\frac{\partial}{\partial t}$	partial derivative w.r.t. time
$\mathbb{R}$	set of real numbers
$\sum$	sum operator

Sub- and Superscripts

i	postsynaptic neuron index
j	presynaptic neuron index
l	layer
pre, post	pre- and postsynaptic contribution subscripts

List of Abbreviations

A	
AGAST	Adaptive and Generic corner detection based on the Accelerated Segment Test
ALV	Average Landmark Vector
ANN	Artificial Neural Network
APL	Anterior Paired Lateral
ATIS	Asynchronous Time Based Image Sensor
B	
BRIEF	Binary Robust independent Elementary Features
C	
CNN	Convolutional Neural Network
CX	Central Complex
D	
DAVIS	Dynamic and Active Pixel Vision Sensor
DID	Descend in Image Distance
DNN	Deep Neural Network
DRA	Dorsal Rim Area
DVS	Dynamics Vision Sensor
E	
EN	Extrinsic Neuron
F	
FAST	Features from Accelerated Segment Test
FOV	Field of View
G	
GNSS	Global Navigation Satellite Systems
I	
IMU	Inertial Measurement Unit
K	
KC	Kenyon Cell
L	
LIF	Leaky Integrate-and-Fire
LLN	Linked Local Navigation
LMC	Lamina Monopolar Cell
M	

MAV	Micro Air Vehicle
MB	Mushroom Body
MBON	Mushroom Body Output Neuron

O

OAST	Optimal Accelerated Segment Test
ORB	Oriented FAST and Rotated Brief

P

PI	Path Integration
PN	Projection Neuron

S

SIFT	Scale-Invariant Feature Transform
SLAM	Simultaneous Localization and Mapping
SNN	Spiking Neural Network
STDP	Spike-timing Dependent Plasticity
SURF	Speeded Up Robust Features

T

TS	Time Surface

U

UV	Ultraviolet

V

VO	Visual Odometry
vPN	Visual Projection Neuron

List of Figures

List of Tables

List of Listings

1

Introduction

Micro Air Vehicles (MAVs) show promising use for applications in hard-to-reach or dangerous environments such as disaster relieve (Delmerico et al., 2019). Being able to autonomously and reliably navigate through such environments poses an important challenge towards the deployability of these systems. Often, one cannot rely on external aid such as GNSS or traditional methods like dead-reckoning systems become inaccurate or impractical. Many insect species are able to tackle these challenges and are highly efficient at it. It is therefore of interest to study how insects are able to perform these tasks and how they can be implemented onboard human-made systems such as MAVs.

Vision is an ideal candidate as part of a navigation solution, as it provides a rich and highly dimensional source of information about the environment. Furthermore, vision sensors have interesting properties such as low power consumption, low weight, and compactness, which makes them an excellent candidate for use on MAVs. Vision-based navigation solutions have evidently evolved much during the past decades. Traditional visual guidance methods like visual(-inertial) odometry — often part of a Simultaneous Localization and Mapping (SLAM) framework — depend on extracting hand-crafted features from the environment (Forster et al., 2014; Mur-Artal and Tardos, 2017; Engel et al., 2014). Recently, learning based image processing with (deep) convolutional neural networks has shown superior performance over hand-crafted feature extractors (Krizhevsky et al., 2012) and has subsequently found its way in guidance applications ranging from optical flow (Hur and Roth, 2020) and depth estimation (Zhao et al., 2020) for augmenting traditional guidance pipelines to end-to-end learning of guidance policies (M. Müller et al., 2019). However, local navigation methods based on visual odometry, suffer from drift in their estimates unless not accounted for in computationally expensive techniques such as loop closure detection or global bundle adjustment. This seriously limits their applicability on MAVs where endurance and autonomy are of importance.

Another approach is to take inspiration from nature, where insects have already solved the navigation problem of navigating through cluttered, unknown environments. They do this in a very energy-efficient manner, which is a result of the manner in which they perceive and neurologically process visual information. Insects have compound eyes with relatively low resolution but which cover an almost panoramic field of view. Brightness changes in the environment asynchronously activate light-sensitive neurons in the compound eyes, this information is eventually encoded in temporally spaced voltage spikes (events) through the optic lobes (underlying networks of interconnected neurons) (Ibbotson et al., 1991; Bausenwein et al., 1992; Sommer and Rüdiger Wehner, 1975) and are further processed in the protocerebrum (Paulk et al., 2009; Menzel and Martin Giurfa, 2001). The combination of this sparse panoramic input and processing visual information as events in a spike-based manner allows visual information to be processed in a highly efficient manner (Tavanaei et al., 2019).

The attractive properties of processing visual information in an asynchronous, spike-based manner have inspired researchers to develop artificial (neuromorphic) counterparts like event cameras, Spiking

Neural Networks (SNNs) and neuromorphic computer chips such as Intel's Loihi (Davies et al., 2018) or IBM's TrueNorth (Akopyan et al., 2015). Event-based vision takes it inspiration from nature and works by asynchronously measuring brightness changes for each individual pixel in the form of events, opposed to capturing images in frames as conventional frame-based cameras do. Consequently, event-based vision sensors have a couple of advantages over conventional cameras (Gallego, Delbruck, et al., 2019): high temporal resolution (order of μs), low latency (order of μs), high dynamic range (140 dB vs. 60 dB of conventional cameras) and low power consumption (order of $10mW$). Their sparse and event-based output lend them very well for subsequent processing with SNNs. SNNs work best when combined with dedicated neuromorphic hardware. Together they have the potential to achieve similar, possibly better, performance than traditional artificial neural networks at a fraction of their power requirements (Pfeiffer and Pfeil, 2018).

The same approach can be applied to distill visual navigation techniques. Over the years, various different models of insect visual navigation have been developed, ranging from pure matching of retinotopic features as proposed in the original snapshot model of Cartwright and T. S. Collett (1983) to holistic encoding of route memory in a model of the insect's mushroom bodies (Ardin et al., 2016). These models however are primarily focused on replicating biological observations and their neural implementations. The approach in this research differs from the one taken by biomimetic work, in the sense that it does not try to mimic or model insects' navigational behavior, but takes *inspiration* from them to fulfill real-world challenges. Insect-inspired visual navigation methods often limit themselves to simulation (Dalen et al., 2018; Dijk, 2017; J. Müller et al., 2018) or do not take advantages of neuromorphic of hardware (Knight et al., 2019; Denuelle and Srinivasan, 2016). It is in our interest to adapt/develop methods that take advantage of the aforementioned aspects and make an effort towards bringing these capabilities to MAVs.

Motivation and Research Question

This thesis aims to evaluate the performance of state-of-the-art neural event-based visual insect-inspired navigation models for route following over large distances (>100 meters). The motivation behind this can be split up in three parts: **1.** Insect-inspired guidance methods based on the Snapshot model have thus far seen little deployment on MAV hardware, partly due to their limited capabilities w.r.t. regular methods **2.** Event-based vision combined with SNNs on neuromorphic hardware has the potential for very efficient computation which could bring visual route guidance to very small (insect-scale Karásek et al. (2018)) platforms **3.** Highly efficient visual guidance for MAV in e.g. search-and-rescue mission could free up valuable computational resources to allow other vital tasks to be preformed **4.** It is of interest to see what the current limitations of such insect-inspired models are in terms of their capability to cover large distances. This can be summarized in the main research objective of this thesis:

> **Evaluating current insect-inspired, neural-based, visual navigation methods with event-based vision over long distances for application onboard micro air vehicles in outdoor environments.**

From which the central question arises:

> **How well do current insect-inspired neural event-based visual navigation methods perform over long (>100 meters) distances?**

These are distilled into the following sub-questions:

- **Given a model, what are the limitations with respect to the amount of route information that can be stored, in other words, how big of an area can it represent?**
- **Which factors govern the performance of neural event-based visual navigation methods?**
- **Which datasets exist that would allow for such comparative evaluations?**

Structure of This Work

The work that is conferred in this thesis consists of four parts. The scientific paper, presented in Part I, contains the main contributions of this thesis.

Part II takes an in-depth look at the relevant literature covering local vision-based guidance, what we can learn from insects, event-based vision, and neural-based guidance methods. First, in Chapter 3, a literature study on local vision-based mapless guidance methods with a focus on insect-inspired navigation is presented. In Chapter 4 an overview is given of insect physiology and behavior related to visual navigation. The basics of event-based vision are discussed in Chapter 5. And finally, Neural-based visual guidance methodologies are presented in Chapter 6. The findings of the literature study are then summarized in Chapter 7.

Subsequently, a number of preliminary experiments investigating the performance of two recent neural familiarity-based insect navigation models is presented in Part III. Chapter 8 lays out the methodology that has been followed through the preliminary experiments and introduces the evaluated models. Chapter 9 evaluates the two introduced models in terms of their efficacy to discern 'familiarity' in real life scenes under varying conditions. These findings are summarized in Chapter 10, from which conclusions are drawn connecting them towards the scientific paper as presented in Part I.

Part IV are the Appendices, which present a detailed account of the implementation of L. Zhu et al. (2020)'s MB model which has been evaluated in Part I.

Part I

Scientific Paper

A novel multi-vision sensor dataset for insect-inspired outdoor autonomous navigation

Jan K.N. Verheyen, Julien Dupeyroux, and Guido C.H.E. de Croon

Micro Air Vehicle Laboratory, Department of Control and Simulation
Faculty of Aerospace Engineering, Delft University of Technology
2629HS Delft, The Netherlands
g.c.h.e.decroon@tudelft.nl

Abstract. Insects have — over millions of years of evolution — perfected many of the systems that roboticists aim to achieve; they can swiftly and robustly navigate through different environments under various conditions while at the same time being highly energy efficient. To reach this level of performance and efficiency one might want to look at and take inspiration from how these insects achieve their feats. Currently, no dataset exists that allows bio-inspired navigation models to be evaluated over long real-life routes. We present a novel dataset containing omnidirectional event vision, frame-based vision, depth frames, inertial measurement (IMU) readings, and centimeter-accurate GNSS positioning over kilometer long stretches in and around the TUDelft campus. The dataset is used to evaluate familiarity-based insect-inspired neural navigation models on their performance over longer sequences. It demonstrates that current scene familiarity models are not suited for long-ranged navigation, at least not in their current form.

Keywords: Long-range navigation · Neuromorphic systems · Event-based Camera · RGB Camera · GPS · GNSS

1 Introduction

To date, some insect-inspired aerial robots have been developed [10, 20] which mimic the flight capabilities of insects and while basic navigating capabilities have already been shown on board such limited platforms [30], their navigational performance falls short compared to their biological counterparts. Recent neural insect-inspired navigational models [2, 3, 11, 39] show promising results over short distances, but lack the capacity for long-ranged navigation. One of the major hurdles holding back high-performance navigation onboard robots is energy-efficient visual processing. Insects' visual system is event-based, where photosensitive cells react *independently* from each other to changes in light intensity and subsequently generate spikes that propagate through the visual system to be processed in their miniature brains. Event cameras are *neuromorphic* vision sensors that mimic that process. Here, pixels take the role of the photosensitive cells and generate events asynchronously. Visual information is thus captured in a stream of events opposed to synchronous frames as taken by traditional cameras. This allows neuromorphic cameras to operate at very high temporal resolution and low latency (in the order of microseconds), very high dynamic range (140dB compared to 60dB of standard cameras), high pixel bandwidth (in the order of kHz), and low power consumption (order of mW) [15]. Processing such information requires novel methods to be developed. Techniques for frame-based visual data generally make use of convolutional neural networks (CNNs) and some approaches have focused on bringing these methods to event-based data [28]. Other more biologically plausible methods involve the use of spiking neural networks (SNNs) since they are biologically more similar to networks of neurons found in animal nervous systems than regular artificial neural networks (ANNs). Implemented on neuromorphic processors such as Intel's Loihi and IBM's Truenorth, SNNs can deliver highly powerful computing at a fraction of the power budget of traditional hardware (CPUs, GPUs), making them promising candidates for implementation on robots. Recent neural insect-inspired navigation methods such as [22] show promising results for implementing different aspects of those methods for visual navigation to real-life challenges.

However, matching insects' capability to efficiently navigate over long distances remains a challenge. Datasets form an important part of training and evaluating such methods. Currently, there are several event-based vision datasets focusing on navigation, covering applications in visual odometry, depth reconstruction, and SLAM but little focusing on insect-inspired navigation. Images, events, optic flow, 3D camera motion, and scene depth in static scenes using a mobile robotic platform are provided in [4]. A large automotive dataset containing over 250000 human-labeled box annotations of cars and pedestrians is presented by [34]. The dataset provided by [38] includes synchronized stereo event data, augmented with grayscale images, and inertial measurement unit (IMU) readings. Ground truth pose and depth images are provided through a combination of a LiDAR system, IMU, indoor and outdoor motion capture, and GPS. The DDD20 [18] dataset consists of an extensive automotive dataset with events, frames, and vehicle human control data collected from 4000 kilometers of mixed highway and urban driving. However, most insects have compound eyes that cover an almost panoramic FOV and this plays an important role in insects' navigational dexterity [16]. Additionally, insects fuse various sensory inputs from their environment together during navigation [14], making datasets that combine sensors valuable sources for training and evaluating such methods. None of the datasets above provide event data captured through an omnidirectional lens enhanced with additional sensors over long distances.

This paper presents two main contributions. First, a dataset containing omnidirectional event camera and IMU data, forward-facing high-resolution footage, and centimeter-level accurate GPS data along with a software package to load, process and manipulate the dataset. The dataset, including the software package, will be made available at `https://github.com/tudelft/NavDataset`. Secondly, an evaluation of three different familiarity-based insect-inspired navigation models from the literature [2, 3, 39] with respect to their performance in long-ranged navigation is presented.

2 The biological principles in insect navigation

2.1 Visual Perception

Insect's compound eyes see in a lower resolution than human eyes but can be arranged to cover an almost panoramic visual field and due to their simpler nature have faster processing times. Compound eyes consist of small individual hexagonally-shaped photoreceptive units, called *ommatidia*, which are arranged to form a faceted surface. Each such ommatidium receives light only from a small angle (1° up to $\pm 20^\circ$ [24] depending on its location on the eye and species) in the visual field, constricting the visual *acuity* of the insect's visual system. When excited by photons, these photoreceptive cells generate electric signals encoding the amount of light it absorbs, which downstream neurons turn into *spikes* that are passed through to the underlying optic lobes [29].

2.2 Insect Navigation Models

Insects are adept navigators capable of maneuvering through cluttered environments and memorizing long routes, e.g. bees have been shown to retrace routes several kilometers in length [12]. As a result, biologists have looked at modeling insects' navigational capabilities to figure out how they realize these feats as well as to better understand their behavior. Cartwright and Collett's [9] seminal snapshot model presented some of the first work concerning honeybee navigation. It hypothesized that honeybees store a single retinotopic *snapshot* of the place that they later want to navigate back to. Other methods include the Average Landmark Vector (ALV) model [23], image warping [13], and rotationally invariant panoramic methods that utilize Fourier-transformed [33] images or other frequency-domain-based methods [32].

The area surrounding the stored snapshot from which agents can successfully return is categorized as the *catchment area*. The extent of the catchment area thus changes depending on various factors such as the deployed navigation technique and the complexity and texture of the environment [37]. In general, the root mean square difference between the stored panoramic snapshot and another panoramic snapshot (also called the *image difference*) changes smoothly in natural scenes in correlation with the distance from the stored snapshot, where it terminates in a sharp minimum [37]. This laid down the theoretical basis for so-called Descent in Image Difference (DID) methods, which follow the declining gradient in image difference towards the stored snapshot, demonstrated in a (still small) 5.5×8.25 environment [26]. Later methods looked at more biologically plausible implementations for processing the visual data through the implementation of ANNs. In the scene familiarity model [3], a route is learned through training a 2-layer feedforward network to memorize snapshots along the route. An SNN-based scene familiarity model, modeled after the mushroom bodies of ants was later formulated by [2].

2.3 Neuromorphic Processing

Event-based Vision Sensors Event cameras are vision sensors that take inspiration from the working principle of the biological retina. Each pixel reacts asynchronously to changes in light intensity. The sensor logs the pixel's location, time (in microsecond resolution), and polarity ('ON' or 'OFF'), and sends it over a digital bus in an Address-event Representation (AER) format [15]. Because event cameras only react to small *changes* in light intensity at individual pixels, visual information is more efficiently conveyed compared to frame-based cameras.

Spiking Neural Networks Analogous to their biological counterparts, artificial neurons in SNNs generate a spike (action potential) if their membrane potential reaches a certain threshold after receiving a series of excitatory spikes from upstream neurons over their synaptic connections. After firing, the neuron lowers its internal voltage to a resting state. For a short time (the so-called refractory period) the neuron will not react to any incoming signals anymore. Various computational models of biological neurons exist to replicate this behavior. The most used neuronal models in artificial spiking neural networks nowadays are the Leaky Integrate-and-Fire (LIF) [31], Spike Response Model (SRM) [21], and the Izhikevich [19] model. The LIF neuron model in Eq. 1 shows how presynaptic spikes $s_j(t)$ arriving from neurons in layer $l-1$ increase (or decrease) — depending on weight matrix $W_{i,j}$, denoting its synaptic connectivity — the neuron's membrane potential $v_i(t)$ (scaled with the time constant λ_v) after which it decays to its resting potential v_{rest} if no more signals arrive. If enough excitatory presynaptic spikes arrive in short succession, the membrane potential will reach a certain internal threshold after which the neuron spikes ($s_i(t)$), resets its membrane potential, and enters a refractory period. Inhibiting presynaptic spikes have the opposite effect and will lower the membrane potential.

$$\lambda_v \frac{dv_i(t)}{dt} = -(v_i(t) - v_{\text{rest}}) + \sum_{j=1}^{n^{l-1}} \left(W_{i,j} s_j^{l-1}(t - \tau_d) \right) \tag{1}$$

Information in SNNs can be encoded in several different manners including position, temporal, rate coding, and subsequent combinations thereof. Learning in SNNs thus takes place in these domains, traditionally with much focus on a mechanism called Spike-Timing-Dependent Plasticity (STDP) [8]. STDP is a (biological) form of Hebbian learning that changes the synaptic strength of neuron connections dependent on their relative spike timing. Learning through backpropagation — the backbone of modern machine learning in ANNs — is not (directly) possible due to the discontinuous nature of spikes. However, a great number of efforts over recent years have shown that back propagation-like algorithms could be successfully applied to SNNs, using a wide range of tricks (surrogate gradient [36], rate-based networks [25], and learning spike times [6] among others).

3 Dataset Design

The following section presents the utilized sensors and how the dataset was collected. The dataset was collected in both rural and urban environments in and around the TUDelft campus. It mainly consists of events and IMU readings captured by a DAVIS240 event camera and video from a GoPro Hero 8 Black along with RTK GNSS positioning data. Video was collected in HEVC encoded MP4 and ROS bag files for the rest. The dataset also provides the same raw data in HDF5 containers. Section 3.1 gives an overview of the dataset collection platform and the acquisition environment. A Python3 package will be made available for converting the bag and HDF5 files to and from various formats, as well as performing the data (pre)processing as elaborated upon in Section 3.2.

3.1 Sensors and Data Acquisition

Table 1: Dataset collected data

Sensor	Characteristics	Container
DAVIS240	240 × 180 pixel DVS AER	bag/HDF5
GoPro Hero 8 Black	1920 × 1080 pixel 60 Hz	mp4 (HEVC)
Ublox ZED-F9P GNSS module	NavPVT 5 Hz Position Accuracy 1.3 cm CEP	bag/HDF5
Intel Realsense d435i	720 × 1280 pixel depth 30 Hz 16UC1	bag

Sensors Table 1 provides an overview of the sensors with their characteristics. The complete logistical overview of the dataset acquisition platform can be seen in Fig. 1A and B. The dataset junction box forms the housing holding the various sensors as well as the Intel Up board computation platform. The Intel Up board runs ROS and is responsible for collecting and time synchronizing the data from the various sensors which are connected over USB2/3 buses. The GNSS antenna was mounted at the back of the bike to minimize interference from the USB3 controllers. A mobile phone with cellular was connected to the Intel Up board by connecting to the phone's wifi hotspot. This allowed for running commands on the Intel Up board over ssh as well as provided the Intel Up board with internet access. This was needed for RTK GNSS positioning; RTCM messages were sent to the ublox ZED-F9P GNSS receiver by connecting to the EUREF-IP network ntrip server allowing for up to centimeter-accurate positioning. The DAVIS240 camera was mounted to an omnidirectional catadioptric lens to achieve omnidirectional vision. The GoPro camera was manually operated, thus a small gap exists between its timing and the rest of the sensors; this has been manually compensated for in the post-processing of the data. An external portable SSD was utilized to offload the collected data after every single run as the internal storage of the Intel Up board was limited. The dataset box was then mounted to a bike (Fig. 1B).

Sequences The dataset consists of in total 12 routes traversed by bike from and to the start point, Fig. 2 gives an overview of the recorded routes. The runs cover both rural and urban environments in and around the TUDelft (Delft, The Netherlands) campus.

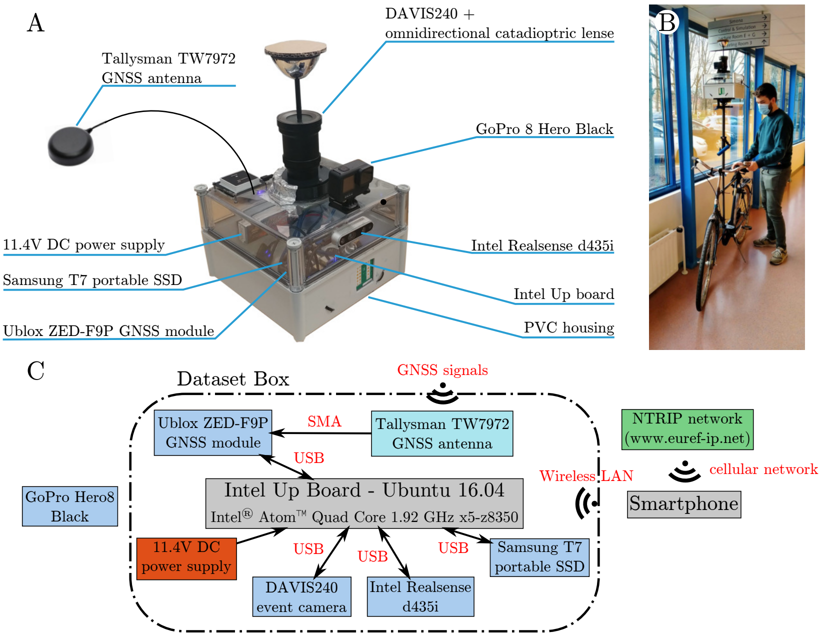


Fig. 1: Dataset acquisition hardware. **A** shows an overview of the various sensors mounted on the dataset box. **B** shows the full setup — the dataset box mounted on a bike to cover the long distances. **C** shows the data flow diagram between the sensors and the central computers.

3.2 Data Processing

Central to this dataset is the data provided by the DAVIS240 equipped with the omnidirectional catadioptric lens. As can be seen in Fig. 3, the omnidirectional lens projects its light on a circular region on the DAVIS240 sensor. In Fig. 3A the direction of the view with respect to the bike's heading is annotated. The approximate location of the capture can be seen in Fig. 3B. This circular projection can easily be masked as it stays fixed with respect to the sensor, subsequent unwrapping results in the view presented by Fig. 3C. These procedures along with a few denoising methods are also included in the software package released alongside the dataset.

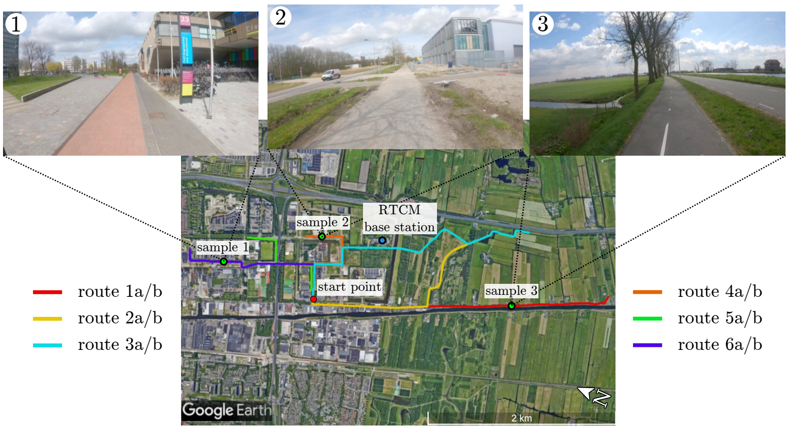


Fig. 2: Map of the routes covered by the dataset. Route ‘a’ indicates runs away from the start point, ‘b’ towards the start point. The routes cover both rural and urban environments as presented by samples 1–3.

4 Experimental study: Evaluating Familiarity-Based Neural Insect Navigation Models

4.1 Introduction

The following section presents the use of the dataset to investigate a few recent neural insect-inspired navigational models. The experiments compare three neural insect navigation models in terms of their performance for long-ranged navigation, namely [3]’s scene familiarity neural network, [2]’s Mushroom Body model, and [39]’s Mushroom Body model. The aforementioned models have been mostly tested in either simulated environments [2, 3] or over very short distances in a controlled environment [39]. This dataset provides an interesting testing ground to evaluate these methods in real-life conditions with visual data inspired by the way insects perceive their environment. The inclusion of frame-based video allows for comparison between frame-based [2, 3] and event-based [39] methods. We are specifically interested in how these methods hold up over longer distances.

4.2 Neural Familiarity-Based Insect Navigation Models

Baddeley et al.’s Scene Familiarity Model The (frame-based) scene familiarity model of [3] consists of an input layer with the same dimensions as the

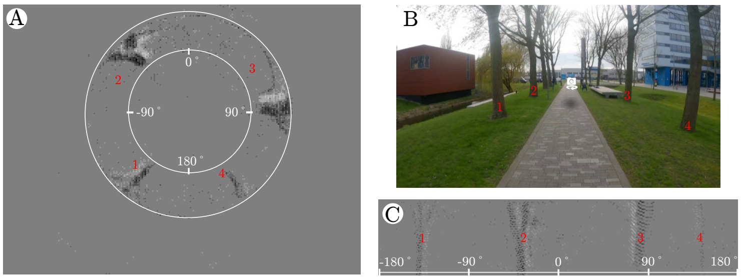


Fig. 3: Example of the omnidirectional events captured by the DAVIS240 and subsequent preprocessing. **A** shows an accumulated event frame as captured by the omnidirectional system, the two concentric circles show the boundaries of the visible field. **B** shows a frame from the GoPro footage, the left snapshot coincides with the camera position in the frame. **C** shows the event stream after masking and unwrapping the events.

number of pixels in the acquired images, which is fully connected by feedforward connections to a novelty layer which consists of tanh activation functions. The information about the input presented by the novelty layer is maximized through weight adaptation following the Information-Maximization (infomax) principle [5]. The infomax principle adjusts the weights of the network in such a way as to maximize the information about the input that the novelty layer presents. This is performed by following the gradient of mutual information; in [3] the natural gradient is utilized to save computation time. By maximization of information through weight adaptation, the output of the novelty layer units is decorrelated, effectively reducing the network's output for sequences that have already been seen. 'Familiar' frames can be discerned after a single training run.

Mushroom Body Models The MBs are relatively large structures in the insect brain that consist of large parallel arrangements of neurons, called Kenyon Cells (KCs), which are sampled by a relatively small amount of extrinsic neurons, also called Mushroom Body Output Neurons (MBONs). The mushroom bodies are known to play an important role in olfactory learning [17]. More recently, the mushroom bodies' role in visual learning has been investigated, revealing direct neural connections between the medulla and mushroom bodies [35]. Recent research has shown that MBs are *necessary* for learned visual navigation [7].

Ardin et al.'s Mushroom Body Model The (frame-based) SNN MB model presented by [2] consists of 360 visual projection neurons (vPNs) that are sparsely connected to 20000 Kenyon cells which connect to a single MBON. Each such vPN can thus activate only a handful of KCs, representing a sparse encoding of information. Learning is performed by lowering the synaptic weights (Long-Term Depression (LTD)) between the KCs and the MBON through STDP. As a result, after training, the MBON's spike rate is lower for familiar views opposed to novel ones. Only LTD is applied, in effect permanently weakening the neurons connection. This limits the model's capacity to memorize long sequences to the amount of depletable weights.

Zhu et al.'s Mushroom Body Model The model presented by [39] (event-based) adapts Ardin et al.'s [2] model based on the finding that 60% of the input synapses of KCs come from other KCs. Instead of performing learning on the weights connecting the KCs to the MBON; when a KC spikes, it inhibits its connection to downstream KCs that spike at a later time based on an STDP rule. The KCs are split up into two groups of 5000 neurons to speed up learning; each solely acting within its group. Additionally, an anterior paired lateral (APL) [1] neuron is included that inhibits the activity of the KC layer.

4.3 Setup

The aforementioned neural networks were trained on sequences of 8, 16, 24, and 32 seconds from a 32-second section of route 1a (see Fig. 2). The respective networks were trained to 'memorize' that stretch of the route. Ensuing runs would result in a lower response from the network for already seen sequences, compared to unseen sequences. During the validation run, the same sequence was injected with unfamiliar sequences over stretches of 4 seconds and presented to the networks. The injected parts were sampled from sections of other routes in the dataset. For the event-based networks, the injected sequences were closely matched to the event rate of the original sequence, to maintain similar levels of activity in the network's layers. The frame-based models were presented with 28 × 8 pixel greyscale histogram equalized frames flattened to a 1D array. Input from the DAVIS240 sensor was max-pooled to 32 × 7 pixels before passing it to the network.

4.4 Results

The results of the experiment can be seen in Fig. 4. Inspired by [39]'s novelty index, we apply a performance index P

$$P = \frac{\sum s_{\text{unfamiliar}} - \sum s_{\text{familiar}}}{s_{\text{total}}} \tag{2}$$

with s the response of the network, to evaluate the model's performance over increasingly longer test sequences, visualized in Fig. 4D. The Infomax scene familiarity [3] model's performance index decreased overall, while Ardin et al.'s [2] stabilized, but both these frame-based models maintain an adequate performance level to still separate familiar from unfamiliar views. This is in stark contrast with Zhu et al's [39] event-based model, which has depleted its 'memory' after about 16 seconds of learning. This deteriorating performance over longer distances is a result of the limited capacity of the networks, as synaptic connections' weights are depleted during training. This could be improved upon by a number of factors. First, the networks were trained with a constant learning rate, this could be tuned for longer distances, although at the cost of lower performance in general. Secondly, one could lower the number of presented frames from 60Hz to lower rates based on some metric of the input data. Further increases in

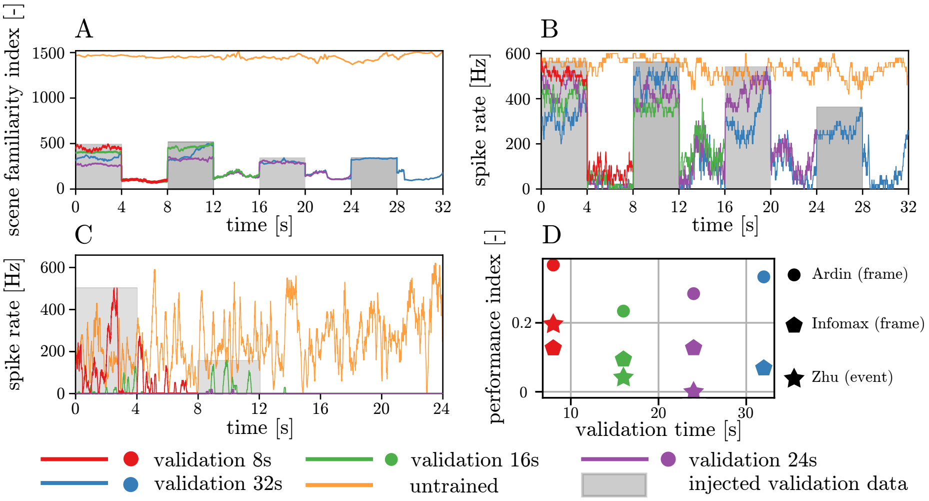


Fig. 4: Insect navigation models response to 8-, 16-, 24-, and 32-second sequences of a section of route 1a. **A** Infomax neural network [3] (frame-based). **B** Ardin et al.'s [2] MB model (frame-based). **C** Zhu et al.'s [39] MB model (event-based). **D** Performance index P (Eq. 2)

the network's size by increasing the amount of KCs remains an option as well, although its computational increase would severely limit the number of deployable robotic platforms. The frame-based method's more stable performance could be a consequence of them having more control over their input through well-established techniques such as normalization, which are less developed for event-based vision. Investigating intrinsically modulating mechanisms such as [27]'s adaptive LIF neuron could perhaps provide more fundamental solutions for this. Furthermore, it is known that insects perform a number of preprocessing steps (including elementary motion detection such as optic flow) in their optic lobes [29] as well as have mechanisms present that adjust the learning and forgetting of 'unnecessary' information [14], worth investigating.

5 Conclusion

The current insect-inspired visual navigation methods still come short compared to their biological counterparts, especially regarding navigating over long distances. This work aims to provide a valuable tool with which further development of neural insect-inspired long-range navigation methods can be accelerated.

References

1. Amin, H., Apostolopoulou, A.A., Suárez-Grimalt, R., Vrontou, E., Lin, A.C.: Localized inhibition in the Drosophila mushroom body. eLife **9** (sep 2020). `https://doi.org/10.7554/eLife.56954`

2. Ardin, P., Peng, F., Mangan, M., Lagogiannis, K., Webb, B.: Using an Insect Mushroom Body Circuit to Encode Route Memory in Complex Natural Environments. PLOS Computational Biology **12**(2), e1004683 (feb 2016). `https://doi.org/10.1371/journal.pcbi.1004683`
3. Baddeley, B., Graham, P., Husbands, P., Philippides, A.: A Model of Ant Route Navigation Driven by Scene Familiarity. PLoS Computational Biology **8**(1), e1002336 (jan 2012). `https://doi.org/10.1371/journal.pcbi.1002336`
4. Barranco, F., Fermuller, C., Aloimonos, Y., Delbruck, T.: A dataset for visual navigation with neuromorphic methods. Frontiers in Neuroscience **10**(FEB), 49 (feb 2016). `https://doi.org/10.3389/fnins.2016.00049`
5. Bell, A.J., Sejnowski, T.J.: An Information-Maximization Approach to Blind Separation and Blind Deconvolution. Neural Computation **7**(6), 1129–1159 (nov 1995). `https://doi.org/10.1162/neco.1995.7.6.1129`
6. Bohte, S.M., Kok, J.N., La Poutré, H.: Error-backpropagation in temporally encoded networks of spiking neurons. Neurocomputing **48**(1-4), 17–37 (oct 2002). `https://doi.org/10.1016/S0925-2312(01)00658-0`
7. Buehlmann, C., Wozniak, B., Goulard, R., Webb, B., Graham, P., Niven, J.E.: Mushroom Bodies Are Required for Learned Visual Navigation, but Not for Innate Visual Behavior, in Ants. Current Biology **30**(17), 3438–3443.e2 (sep 2020). `https://doi.org/10.1016/j.cub.2020.07.013`
8. Caporale, N., Dan, Y.: Spike Timing–Dependent Plasticity: A Hebbian Learning Rule. Annual Review of Neuroscience **31**(1), 25–46 (jul 2008). `https://doi.org/10.1146/annurev.neuro.31.060407.125639`
9. Cartwright, B.A., Collett, T.S.: Landmark learning in bees - Experiments and models. Journal of Comparative Physiology **151**(4), 521–543 (1983). `https://doi.org/10.1007/BF00605469`
10. de Croon, G., de Clercq, K., Ruijsink, R., Remes, B., de Wagter, C.: Design, Aerodynamics, and Vision-Based Control of the DelFly. International Journal of Micro Air Vehicles **1**(2), 71–97 (jun 2009). `https://doi.org/10.1260/175682909789498288`
11. Dupeyroux, J., Serres, J.R., Viollet, S.: Antbot: A six-legged walking robot able to home like desert ants in outdoor environments. Science Robotics **4**(27), eaau0307 (2019)
12. Eckert, J.E.: The flight range of the honeybee. Journal of Agricultural Research **47**(5) (1933)
13. Franz, M.O., Schölkopf, B., Mallot, H.A., Bülthoff, H.H.: Where did I take that snapshot? Scene-based homing by image matching. Biological Cybernetics **79**(3), 191–202 (oct 1998). `https://doi.org/10.1007/s004220050470`
14. Freas, C.A., Schultheiss, P.: How to Navigate in Different Environments and Situations: Lessons From Ants. Frontiers in Psychology **9**(MAY), 1–7 (may 2018). `https://doi.org/10.3389/fpsyg.2018.00841`
15. Gallego, G., Delbrück, T., Orchard, G., Bartolozzi, C., Taba, B., Censi, A., Leutenegger, S., Davison, A.J., Conradt, J., Daniilidis, K., Scaramuzza, D.: Event-based vision: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(1), 154–180 (2022). `https://doi.org/10.1109/TPAMI.2020.3008413`
16. Graham, P., Philippides, A.: Vision for navigation: What can we learn from ants? Arthropod Structure & Development **46**(5), 718–722 (2017). `https://doi.org/https://doi.org/10.1016/j.asd.2017.07.001`
17. Heisenberg, M., Borst, A., Wagner, S., Byers, D.: Drosophila Mushroom Body Mutants are Deficient in Olfactory Learning. Journal of Neurogenetics **2**(1), 1–30 (jan 1985). `https://doi.org/10.3109/01677068509100140`

18. Hu, Y., Binas, J., Neil, D., Liu, S.C., Delbruck, T.: DDD20 End-to-End Event Camera Driving Dataset: Fusing Frames and Events with Deep Learning for Improved Steering Prediction. In: 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC). pp. 1–6. IEEE (sep 2020). https://doi.org/10.1109/ITSC45102.2020.9294515
19. Izhikevich, E.: Simple model of spiking neurons. IEEE Transactions on Neural Networks **14**(6), 1569–1572 (nov 2003). https://doi.org/10.1109/TNN.2003.820440
20. Jafferis, N.T., Helbling, E.F., Karpelson, M., Wood, R.J.: Untethered flight of an insect-sized flapping-wing microscale aerial vehicle. Nature 2019 570:7762 **570**(7762), 491–495 (jun 2019). https://doi.org/10.1038/s41586-019-1322-0
21. Kistler, W.M., Gerstner, W., Hemmen, J.L.v.: Reduction of the Hodgkin-Huxley Equations to a Single-Variable Threshold Model. Neural Computation **9**(5), 1015–1045 (07 1997). https://doi.org/10.1162/neco.1997.9.5.1015
22. Knight, J.C., Sakhapov, D., Domcsek, N., Dewar, A.D., Graham, P., Nowotny, T., Philippides, A.: Insect-Inspired Visual Navigation On-Board an Autonomous Robot: Real-World Routes Encoded in a Single Layer Network. In: The 2019 Conference on Artificial Life. pp. 60–67. MIT Press, Cambridge, MA (2019). https://doi.org/10.1162/isal_a_00141.xml
23. Lambrinos, D., Möller, R., Pfeifer, R., Wehner, R.: Landmark Navigation without Snapshots: the Average Landmark Vector Model. In: Elsner, N., Wehner, R. (eds.) Proc. Neurobiol. Conf. Göttingen. p. 30a. Georg Thieme Verlag (1998)
24. Land, M.F.: Visual acuity in insects. Annual Review of Entomology **42**(1), 147–177 (jan 1997)
25. Lee, J.H., Delbruck, T., Pfeiffer, M.: Training deep spiking neural networks using backpropagation. Frontiers in Neuroscience **10**(NOV), 508 (2016). https://doi.org/10.3389/FNINS.2016.00508/BIBTEX
26. Möller, R., Vardy, A.: Local visual homing by matched-filter descent in image distances. Biological Cybernetics **95**(5), 413–430 (oct 2006). https://doi.org/10.1007/s00422-006-0095-3
27. Paredes-Valles, F., Scheper, K.Y.W., De Croon, G.C.H.E.: Unsupervised Learning of a Hierarchical Spiking Neural Network for Optical Flow Estimation: From Events to Global Motion Perception. IEEE Transactions on Pattern Analysis and Machine Intelligence **8828**(c), 1–1 (2019). https://doi.org/10.1109/TPAMI.2019.2903179
28. Rueckauer, B., Lungu, I.A., Hu, Y., Pfeiffer, M., Liu, S.C.: Conversion of Continuous-Valued Deep Networks to Efficient Event-Driven Networks for Image Classification. Frontiers in Neuroscience **11**(DEC), 682 (dec 2017). https://doi.org/10.3389/fnins.2017.00682
29. Sanes, J.R., Zipursky, S.L.: Design Principles of Insect and Vertebrate Visual Systems. Neuron **66**(1), 15–36 (apr 2010). https://doi.org/10.1016/j.neuron.2010.01.018
30. Scheper, K.Y., Karasek, M., De Wagter, C., Remes, B.D., De Croon, G.C.: First Autonomous Multi-Room Exploration with an Insect-Inspired Flapping Wing Vehicle. In: 2018 IEEE International Conference on Robotics and Automation (ICRA). pp. 5546–5552. IEEE (may 2018). https://doi.org/10.1109/ICRA.2018.8460702
31. Stein, R.B.: A theoretical analysis of neuronal variability. Biophysical Journal **5**(2), 173–194 (1965). https://doi.org/https://doi.org/10.1016/S0006-3495(65)86709-1
32. Stone, T., Mangan, M., Wystrach, A., Webb, B.: Rotation invariant visual processing for spatial memory in insects. Interface Focus **8**(4), 20180010 (aug 2018). https://doi.org/10.1098/rsfs.2018.0010

33. Stürzl, W., Mallot, H.: Efficient visual homing based on Fourier transformed panoramic images. Robotics and Autonomous Systems **54**(4), 300–313 (apr 2006). `https://doi.org/10.1016/j.robot.2005.12.001`
34. de Tournemire, P., Nitti, D., Perot, E., Migliore, D., Sironi, A.: A Large Scale Event-based Detection Dataset for Automotive. arXiv preprint (jan 2020)
35. Vogt, K., Aso, Y., Hige, T., Knapek, S., Ichinose, T., Friedrich, A.B., Turner, G.C., Rubin, G.M., Tanimoto, H.: Direct neural pathways convey distinct visual information to Drosophila mushroom bodies. eLife **5**(APRIL2016), 1–13 (apr 2016). `https://doi.org/10.7554/eLife.14009`
36. Wu, Y., Deng, L., Li, G., Zhu, J., Shi, L.: Spatio-temporal backpropagation for training high-performance spiking neural networks. Frontiers in Neuroscience **12**(MAY), 331 (may 2018). `https://doi.org/10.3389/FNINS.2018.00331/BIBTEX`
37. Zeil, J., Hofmann, M.I., Chahl, J.S.: Catchment areas of panoramic snapshots in outdoor scenes. Journal of the Optical Society of America A **20**(3), 450 (mar 2003). `https://doi.org/10.1364/JOSAA.20.000450`
38. Zhu, A.Z., Thakur, D., Ozaslan, T., Pfrommer, B., Kumar, V., Daniilidis, K.: The Multivehicle Stereo Event Camera Dataset: An Event Camera Dataset for 3D Perception. IEEE Robotics and Automation Letters **3**(3), 2032–2039 (jul 2018). `https://doi.org/10.1109/LRA.2018.2800793`
39. Zhu, L., Mangan, M., Webb, B.: SPATIO-temporal memory for navigation in a mushroom body model. In: Lecture Notes in Computer Science, vol. 12413 LNAI, pp. 415–426. Springer International Publishing (2021). `https://doi.org/10.1007/978-3-030-64313-3_39`

Part II

Literature Study

2

Motivation and Scope

The goal of the literature study is to explore and get a broader understanding of the relevant literature that covers the scope of this thesis. Towards this purpose, the following motivation and goals had been initially set at the start of the literature study, which evidently changed over the course of this thesis (see Chapter 1).

The ultimate goal of this thesis was to theorize/adapt a novel neural event-based visual insect-inspired navigation model for route following and evaluate its performance for long-ranged (more than 100 meters) navigation. From which the central question arises:

How can a neural-based, insect-inspired guidance model, relying on event-based vision, be developed to achieve parsimonious visual guidance over long distances (> 100m) onboard a micro air vehicle?

As the method will concern itself with one-shot learning of previously traversed routes, it will be of interest which learning architectures lend themselves for such purpose. Another requirement is that the method is able to run onboard a MAV in realtime for it to be usable, this computational parsimoniousness is of importance. Finally, when a method has been developed it would be of interest to investigate its capabilities in route following, and subsequently investigate how large of an area it can 'represent'. These are distilled into the following sub-questions:

- **Which neural network architecture, interfacing with event-based vision, combined with a guidance policy allows for one-shot learning of visual routes in novel natural scenes?**
- **What are the minimum network requirements for achieving this, and can this be run in realtime on board a micro air vehicle?**
- **Given the neural model, what are the limitations with respect to the amount of route information that can be stored, in other words, how big of an area can it represent?**

Towards the purpose of answering these questions, a study comprising relevant and state-of-the-art literature is therefore presented in the following chapters. Chapter 3 presents an overview of local vision-based mapless guidance methods with a focus on insect-inspired navigation. Chapter 4 dives into insect physiology and behavior related to visual navigation. Event vision basics and processing methods are covered in Chapter 5. Finally, Chapter 6 presents neural-based visual guidance methods. The findings of the literature study are then summarized in Chapter 7.

3

Local Vision-Based Guidance

In this Chapter, navigation methods focusing on MAV monocular visual(-inertial) navigation without the need for prior maps of the environment will be reviewed. Map-building-based (SLAM) navigation will be briefly discussed in Section 3.1. Mapless navigation models, including insect-inspired models, will be reviewed in Section 3.2. Finally, studies on the catchment area of visual snapshots will be reviewed in Section 3.2.3.

3.1. Visual(-Inertial) Odometry and SLAM

SLAM has steadily become the de facto technology for navigation in unknown/cluttered environments where one can not rely on GPS data. Early SLAM methods were filter-based (e.g. Kalman filter), where localization and mapping are done simultaneously. A filter estimates the camera pose along with the state of all the landmarks that are detected in its environment. This clearly becomes a problem when covering larger areas or when tracking a lot of features at the same time. Therefore, the introduction of keyframe-based methods was made. Keyframe-based methods separate the task of SLAM up in two parts:

1. Camera pose estimation is performed on every frame and uses a subset of the map.
2. A map of the environment is maintained but only updated on keyframes. Keyframes are tagged based on criteria like: significant change of pose measurements, detection of a certain amount of previously unseen features, time elapsed, ...The map is updated using techniques like pose graph optimization and local/global bundle adjustment.

Many of the latest SLAM methods are keyframe-based. One can generally split SLAM methods into indirect and direct methods, and methods that form a hybrid between the two (Younes et al., 2017; Huang, 2019). Indirect methods extract and track features from the environment while direct methods act directly on raw pixel values.

3.1.1. Indirect Methods

Indirect or feature-based methods are more robust and mature than the direct methods. They make use of features. Feature detectors try to find features in a scene that are invariant to illumination and viewpoint changes and can deal with noise and motion blur. Commonly used feature extractors include but are not limited to the Harris detector, Shi-Tomasi corners, Difference of Gaussian, Features from Accelerated Segment Test (FAST), Adaptive and Generic corner detection based on the Accelerated Segment Test (AGAST) and Optimal Accelerated Segment Test (OAST). Trade-offs have to be made between robustness, computational cost and speed—here typically, not all can be met and depending on the platform and utilization different methods have to be used. After the extraction of features in a scene, different feature descriptors are utilized, of which Binary Robust independent Elementary Features (BRIEF), Speeded Up Robust Features (SURF), Scale-Invariant Feature Transform (SIFT) and Oriented FAST and Rotated Brief (ORB) are some of the most commonly used. They again have differences in speed and robustness to changes in rotation and scale (Krig and Krig, 2014). A major overhead in the real-time applicability of feature-based SLAM is the detection and extraction of features. An advantage of indirect methods is that they are more robust to larger baseline movements in between frames. The main difficulty nowadays lays in finding the right trade-off between robustness, computational efficiency and speed for the right application.

3.1.2. Direct Methods

Direct methods act directly on individual pixels of frames. Dense methods utilize all the pixel in the frame whereas semi-dense methods only use those pixels for which a significant image brightness gradient exists. They work by minimizing the intensity error between camera frames and utilizes as a basis the brightness consistency constraint:

$$J(x, y, t) = I(x + u(x, y), y + v(x, y), t + 1) \tag{3.1}$$

where x and y are the coordinates of the pixel on the image and u and v are the displacement functions that signify the movement of a pixel (x, y) from image I to J and t is time. The brightness consistency constraint assumes that any object in a scene will not see a significant change of illumination when seen from a different point of view. The main advantage of using direct methods is that they can also track movement even in low-textured environments and are able to deal with motion blur—cases in which indirect methods struggle to find features. However, for the brightness consistency constraint to hold, a good state initialization and high frame rate are required and measures have to be taken to overcome changes in scene illumination. Furthermore, the calculation of the photometric error for each individual pixel is expensive and real-time computation has only recently become plausible due to advancements in parallel processing and the introduction of semi-dense inverse depth filtering (Younes et al., 2017).

3.2. Mapless Navigation

When close enough to a goal location, one can visually navigate to that location by comparing an image, or a holistic representation of the image taken at that location, with the current view and relating that to a direction of travel. This can be utilized for homing to a single location, but also multiple snapshot can be used to stitch a route together, which is then visually followed. Or, as proposed by (Baddeley et al., 2012), one could simply follow the most familiar route. Following (Möller and Vardy, 2006), mapless navigation methods that solely rely on monocular visual intensity information can be divided into two categories: correspondence and holistic methods.

3.2.1. Correspondence Methods

Correspondence methods try to match regions in the stored snapshot with the current view by computing a vector that would match the transformation of a region from the current view to a matched region in the stored snapshot. Multiple regions can be matched to give an estimate for the direction one should navigate to, to end up at the goal location. Correspondence methods can be subdivided into (Möller and Vardy, 2006):

- differential flow methods, e.g. (Vardy and Moller, 2005)
- matching methods
  - without feature preselection, e.g. block, intensity and gradient matching ((Argyros et al., 2005))
  - with feature preselection, e.g. Cartwright and T. S. Collett (1983)'s original snapshot model

Differential flow methods use Taylor approximations of the intensity or gradient correspondence functions between frames to compute the direction of travel. The original snapshot model of bee navigation proposed by Cartwright and T. S. Collett (1983), uses pre-programmed matches with dark and bright regions in the (controlled) environment to compute a desired homing vector. Other methods with feature preselection use e.g. edges, blobs ...to determine a guidance direction. Methods without feature preselection match single pixels or blocks of pixels between frames, and originate from optic flow methods. Correspondence methods have been successfully used in applications ranging from mobile cleaning robots (Vardy and Moller, 2005) to visual road navigation (Pink et al., 2009). Many early models of insect-inspired visual navigation are essentially correspondence methods, where controlled experiments were set up to serve as a test bed for the model of insect navigation in question. Correspondence methods have not recently seen much development in applications for direct visual guidance. Visually extracted features are however extensively used in SLAM methods as discussed in Section 3.2.

The Snapshot Model

The seminal work presented by Cartwright and T. S. Collett (1983) has formed the basis for much of the research in insect navigation models. In their experiments, they trained bees to navigate to a certain location in a room designated by one or more landmarks which were then subsequently moved to observe their behavior. Based on the reaction of the bees a couple of hypotheses were put forward and tested for plausibility in the snapshot model, namely:

- Bees search in the expected location of the food
- Bees learn the apparent size of a single landmark
- Bees see edges
- Bees notice horizontal and vertical extents

Cartwright and Collett's Snapshot Model extracts predefined features (edges) from the projected landmarks on the retina and matches them with the closest ones stored in the snapshot. The guidance vectors that minimize the mismatch of these edges, are used to guide the agent to move in that direction as to finally match the retinal view with the snapshot and end up in the right position. However, this method only worked robustly when an external compass would orient the view in the same direction as the original snapshot and required filtering out distant landmarks.

3.2.2. Holistic Methods

In contrast to correspondence methods, holistic methods make use of holistic representations of an image and do not work on matched regions between images. Holistic methods can be subdivided into (Möller and Vardy, 2006):

- image warping methods (Franz et al., 1998; Möller, Krzykawski, et al., 2010; Möller, 2012)
- parameter methods, e.g. Average Landmark Vector (ALV) (Lambrinos et al., 1998), Fourier-amplitude model (W. Stürzl and Mallot, 2006) and scene familiarity (Baddeley et al., 2012)
- descend in image distance methods (Möller and Vardy, 2006)

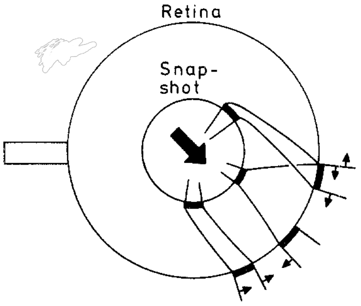


Figure 3.1: Landmark feature matching in the original Snapshot Model. Edges on the snapshot are matched to the closest ones present on the retina. Vectors point in the direction that would minimize the difference in bearing of the edges. Vectors are summed to compute a heading. Retrieved from Cartwright and T. S. Collett (1983).

Many insect-inspired guidance models find themselves in the group of parameter methods. Parameter methods reduce images to a parametric description. This parametric description can then be used to derive a guidance vector.

The Average Landmark Model

The ALV model as proposed in (Lambrinos et al., 1998) and later demonstrated in Dimitrios Lambrinos et al. (2000) starts from the notion that it is not necessary to physically store a snapshot as proposed in (Cartwright and T. S. Collett, 1983), but rather that the representation of a vector that is the summation of all vectors pointing to landmarks (the ALV) suffices to perform successful visual homing. Only the ALV at the home location is stored and ALVs at other locations can be computed and compared to the original ALV and subsequently a homing vector can be computed minimizing the difference in ALVs. The ALV model requires the extraction of features (e.g. edges, corners, ...) as a representation for landmarks and additionally an external compass to keep track of the original direction of the ALV. The external compass required for ALV navigation can be provided by polarized skylight (R. Wehner and Räber, 1979; Labhart and Meyer, 1999; Zeil, W. A. Ribi, et al., 2014; Dupeyroux et al., 2019). This methodology was later applied on a gantry robot in (Smith et al., 2007). Opposed to using a single ALV at the goal location, several ALVs were 'linked' together to form a network of connected ALVs: Linked Local Navigation (LLN). New ALVs were stored when the number of perceived landmarks changed compared to the previous time-step. The environment was a small 3 m x 2 m x 2 m volume with black/dark-gray cardboard tubes scattered over the area, which allowed for simple extraction of landmarks. The LLN framework showed better performance over larger distances than (Dimitrios Lambrinos et al., 2000) and allowed for the construction of indirect routes to the goal. These methods operated in a fairly simple environment with predetermined features however.

Image Warping Methods

Image warping methods try to transform the current image through a simulated transformation in such a way as to resemble the image at the goal location. The transformation that best resembles (e.g. euclidean distance of pixel values) the stored image is used for the computation of the guidance maneuver. The original image warping method was introduced by (Franz et al., 1998), which used a 1D representation of panoramic images to derive a guidance vector. The main assumption that allowed for the cultivation of this method is that all landmarks are at the same distance from the snapshot location. Nonetheless, this methodology performed well even in environments that violated this assumption. This methodology was subsequently used as a basis for further developments in visual homing and was later extended to 2D imagery by Möller, Krzykawski, et al. (2010) which allowed to account for the difference in elevation height of landmarks on the retina when moving through the scene. In (Möller, Krzykawski, et al., 2010), instead of assuming all landmarks to be at an equal distance from the snapshot, it is assumed that all pixels within a single columns are at the same distance. Additionally, the addition of compass cues to accelerate the search and coarse-to-fine search with dilation was investigated for increased performance. The authors noted that the method is computationally cheap enough for application on domestic cleaning robots. Extending this methodology for application in 3D navigation is limited, as computational requirements will be considerably higher. As with many methods based on the original snapshot theory, image warping suffers from occlusion of landmarks and changes in illumination. This method is unlikely to be used by insects, as mental warping of images requires considerable computation power and is unlikely to be present in the simple brain of insects, therefore a biologically plausible method was proposed in (Möller, 2012) using visual prediction. Möller (2012) takes inspiration from the observation that ants collect multiple snapshots in different orientations at the nest location and assumes that they derive a homing direction by predicting how the current view would change under a certain translatory movement. This predicted view is compared with all stored snapshots, for which the best matching heading is found in each snapshot, and subsequently the heading direction is chosen as the best matched heading over all snapshots. The image distortion prediction is performed by a predictor network with only local connections to sustain biological plausibility. This model reaches very good performance in textured indoor environments but fails in environments with easily discernible landmarks.

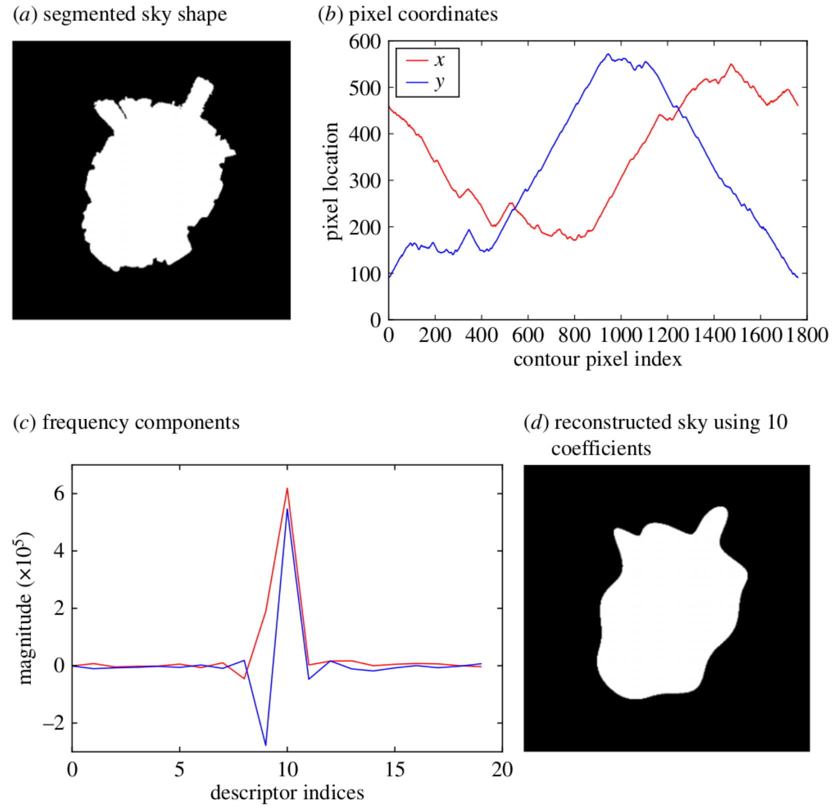
Rotation Invariant Panoramic Guidance

Another methodology is to use rotation invariant representations for images, eliminating the need for an external compass reference. In W. Stürzl and Mallot (2006), the model from (Franz et al., 1998) is transformed into Fourier space. Both use an omnidirectional 1D strip of grayscale values. Phase components of the Fourier transformed 1D strip were subsequently used to estimate the difference in rotation between the goal snapshot and the current view. Lower frequencies components were useful in more robust, long-ranged homing but lacked precision compared to the use of higher frequency components. They therefore proposed a coarse-to-fine strategy, where only low frequency components were used initially, and higher frequency components were added when homing failed. In Stone et al. (2018), Zenrike moments were used to describe and match skyline segmented images, an important cue for insect navigation (Differt and Möller, 2015), of the goal location and the current view. Panoramic snapshots of the skyline were wrapped to form a stitched-together image of the skyline as shown in Figure 3.2.

A larger catchment area was observed when using frequency and Zenrike moment encoded images compared to pure retinotopic matching (image difference) and showed robustness to the orientation in which the snapshots were taken. Zenrike moments and Fourier-based transforms of panoramic snapshots still suffer when variance in roll and pitch is present when capturing snapshots.

The Scene Familiarity Model

The work by Baddeley et al. (2012) presented a shift in insect-inspired models of visual guidance and propose a scene familiarity model for desert ant route navigation. The scene familiarity hypothesis stems from the notion that views that were previously seen on route are more familiar than other views and by following a scanning routine and subsequently moving in the direction that seems most familiar, successful route-following and homing can be achieved. The familiarity of a route is determined by pixel-wise comparison of all the scenes in memory with the current view; later an artificial neural network (Infomax) was trained to perform familiarity discrimination. The scene familiarity model provides a proof-of-concept that an Artificial Neural Network can learn multiple independent routes without the need for odometry, compass cues or dividing the route into multiple waypoints after a single training run. The Infomax model for discriminating familiarity is depicted in Figure 3.3. The Infomax model of (Baddeley et al., 2012) consists of an input layer with the same dimension as the amount of pixels in the acquired images, which is fully connected by feedforward connections with a novelty layer which consists of tanh() activation functions. The amount of novelty functions in the novelty layer can be chosen arbitrarily, although studies of visual pathways in honeybees suggest that routes are sparsely encoded from a small number of neurons in the antennal lobes to a large amount of Kenyon Cells (KCs) in the MBs, which are thought to play a role in visual route memory (see Section 4.2). Weights are


(*a*) segmented sky shape

(*b*) pixel coordinates

(*c*) frequency components

(*d*) reconstructed sky using 10 coefficients

Figure 3.2: Fourier descriptors used for description of the skyline, retrieved from Stone et al. (2018). (a) Segmented skyline as a closed shape. (b) Image coordinates of the shape of the edge as a function of x and y. (c) Frequency components of the functions in (b). (d) Reconstructed sky shape using the first 10 Fourier descriptors.

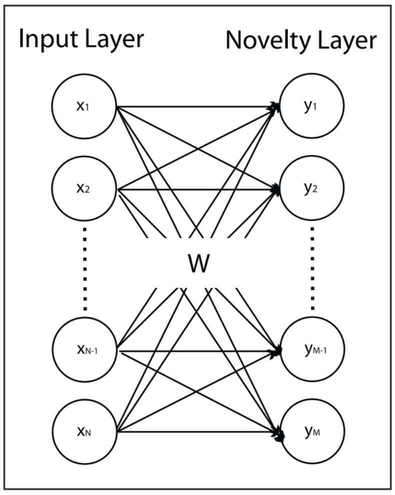


Figure 3.3: The Infomax neural network, retrieved from Baddeley et al. (2012).

initialized over the interval $[-0.5, 0.5]$ and normalized to a distribution with mean 0 and standard deviation 1. The network is subsequently trained using the Information-Maximization (Infomax) approach presented in (Bell and Sejnowski, 1995). The Infomax principle adjusts the weights of the network in such a way as to maximize the information about the input that the novelty layer presents. This is performed by following the gradient of mutual information; in Baddeley et al. (2012) the natural gradient is utilized to save computation time. By maximization of information through weight adaptation, the output of the novelty layer units are decorrelated. After a single pass of all the collected views a measure of familiarity can already be discerned. The M novelty unit inputs h_i are computed as:

$$h_i = \sum_{j=1}^{N} w_{ij} x_j \tag{3.2}$$

where w_{ij} are the weights of the feedforward connections, x_j the j th input and N the number of input units. The output of the novelty units, y_i is subsequently computed by:

$$y_i = \tanh(h_i) \tag{3.3}$$

The following learning rule is applied for weight adaptation:

$$\Delta w_{ij} = \frac{\eta}{N}(w_{ij} - (y_i + h_i) \sum_{k=1}^{N} h_k w_{kj}) \tag{3.4}$$

where η is the learning rate. The output of the Infomax network is computed by:

$$d(\boldsymbol{x}) = \sum_{i=1}^{M} |h_i| \tag{3.5}$$

Depending on the application, a threshold can be set on the value for $d(\boldsymbol{x})$ to discriminate between new and familiar views. In (Baddeley et al., 2012), the most familiar view is used, and thus no threshold had to be set.

Using a Model of the Mushroom Bodies to Encode Route Memory

Although realistic ant-like behavior was observed in the work by Baddeley et al. (2012) and a biologically plausible learning rule for Infomax was derived in Hayakawa et al. (2014), the Infomax neural network likely does not have physical resemblance to the actual neural processing in the insect brain. Therefore, Ardin et al. (2016) proposed an insect visual guidance model utilizing the Mushroom Bodies as model for the neural architecture (see Section 4.2), possibly forming the neural substrate for the Scene Familiarity model of Baddeley et al. (2012). The model was tested in simulation in an ant-like environment in which an almost panoramic (296×76 degrees) view of 10×36—after inverting the intensity and histogram equalizing the original 19×74 view—is provided as an input to the MB network. This was performed in such a way as to closely resemble the visual perception field of the ant *Melophorus bagoti*. The images are subsequently sampled by 360 Visual Projection Neurons (vPNs), after which each KC of the 20000 KCs receives input from 10 random vPNs. This entails a sparse encoding of visual stimuli to the MB. The 20000 KCs all branch onto a single Extrinsic Neuron. Using the principle of STDP, synapses are tagged and their strength is permanently decreased such that previously seen views will no longer activate the Extrinsic Neuron (EN). The activity of the EN can thus be seen as a sense of the novelty of a newly presented view. The network presented in (Ardin et al., 2016) used 20000 KCs and was able to store 375 views before confusion between novel and 'stored' images occurred. This translates to a route of about 37.5 m when taking snapshots every 10 cm. The honeybee, which possesses about 200000 KCs could potentially store much more using an analogous method. Other ways to increase capacity would be through (Ardin et al., 2016):

- additional ENs
- more states per synapse
- probabilistic instead of deterministic synapse switching
- preprocessing images

Theoretically and through empirical testing (Ardin et al., 2016), it was determined that for the current network architecture an error rate of $P_{\text{error}} = 0.01$ when 'storing' more than 375 images should be expected. When taking a view at the spike response of the network when storing more images (Figure 3.5), one can still clearly distinguish, based on the spikes, off-route views from on-route views.

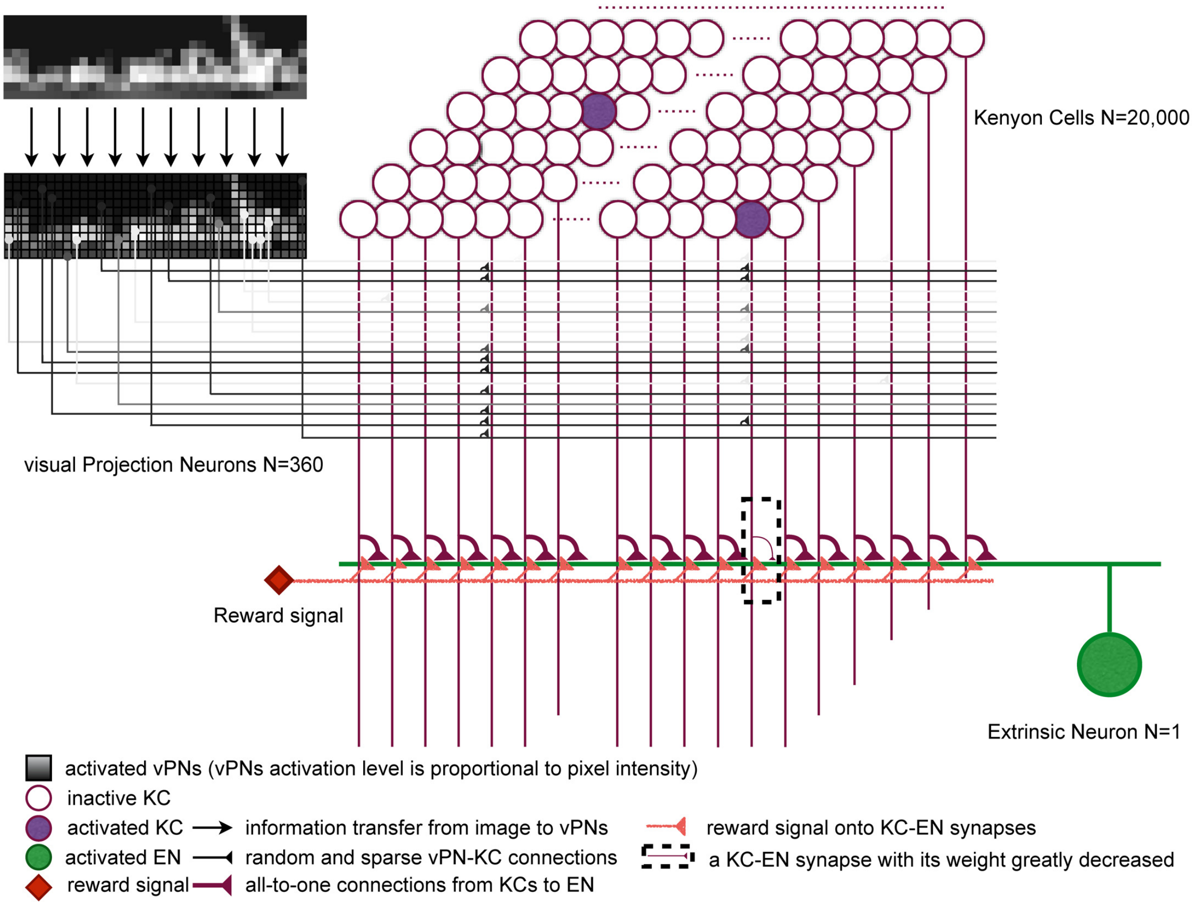

Kenyon Cells N=20,000
visual Projection Neurons N=360
Reward signal
Extrinsic Neuron N=1
activated vPNs (vPNs activation level is proportional to pixel intensity)
inactive KC
activated KC
activated EN
reward signal
information transfer from image to vPNs
random and sparse vPN-KC connections
all-to-one connections from KCs to EN
reward signal onto KC-EN synapses
a KC-EN synapse with its weight greatly decreased

Figure 3.4: Architecture of the MB model, retrieved from (Ardin et al., 2016). Images are sampled by the (360) vPNs which are connected to the (20000) KCs to form a sparse encoding. Each KC takes input from 10 different vPNs and only fires when several vPNs are activated. All KCs branch onto a single EN. Route memory is encoded through anti-Hebbian learning with STDP. After training, the EN will react little to familiar images.

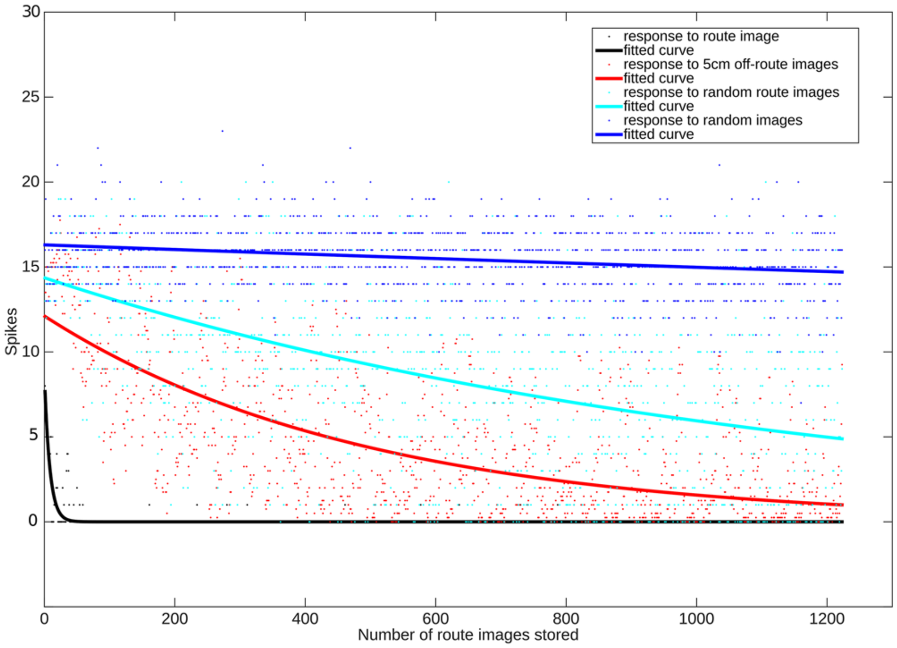


Figure 3.5: Capacity of the MB model to distinguish familiar and novel views as additional route information is stored. Retrieved from Ardin et al. (2016)

Opponent Processes for Visual Guidance

The navigational familiarity-based models as introduced by Baddeley et al. (2012) was adapted by Le Möel and Wystrach (2020) to also include anti-goal views. This view on ant perception was brought about as the original scene familiarity model could not explain certain recently observed behaviors in ants. The major drawback of the familiarity-based models is that recognition of a familiar view relies on a scanning behavior over a range of angles (as a direction of movement can not be computed directly but is chosen as the most familiar view over the range of head directions), which does not comply with behavior in ants—which only perform elaborate scanning when confronted with visual uncertainty (Wystrach, Philippides, et al., 2014)—and additionally introduces more computation effort as many different views at a single location have to be evaluated; scene familiarity does not correlate with directional error. Inspired by the discovery that ants can exhibit both attractive and repulsive learning (Wystrach, Buehlmann, et al., 2020; Murray, Kócsi, et al., 2020) and that MBs possibly form the neural substrate for such learning (Felsenberg et al., 2018; Aso et al., 2014), their approach was to present both goal and anti-goal views. In this way, the current view of an agent does correlate with directional error and subsequently a single view suffices to derive a direction of travel as seen in Figure 3.6. Familiarity was computed by calculating the global root-mean-square pixel difference between the current view and all views in the memory bank and retaining the lowest mismatch value, resulting in a measure for unfamiliarity. These values were then scaled to the range $[0 : 1]$ and subtracted from 1 to obtain a measure for familiarity. Views are stored in respectively attractive and repulsive memory banks for which an opponent familiarity value is deducted in the following manner:

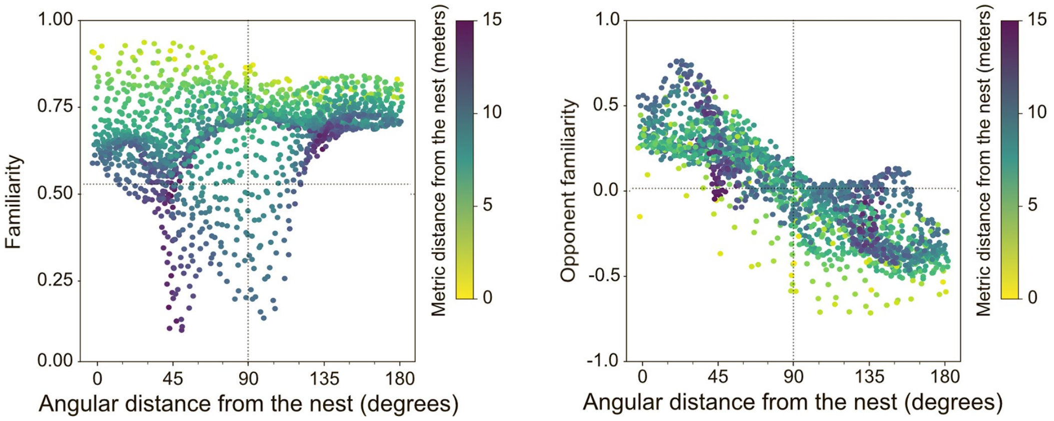


Figure 3.6: Scatter plot of the familiarity values against the angular distance between the north facing view and the nest direction. Angular distance of 0 degree means that the north facing view is pointing towards the nest. Euclidean distance of the position of the view from the nest is shown in the color map. Retrieved from Le Möel and Wystrach (2020).

$$\text{opponent familiarity} = \text{attractive familiarity} - \text{repulsive familiarity} \tag{3.6}$$

and the turn amplitude of the agent is deducted by:

$$\text{turn amplitude(deg)} = \text{baseline} - (\text{gain} \times \text{opponent familiarity}) \tag{3.7}$$

where the baseline is a fixed parameter that determines the extent of the turn amplitude and the gain a value to convert the familiarity into a turn amplitude.

Simulations with an ant agent in a reconstructed natural environment showed the model to be robust to:

- Decoupled memories: attractive and repulsive memories were acquired at different locations.
- Lower resolution eyesight.
- Noisy learning angles: uniform noise of ±90 degree was added to the view direction when acquiring goal and anti-goal views.
- Small learning walks: learning walks in a radius of 10 cm surrounding the nest opposed to a 2 m radius when 'walking' slower.

- Half as many memories: half the amount of views, because both goal and anti-goal views need to be stored.

3.2.3. Catchment Area of Panoramic Snapshots

Many methods of visual mapless navigation rely on the fact that the image difference function (calculated by computing the root-mean-square difference of pixel values) varies smoothly in natural, outdoor scenes, as reported in the seminal paper of Zeil, Hofmann, et al. (2003). Descend in Image Distance (DID) methods make use of this through descending in the direction that minimizes the image difference, which leads to homing behavior. Other methods rely on the same principle, but in a different manner. The scene familiarity model for example assumes that moving in the direction that seems most familiar—where there is the least difference between current and goal view—allows for successful route guidance. The performance of these methods varies greatly with the amount of information that is present in the scene (J. Müller et al., 2018).

Of great interest is the so-called 'catchment area/volume': the area/volume surrounding a snapshot where successful homing to the snapshot location can be achieved by an agent, through DID. In Zeil, Hofmann, et al. (2003), the question was asked: 'how different is the visual world when viewed from neighboring vantage points, and is this difference correlated with, and does it vary smoothly with, physical distance?'. This was performed by assessing the global image difference functions natural outdoor scenes along the edge of a small forest that included variance in shadow contours and motion of vegetation due to wind moving overhanging branches. Snapshots were recorded at intervals of 10 cm inside a 0.7 m unit cube and additionally at 1 m intervals over a 1 × 3 m area to assess the catchment area of natural scenes, of which the second experiment can be seen in Figure 3.7. From analyzing the root mean squared image difference surface, it is seen that in natural scenes, image difference functions appear to be smooth, without pronounced local minima, reaching similar values surrounding the reference image location. This smoothness of image difference functions depends most likely on the spatial-frequency distribution, where a broad variance in contrast, object distance and angular size contribute to its smoothness (Zeil, Hofmann, et al., 2003). This is less observed in indoor (artificial) scenes, making DID and familiarity methods less powerful in these circumstances. Later studies by Murray and Zeil (2017) confirmed this methodology to work in 3D scenes as well, where the respective catchment volumes were analyzed. Apart from the finding that the same methodology extends to 3D scenes, an increase in catchment volume is seen for snapshot at higher altitudes. Much of the deterioration in snapshots taken at lower locations is probably a cause of the inherent noise due to higher texture regions (like grass) near the ground. Wolfgang Stürzl and Zeil (2007) extended the work of (Zeil, Hofmann, et al., 2003) by evaluating the contribution of depth structure and contrast in the scene to the smoothness of the image difference functions in outdoor scenes. As previously discussed, direct computation of image difference functions suffers from changes in illumination, dynamic movements in scenes and features like shadows. It was shown that after contrast normalization, the image difference function seems to rely almost entirely on the depth structure of the scenes, making it robust to (small) dynamic

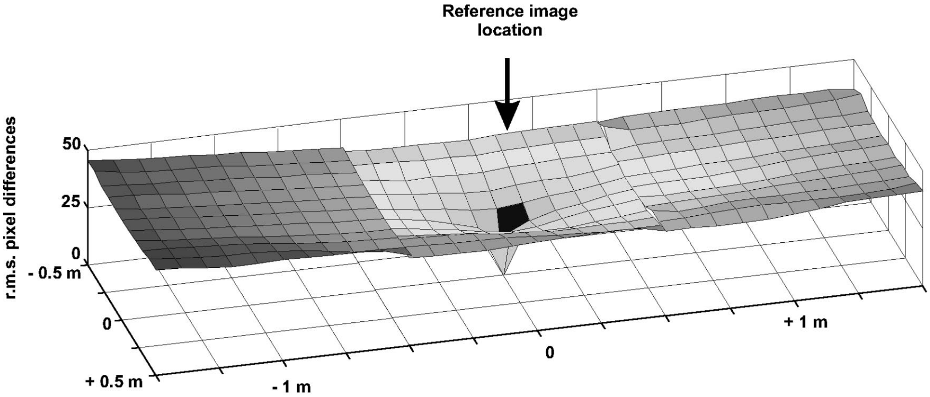


Figure 3.7: Horizontal extent of the image difference function, retrieved from Zeil, Hofmann, et al. (2003)

changes in the environment. Dealing with the inherent noise present in natural scenes and in the processing of information seems to be crucial for successful navigation (Cheung and Vickerstaff, 2010). Many insect-inspired visual navigation methods are evaluated in simulation which often lack the rich texture, color, luminance contrast, depth variance and noise present in natural outdoor scenes (Zeil, Hofmann, et al., 2003). Furthermore, 3D scenes in simulation that try to accurately capture and model natural environments, still show deficits in representing those environments (Wolfgang Stürzl, Grixa, et al., 2015). As a result it is important to perform real-life experiments to evaluate the performance and applicability of these algorithms.

4

Insect Physiology and Behavior

The physiology and behavior related to visual navigation in insects, specifically in the ant and bee, will be handled in this Chapter. Insects' visual perception is covered in Section 4.1. Next, their neuroanatomy related to visual processing and navigation is tackled in Section 4.2. In Section 4.3 the basics of insects' learning walks and flight will be covered as they play an important part in their navigational toolkit.

4.1. Insect Visual Perception

The camera-type eyes found in terrestrial vertebrates (Figure 4.1 a) work by focusing light that passes through the cornea and lens and which, via refraction, is projected on the retina, where an image is formed. The sharpest, high resolution color vision is generated at the macula lutea—also called the yellow spot—where there is a large concentration of cone cells, which have a low sensitivity to light but confer color vision. In the peripheral view, there is a greater abundance of rods, which have higher sensitivity to light but do not confer color vision and offer a lower resolution due to their relatively lower concentration on the retina opposed to the macula. This results in a low resolution peripheral sight, sensitive to motion and low brightness with very sharp vision only in a small area.

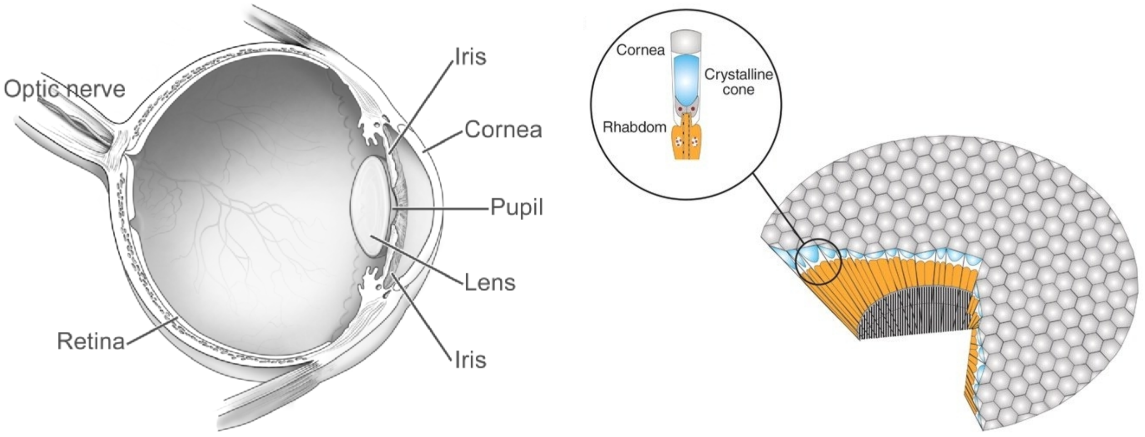
The main visual organ of insects is the 'compound eye' (Figure 4.1 b) and works quite differently. A compound eye is composed of numerous ommatidia (see also Section 4.1.3) that are stacked in a hexagonal pattern. Instead of a single beam of light that is focused on a retina, light enters through the different facets (part of the ommatidium) that compose the outer edge of the compound eye and covers different patches along the visual field, which are subsequently combined in the insect brain to form an image. Alongside their pair of compound eyes, many insects, including ants and bees, also possess dorsal ocelli. Dorsal ocelli are 'simple eyes' in the sense that they do not possess a complex retina. Ocelli likely play a role in sensing polarized skylight as a compass cue (Berry et al., 2011) and horizon detection for flight control (Mizunami, 1995), but as their function in visual navigation is less understood (Zeil, W. A. Ribi, et al., 2014; Kelber and Somanathan, 2019), and are thought to be too blurry for landmark recognition they will not be covered here any further.

4.1.1. Compound Eyes vs. Camera-type Eyes

Compound vision is more blurry compared to camera-type vision. This is due to the fact that it is physically difficult to fit a high number of physically separated ommatidia (see Section 4.1.3), ergo pixels, in compound eyes compared to camera-type eyes, where the amount of rods and cones on the retina determine the sharpness of vision. E.g. dragonfly—which are thought to have some of the best vision in insects—have 'only' up to 30000 ommatidia per eye, while an average human retina has about 4.6 million cones and 92 million rods.

Compound vision offers a couple advantages over camera-type eyes however, namely:

1. Due to their physical arrangement, compound eyes can cover a field of view of up to almost 360 degrees, this panoramic view, along with low-resolution sight, plays an important role in facilitating robust visual navigation (Wystrach, Dewar, et al., 2016).
2. Compound eyes in insects posses higher flicker fusion rates (up to 350 Hz (Ruck, 1958)) than humans (around 30 Hz) resulting in being able to better detect (fast) movement. This higher flicker rate is achieved due to the dynamics of the ommatidia as well as less visual information that needs to be processed in the insect brain.



(**a**) Schematic overview of human eye[1]

(**b**) Schematic overview of a part of a compound eye, encircled is (part of) a single ommatidium, adapted from L. P. Lee and Szema (2005) (AAAS: Science)

Figure 4.1: Structural difference between camera-type eyes and compound eye types.

4.1.2. Compound Eye Types

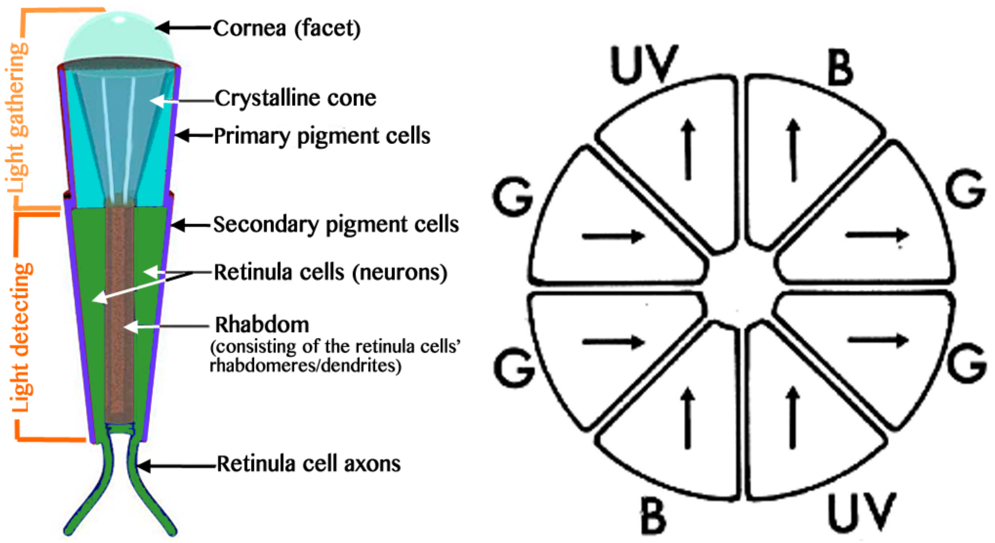
There exist roughly three types of compound eyes: the apposition (photopic vision), the optical superposition (scotopic vision) and the neural superposition compound eyes (found in many dipteran flies) (Cheng et al., 2019). In the apposition compound eye, each ommatidium consists of a single corneal lens focusing a single beam of light through the rhabdom whereas in the optical superposition compound eyes, in low-light conditions, this light beam is also received by neighboring rhabdoms thus increasing overall sensitivity to light. This is achieved through the presence of a crystalline tract. At night, the neighboring secondary pigment cells move towards the ends of the cell and thus light can pass through the crystalline tract to the neighboring rhabdoms, resulting in superimposed images for each ommatidium. In the neural superposition compound eye, there are several rhabdoms (6–9) in each ommatidium, which receive light from different corneal lenses and each optical nerve subsequently collects signals from different rhabdoms (Cheng et al., 2019). Apposition compound eyes are present in diurnal insects; optical superposition compound eyes are present in nocturnal insects.

In Figure 4.1 b, a fairly regularly structured visual organ can be seen, but because each ommatidium is practically equivalent to a single pixel and their is natural variation in the properties of these ommatidia, sensitivity and resolution can vary considerably within a compound eye.

[1] National Eye Institute, National Institutes of Health. (`https://commons.wikimedia.org/wiki/File:Human_eye_diagram-sagittal_view-NEI.jpg`), 'Human eye diagram-sagittal view-NEI', marked as public domain, more details on Wikimedia Commons: `https://commons.wikimedia.org/wiki/Template:PD-US`

4.1.3. The Ommatidium

Although there exists a great variety in the composition of compound eyes between and within different insect species, the building blocks are mostly the same: up to thousands of patches of ommatidia with each a corneal lens, crystalline cone and a rhabdom, encapsulated by pigment cells to block light from entering neighboring ommatidia. The structure of a single ommatidium is visualized in Figure 4.2 a.



(**a**) Structure of a single ommatidium[2]

(**b**) Overview of the retinula cells in the ommatidium of a bee, each cell is sensitive to a certain part of the visual spectrum as denoted by G = Green, B = Blue and UV = Ultraviolet. The arrows indicate the direction of polarization the photoreceptive cells are sensitivity to. An additional ninth photoreceptor is located in the center but is not shown here. Retrieved and adapted from Pye (2018)

Figure 4.2: Overview of the structure of ommatidia in bees.

The ommatidium (of apposition compound eyes) can be separated into two segments:

1. The outer, light-gathering segment, which consists of:
   - a cornea
   - a crystalline cone
2. The lower, light-sensing segment (rhabdom), which consists of:
   - 7–9 retinula cells
   - microvilli (dendrites of the retinula cells) that form the rhabdomeres and converge to form the rhabdom
3. Each ommatidium is surrounded by pigment cells, which block light from passing to neighboring ommatidia, thus restricting the view of each ommatidium to a certain angle in the view field. (The secondary pigment cells can move towards the cell ends during the night in optical superposition vision.)

The light-gathering segment collects and focuses incoming light which then passes through the rhabdom underneath. The rhabdom consists of a number of rhabdomeres (7–9) that are made up of a large number of parallel microvilli, which are dendrites of the surrounding retinula cells. These microvilli are sensitive to light in different spectra—in e.g. bees: Ultraviolet (UV), blue and green—but also different polarizations (in line with their orientation), as visualized in Figure 4.2 b. Early studies conveyed the idea that the ommatidia in the main part of the compound eyes contain identical sets of spectral receptors, but with the advent of the sequencing of the honey bee genome and the availability

[2] Retrieved from `http://www.bio.miami.edu/dana/360/360F19_11c.html` which was adapted from `https://cronodon.com/BioTech/Insect_Vision.html`

of new molecular tools that allowed more detailed analyses, this changed to a more heterogenous view of the compound eye (Avarguès-Weber et al., 2012). Mainly three different types of ommatidia were identified, all of them having six green receptors:

1. type I: 44% of ommatidia; 1 additional UV and blue receptor
2. type II: 46% of ommatidia; 2 additional UV receptors
3. type III: 10% of ommatidia; 2 additional blue receptors

An additional ninth receptor recedes at the base of the ommatidia but its spectral sensitivity is uncertain (Wakakuwa et al., 2005). The prevalence of green sensitive photoreceptors is due to them playing a role in both chromatic as well as achromatic visual pathways (M. Giurfa et al., 1996; Martin Giurfa et al., 1997). Green sensitive photoreceptors also exhibit faster response time compared to blue and UV photoreceptors (Skorupski and Chittka, 2010), indicating their role in fast achromatic vision. Blue and UV photoreceptors are in less of an abundance. Most hymenopteran insects have developed a region with a prevalence of type II ommatidia where UV receptors are solely orientated perpendicularly to each other, and thus is specialized in detecting polarized light. This region is present in the most dorsal part of their compound eyes, the so-called Dorsal Rim Area (DRA), and plays an important role in polarization vision that is used as part of a skylight compass (Zeil, W. A. Ribi, et al., 2014; Labhart and Meyer, 1999; Wehner, 2003). The type III ommatidia (and thus blue receptors) are in higher concentration in the anterior ventral region, where it is thought to play a role in the detection of ventral targets that contrast with the blue photoreceptors (Lehrer, 1999). Blue and UV photoreceptors seem to project only and directly to the medulla while the green photoreceptors project only to the lamina (Dyer et al., 2011).

4.2. Insect Neuroanatomy

The visual information that is captured by the compound eyes passes through several structures along its path through the insect brain. Visual information in the form of electro-chemical signals generated by light passing through the retinal cells, is passed through the basement membrane by the retinula cells' axons which form the optic nerve. Each of those signals pass through the optic lobes (the lamina, medulla and lobula) and subsequently to the protocerebrum (the visual center). Mainly two regions in the protocerebrum and their visual pathways have been identified that play an important role in navigation, namely the Central Complex (CX) and the MB. Where the MBs have been identified as forming the neural substrate for associative learning (Aso et al., 2014). The understanding of the neural circuitry of insects mostly stems from research on the fruit fly *Drosophila Melanogaster* and honeybee *Apis Mellifera*. Neural structures like the optic lobes, CX and MB are present in all insects but are sometimes theorized to play different roles depending on species and specialization, e.g. the potential role of the Mushroom Bodies in navigation in hymenoptera like ants and bees, opposed to a presumably more primary olfactory role in the fruitfly (Fahrbach, 2006). Universal visual processing structures in the insect visual system will be discussed in general while more focus will be given to the neuropils related to visual navigation in ants and bees, namely the Mushroom Bodies.

4.2.1. Optic Lobes

The optic lobes (the lamina, medulla and lobula) are responsible for the pre-processing of visual signals in the insect brain and are the first stops along the way to the protocerebrum.

The first layer where visual information is processed is the lamina (denoted by LA in Figure 4.3). Not all axons of the retinula cells in the ommatidia pass their signals to the lamina, e.g. in the honeybee, solely the green photoreceptors (R2,R3,R4,R6 and R7) are connected to the monopolar neurons of the lamina (Sommer and Rüdiger Wehner, 1975). It is theorized that the lamina's main function is to provide better visual contrast and elementary motion detection. Monopolar cells in the lamina (Lamina Monopolar Cells (LMCs)) are observed to have an excitatory response to signals that are in the center of

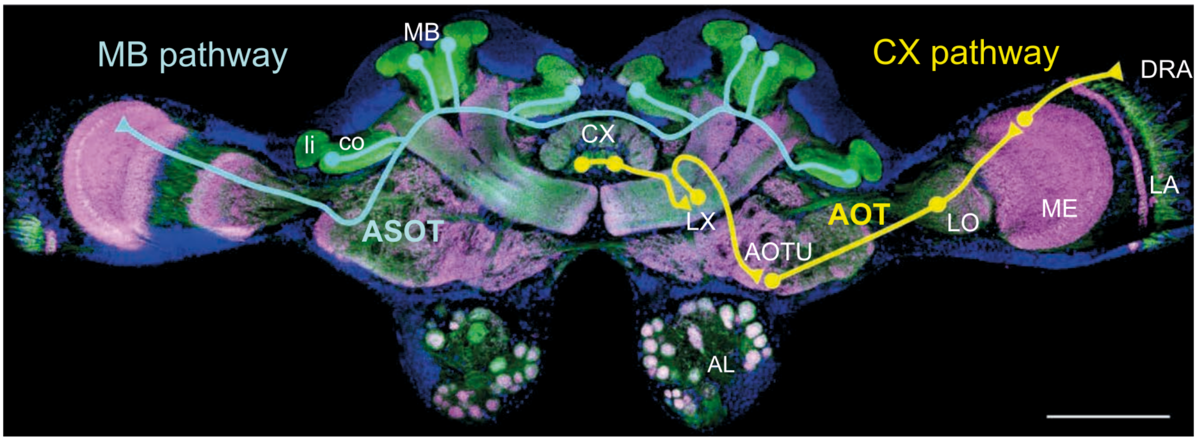


Figure 4.3: Two of the main pathways in the brain of the desert ant *Cataglyphis*. Retrieved from Rössler (2019).

their receptive field (the receptive field being the volume surrounding the neuron where a stimulus generates an influence on the neuron's spiking rate) while stimuli that are in the periphery of the receptive field invoke inhibitory signals. This mechanism is known as lateral inhibition (Zettler and Järvilehto, 1972) and has as effect that the insect visual system will respond less to global illumination changes and more to visual patterns that excite the center of the receptive field of the monopolar cells of the lamina. The lamina neuropil is made up of optical cartridges that receive input from the ommatidia's—the ones are positioned directly atop the cartridges—nine photoreceptor axons in addition to axons of four different LMCs. Between cartridges there exist tangential, centrifugal and horizontal connections. The retinotopic organization is thus preserved, as the spatial configuration of cartridges and the overlying ommatidia is constant throughout the lamina (Avarguès-Weber et al., 2012).

Projections from the lamina are subsequently sent to the second layer, the medulla, (Strausfeld, 1976) (denoted by ME in Figure 4.3). In *Apis Mellifera* the three remaining axons (R1, R2 and R9) from the ommatidia bypass the lamina and also connect to the medulla where cells are organized in columns. The medulla is thought to play a role in color vision (Bausenwein et al., 1992) but also in motion detection (Ibbotson et al., 1991). The medulla consists of eight different laminated layers (W. Ribi and Scheel, 1981), orthogonal to the columns. Between these columns many more horizontal connections exist than in the lamina. Furthermore, neurons in the medulla show opponent excitation or inhibition to different photoreceptor-type inputs. These so-called color-opponent neurons can thus generate inhibitory or excitatory responses depending on the combination of inputs from the three different types of photoreceptors (Hertel and Maronde, 1987). This opponent process is thought to represent the neural substrate for color vision (Avarguès-Weber et al., 2012).

The third and final optic lobe layer is the Lobula (denoted by LO in Figure 4.3). Again, color-opponent neurons seem to be present in this layer (Hertel and Maronde, 1987). Additionally, spatially sensitive opponent neurons are present in the lobula that have the same kind of opponent excitation-inhibition response as in color-opponent neurons, but react differently depending on where the signal comes from or in which direction the signal is propagated (Hertel and Maronde, 1987; Avarguès-Weber et al., 2012).

Signals from the optic lobes are subsequently passed mainly to the calyxes of the mushroom bodies and the central complex (Ehmer and Gronenberg, 2004).

4.2.2. Central Complex

The Central Complex seems to be responsible for the integration of many different (pre-processed) signals (coming from e.g. the optic lobes), in order to keep track of the insect's body state and perform path integration (Hoinville and Rüdiger Wehner, 2018). This information is used to send signals to the legs and or wings for positioning itself in the world (Webb and Wystrach, 2016). The CX is also thought to play a role in the detection of polarized skylight for use in a celestial skylight compass as a visual pathway from the DRA to the CX exists (Paulk et al., 2009). The CX has not been connected to visual navigation however, although a study proposes the integration of signals from the MB (which is known to play a role in visual navigation) in the CX which is then translated to locomotion control (X. Sun et al., 2019).

4.2.3. Mushroom Body

Mushroom Bodies on the other hand, possess a more parallel structure which would suite them well for parallel processing and memory storage and presumably play an important role in olfactory and storing/learning visual memories (Menzel, 2014; Webb and Wystrach, 2016; Hoinville and Rüdiger Wehner, 2018). The mushroom body consists of a large amount of Kenyon cells (KCs; intrinsic neurons). The dendrites of the KCs form the calyx, and their axons run through the pedunculus which then split into two structures, called the α and β lobes (Heisenberg, 2003; Ardin et al., 2016). Panoramic images from the compound eyes travel through the optic lobes to the MB where these signals are sampled by Projection Neurons (PNs) and subsequently are sent to KCs in the calyx. Multiple PNs output to single KCs, which only react to certain combinations of inputs. The about 200000 KCs in the bee MB project to only a few Mushroom Body Output Neurons (MBONs), thus presenting a sparse encoding of visual data. This sparse coding allows the MBs to store many different representations of olfactory cues but presumably also visual scenes and has inspired methods that implement SNNs based on the MB architecture coupled with biologically plausible learning (e.g. Hebbian learning like STDP) as a model for ant visual navigation (Le Möel and Wystrach, 2020; Ardin et al., 2016).

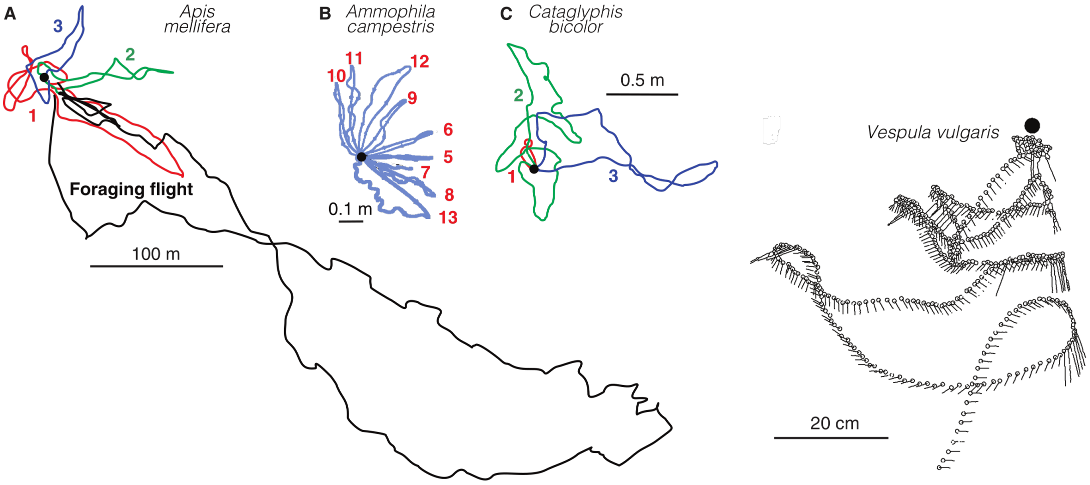
4.3. Learning Walks and Flights

Learning walks and flights are an important part of the navigational behavior in ants and bees, and are typical of hymenopteran central foragers (Zeil and Fleischmann, 2019). The two form a close relationship with each other, as systematic exploration of an area can help tremendously in returning to the nest form distant locations. Bees and ants are observed to exploit features like depth structure, prominent landmarks and color among possibly other cues to aid in recognition of the environment.

4.3.1. The First Walks and Flights

Learning walks and flights constitute the first navigational act of hymenopteran central foragers when leaving the nest. When *Cataglyphis* ants transition to a foraging role they spend the first 2–3 days taking exploratory walks around the nest in which they map the surroundings of the nest. The first walk remains very close to the nest, such that they can rely on their Path Integration (PI) system, and subsequent walks venture further and cover different compass directions around the nest (Thomas S. Collett and Zeil, 2018). The same kind of behavior is observed across different central-foraging species. The first exploratory flights of the bee *Apis mellifera*, the wasp *Ammophila Campestris* and the ant *Cataglyphis Bicolor* can be seen in Figure 4.4 a.

Typical during such exploratory walks/flights are saccadic movements that expand outwards of the nest location in increasingly bigger arcs/circles (Thomas S. Collett and Zeil, 2018). During these learning walks/flights the insect will look regularly back at the direction of the nest, even at locations where the nest is not directly visible. *Cerceris* wasps tend to fixate their view to the nest at the end of such arcs and in between keep the nest centered between about 45 and 60 degrees of the field of view on either side of its pair of compound eyes (Zeil, 2012) (Figure 4.4 b). Similar behavior is seen in ants but variability exists between species. Another influencing factor is the environment in which the nest is located, specifically the depth structure—which is shown to be influential on the smoothness of the catchment area (Wolfgang Stürzl and Zeil, 2007), see Section 3.2.3—is of concern, where bumblebees and ground-nesting wasps will orient their arcs opposite the direction of prominent landmarks close to the nest (Thomas S. Collett and Zeil, 2018). The way that the surroundings of the nest are learned is also reflected in how they return, as shown in the wasp *Cerceris*: they will return in roughly the same di-



(**a**) Learning walks/flights of different insect species.

(**b**) Saccadic learning walk of the wasp *Vespula vulgaris*.

Figure 4.4: Insect learning walks/flights. (a) Learning walks/flights of **A** the bee *Apis Mellifera*, **B** the wasp *Ammophila Campestris* and **C** the ant *Cataglyphis Bicolor*. In red: first learning walk/flight, in green: second walk/flight and in blue: the third walk/flight. Adapted from Thomas S. Collett and Zeil (2018). (b) Saccadic learning path of the wasp *Vespula vulgaris*; top-down view, retrieved from Thomas S. Collett and Zeil (2018).

rection as how the views were originally acquired. This shows the close relationship between learning walks/flights and the subsequent foraging trips that follow.

4.3.2. Lifelong Learning

After getting to know the immediate surroundings of the nest, hymenopteran central foragers will leave for (long) foraging trips; getting increasingly better at finding their way to and from feeding grounds the more experienced they are (Zeil and Fleischmann, 2019). The same techniques of the first learning walks are employed throughout the rest of their foraging lives, but are triggered depending on circumstances. For example *Myrmecia croslandi* ants have been observed to explore the neighborhood of the nest opposite the direction they depart to for foraging trips, presumably to make sure that they can return to the nest when overshooting the nest upon return, which occurs regularly (Jayatilaka et al., 2018). Learning happens continuously for the rest of their foraging life, where returning to the nest from different directions provides learning insects with validating results or triggers learning behavior when having difficulties to home. Foraging insects are triggered to perform additional learning maneuvers upon having difficulties or failing to home to their nest. Considerably more time (about four times more) is spent by the bumblebee *Bombus terrestris* exploring the nest than when arriving at a newly discovered flower, even when the surroundings are the same (Robert et al., 2018). This is quite logical, as getting safely back to the nest is probably of higher importance than being able to return to a single flower; this also shows that insects can exhibit remarkably quick (one-shot) learning of new visual scenes.

5

Event-Based Vision

In disaster-relieve areas, there is often no prior map and the scene can change dynamically or go from light to dark quickly. These are areas in which an event-based camera can excel over conventional cameras. First, the working principles of event-based vision sensors will be depicted in Section 5.1. Its advantages and challenges compared to frame-based vision will be discussed. In Section 5.2 event representations and how to process individual events are discussed. Much of the following Sections are based on the excellent review on event-based vision of Gallego, Delbruck, et al. (2019).

5.1. Working Principles of Event-Based Vision Sensors

Event cameras capture brightness changes asynchronously for each individual pixel. Event cameras output a timestamped stream of events with information about the location of the brightness change and its sign (Gallego, Delbruck, et al., 2019). The sensor's pixels act on the log intensity of the change in brightness, as set by a change threshold. Each individual pixel captures the log intensity of the latest event and send a new event when there is a change in brightness that is bigger than the threshold. There are a few different event-based vision sensors on the market. Namely, the Dynamics Vision Sensor (DVS) (Lichtsteiner et al., 2008), the Asynchronous Time Based Image Sensor (ATIS) (Posch et al., 2011) and the Dynamic and Active Pixel Vision Sensor (DAVIS) (Brandli et al., 2014) are the most commonly used. The DVS camera is based on a frame-based silicon retina design and only outputs brightness changes in the form of 'ON' and 'OFF' events. The ATIS sensor additionally outputs absolute brightness levels, but has more difficulties in dark, dynamic scenes as the pixels can get saturated. The DAVIS sensor combines an active pixel sensor with a DVS in the same pixel. The main advantages of event cameras are summarized:

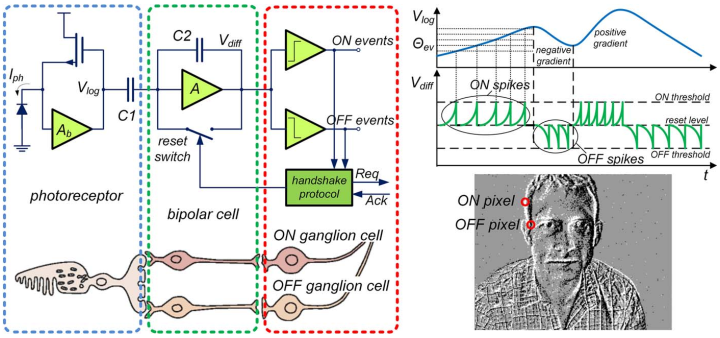


Figure 5.1: Schematic of the operation of DVS pixel, converting brightness changes into events. Adapted from (Gallego, Delbruck, et al., 2019).

1. High temporal resolution (order of μs). Every brightness change is detected very fast and timestamped with a 1 MHz clock. As such, the camera does not suffer from motion blur.
2. Low latency (order of sub-millisecond). Pixel events are transmitted almost immediately after the time the change in brightness is detected.
3. High dynamic range ($>$ 120 dB vs. 60 dB of conventional cameras). Ability to be deployed in very dimly lit and very bright scenes, making it excellent for outdoor use throughout the day or in environments where there is little control of the lighting.
4. Low power consumption (order of 10) mW. Due to event-based cameras only acting on brightness changes, any redundant (static) information is discarded. This does not only influence sensor power consumption, but also the processing of visual information as only necessary data is passed.

Event-based vision compels to take a different approach to visual processing as there is a fundamental difference between the data output of event cameras compared to frame cameras. This has an effect that traditional theories can not be applied, and need to be adjusted/reimagined. Quite importantly, event cameras suffer from the same inherent noise of photons and non-ideal circuitry that is found in all vision sensors. Due to the novel nature of event cameras, this is more pronounced, as the process of quantizing temporal contrast is not yet fully understood and in order to overcome this difference, different methods will have to developed.

5.1.1. Event Generation Model

Events are represented as incremental pixel brightness changes that are represented by their position, time and polarity:

$$e_k \doteq (\boldsymbol{x_k}, t_k, p_k) \tag{5.1}$$

where t_k is the time at which the event occurred, $p_k \in \{-1, 1\}$ the polarity of the event ('ON', 'OFF'), and $\boldsymbol{x_k}$ the position of the pixel:

$$x_k \doteq (x_k, y_k)^T \tag{5.2}$$

Event cameras react on the log of the photocurrent:

$$L \doteq \log I \tag{5.3}$$

An event occurs when the change of the log intensity reaches a certain threshold $\pm C$, which can be set by the user using the pixel bias current (Lichtsteiner et al., 2008):

$$\Delta L(x_k, t_k) \doteq L(x_k, t_k) - L(x_k, t_k - \Delta t_k) = p_k C \tag{5.4}$$

Typical values for C are between $10\% - 50\%$ illumination change (Lichtsteiner et al., 2008; Son et al., 2017). A trade-off has to be made between sensitivity of the sensor and the amount of noise that will result because of the increase in sensitivity.

For small increments in time Δt_k, the increment in brightness change can be approximated by a Taylor's expansion, such that any event portrays information about its temporal derivative:

$$\Delta L(x_k, t_k) \approx \frac{\partial L}{\partial t}(x_k, t_k)\Delta t_k = p_k C \tag{5.5}$$

This interpretation can be utilized to give physical meaning to the otherwise binary 'ON' and 'OFF' events.

Under constant lighting conditions, one can proof, by linearizing Equation 5.4 and constant brightness assumption that:

$$\Delta L \approx -\nabla L \cdot \boldsymbol{v} \Delta t \tag{5.6}$$

Meaning, if the motion is parallel to the edge, no events are generated ($\boldsymbol{v} \cdot \nabla L = 0$). When the motion is perpendicular to the edge ($\boldsymbol{v} \cdot \nabla L$ has its maximum value), the highest rate of events is generated.

The above derivations are an idealized model for when events are generated. Due to sensor noise, which is influenced by the design and the operating conditions, and influence of the illumination of the scene the process behaves in a stochastic fashion. Several approaches exist for taking into account the stochastic nature of the DVS sensor output(Gallego, Delbruck, et al., 2019):

- Contrast sensitivity C can be approximated by a normal distribution centered around the mean value of C with a standard deviation σ of typically 2–4% (Lichtsteiner et al., 2008).
- One can take into account the image gradient of the scene, as most events are created by this image gradient (Censi and Scaramuzza, 2014).
- Understanding of the temporal behavior of event data is preliminary. Most noise filtering assume that real events are more spatially correlated than noise as they occur due to real objects (Czech and Orchard, 2016).

5.2. How To Process Events

Event cameras have high temporal resolution and low latency. One can take two different approaches to act on events: the state of the system is updated immediately when an event occurs (taking advantage of the aforementioned properties), or processing grouped events over a certain time span. One event in itself can not provide enough information though, and as such it is important to also capture information about past events. Furthermore, another differentiation can be made between model-based and model-free approaches. Additionally, one can differentiate between types of objective/loss functions: geometric- or temporal- or photometric-based. In the following sections, focus is given on representations and methods that directly work on the interesting properties of event-based vision over frame-based vision, namely its low latency and sparse representation (Gallego, Delbruck, et al., 2019).

5.2.1. Event Representation

Events can be represented in different manners depending on the designated use case and available techniques. The most straightforward method is to process individual events as they arrive, other methods focus on grouping events based on their spatiotemporal information:

- **Individual Events:** each event $e_k \doteq (x_k, t_k, p_k)$ is used in event-by-event processing. This is mostly performed asynchronously on probabilistic filters and Spiking Neural Networks. The filters or network keep or get additional information from past events which is combined with the current input to generate a new output.
- **Packet Events:** in an event packet, events $E \doteq \{e_k\}_{k=1}^{N_e}$ are grouped according to their spatiotemporal proximity. Here it is important to select the right value for N_e depending on the speed of the motion of the image.
- **Event frame or 2D histogram:** here all events in the same spatio-temporal neighborhood are accumulated over time and displayed as a regular 2D image. This loses a lot of the advantages of event-based vision but has the advantage of being compatible with conventional computer vision algorithms and allows for easy interpretation of the data.
- **Time Surface (TS):** a TS is similar to the event frame in the sense that it maps the event camera output to a 2D map. Each pixel is expressed as an intensity value, where more recent event activity correlates with a higher intensity value. They are great for showing the rich temporal information captured by the event cameras. Only temporal information is portrayed by TS.
- **Voxel Grid:** voxels to represent events in 3D space-time. Each voxel portrays a pixel's location over a certain time-frame. This preserves temporal information better with respect to event frames.
- **3D point set:** where proximate spatio-temporal events are grouped per voxels in Voxel grids, the 3D point set preserves information of individual events, $(x_k, t_k, p_k \in \mathbb{R}^3)$. Plane fitting can be used to derive optic flow.

- **Points sets on image plane:** of use when tracking edges.
- **Motion-compensated event image:** the motion of an edge can be estimated by warping events to a reference time and maximizing their alignment which produces a sharp image. Motion-compensated event images have applications in feature tracking, as motion-invariant edges are revealed.

5.2.2. Event Processing

In general, there is no single processing step that encapsulates all the necessary action required for processing event data. That is, data is mostly pre-processed to representations that suite their application. E.g. images are reconstructed from event cameras that are then fed to traditional high-performance computer vision algorithms. However, this defeats the purpose of event based vision in a sense, as the high temporal resolution and sparsity of visual data is lost. Other approaches transform event data to the aforementioned representation (Section 5.2.1), perform feature extraction, which are then fed to Artificial Neural Networks (ANNs). Deep Neural Networks (DNNs) can also be exploited to directly extract features and process data. One-by-one processing of events can be performed with SNNs (on neuromorphic hardware). Filters play an important role in one-by-one event processing too, where the state can be updated based on single events in continuous time (Gallego, Delbruck, et al., 2019). Filters allow to seamlessly fuse additional data from other sensors (sensor fusion) for better and more robust results. Deterministic filters find their application more in performing operations on event data to prepare them for further processing steps. They have been used for but not limited to use in noise reduction, brightness filtering, image reconstruction and feature extraction. Probabilistic filters like Kalman and particle filters have been used mostly for pose tracking, SLAM and Visual (Inertial) Odometry. ANNs are used in a wider range of topics and can be used for end-to-end learning. Parts of the network (mostly related to feature extraction) can be trained in an unsupervised fashion which are then fed to supervised classifiers which need labeled data.

6

Neural-Based Visual Guidance

This Chapter will focus on the use of neural networks for monocular visual guidance and its implementations in simulation and/or robots. Both frame-based and event-based methods will be discussed. Mainly three different neural-based visual guidance approaches can be distinguished with each its (dis)advantages. Section 6.1 discusses neural-based visual odometry. Section 6.2, handles neural-based route-following, including path-following and visual homing schemes.

6.1. Neural-based Visual Odometry

Traditional feature-based visual odometry relies on tracking and matching salient features between consecutive frames using feature descriptors. The camera pose and environment structure are roughly recovered through epipolar geometry, and later fine-tuned through minimization of the reprojection error. Direct visual odometry methods do not rely on tracking and matching features but estimate pose and structure directly through matching image intensity values with the local intensity gradient. Monocular Visual Odometry (VO) methods however suffer from not having a direct estimate of the depth structure of the scene. Early methods required initialization from a known position e.g. at a fixed distance from a plane. Longuet-higgins (1981) solved this by algebraically eliminating depth from the initialization problem, however this results in an unknown scale for translation and scene structure. The scale can be recovered by assuming the scene to be planar (use of Homography matrix), non-planar (use of Essential matrix) and iteratively estimating these matrices through matching new features. A third method was introduced by initializing the scene with random, high variance depth values, that subsequently converges after iterative matching; this method is not guaranteed to converge however (Younes et al., 2017).

6.1.1. Deep Learning for Monocular Depth Estimation

As a way to overcome the depth ambiguity problem, deep learning based monocular depth estimation methods have been proposed in recent years (see Figure 6.1). Deep learning methods based on convolutional, recurrent, auto-encoder or generative adversarial networks show promising results in estimating the depth structure from a sequence of or a single monocular image and are thoroughly reviewed in (Zhao et al., 2020). Neural-based depth estimation can be generally split up into supervised, unsupervised and semi-supervised methods, using—in general—respectively, known dense depth maps, geometric constraints between frames and stereo-imagery (Zhao et al., 2020).

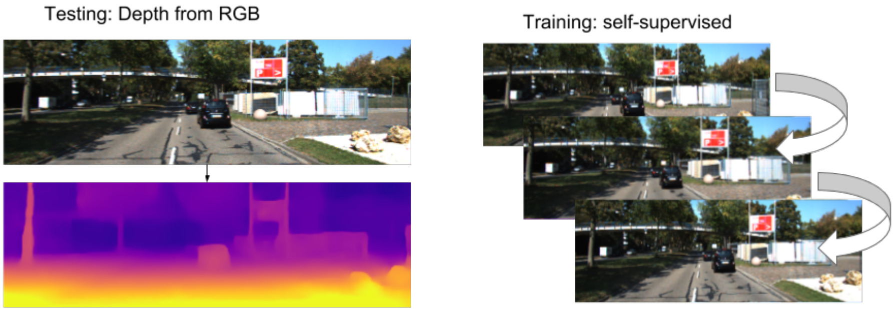


Figure 6.1: Unsupervised deep learning based monocular depth estimation, retrieved from Casser et al. (2019).

Supervised Learning of Depth

Deep learning based depth estimation was first introduced by (Eigen et al., 2014). Eigen et al. (2014) trained their network on accurate ground truth depth information and used a simple loss defined as the difference between the predicted and the real depth map and employed a coarse- and fine-scale network to first predict coarse depth and later refine these results with the fine-scale network. Ensuing, due to its success in image classification, ResNet (He et al., 2016) networks were introduced in estimating image depth (Laina et al., 2016). Many models since then have utilized pre-trained network architectures like but not limited to ResNet (He et al., 2016), MobileNet (Howard et al., 2017) and VGG (Simonyan and Zisserman, 2014) which are often pre-trained on datasets like ImageNet (Deng et al., 2009).

Semi-supervised Learning of Depth

Semi-supervised methods like (Laidlow et al., 2019; Amiri et al., 2019), utilize mostly stereo imagery—(Kuznietsov et al., 2017) augment this additionally with sparse LiDAR data. An inverse depth map is computed from the left (or right) image, from this inverse depth map the other image is predicted through reverse image warping and the reconstruction error is used as for the learning update. Additional consistency constraints are added to constrain the consistency of the disparity between the left and right frame (Garg et al., 2016; Godard et al., 2017). Many other methods like the addition of semantics (Ramirez et al., 2018) are added but not handled further here; for a comprehensive overview one is advised to look at (Zhao et al., 2020).

Unsupervised Learning of Depth

Collecting large datasets with detailed ground truth dense depth maps or calibrated stereo imagery poses more effort and with the better availability of monocular image streams, unsupervised methods have gained increasing interest from the research community. Unsupervised methods like (Yin and Shi, 2018; Ye et al., 2018; A. Z. Zhu et al., 2019; Feng and Gu, 2019; Casser et al., 2019; Chen et al., 2019) however suffer from the same scale ambiguity and inconsistency of traditional monocular VO methods, and show generally worse performance than (semi-)supervised methods. Often, ego-motion (Feng and Gu, 2019) and optical flow are learned (Yin and Shi, 2018; Ye et al., 2018; A. Z. Zhu et al., 2019; Casser et al., 2019; Chen et al., 2019) in parallel in order to be able to separate rigid and non-rigid parts from the scene. Furthermore Casser et al. (2019) propose an approach that discerns dynamic objects from the scene to get better results and does not make use of optical flow.

Deployability to Novel Scenes

Deep monocular depth estimation algorithms are mostly trained on large datasets like KITTI (Geiger et al., 2012), NYU Depth (Silberman et al., 2012), Cityscapes (Cordts et al., 2016) and Make3D (Saxena et al., 2008) and show excellent performance when tested on these datasets themselves, but exhibit severe performance degradation when deployed in other domains if no adaptation is performed. Transfer learning to different domains has gained traction recently to facilitate deployment 'in the wild'. Chen et al. (2019) for example take advantage of the fact that self-supervised learning eliminates the asymmetry between training and testing and perform online optimization to learn the new structural constraints (image intensity, flow, camera motion and depth). They propose online optimization through the fine-tuning of parameter weights of the original model (PFT) or through output fine-tuning (OFT) which only optimizes the output without recomputing all the network weights and instead uses a self-supervised loss function, achieving an order of magnitude speed-up compared to PFT.

Real-time Computation on Embedded devices

Deployability on embedded systems still remains a challenge for many of the proposed methods as they rely on deep neural networks with millions of parameters. Recently, efforts have been put forward to be able to run those systems on embedded devices like phones and drones. Wofk et al. (2019) propose a method based on MobileNet, which is optimized to run on mobile devices. MobileNet decomposes traditional $n \times m \times m$ convolutional layers into n $m \times m$ and a 1×1 pointwise layer such that each convolutional layer convolves with a single channel; resulting in considerable loss in latency (increased inference speed) (Wofk et al., 2019). Furthermore, a decoder network consisting of 5 convolutional neural networks is deployed to increase output resolution and perform dense depth prediction. The same methodology as MobileNet is deployed, depth-wise decomposition, and nearest-neighbor interpolation is performed *after* convolution to lower the resolution of feature maps (Wofk et al., 2019). Additionally, feedforward connections from the encoder are added to layers in the decoder to help with the reconstruction of features that might have gotten lost due to the compression to lower resolution by the encoder. Finally, network pruning with NetAdapt (T.-J. Yang et al., 2018) is performed to identify and remove superfluous feature channels. NetAdapt removes features until a certain performance-complexity tradeoff has been reached. This allows the network to run at 178 fps on an NVIDIA Jetson TX2 GPU and at 27 fps on its CPU, unlocking real-time onboard processing capabilities for micro aerial vehicles. Aleotti et al. (2020) explore the use of existing estimators to be able to run on smartphones (iPhone XS). Along with the study presented in (Peluso et al., 2019), to the author's knowledge, these are some of the only limited publicly available examples of deep learning based monocular depth estimation methods that are able to run on low-power embedded systems.

6.1.2. Deep Learning for Optical Flow Estimation

Optical flow estimation plays an important role in visual navigation and is widely used for obstacle avoidance, autonomous landing maneuvers and as part of visual odometry and SLAM pipelines. Classical methods for optical flow estimation are energy-based. That is, energy is minimized with respect to a brightness constancy term between temporally matching pixels and a spatial smoothing term to prioritize pixels to move in similar directions:

$$\arg\min_{\boldsymbol{u}} E(\boldsymbol{u}) = \arg\min_{\boldsymbol{u}} \int \left((I_x u + I_y v + I_t)^2 + \alpha^2 (\|\nabla u\|^2 + \|\nabla v\|^2) \right) dx\, dy \tag{6.1}$$

with E the energy reward function, u, v optical flow components in respectively the x and y direction, I_x, I_y the image brightness of a pixel at position (x, y) and α a scaling factor (smoothing term). This methodology was introduced by (Horn and Schunck, 1981) and later energy-based optical flow models have adapted the energy model to achieve better results. Concurrently with the surge in deep learning for depth estimation, deep learning for optical flow estimation gained popularity as well, often being implemented alongside each other to complement their respective weaknesses. For example, having knowledge about the depth of a scene helps with separating fore- and background, which helps in determining optic flow near the border of occluding objects, where traditional optical flow methods lack performance. Deep learning methods for optical flow have been comprehensively reviewed in (Hur and Roth, 2020).

Deep CNNs as Feature Extractors

Early methods focused on using Convolutional Neural Networks (CNNs) purely as feature extractors and descriptors which were then fed to classical energy-based methods for extracting optical flow.

These methods achieved better results than the state-of-the art classical energy-based methods as CNNs are able to extract better feature descriptors than the handcrafted ones. The networks mostly utilized a Siamese architecture (two identical parallel networks), where a single image is fed to each of the networks. The feature vector that each network outputs is compared and matched. Mostly positive (with ground truth optical flow) and negative samples are used in a supervised manner for training, where the L_2 loss is computed to both minimize distance between positive samples and maximize distance between negative samples (Hur and Roth, 2020).

Supervised Learning of Optical Flow

At the same time, end-to-end learning approaches were explored to estimate optical flow. The first deep learning end-to-end optical flow architecture, FlowNet, was proposed by Dosovitskiy et al. (2015), and set the stage for subsequent further developments and fine-tuning of deep learning based optic flow estimators. Dosovitskiy et al. (2015) deployed two networks, FlowNetS and FlowNetC, both with the typical encoder-decoder architecture, as seen in Figure 6.2. FlowNetS uses concatenated image pair inputs to output optical flow directly, while FlowNetC uses each image separately and extracts features which are then used to construct a cost volume. FlowNet is trained in a supervised way and because of limitations in acquiring accurate ground truth optical flow in natural scenes, the authors pre-trained their network on a synthetic dataset. Due to the supervised way of training, this did not generalize well to real life datasets and achieved subpar performance compared to classical energy based networks. Furthermore, the network was quite large (over 70 million parameters) and thus inference rate remains slow for use on embedded devices. The network proposed by (Ranjan and Black, 2017), deployed a spatial pyramid network (SPyNet) which only required 1.2 million parameters while achieving better performance than FlowNet. The pyramidal structure recursively refines the estimation of the optical flow and thus naturally deals with the computation of optical flow for larger displacement, its architecture can be seen in Figure 6.3. Later, FlowNet2 (Ilg et al., 2017) was proposed, which stacked multiple (modified) FlowNet architectures to refine the optical flow output and achieved a 50% increase

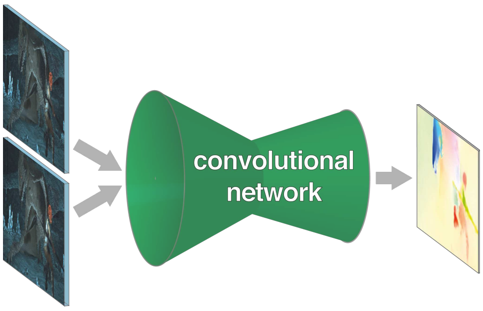


Figure 6.2: FlowNet, uses an encoder-decoder architecture to first compress information and later refine it, retrieved from Dosovitskiy et al. (2015)

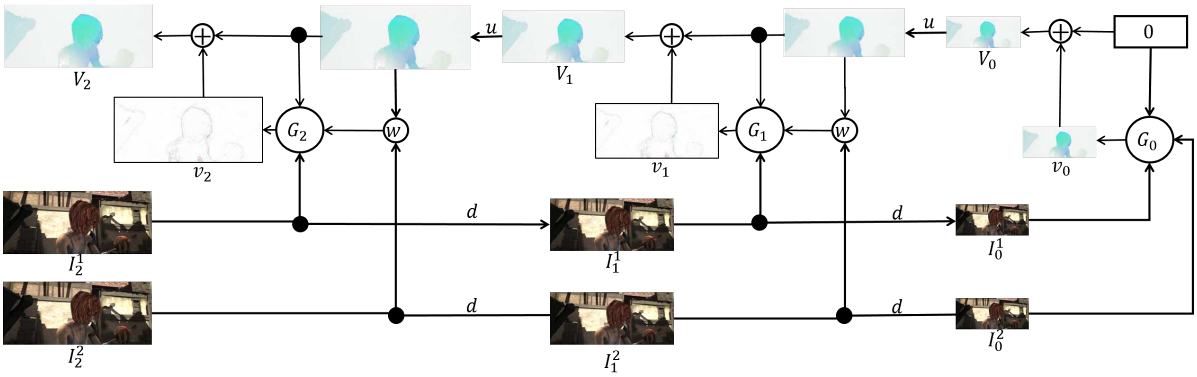


Figure 6.3: The pyramidal structure of SPyNet. SPyNet computes residual optical flow at each level, the residual flow of higher levels is passed down to lower levels where the results are refined. Retrieved from Ranjan and Black (2017)

in performance over FlowNet and now achieved comparable performance to state-of-the-art energy-based methods. Later supervised methods like PWC-Net (D. Sun et al., 2018) and LiteFlow (Hui et al., 2018) further reduced model size, inference time and increased performance. Further improvements were mainly achieved through variations on existing model architectures like PWC-Net: e.g. iterative residual refinement (Hur and Roth, 2019) and the introduction of a 4D volumetric output representation (G. Yang and Ramanan, 2019). The current state-of-the-art is presented by Bar-Haim and Wolf (2020) (ScopeFlow), which uses the network architecture of PWC-Net as its backbone and focuses on better training techniques. By augmenting scoping sampling with larger scopes (crops and zoom-out) and more careful selection of where to crop alongside relaxation of regularization and augmentation during the training process increased results considerably.

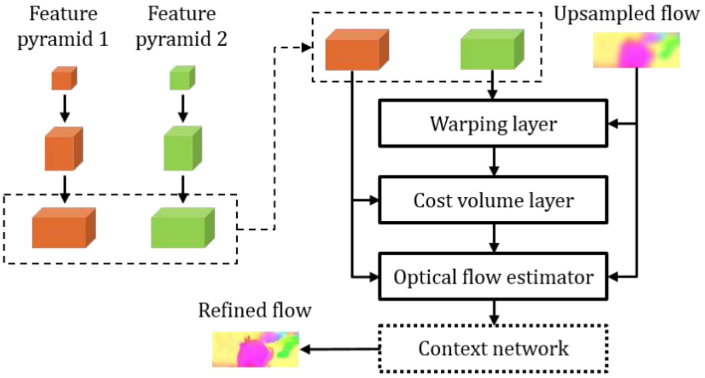


Figure 6.4: PWC-Net utilizes a parallel pyramidal structure to retrieve image features of two images. The second image is warped with the optical flow computed at the previous pyramid level. Next, a cost volume is computed by comparing the first image features to the second. Afterwards, the optical flow is estimated by comparing image features of the first layer, the cost volume and the upsampled flow of the previous pyramid level. An optional context network is added for further refinement. Retrieved from D. Sun et al. (2018)

Unsupervised Learning of Optical Flow

The main drawback of supervised optical flow estimation methods is that they need extensive high quality datasets with labeled data for training. As real life ground truth data datasets for dense optical flow are difficult to obtain, artificial datasets are used for initial training. Methods that use artificial scenes for training should take caution when transitioning to real life deployment, and many methods have indeed been introduced to deal with this, mostly through regularization techniques such as affine transformations, like scoping (Bar-Haim and Wolf, 2020), to include more motion patterns. Subsequent fine-tuning on the testing data set helps with performance. Unsupervised learning provides another approach to dealing with the scarcity of labeled optical flow datasets.

The difficulty in unsupervised learning lays more in the design of the loss function compared to supervised methods. Ahmadi and Patras (2016) introduced the first unsupervised deep convolutional neural network for estimating optical flow and achieved similar performance to FlowNet. They used the brightness constancy equation underlying the classic optical flow methods as a loss function. Other methods also included the smoothness constraint on the loss function (Jason et al., 2016; Ren et al., 2017). Later methods explored taking occlusions into account through calculating both forward and backward motion and using the disocclusion mask of the backwards flow as an occlusion mask for the forward flow (Janai et al., 2018). P. Liu et al. (2019) present the current state-of-the-art in unsupervised learning of optical flow, called SelFlow. SelFlow, utilizes a student-teacher network where the teacher network tries to provide better information on pixels in occluded regions. The teacher network output is superpixelized and noise is added to random super pixels as to mimic the effects of occlusion by that super pixel on neighboring pixels. The student network then can be trained to predict optical flow in occluded regions as the ground truth optical flow in the 'occluded' superpixels are known.

6.1.3. Neural Event-based Vision for Visual Odometry

Event-based vision algorithms are still in their early stages due to the recency of the availability of event based vision sensors to the wider research community. The remarkable characteristics (low latency, high dynamic range, non-redundant data and low power consumption) of event-based vision sensors have also attracted research into applications for visual odometry. Early works mostly focused on using methods that are known to work very well on traditional frame-based cameras (e.g. convolutional neural networks) and adapted event representations to fit these traditional methods, however often losing the very advantages of event-based vision in the process. Ye et al. (2018) present the first fully event-based deep learning pipeline for odometry estimation. They do not make use of intensity images as a supervisory signal and instead use photoconsistency assumptions on the event data itself. They use event images as representation for event data however and lose valuable temporal information in the process. A. Zhu et al. (2018) present the first event-based unsupervised optical flow estimator. They make use of the traditional encoder-decoder architecture to extract optical flow estimates. The network is self-supervised by photoconsistency assumptions from regular intensity images. In follow-up work, A. Z. Zhu et al. (2019) make use of a different event representation to retain more temporal resolution by discretizing the time domain, which they call the discretized event volume. Furthermore, they make use of the motion compensation technique through contrast maximization introduced by (Gallego, Rebecq, et al., 2018). However, they use stereo imagery, making it less applicable to this project's use case. Mitrokhin et al. (2019) focused on the segmentation and pose estimation of dynamic objects. In order to achieve this, depth, optical flow and egomotion are estimated simultaneously. A shallow neural network was developed to perform the same task with similar performance, but failing to produce reliable motion estimates.

SNNs naturally lend themselves to processing event data, being able to deal with its sparse event (spike) inputs. Combined with neuromorphic hardware, this has the potential to achieve very low power, low latency visual navigation. Full 6-DOF odometry/SLAM using event-based vision and neuromorphic hardware has not yet been demonstrated and poses an interesting challenge for the foreseeable future. For now, early methods combining event-based vision with SNNs focus on solving sub-problems like optical flow estimation (Paredes-Valles et al., 2019; C. Lee et al., 2020), constraint pose estimation (Gehrig et al., 2020) or constrained navigation (Kaiser et al., 2016; Kreiser et al., 2018). Application are still limited to specific cases, e.g. divergence-based landing of a MAV (Hagenaars et al., 2020), which shows great potential however as control commands from only a single spiking neuron are shown to be able to perform safe landings.

6.1.4. End-to-end learning of Monocular Visual Odometry

The previously discussed methods have subsequently been used in neural visual odometry pipelines, where depth and optical flow are used collaboratively to augment results for egomotion estimation. Yin and Shi (2018) propose the first unsupervised end-to-end visual odometry model, GeoNet, that estimates depth, optical flow and camera pose while being able to handle non-rigidity and occlusions. GeoNet consists of three neural networks: DepthNet, PoseNet and ResFlowNet. The DepthNet and PoseNet are used to reconstruct the rigid structure of the scene which is then fed to the ResFlowNet which computes both forward and backward optical flow. For an overview of the GeoNet architecture, see Figure 6.5. As the network shows capabilities to learn high level features in the scenes, state-of-the-art performance is reached but this also has as a result that this method generalizes less well to novel scenes. Chen et al. (2019) make use of self-supervised learning in their Geometric Learning Net (GLNet). Chen et al. (2019) use loss functions that capture the photometric constraints as a way of self-supervising the network. Additionally, camera intrinsics are estimated with a neural network which allows their method to generalize better to uncalibrated cameras. The combination of optimizing the network output, subject to the geometric and photometric constraints can be seen as a proxy for global bundle adjustment and increases results considerably. They propose two methods for transfer learning: one through finetuning the full network and one where only the output of the network is optimized through using the gradient of the output loss. Feng and Gu (2019) (SGANVO) breaks from the tradition of using encoder-decoder type network architectures and makes use of stacked generative adversarial networks. Stacked networks have been shown to increase accuracy in predicting optical

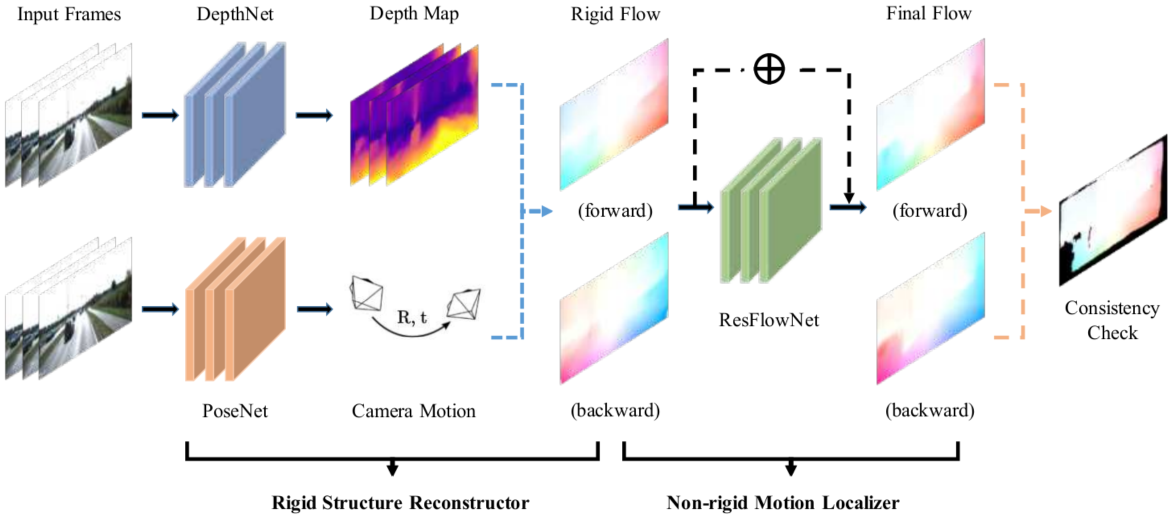


Figure 6.5: Overview of GeoNet structure. GeoNet consists of a rigid structure reconstructor that constructs a depth map and recovers the pose of the camera and a non-rigid localizer for localizing dynamic objects. A consistency check applied to the bidirectional flow is performed to deal with occlusions and non-Lambertian surfaces. Retrieved from (Yin and Shi, 2018).

flow (Ilg et al., 2017) while retaining similar structure and is here used to compute depth and ego-motion in the lowest layer from spatial features in higher layers. The addition of recurrent connections allows the architecture to capture dynamic phenomena in image sequences. SGANVO is able to achieve comparable results to state-of-the-art methods (slightly worse than GLNet (Feng and Gu, 2019)).

Other methods try to directly derive visual odometry from visual information, without making use of intermediate estimation like scene depth or optical flow. The first neural based method was proposed in Mohanty et al. (2016), which used ground truth poses from the KITTI dataset to regress odometry directly from images. They deployed two parallel AlexNet-like architectures; their output was concatenated to generate fully connected layers. Li et al. (2018) deploy an unsupervised learning scheme to recover both scaled depth estimates and odometry that is trained using supervisory stereo images (Figure 6.6). During inference, only monocular image sequences are provided. The projective photometric

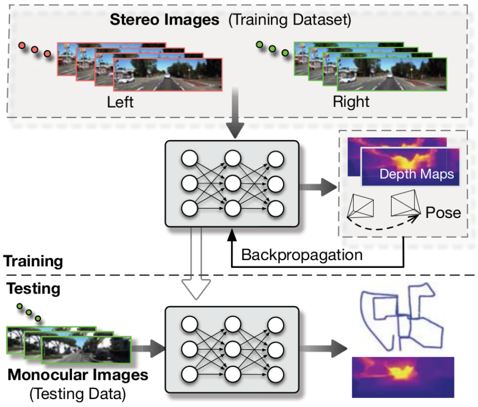


Figure 6.6: Overview of the UnDeepVO architecture. UnDeepVO recovers a depth map from stereo imagery for scale recovery. In parallel, the camera pose is estimated. Retrieved from Li et al. (2018)

error left-right stereo image pairs are used as a photometric loss. Additionally, poses are estimated for both stereo image pairs, which should result in the same estimated pose and can thus be used as a loss. A photometric loss between consecutive frames was applied with an additional 3D geometric registration loss, which uses the depth information from the depth estimation network to derive a loss between consecutive frames. This method has the advantage of not needing labeled ground truth data but still shows degraded performance when deployed in novel scenes as stereo imagery is required to recover the scale and depth estimation in their model.

6.2. Neural-based Route Following

Neural-based route following is split up into three distinct categories. The first one (Section 6.2.1) concerns itself with following tracks. Track-following discerns itself from other route-following methods by having priorly known features/objects in the environment that can be tracked as to successfully follow a route. Path following (Section 6.2.2), in contrast, also attempts to follow a route by tracking features but these features are often difficult to classify, e.g. following a trail through the woods. Finally, insect-inspired methods (Section 6.2.3) are presented that present a more generic route-following, solely relying on views (also called snapshots) perceived during a previous run, which do not allow for specific route-dependent features to be tracked.

6.2.1. Track Following

Fueled by competitions like the IROS Autonomous Drone Race (Moon et al., 2017) and the AlphaPilot Innovation Challenge, methods for autonomous drone racing has seen much development. In these drone races, a drone has to autonomously navigate as quickly as possible through a sequence of gates. Many of the algorithms have thus focused on detecting and locating these gates (Figure 6.7), where its pose is estimated and a local control law (VIO) is utilized to navigate through the center of the gates. Neural-based approaches (Kaufmann et al., 2018; Jung et al., 2018; Loquercio, Kaufmann, et al., 2020) have shown promising results towards this cause. Jung et al. (2018) propose a deep learning model, based on Single-Shot Detection (SSD) (W. Liu et al., 2016). They adapted the SSD model by using AlexNet as a backbone and removing redundant layers to achieve realtime computation on an NVIDIA TX2 board. They combined the gate detection algorithm with a line-of-sight control law to navigate through the centers of the gates. One limitation was that constant line-of-sight of the next gate was required to successfully navigate. Kaufmann et al. (2018) solved this by implementing an extended Kalman filter to estimate gate poses. The estimated poses of the gates are used in a receding-horizon trajectory planner for navigation. The convolutional neural network is based on the DroNet architecture (Loquercio, Maqueda, et al., 2018) to realize realtime inference. Because gate poses are estimated by the EKF and waypoints are generated for each gate, the drone does not require line-of-sight with the next gate and can handle displacements of the gates, as long as a single demonstration flight can be carried out. Loquercio, Kaufmann, et al. (2020) employ a similar setup as (Kaufmann et al., 2018) and train their model in simulation such that it is able to transfer to the real world without adaptation. To achieve this zero-shot sim-to-real transfer, extensive domain randomization was performed during simulation, with changes in illumination, texture of the background and the appearance of the gates to pass through. They make use of an expert policy in simulation that follows a minimum-snap trajectory through the gates and outputs ground truth normalized image coordinate reference directions and a desired speed. To deal with deviation from the trajectory that could arise during real-world flight, a

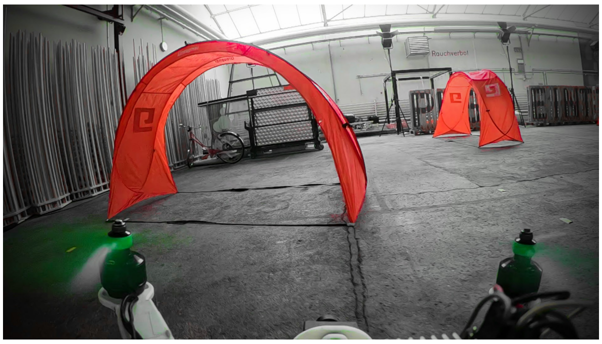


Figure 6.7: Typical drone racing setup: a drone needs to navigate through (dynamically moving) gates on a track. Retrieved from Kaufmann et al. (2018)

variation on DAgger (Ross et al., 2011) is used, which recovers the drone using the expert policy, when deviating too far from the reference trajectory. After the drone can reliably complete the trajectory, the margin where the expert policy is used for recovery is increased. This allowed the drone to behave more robustly when following the trajectory. To deal with dynamic movement of gates, the network was trained simultaneously on multiple static variations of the same track layout.

6.2.2. Path Following

The work presented by Giusti et al. (2016) focuses on following an inconspicuous forest trail through the mountains. Many previous methods had focused on navigating along more clearly defined paths like paved roads, or assessed general traversability. They train a deep convolutional neural network on over 17000 image frames that were collected during approximately 7 km of hiking trails with varying weather and illumination conditions (not during twilight to avoid motion blur). Images along the trail were collected by three cameras mounted on the head of a hiker, with one of the cameras looking forward in the direction of walking and the other two pointing 30 degrees to the left and right. The images collected in the center were labelled for learning as 'go forward', while the other were labelled as 'go left/right' for respectively the right/left pointing cameras. The network thus learns to map perceived views to discrete actions that keep the drone along the trail. The network consists of successive pairs of convolutional layers followed by max-pooling layers followed by a fully connected layer to the three layers, with each a softmax activation function. The whole model ran fully onboard at 15 fps on a custom-designed drone with an Odroid-U3 computer that runs both the deep neural model and a visual odometry pipeline. The proposed methods showed human-level classification performance and was able to navigate along several hundred meters of previously unseen forest trails, but failed however at parts of the trail with little space on either side. Smolyanskiy et al. (2017) based their work on (Giusti et al., 2016), but introduced three additional classes that denote lateral shifts: shift left, centered and shift right. This learns the drone to fly near the center of the trail, resulting in less issue with narrow trails as reported by (Giusti et al., 2016). Furthermore, they introduced a new network architecture, TrailNet (see Figure 6.8), based on ResNet-18, but without batch normalization, and using shifted ReLU instead of regular ReLU.

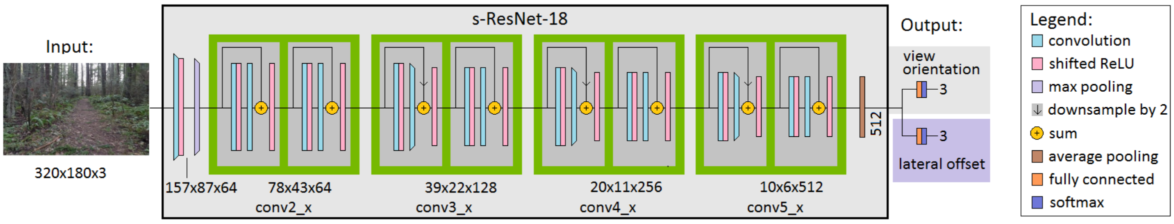


Figure 6.8: TrailNet Architecture, based on ResNet-18. Smolyanskiy et al. (2017) replace the regular ReLUs with shifted ReLUs and do not use batch normalization. Low resolution images of the forrest trail are used to train the network to discern which orientation or lateral offset is necessary to stay on the trail. Retrieved from Smolyanskiy et al. (2017).

Additionally, an object detection model was implemented alongside with a visual odometry pipeline that computes a semi-dense depth map for obstacle detection and avoidance. Their model was overfitted and produced very high confidence values for the different classes, this had as a result that the drone switched between movement classes too late. Therefore, an extra entropy award term was added to the loss function, which consists of the aforementioned entropy award, a cross-entropy loss and a side swap penalty. All modules were run in realtime on a Jetson TX1 at 30 Hz, allowing for reliable autonomous navigation on more than 1 km of forest trails.

6.2.3. Neural Insect-Inspired Navigation

(J. Müller et al., 2018) utilized Ardin et al. (2016)'s Mushroom Body Model for evaluating route following performance in three different simulated environments and extended the model to accommodate behavioral context as an input. The addition of behavioral context as input was accomplished through the addition of an array of activations (context-dependent projection neurons (cPNs)) in parallel to the visual input, that constituted 30% of the total network input. These cPNs represented a categorical internal state of the honeybee (outbound/inbound flights) and were used to give the agent the (biologically plausible) capacity to distinguish between ambiguous routes. An overview of their architecture is given in Figure 6.9. The agent was tested in simulation in three different environments: a flat world

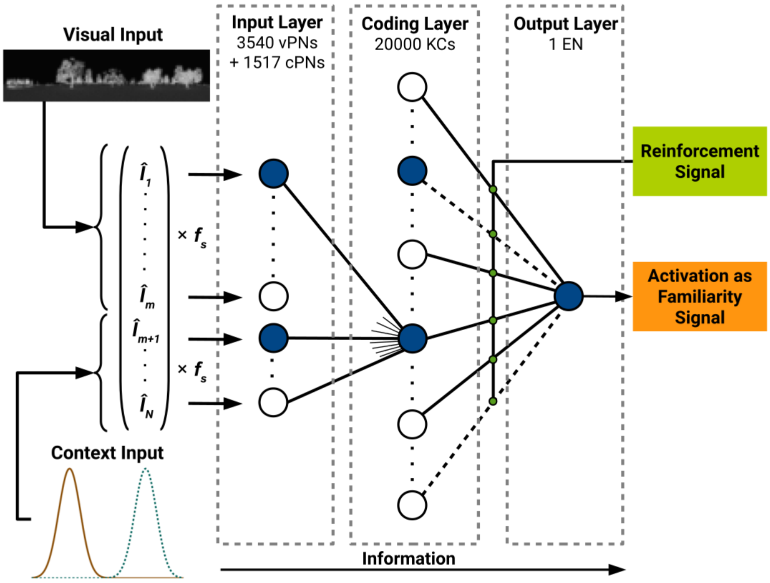


Figure 6.9: J. Müller et al. (2018)'s neural model, which is based on (Ardin et al., 2016)'s MB circuit. cPNs are added to provide context to the network such that it can discern ambiguous routes. Retrieved from J. Müller et al. (2018).

and a low (LD) and high (HD) density environment with trees, bushes and rocks scattered over the 3D environment. A series of experiments was performed that compared navigational performance in the flat world for respectively a straight route and a detour, a series of differently shaped routes in the LD and HD environments and finally an experiment to evaluate the performance of the addition of the cPNs in discriminating between two routes that share a part of their path. J. Müller et al. (2018) note that the model performed best in the LD (uncluttered) environment. In uncluttered environments, relatively more information about the environment can be encoded in the KCs. In a feature-deprived environment however, e.g. the flat world, almost everything looks familiar and performance worsens again. Additionally, results indicate that travelling along extended landmarks (rivers, roads, ...) improves guidance performance but when not many additional features in the environment are present this could result in the agent getting stuck to the extended landmark. The addition of context cues (through the cPNs) allows the agent to discriminate between different routes that share a part of their path.

Knight et al. (2019) implemented the Infomax neural network of (Baddeley et al., 2012) on a custom robotic platform equipped with a Jetson TX1 computer, showing that their model could also be implemented in real natural scenes in realtime. The neural scene familiarity model showed more robust performance compared to using a 'Perfect Memory' (move in the direction of the minimum of Rotational Image Difference function) in some cases. The experiments were conducted over small distances however and required scanning in multiple direction. Additionally, the model took 500 ms to run on the CPU of the Jetson TX1, which would not be suitable for MAV navigation, unless operating at slow speeds.

Nowak and Stewart (2019) propose a spiking neural model for desert ant visual navigation. The route is split into multiple segments, where the agent navigates between waypoints by means of the average landmark vector model and utilizes local vector navigation when leaving a waypoint. Their model assumes that knowledge about the average landmark vector at each waypoint is already present and uses the spiking net's output as an indicator for steering to a waypoint and signalling when a waypoint has been left. Furthermore, despite that computations are performed by a spiking net, there is little resemblance to known neural structures in the desert ant's brain (they do use a model of the basal ganglia, which appears to play a prominent role in vertebrate navigation along a habitual route) and rely on careful gain selection to balance out the attraction of the ALV model and the local steering vector. However, it shows that spiking nets can be successfully deployed to perform view-based navigation tasks.

7

Literature Synthesis

This Chapter provides a synthesis of the literature study that has been carried out. The goal of this study was to give an overview of and review insect(-inspired) vision-based navigation models for route following. The literature study focused on several aspects. First, local vision-based guidance models that do not rely on global navigation satellite systems nor pre-made maps were discussed, with the goal to give an overview of the relevant techniques that are employed. Also, the catchment area of snapshots was discussed, which serves as an important factor in evaluating snapshot-based navigation methods. Next, insect physiology and behavior of ants were presented which gave insight into how insect capture and process visual information relevant to their navigational capabilities. Navigational behavior, specifically learning walks/flights, where discussed as they form an intrinsic part of insects' navigational strategies and facilitate their navigational performance. Additionally, basic principles of event-based vision and how to represent and process them were discussed. Finally, recent developments in neural-based visual navigation methods were discussed. These neural-based implementations show promising results in various navigational applications, and have started to outperform traditional methods with comparable computational requirements and the promise of even more parsimonious methods through neuromorphic computing. The use of (spiking) neural networks for insect-inspired visual navigation on robotic platforms was shown to be often limited to simulation or small navigation tasks.

7.1. Local Vision-Based Guidance

Navigation in GNSS-denied environments or without a prior map requires local navigation strategies. Local vision-based guidance methods can be generally split up into map-building (SLAM), and mapless navigation. SLAM simultaneously constructs a map of the environment and navigates through it by estimating the camera pose and matching and tracking features in the environment. Keyframe-based methods have become the most popular in state-of-the-art SLAM architectures (Younes et al., 2017; Huang, 2019), which either act directly on the individual pixels of frames (direct methods (Engel et al., 2014; Caruso et al., 2015)) or indirect methods (Mur-Artal, Montiel, et al., 2015; Leutenegger et al., 2015; Qin et al., 2018) which use features descriptors (BRIEF, SURF, SIFT, ORB) to match distinct features between keyframes. The requirements of constant pose estimation and successful matching of features requires considerable computational resources and can fail in challenging environments with fast motion, low texture and changes in illumination. Furthermore, it is debated whether insects retain metric/topological internal maps or solely rely on a more parsimonious implementation such as merely matching geometric memories to directions of travel (Cheung, M. Collett, et al., 2014; Webb, 2019), although literature seems to indicate the latter (Rüdiger Wehner, 2008).

Mapless navigation methods do not build or make use of metric/topological maps but instead relate the current experienced view, or a holistic representation of it, to a view experienced at the goal location (also called *snapshot*) and subsequently derive a direction of travel. Following (Möller and Vardy, 2006), monocular vision-based methods can be divided into correspondence and holistic methods. Correspondence methods, analogous to (in)direct methods, compute the vector that would match the transformation of matched regions in a stored snapshot to regions in the current view. The original snapshot model of bee navigation (Cartwright and T. S. Collett, 1983) belongs to this group and matches edges of dark regions on a panoramic 1-dimensional snapshot to derive a travel direction. Other correspondence methods like differential flow methods (Vardy and Moller, 2005) and methods without feature selection found their way to applications in mobile cleaning robots (Vardy and Moller, 2005) and visual road navigation (Pink et al., 2009). Holistic methods like image warping (Franz et al., 1998; Möller, Krzykawski, et al., 2010; Möller, 2012), parameter methods (Lambrinos et al., 1998; Baddeley et al., 2012; W. Stürzl and Mallot, 2006) and descent in image distance methods (Möller and Vardy, 2006) act on a holistic representation of the image. The main advantage of this is that visual information is represented in a more parsimonious manner, which allows for relatively simple deduction of navigational strategies. The power lays in the way that this visual information is captured. The average landmark model (Lambrinos et al., 1998) for example only captures the average of all the vectors pointing to landmarks at the snapshot location and thus only knowledge about the average landmark vector at other positions and a strategy for minimizing this difference is needed. Image warping works well in the case of 1D visual strips (Franz et al., 1998), but gets computationally very intense for 3D scenes and requires dedicated methods for taking advantage of the image warping methodology (Möller, Krzykawski, et al., 2010; Möller, 2012). Rotation invariant methods using Fourier transformed (for 1D panoramas) images and Zenrike moments (for 2D segmented sky panoramas) were also explored as other methods require an external compass for view alignment. Then the scene familiarity (Baddeley et al., 2012) and its neuromorphic implementation by Ardin et al. (2016) showed that a strategy that involves moving in the most familiar direction can result in successful route following. Although their implementations are different (Ardin et al. (2016) use an Infomax net to decorrelate outputs and Ardin et al. (2016) a sparse encoding of information) they rely on the same underlying principle. An extension on this notion was presented by (Le Möel and Wystrach, 2020) that uses both goal *and* anti-goal views as an effort to eliminate the required scanning behavior of Ardin et al. (2016) and Baddeley et al. (2012). In this way a current view, when compared to the two attractive and repulsive memory banks, correlates with directional error.

Work by Zeil, Hofmann, et al. (2003) showed that image difference functions (the root-mean-square difference of pixel values) varies smoothly in natural, outdoor scenes. By descending in the direction of minimal image difference, one can navigate towards a snapshot. In this case the image difference function serves as a proxy for the visual information that is present in the scene, which all mapless guidance methods inadvertently use. This proxy is thus interesting as a way of capturing the information content that is present in scenes for navigation. A key parameter is the so-called Catchment Area/Volume, denoting the area/volume where an agent can successfully return to the snapshot through DID. From analyzing the root mean squared image difference surface, it is seen that in natural scenes, image difference functions appear to be smooth, without pronounced local minima, reaching similar values surrounding the reference image location. This smoothness of image difference functions depends most likely on the spatial-frequency distribution, where a broad variance in contrast, object distance and angular size contribute to its smoothness (Zeil, Hofmann, et al., 2003). Many insect-inspired visual navigation methods are evaluated in simulation which often lack the rich texture, color, luminance contrast, depth variance and noise present in natural outdoor scenes (Zeil, Hofmann, et al., 2003). Furthermore, 3D scenes in simulation that try to accurately capture and model natural environments, still show deficits in representing those environments (Wolfgang Stürzl, Grixa, et al., 2015). Therefore, it is important to perform real-life experiments to evaluate the performance and applicability of these algorithms.

7.2. Insect-Inspired Perception, Processing and Behavior

The main visual organ of insects is the 'compound eye', which consists of numerous patches (ommatidia) that are stacked in a hexagonal pattern. Each of those ommatidia has a single corneal lens that focuses light on the underlying light-sensitive retinula cells. These retinula cells react asynchronously to luminance changes in a spike-based manner, where each cell is sensitive to a certain wavelength (Lebhardt and Desplan, 2017) and polarization of light (Zeil, W. A. Ribi, et al., 2014). Specialized regions such as the DRA focus on specific functions (polarization vision in the case of the DRA). The relatively low resolution view, that is a result of the physical spacing of the ommatidia, combined with a field of view of up to almost 360 degrees plays an important role in facilitating robust visual navigation (Wystrach, Dewar, et al., 2016). Furthermore, compound eyes possess a very high flicker fusion rate (350 Hz) (Ruck, 1958) as result of the sparse input to the insect brain, which allows them to react very quickly to visual scene changes.

Processing of visual information happens largely in the optic lobes and protocerebrum. The optic lobes (lamina, medulla, lobula) are responsible for pre-processing (contrast enhancement (Zettler and Järvilehto, 1972), color vision (Bausenwein et al., 1992), motion detection (Ibbotson et al., 1991)) of visual signals before they pass to the protocerebrum. The Central Complex has been associated with the integration of different visual signals for navigation (Hoinville and Rüdiger Wehner, 2018), where it presumably combines path integration, polarization vision and signals from the MBs (X. Sun et al., 2019) to orient itself in the environment (Webb and Wystrach, 2016). The MBs form a parallel structure of neurons (Kenyon Cells) that are thought to be involved in associative learning (Aso et al., 2014) and connected to olfactory learning in houseflies (Szyszka et al., 2005). This is also thought to form the neural substrate for visual encoding of route memories (Ardin et al., 2016), due to its parallel structure.

The attractive properties of processing visual information in an asynchronous, spike-based manner can be captured through its artificial (neuromorphic) counterparts like event cameras, SNNs and neuromorphic computer chips such as Intel's Loihi or IBM's TrueNorth. Event cameras, like compound and camera-type eyes, capture brightness changes asynchronously and on a per-pixel basis. Collecting visual information in this manner requires the development of novel techniques to represent and process events. Inspiration in terms of (pre-)processing this data can be taken from their neurological implementation in insects. Their artificial counterpart, SNNs, are subsequently a natural fit for processing the sparse and event-based output of event cameras; combined with dedicated hardware they possess the capability to achieve similar, possibly better, performance than traditional artificial neural networks at a fraction of their power requirements (Pfeiffer and Pfeil, 2018).

Navigation strategies also play an important role in the successfulness of guidance methods. A technique commonly deployed by insects is the use of learning walks and flights. Central-place foraging insects are typically seen to leave the nest in a series of saccadic movements that expand outwards of the nest in increasingly bigger arcs/circles while regularly looking back at the nest (Thomas S. Collett and Zeil, 2018). Insects are also seen to orient these arcs/circles depending on landmarks close to the nest (Thomas S. Collett and Zeil, 2018). Acquisition of visual memories is not only limited to the early stages but extend itself throughout the life of the insects. Foraging insects are triggered to perform additional learning maneuvers upon having difficulties or failing to home to their nest. Another technique is to explore the neighborhood of the nest opposite the direction of departure, presumably to make sure that they can return to the nest when overshooting the nest (Jayatilaka et al., 2018). Navigational behavior and performance seems to be closely integrated with each other, and therefore it would be of interest to investigate its applicability to MAV navigation.

7.3. Neural-Based Visual Guidance

The use of ANNs for estimating depth, optical flow and odometry have recently overtaken traditional methods in terms of performance. Mainly deep convolutional neural networks based on architectures like ResNet (He et al., 2016), VGG (Simonyan and Zisserman, 2014) and MobileNet (Howard et al., 2017) are deployed. Deep learning based methods showed its capability of learning depth information from scenes and resolving the depth ambiguity problem, which traditional Visual Odometry methods suffered from. Early methods focused on using supervised learning with labeled data (Eigen et al., 2014). Due to the difficulties in obtaining high quality ground truth depth estimates, methods shifted to semi-supervised (Amiri et al., 2019), and later unsupervised methods (Casser et al., 2019).

Early methods of determining optical flow mostly focused on replacing parts of traditional methods (e.g. feature extraction) that could then be fed to traditional pipelines (Güney and Geiger, 2017). This was later replaced with direct estimates for optical flow. Analogous to depth estimation, these methods made use of supervised learning in early stages (Dosovitskiy et al., 2015) and later unsupervised learning (Ahmadi and Patras, 2016). In order to improve performance, often egomotion and optical flow are learned simultaneously (A. Z. Zhu et al., 2019; Yin and Shi, 2018; Chen et al., 2019). Other methods focus on directly estimating visual odometry (Mohanty et al., 2016; Li et al., 2018). Although better performance was achieved over traditional methods, they still often lack performance in novel scenes. In order to cope with this, many methods first train on artificial datasets and later transition to real world datasets like KITTI (Geiger et al., 2012). For deployment to novel scenes, Chen et al. (2019) propose online optimization through fine-tuning of parameter weights of their original model (PFT) or through output fine-tuning (OFT) which only optimizes the output without recomputing all the network weights. Furthermore, methods often require powerful hardware, although models have been proposed that are able to run reliably on embedded devices (Wofk et al., 2019; Aleotti et al., 2020; Peluso et al., 2019). These are often achieved through extensive optimization and network pruning. Full 6-DOF odometry/SLAM using event-based vision and neuromorphic hardware has not yet been demonstrated and poses an interesting challenge for the foreseeable future. For now, early methods combining event-based vision with SNNs focus on solving sub-problems like optical flow estimation (Paredes-Valles et al., 2019; C. Lee et al., 2020), constraint pose estimation (Gehrig et al., 2020) or constrained navigation (Kaiser et al., 2016; Kreiser et al., 2018).

Neural-based methods have also found their way in route following. Traditional VO pipelines are combined with the tracking of gate poses in (Kaufmann et al., 2018; Jung et al., 2018; Loquercio, Kaufmann, et al., 2020) for drone racing. Giusti et al. (2016) and Smolyanskiy et al. (2017) present a model for following an inconspicuous forest trail over great distances, where essentially the perception of the trail was mapped to discrete actions that would steer the drone to remain on track. Neural-based insect-inspired methods remain mostly limited to simulation (J. Müller et al., 2018; Nowak and Stewart, 2019), except Knight et al. (2019) who implemented their scene familiarity model on land-based robot and reported real world performance over limited distances.

The advent of event-based vision promises computationally sparse bio-inspired methods for insect-inspired visual guidance, but these have been only applied in limited cases. Insect-inspired navigation models that have been implemented on robotic hardware have also often limited themselves to fairly short ranges. The development of a real world event camera dataset that allows for the comparison of different insect-inspired approaches in terms of performance, robustness and parsimoniousness over long distances followed by an evaluation between neural-based insect-inspired methods seems crucial towards bridging this gap. Part III will implement and evaluate these networks on smaller datasets in order to better understand their workings and limitations and provide a basis for their further testing on the large dataset.

Part III

Preliminary Evaluation of Neural Insect-Inspired Familiarity-Based Navigation Models

8

Methodology

Much research in using (spiking) neural networks for insect-inspired visual guidance are limited to implementations in simulation (Baddeley et al., 2012; Ardin et al., 2016; J. Müller et al., 2018; Le Möel and Wystrach, 2020). However, real life natural scenes are considerably different from simulation (see Section 3.2.3) and thus it is of interest how these models behave in natural scenes. Knight et al. (2019) show that this methodology works in natural scenes, but limit themselves to cover relatively small distances. These preliminary experiments intent to provide a better understanding of recent neural insect-inspired scene-familiarity based navigation models such that they can be evaluated on the real life dataset as presented in Part I.

First, the outline of the experiments are given in Section 8.1. In Section 8.2, the software and tools used for constructing and running the (spiking) neural networks are discussed. An overview of the datasets that have been used in the preliminary experiments are detailed in Section 8.3. In Section 8.4, the image processing that is performed before the frames are passed to the neural networks is described. Section 8.5 introduces two recent insect-inspired familiarity-based neural navigation models which will be used for further evaluations. Chapter 9 presents a preliminary evaluation of the models introduced in Section 8.5 in terms of their performance for long-ranged navigation. A report of the preliminary experiments and a discussion of their results and implication towards this thesis are presented in Chapter 10.

8.1. Outline

As there exists limited research comparing neural familiarity-based insect-inspired visual guidance models for MAVs, this preliminary research focuses on evaluating the neural networks proposed by Baddeley et al. (2012) and Ardin et al. (2016) on vision data from real life natural scenes. Of special concern is how well these models perform in natural scenes, and which learning strategy and network configuration should be deployed to optimize the use of resources while retaining adequate performance. Of special concern are the speed, capacity and accuracy of the networks as these elements are vital towards their applicability onboard limited platform such as a MAV. The effect of parameters such as image resolution, image aspect ratio and learning rate will be explored.

8.2. (Spiking) Neural Network Frameworks

To run the three neural (spiking) models, two different frameworks are used. PyTorch[1] is used for the non-spiking neural network due to its highly modular and dynamic design, accelerated computing

[1] `https://github.com/pytorch/pytorch`

and CUDA support and rich resources in terms of libraries and tools. For simulating the (spiking) mushroom body models, an adaptation[2] of Bas Buller's SNN simulator framework PySNN[3] (which is build upon PyTorch) is used to support the simulation of Izhikevich neurons and the learning rule as presented in Ardin et al. (2016). The PySNN framework allows a user to set up a SNN by defining the neuron's dynamics as a pysnn.Neuron module and the connection between layers as pysnn.Connection modules (opposed to the single nn.Module in PyTorch). As PySNN uses PyTorch as basis, the same workflow for non-spiking and SNN models can be used, while retaining all the advantages of PyTorch.

Listing 8.1: Simulation and learning of a Mushroom Body model in the PySNN framework

```python
import torch
import models.MBModel as MBModel
from pysnn.learning import IzhSTDP

# simulation setup
dt = 1 # timestep in milliseconds

# configure Mushroom Body model
n_in = 360 # number of visual projection neurons
n_hidden = 20000 # number of Kenyon Cells
n_out = 1 # number of extrinsic neurons
MB = MBModel.Network(n_in, n_hidden, n_out) # initiate MB model

# generate 10 input sequences
MB_input = torch.rand(10, n_in)

# learning setup
layers = MB.layer_state_dict()
tau_c = 40.0
tau_d = 20.0
A_plus = A_min = -1.0
tau_plus = tau_min = 15.0
learning_rule = IzhSTDP(layers, tau_c, tau_d, A_plus, A_min, tau_plus, tau_min)

# simulate network for 50 milliseconds
for input in MB_input:
    for t in range(int(50 / dt)):
        BA = 0
        if t == 40:
            BA = 0.5
        MB(input.view(1,1,-1))
        learning_rule.step(BA)
    MB.reset_state()
```

8.3. Event Vision Datasets for Scene Familiarity

Scene-familiarity based insect navigation models are generally trained using a series of views captured at various points surrounding and pointing towards a 'home' location. They are subsequently evaluated by examining their response to a rotation on the spot at an unseen location. The network should ideally show a clear dip in its response when facing in the direction of the goal. As little such datasets exist, two (Home and Cyberzoo) datasets were collected and used for the preliminary evaluations re-

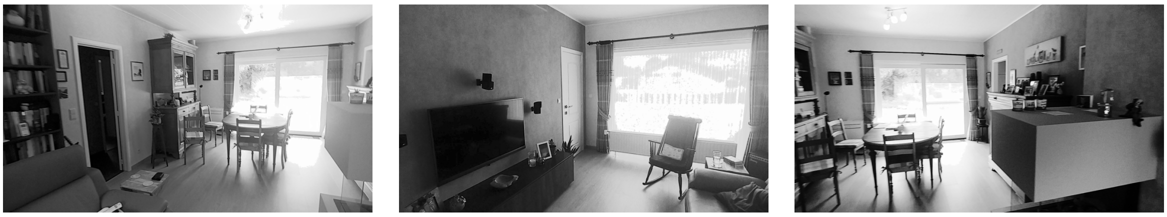
[2] https://github.com/JanVerheyen/PySNN-Izhikevich

[3] https://github.com/BasBuller/PySNN

garding the performance of the Infomax network and the Mushroom body model. Both datasets consist of goal, anti-goal and test views. The goal views are directed towards the goal (left chair in the Home dataset and left pole in Cyberzoo dataset), the anti-goal views are oriented in the opposite direction. The test footage was recorded by rotating 360 degrees on the spot. The Home dataset was recorded with an LG V30+ phone and the Cyberzoo dataset was recorded with the onboard camera of a Parrot bebop 2 drone, along with events from a DAVIS240 sensor.

8.3.1. Home Dataset

The 'Home' dataset consists of only video shot by a phone. The Home dataset consists of high dynamic range (low-brightness inside and external sunlight through the window) and textured (the bookshelf, etc.) footage. A number of sample views can be seen in Figure 8.1.



(**a**) Home, Goal (**b**) Home, Anti-Goal (**c**) Home, Test

Figure 8.1: Samples of the goal, anti-goal and test views from the Home dataset

8.3.2. Cyberzoo Dataset

A small dataset was collected in the MAVLab Cyberzoo that consists of goal- and antigoal-oriented trajectories in five distinct locations around a central goal post. At three other locations, 360 degree scans were performed to assess the response (performance) of the networks. An overview of the trajectories can be seen in Figure 8.2.

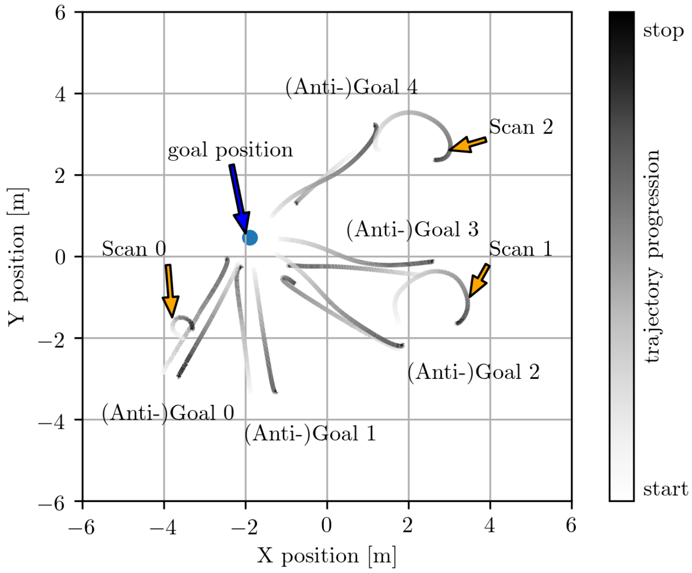
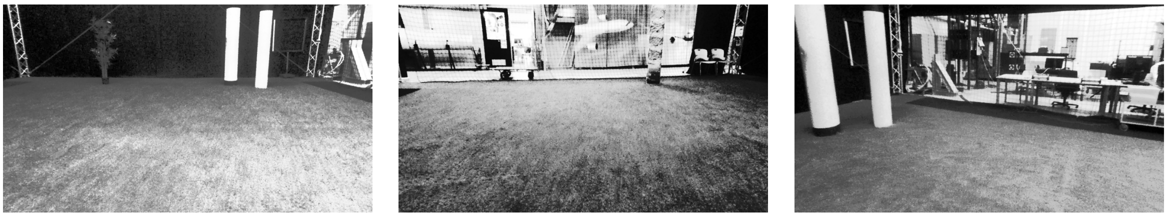


Figure 8.2: Cyberzoo experiments setup. Drone position during dataset acquisition in the Cyberzoo. Time is encoded as color (light red → dark red = start → stop).

In Figure 8.3 you can see samples of both the Cyberzoo datasets for respectively the goal, anti-goal and test views. The Cyberzoo dataset was shot in the Cyberzoo of TUDelft's MAVLab and consists of more recurrent texture (artificial grass, black curtain, ...).



(**a**) Cyberzoo, Goal (**b**) Cyberzoo, Anti-goal (**c**) Cyberzoo, Test

Figure 8.3: Samples of the goal, anti-goal and test views from the Cyberzoo dataset

8.4. Image Pre-processing

In this section, an overview of the image pre-processing steps that are performed on the datasets is given. The datasets (Section 8.3) consist of 1080p color videos in '.mp4' format with H.264 encoding. Each of these videos were split into individual frames, which were subsequently processed as described in Baddeley et al. (2012) and Ardin et al. (2016) using OpenCV:

1. conversion to gray scale with cv2.cvtColor
2. histogram equalization with cv2.equalizeHist for contrast enhancement
3. resize to desired (lower) resolution with cv2.resize using the cv2.INTER_AREA option
4. normalize pixel brightness values (from $[0, 255]$) to the range $[0, 1]$

These images where then saved by storing them as '.png' files. The only additional processing was performed for the SNN model of the mushroom body (Section 8.5.2). Following Ardin et al. (2016), image inputs for the mushroom body model are normalized by dividing each pixel value by the square root of the sum of squares of all pixel values, after which they are scaled by a scaling factor such that a specified amount of Kenyon Cells are activated given by Equation 8.11. An example image after preprocessing can be seen in Figure 8.4.

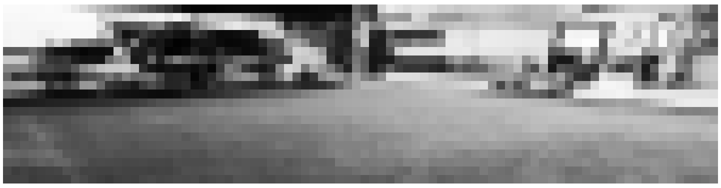


Figure 8.4: Example image from the Cyberzoo after preprocessing (resolution = 84×21).

8.5. Neural Familiarity-Based Insect-Navigation Models

The following section covers the implementation of Baddeley et al. (2012)'s Infomax scene familiarity model and Ardin et al. (2016)'s MB model.

8.5.1. Infomax Neural Network

The Infomax neural network already been covered in detail in Section 3.2.2 of Part II. One note can be made however on the implementation of the weight adaptation which has been slightly modified:

$$\Delta w_{ij} = \frac{\eta}{N \cdot \boldsymbol{M}}(w_{ij} - (y_i + h_i)\sum_{k=1}^{N} h_k w_{kj}) \tag{8.1}$$

Note that in this notation the normalization term $\frac{1}{N}$ in Baddeley et al. (2012)'s work has been replaced by $\frac{1}{N \cdot M}$. The original term required fine-tuning of the learning rate depending on the number of input units N and the amount of novelty unit inputs M to guarantee convergent behavior of the learning step. This novel notation stems from the fact that $(y_i + h_i)$ contains M elements and $(\sum_{k=1}^{N} h_k w_{kj})$ N elements which requires a factor $N \cdot M$ for normalization.

8.5.2. Mushroom Body Model — Ardin et al. 2016

The model presented by Ardin et al. (2016) makes use of Izhikevich spiking neurons (Izhikevich, 2003) where changes in the membrane potential v(mV) are modelled by:

$$C\dot{v} = k(v - v_{\text{rest}})(v - v_t) - u + I(t) + [\zeta \sim N(0, \sigma)] \tag{8.2}$$

$$\dot{u} = a(b(v - v_{\text{rest}}) - u) \tag{8.3}$$

where C is the membrane Capacitance, v_r the resting membrane potential, v_t a threshold potential, I the input current, $\zeta \sim N(0, \sigma)$ Gaussian white noise and a, b, c, d and k are model parameters that control the characteristics of the neurons. The membrane potential v and recovery current u are reset if the membrane potential v_t is exceeded:

$$\begin{cases} v \leftarrow c \\ u \leftarrow u + d \end{cases} \tag{8.4}$$

The input current is modelled by:

$$I = gS(v_{\text{rev}} - v) \tag{8.5}$$

where g(nS) is the maximal synaptic conductance, $v_{\text{rev}} = 0$ is the reversal potential and S the amount of active neurotransmitter. S is given by:

$$\dot{S} = -\frac{S}{\tau_{\text{syn}}} + \phi\delta(t - t_{\text{pre}}) \tag{8.6}$$

where ϕ is a quantile of the amount of neurotransmitter released after the occurrence of a pre-synaptic spike, τ_{syn} is the synaptic time constant, t_{pre} the time at which the pre-synaptic spike occurred and δ the Dirac delta function. During learning, the weights g are altered using a modified three-factor rule:

$$\dot{g} = cd \tag{8.7}$$

where c is a synaptic tag which serves as a transient eligibility trace and d is the extracellular concentration of biogenic amine:

$$\dot{d} = -\frac{d}{\tau_d} + BA(t) \tag{8.8}$$

where $BA(t)$ is the amount of biogenic amine released at time t, depending on the reinforcement signal and tau_d is the time constant of the decay of concentration d. The synaptic tag c is modelled by STDP:

$$\dot{c} = -\frac{c}{\tau_c} + STDP(t_{\text{pre}} - t_{\text{post}})\delta[(t - t_{\text{pre}}) * (t - t_{\text{post}})] \tag{8.9}$$

where $\delta(t)$ is the Dirac delta function, t_{pre} the time of a pre-synaptic spike, t_{post} the time of post-synaptic spike and τ_c the time constant for the decay of synaptic tag c. STDP acts as follows on synaptic tag c:

$$STDP(t_{\text{pre}} - t_{\text{post}}) = \begin{cases} A_+ e^{\frac{t_{\text{pre}} - t_{\text{post}}}{\tau_+}} & \text{, if} \quad t_{\text{pre}} - t_{\text{post}} < 0 \\ 0 & \text{, if} \quad t_{\text{pre}} - t_{\text{post}} = 0 \\ A_- e^{\frac{t_{\text{pre}} - t_{\text{post}}}{\tau_-}} & \text{, if} \quad t_{\text{pre}} - t_{\text{post}} > 0 \end{cases} \tag{8.10}$$

Table 8.1: Neuronal and synaptic parameters of Ardin et al. (2016)'s Mushroom Body model.

	Neuron Properties			
	PN	KC	MBON	unit
neuron number n	360	20000	1	-
resting potential v_{rest}	-60.	-85.	-60.	mV
threshold voltage v_{thresh}	-40.	-25.	-40.	mV
model parameter a	0.3	0.01	0.3	-
model parameter b	-0.2	-0.3	-0.2	-
model parameter c	-65.	-65.	-65.	mV
model parameter d	8.	8.	8.	mA
model parameter C	100.	4.	100.	-
model parameter k	2.	0.035	2.	-
noise ξ	$N(0, 0.05)$	$N(0, 0.05)$	$N(0, 0.05)$	mA

	Synapse properties			
	input to PN	PN to KC	KC to EN	unit
connectivity	one-to-one	10 per KC	fc	-
synaptic weight g	1.	1.	1.	-
neurotransmitter quantile ϕ	50.	0.93	8.	-
synaptic time constant τ_{syn}	1.8	3.	8.	ms
synaptic tag time constant τ_c	-	-	40.	ms
biogenic amine time constant τ_d	-	-	20.	ms

where $A_\pm$ are the magnitudes of synaptic change due to either long-term potentiation and depression and $\tau_\pm$ the time constants. An anti-Hebbian learning rule is applied, ensuring the tag c is always negative such that the network's tagged weights quickly decline to zero. In Table 8.1 an overview is given of the configuration of the parameters of the MB model as presented in Ardin et al. (2016).

Ardin et al. (2016) use 360 visual projection neurons that are sparsely connected to 20000 Kenyon cells which terminate on a single extrinsic neuron. The main working principle behind this network is the sparse projection of the visual projection neurons to the Kenyon cells, where especially their ratio is of importance. Ardin et al. (2016) use a constant scaling factor on image inputs of 5250, in order to activate about half of the visual projection neurons which in their turn activate about 2% of the Kenyon cells (see Ardin et al. (2016, Fig. 5)). For different image input sizes, the same ratio should be adhered to. To make sure that the Kenyon cells' activity remains the same ($\approx$ 2%) however, the input current will have to be scaled the right amount. The following empirical relationship for the scaling factor was found to produce a KC activity of approximately 2%:

$$\text{Scaling factor} = 267.14 \cdot \sqrt{N_{\text{PN}}} + 240 \tag{8.11}$$

with N_{PN} the number of visual projection neurons.

8.5.3. Training

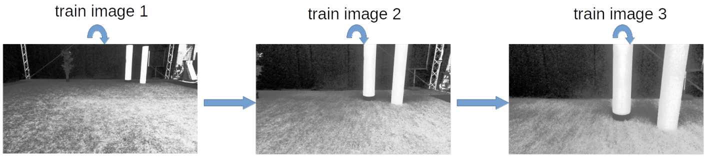


Figure 8.5: Scene familiarity based neural network training procedure. Image 1 is shown, after which the weights are adapted. Next, image 2 is presented to the network and the weights adapted, and so on.

The Infomax neural network's weights are adapted according to Equation 8.1 for each image. The (spiking) MB model is trained by presenting images for 50 milliseconds — this activates the vPNs, KCs and MBON and tags KCs that are firing — after 40 milliseconds biogenic amine (Equation 8.8) is released which reduces the synaptic connection strength between the KCs and the MBON over a time span of 10 milliseconds. The MB model state is reset, after which the second image is shown, etc.

9

Spiking Neural Networks for Familiarity-based Guidance in Natural Scenes

This chapter focuses on evaluating the Infomax and Mushroom body neural networks for use in natural scenes. First, the concept of natural scenes and its relation to and evaluation of the datasets is presented in Section 9.1. Following, the networks will be evaluated with respect to their aspect ratio and resolution in Section 9.3 and 9.2 respectively. The learning strategy will be handled in Section 9.4, finally covering their performance in Section 9.5.

9.1. Information Content of Natural Scenes

The visual systems of vertebrates and insects are adapted to the environment in which they live, which mostly consists of natural scenes. Natural images can thus be considered of as images with a statistical structure to which those visual systems are best adapted to. They can therefore also come from artificial sources (e.g. simulation, drawings, ...) (Dyakova, 2017). Natural *scenes* (which are natural images) represent images from nature. These images can be described in order of their statistics: first-order, which describes images using solely pixel brightness values regardless of their position, and second-order, which take into account the pixels' spatial relationship. The RMS-contrast (first-order) gives a measure for the contrast in an image:

$$RMS = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i-\mu)^2} \tag{9.1}$$

Figure 9.1 shows the RMS-contrast of the frames in datasets 'Cyberzoo' and 'Home' (Section 8.3). It can be seen that there is an about 7.5% increase in average RMS frame contrast in the Home dataset over the Cyberzoo dataset. This result indicates that there is more variability in the Home dataset pixels' values than in the Cyberzoo dataset, requiring a neural network to need more capacity to store the visual information present in the scene. To get more insight into the spatial relationship of the datasets, one has to look at the frequency domain. The 2D Fourier transform of a brightness image is defined as:

$$F(u,v) = \frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)e^{\frac{-\pi j(ux+vy)}{N}} \tag{9.2}$$

where $f(x,y)$ is the 2D array of brightness values, u and v are respectively the number of horizontal and vertical cycles that fit into a single Period of the frame, $F(u,v)$ the Fourier matrix and N the number of

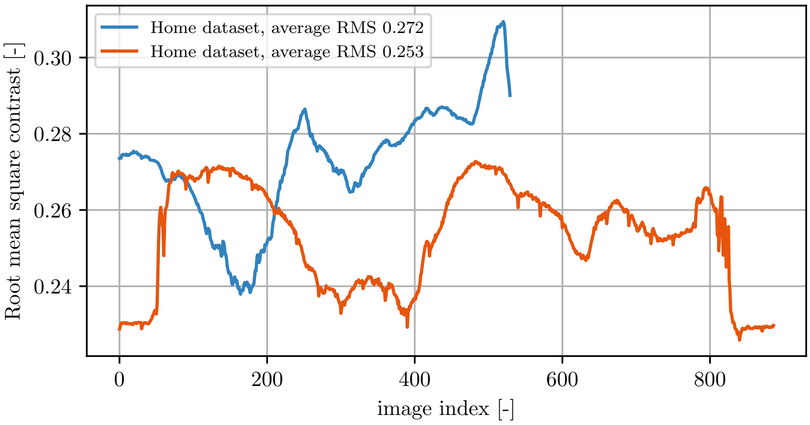


Figure 9.1: Root-mean-square image contrast of the Home and Cyberzoo datasets. The Home datasets shows overall the most image contrast (and thus more information).

pixels. One can subsequently compute the amplitude A and the phase ϕ:

$$A = \sqrt{Re(u,v)^2 + Im(u,v)^2} \tag{9.3}$$

$$\phi = \text{arctan}\Big(\frac{Im(u,v)}{Re(u,v)}\Big) \tag{9.4}$$

where Re and Im are the real, respectively the imaginary part of the Fourier matrix. The amplitude spectrum of natural images can be quantified by a single value, the slope constant α of the orientation averaged amplitude. A higher (lower) $alpha$ value shows there is less (more) fine detail in a scene. On a log-log scale, a linear relationship exists between the spatial frequency (often expressed in terms of cycles/image) and the amplitude (Field and Brady, 1997):

$$A(f) = \frac{c}{f^\alpha} \tag{9.5}$$

Typical natural images have a slope constant that lies in between 0.8 and 1.5, peaking around 1–1.2 (Tolhurst et al., 1992). Figure 9.2 shows that the Home dataset has considerably higher amplitudes for the lower spatial frequencies and similar values for the higher frequencies. This again illustrates the higher diversity of input from the Home dataset, which will require more neural capacity, ergo a bigger network, to store the information adequately. This shows that for equivalent performance, network capacity (size) will need to be adjusted to match the information content of the scene. On the other

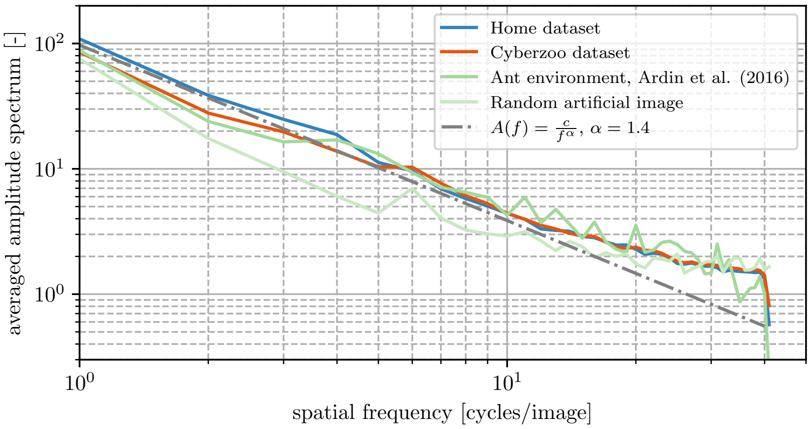


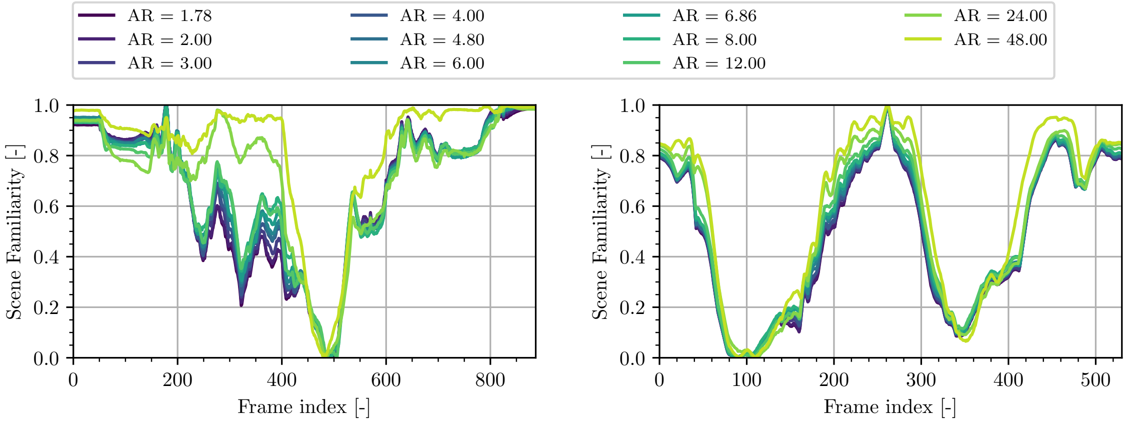
Figure 9.2: Orientation averaged amplitude spectrum of the Home and Cyberzoo datasets, the ant environment used in Ardin et al. (2016), and a random artificial image. An amplitude spectrum with a slope of 1.18 is shown as a reference.

hand, one could say that scenes with a lot of visual information require less information to be stored per view as it is easier to differentiate between different views within this scene. Highly repetitive (low visual information content) scenes would then require more information to be stored as each individual view is not that much different from another view. Finding how much visual information is required to differentiate (in a practicable manner) between views within a scene and how much capacity is needed to store all these views is crucial for the success of this project. As a next step, the extent to which horizontal visual information contributes to navigational performance is evaluated, in a search to minimize the amount of vertical visual information needed for differentiating between views within a scene.

9.2. Aspect Ratio

Many insects primarily rely on path integration for estimating the distance to their nest. It is thus hypothesized that the visual system is mainly used for orientation/directionality in navigation (is also used for other purposes: e.g. wasps are known to use the location of visual landmarks for pinpointing the nest location (Thomas S. Collett and Zeil, 2018)). Insects have a very high field of view (some almost 360 degrees) with more ommatidia (= 'pixels') covering the horizontal than the vertical extent of their view. The above observations have led to the hypothesis that insects mainly rely on visual information in the horizontal field of view for guidance to aid in finding the right direction/orientation; which is shown to work well in e.g. Franz et al. (1998) and W. Stürzl and Mallot (2006). This makes sense as visual information w.r.t. orientation is mainly found in the horizontal extent of pictures. If less pixels in the vertical direction are required for successful navigation, this would benefit the success of deploying the algorithms onboard MAVs as a smaller network would be required.

The Home and Cyberzoo datasets are thus analyzed to investigate the influence of the horizontal field of view on navigation performance. This is done by changing the aspect ratio of the original dataset. Both datasets were captured in a standard 16:9 aspect ratio. Aspect ratios ranging from 16:9 to 48:1 were analyzed. Starting from a resolution of 48×27 (AR = 16:9), the video aspect ratio is modified to 2:1, 3:1, 4.8:1, ...(see legend Figure 9.3) with accompanying video resolution $48 \times 24, 16, 12, ...$ Hence, the same amount of horizontal pixels is used and the same amount of vertical information, but this vertical information is compressed into fewer pixels. Each frame in the 'test' dataset is evaluated using the global minimal sum square error method. The method works as follows: **1.** A database is made of all the 'goal' images **2.** Each image in the 'test' dataset is evaluated in terms of pixel-wise sum square difference ('test' database consists of a rotation on the spot, see Section 8.3) **3.** The image in the 'goal' database with the lowest sum of the square of pixel differences is registered **4.** The results are normalized to the interval $[0, 1]$ **5.** 1 minus this result is what is used as 'scene familiarity' measure. This method will from now on be called the 'perfect memory' method. Figure 9.3 shows the results of this analysis.



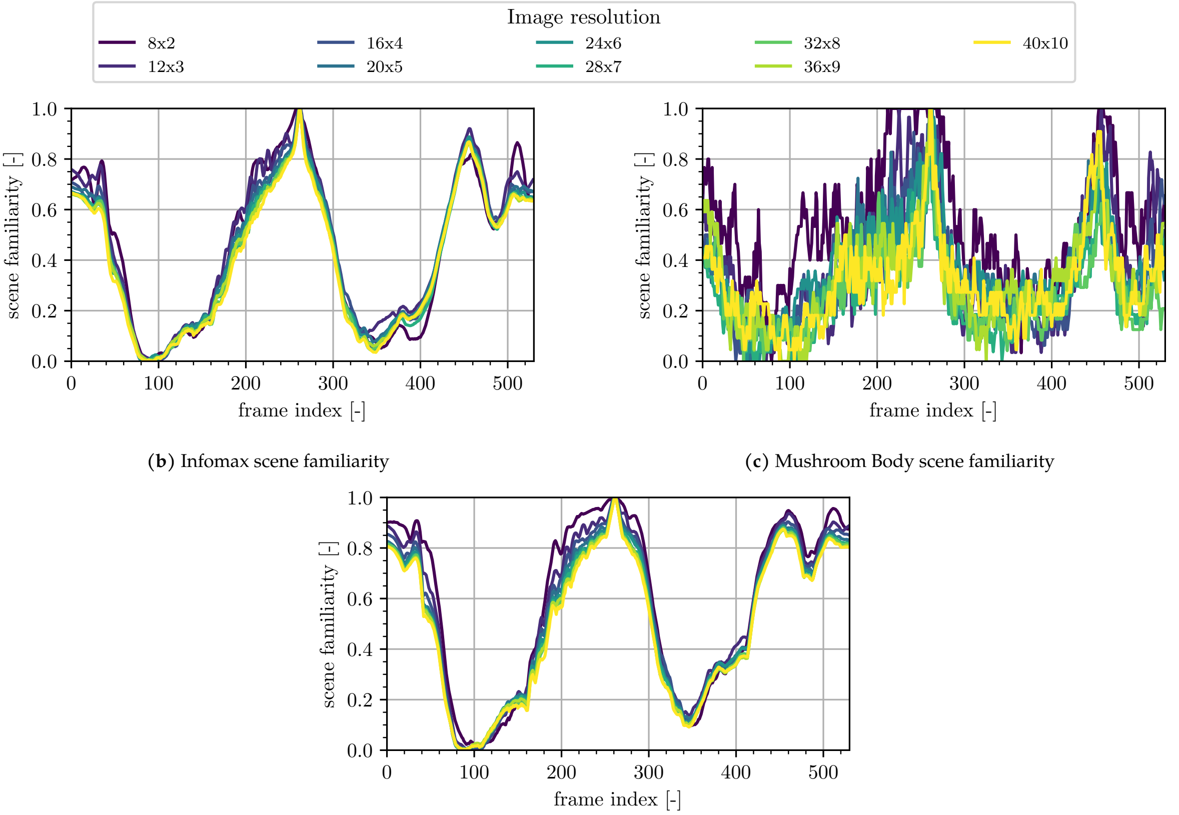
(**b**) Cyberzoo dataset (**c**) Home dataset

Figure 9.3: Influence of image aspect ratio on scene familiarity curve for the Home and Cyberzoo dataset.

It can be seen that, irrespective of the dataset, the aspect ratio of the image does not influence the shape of the scene familiarity curve considerably for aspect ratios of up to 12:1 (48×4). This remains the same for the Home dataset for aspect ratios greater than 12:1. For the Cyberzoo dataset however, a considerable shift occurs for aspect ratios greater than 12:1. The maximal scene familiarity is not reached anymore at ≈ 190 but much later, around the intervals $[300, 400]$ and $[600, 800]$ which illustrates the degradation of the method in this case. This is likely a result of the more repetitive textures and layout in the Cyberzoo environment, which requires additional vertical information to differentiate between frames. The Home dataset on the contrary consists of more varying textures and layout and thus does not suffer from the same degradation. These experiments show that only little information is needed in the vertical direction to achieve similar results to using all visual information, very high aspect ratios however will result in degradation in certain (more repetitive) environments. Aspect ratios of around 10:1 seem to work well for these environments, with similar results expected for other natural scenes.

9.3. Resolution

The influence of image resolution on navigational information content is another important factor. Ideally, image input resolution should be as small as possible to save computational cost. Figure 9.4 shows the influence of image resolution on the shape of the scene familiarity response for the Infomax and Mushroom body model networks trained on the Home dataset with 50 images.



(b) Infomax scene familiarity

(c) Mushroom Body scene familiarity

(d) Perfect memory scene familiarity

Figure 9.4: Comparison of model performance with respect to image resolution. 50 images with an aspect ratio of 4 were used for training. 'Home' dataset.

The results indicate that the Infomax and perfect memory methods do not suffer much from lower input image resolution. The Mushroom Body model however sees some degradation w.r.t. performance for lower input image resolutions. Although the shape of the response remains mostly similar, the maxima and minima of the response are not the same. Especially around the 'true' maximum (frame 260), the lower resolution model returns multiple other maxima. This has probably mostly to do with the fact that the lower resolution and thus smaller sized (e.g. image input 8×2 results in a MB model with 16 vPNs, 889 KCs and 1 EN) MB model can store less visual 'memories', as illustrated in Ardin et al. (2016, Fig. 5). Using more than 50 training images would make the resolution of 8×2 inoperable. Depending on the application, a larger network would thus be needed. Larger networks require more computational power however which would limit the navigational speed of the MAV, making it ultimately inoperable too. In general the following chain of events is linked together: navigation update speed $\rightleftarrows$ process speed $\rightleftarrows$ network size $\rightleftarrows$ storage capacity $\rightleftarrows$ navigation distance. This makes the amount of views that needs to be 'stored' in the network for reliable performance a very important factor. This is investigated in the following section.

9.4. Learning Strategy

Learning walks form an essential part of insects' navigational strategy and capabilities (Section 4.3). Crucial in learning a route representation is how many views are needed in order to achieve an accurate estimate for which direction to take at a newly presented view away from the goal. Baddeley et al. (2012) train on views that are spaced apart 2 cm for the learning walks and 4 cm for the outbound routes. Ardin et al. (2016) uses views that are distanced 10 cm apart for learning and Le Möel and Wystrach (2020) use 25 views along a nest-centered spiral of length 0.5, 2 and 8 meters. However, intuitively, it appears that fewer views could be needed to 'memorize' a route as work from Denuelle and Srinivasan (2016) shows. Much of this discrepancy has to do with scale: small obstacles on the ground are irrelevant for navigating drones but form sizeable landmarks for e.g. ants which have to navigate in between them, requiring less spacing between snapshots. In order to investigate the amount of views that would be necessary to achieve a sense of directionality, the Infomax network was trained on the same dataset but with a varying amount of frames. Results for varying the amount of training images, n, in the Cyberzoo dataset for $n = 200$, $n = 50$, $n = 10$ and $n = 1$ images are shown in Figure 9.5.

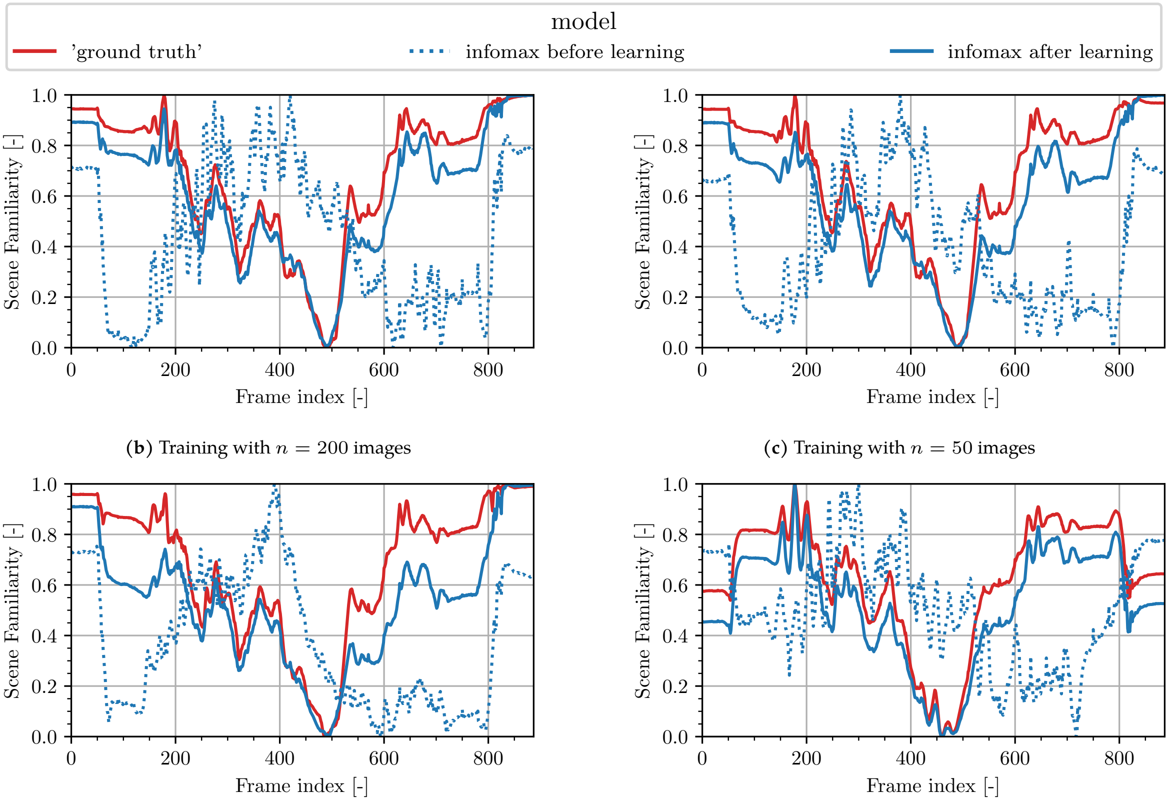
At first glance, Figure 9.5 shows that for different amounts of training images, a similar scene familiarity curve is achieved. Only 1 image actually suffices[1] to find the most familiar frame pointing to the goal location (frame 190), although it also shows for $n = 10$ that crucial information can be missed by different spacing of the training views as it misses the most familiar view. A similar story can be seen for the mushroom body model in Figure 9.6. Here the response for $n = 200$ and $n = 1$ are compared. Again, a single view suffices if taken at the correct position.

Figure 9.7 shows that when extended for a range of resolutions and different datasets, one can see that there is no real advantage in using more images than necessary. One can also see that for very low resolutions there is indeed a certain degradation in performance, but not much more performance is gained with resolutions greater than 24×6.

9.5. Learning the Goal Location

To assess whether the network can not only 'memorize' a sequence of data, but asses the 'familiarity' of a scene, $360°$ scans at two different locations in the vicinity of the goal-oriented runs (scan 0 and 1 in Figure 8.2) were performed to assess this capability. Figure 9.8 shows the normalized difference in spike rate between the network before learning and after learning. While the goal location can be retrieved from the graphs presented in Figure 9.8, it doesn't show a convincingly large difference compared to the output at other time intervals. Its implications will be discussed in Section 10.1.

[1] The room in which the Cyberzoo dataset was collected spans an area of about 5 by 5 meter, resulting in a spacing of about 5 m between images.



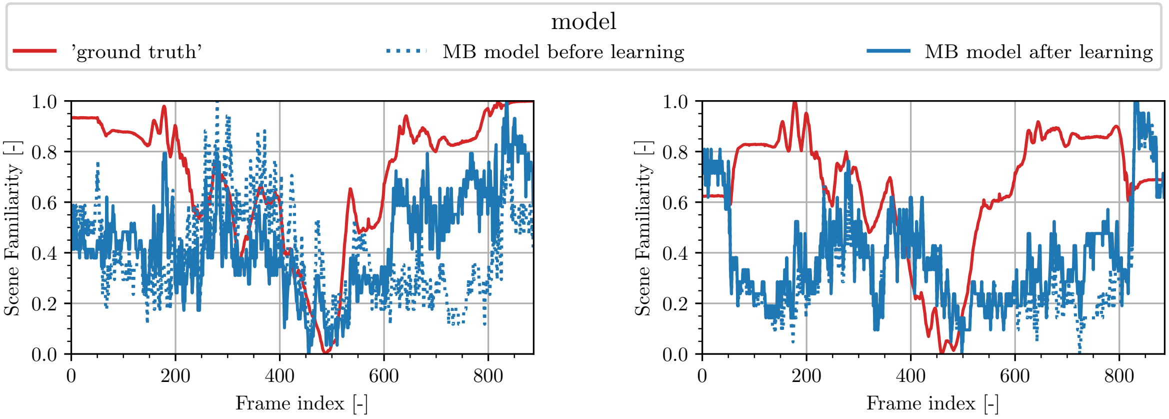
(**b**) Training with $n = 200$ images

(**c**) Training with $n = 50$ images

(**d**) Training with $n = 10$ images

(**e**) Training with $n = 1$ image

Figure 9.5: Influence of the amount of training images on Infomax scene familiarity shape and performance. Shown are the response of the neural network before training, after training and the ground truth response as found with the 'perfect memory' method. Image frames are of resolution 40×10.



(**b**) Training with 200 images

(**c**) Training with 1 image

Figure 9.6: Influence of the amount of training images on Mushroom Body scene familiarity shape and performance. Shown are the response of the neural network before training, after training and the ground truth response computed with the'perfect memory' method. Image frames are of resolution 24×6.

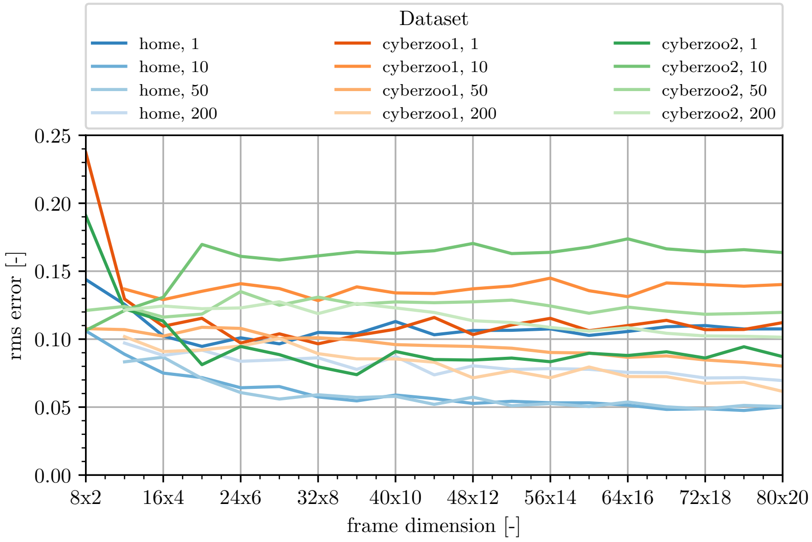
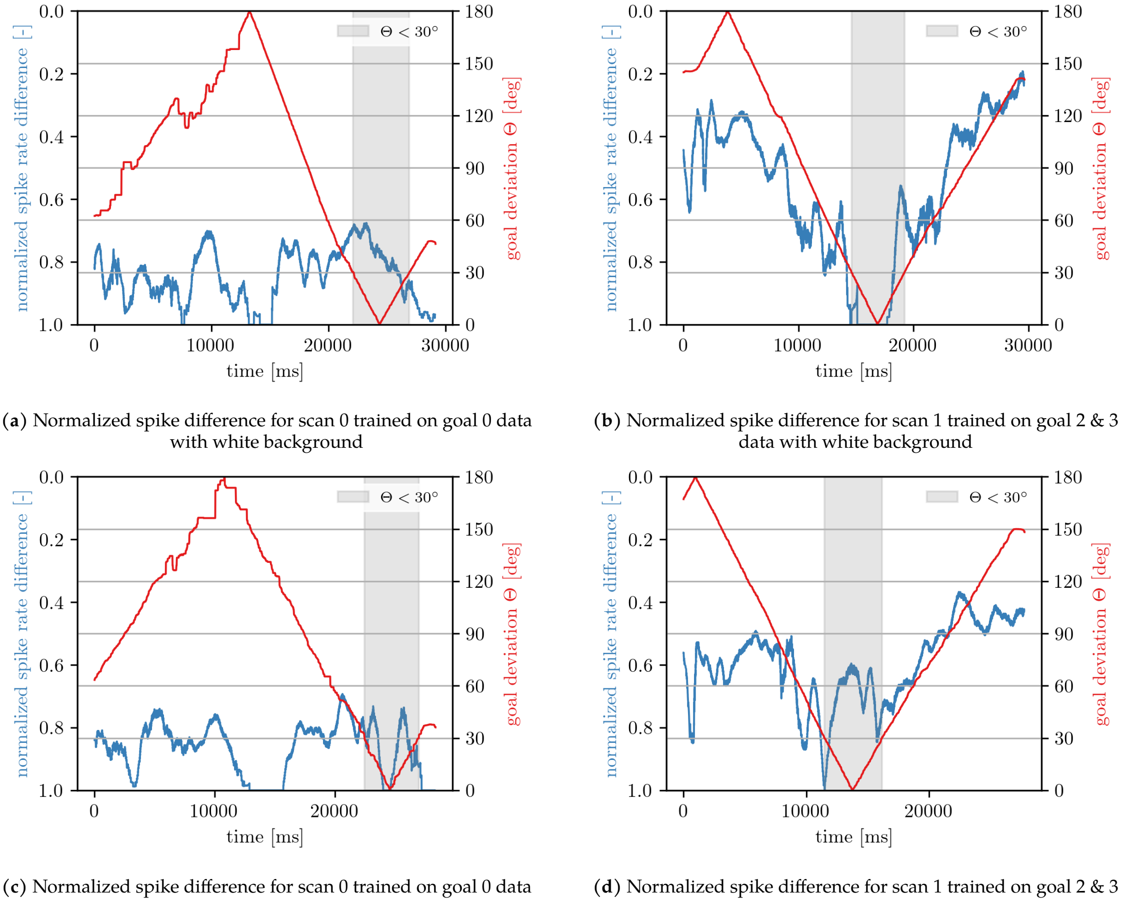


Figure 9.7: RMS error of the Infomax response compared to the 'ground truth' global minimal distance method, depending on the image resolution for different amounts of training images. E.g. home, 1 is dataset 'Home' using 1 image for training.



(**a**) Normalized spike difference for scan 0 trained on goal 0 data with white background

(**b**) Normalized spike difference for scan 1 trained on goal 2 & 3 data with white background

(**c**) Normalized spike difference for scan 0 trained on goal 0 data with textured background

(**d**) Normalized spike difference for scan 1 trained on goal 2 & 3 data with textured background

Figure 9.8: Normalized spike rate difference of the MB model compared to orientation towards the goal post for the Cyberzoo dataset with for white (a and c) and textured (b and d) background panels.

10

Discussion of the Preliminary Experiments

The preliminary experiments presented in Chapters 8 and 9 have covered a series of tests that evaluate two recent neural insect-inspired familiarity-based navigation models. This chapter serves to draw a couple of conclusions towards the feasibility of this thesis as presented in Chapter 1.

Section 10.1 discusses the influence of parameters such as image aspect ratio and resolution, learning strategy on network performance, as well as the correlation between network output and the view's deviation from the home location. Section 10.2 goes into the implications of the presented results and its effect on the work presented in Part I.

10.1. Influence of Visual Input on Network Performance

Recent insect-inspired visual guidance models (Baddeley et al., 2012; Ardin et al., 2016) show that small neural networks can discriminate familiar from non-familiar views as part of a so-called scene familiarity model (Sections 3.2.2). To optimize their efficiency, optimal use should be made of the visual input of the scenes they try to 'memorize'. Parameters such as input resolution and the amount of vertical versus horizontal information (Aspect Ratio) play an important role.

Insects have compound eyes that span a large horizontal (sometimes up to almost 360 degrees) field of view. It makes sense to think that visual navigation information that is important with respect to orienting oneself should be mostly present in the horizontal field of view and to a lesser extent in the vertical field of view. Section 9.2 confirms this notion by showing that images with an aspect ratio of up to 10:1 (meaning 10 times more pixels in the horizontal than the vertical direction) still produce reliable results. Image input resolution directly impacts the size of the networks and hence its computational requirements (see Sections 8.5.1 and 8.5.2). Section 9.3 indicates that the input resolution of the network is not as much of a concern as is the resulting network's size. The smallest size networks struggle to memorize more than about 50 scenes before degrading their performance.

Section 9.4 shows that careful consideration should be taken in selecting training views. Simply having a constant distance between 'snapshots' seems to not be enough to take full advantage of the network's capacity. Views can remain quasi the same over long distances, or change drastically over only a few meters, requiring respectively less or more snapshots to be taken. A metric for the difference in 'familiarity' between locations could determine when to best take a snapshot that is used to memorize the route, see e.g. (Denuelle and Srinivasan, 2016).

Section 9.5 shows that differentiating between sequences of familiar and unfamiliar views is a considerably more difficult task than giving a measure of familiarity of an unseen view towards a goal location (the goal location being the area where the sequence of training views was directed towards). This can be attributed to several factors. **1** Scene structure of the Cyberzoo experiment: visual references change dramatically in cluttered environments, requiring more exploring around the goal location for robust navigation. **2** Antigoal views often had the white/textured panels as a primary reference. These 'landmarks' are close and change a lot with movement. Having more distant visual landmarks would be beneficial. **3** Use of a small Field of View (FOV): the standard camera onboard the Bebop Parrot 2 drone was used, with a relatively small FOV. A small FOV, high resolution, image is useful in matching and pinpointing specific features but is less useful for extracting general scene information. A larger FOV presents more contextual information that is present in the scene and will increase performance, especially for finding the right *direction*.

10.2. Implications of the Preliminary Experiments

The preliminary experiments indicate that the models presented in Baddeley et al. (2012) and **Arind2016** can indeed discern familiar from unfamiliar frames within a single sequence. Recovering a goal location from a new unseen view remains a challenge however and probably requires a more dedicated learning strategy for reliably learning that goal location. Experiments covering the effect of the input view's resolution and aspect ratio give promising results for utilizing very high aspect ratio, low resolution (ergo omnidirectional) views. Furthermore, it indicates that the network's performance is predominantly limited by its size rather than the dimensions of its input. This thesis focuses on evaluating over how long of a distance such models can provide valuable navigational information. Evaluating such methods over long distances requires the appropriate dataset. Ideally one which includes low resolution omnidirectional vision (close to the insect's actual visual perception) in natural scenes combined with accurate positioning and pose information for evaluating the navigational performance. Furthermore, this thesis has as additional goal to use event-based vision. As no such datasets exist, it was chosen to develop a novel dataset, which is presented in Part I.

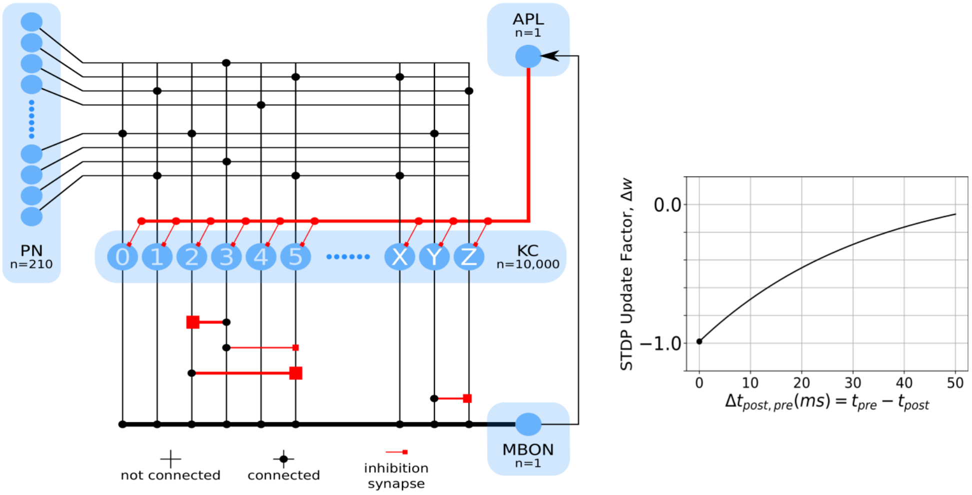
Part IV

Appendices

A

Mushroom Body Model — Le Zhu et al. 2020

This chapter covers the MB model presented by L. Zhu et al. (2020) and as discussed in Part I. L. Zhu et al. (2020) adapt the model presented in Ardin et al. (2016) based on the finding that 60% of the input synapses of KCs come from other KCs. Instead of learning on the weights connecting the KCs to the MBON, when a KC spikes, it will inhibit its connection to downstream KCs that spike at a later time based on the STDP rule as defined in Figure A.1 (b). The KCs are split up in two groups of 5000 neurons for learning; each solely acting within its group. Additionally, an Anterior Paired Lateral (APL) (Amin et al., 2020) neuron is included that inhibits the activity of the KC layer. This model is visualized in Figure A.1 (a).



(**a**) Schematic overview of the MB model

(**b**) The STDP learning rule.

Figure A.1: L. Zhu et al. (2020)'s MB model and STDP update rule.

L. Zhu et al. (2020) utilize different modified versions of the Leaky Integrate-and-Fire (LIF) neuron. Let i indicate a post-synaptic neuron from layer l with neurons $0,\ 1,\ 2,\ \ldots,\ n^l$ and j a pre-synaptic neuron from layer $l-1$ with neurons $0,\ 1,\ 2,\ \ldots,\ n^{l-1}$

$$\tau_v \frac{dv_i(t)}{dt} = -(v_i(t) - v_{\text{rest}}) + \alpha i_i(t) \tag{A.1}$$

$$i_i(t) = \sum_{j=1}^{n^{l-1}} \left(W_{i,j} s_j^{l-1}(t-\tau_d)\right) \tag{A.2}$$

with v_i the membrane potential of neuron i, τ_v the time constant of the membrane potential, v_{rest} the resting potential and $i_i(t)$ the forcing function. The forcing function depends on the synaptic efficacy, 'weight' $W_{i,j}$, of the pre-synaptic connections and the pre-synaptic spike train $s_j^{l-1}(t-\tau_d)$ with delay τ_d which captures the time for the electrochemical signal to pass through its synaptic connections. L. Zhu et al. (2020) use slight modifications of this model for the different neurons in their model of the MB.

The PNs follow the LIF notation but have an additional adaptive threshold which varies in the following sense:

$$\tau_{\text{thresh}} \frac{dv_{\text{thresh}}(t)}{dt} = -(v_{\text{thresh}}(t) - v_{\text{rest}} + 1) \tag{A.3}$$

The time constant τ_{thresh} is tuned such that the PNs spike at about 4 Hz while at rest (no input received). When a PN resets, its threshold gets updated:

$$v_{\text{thresh}} = v_{\text{thresh}} + 20 \tag{A.4}$$

The KCs as well as the APL neuron use the regular formulation of the LIF neuron. The MBON is also an adaptation of the LIF model and its membrane potential is described by:

$$\tau_v \frac{dv_i(t)}{dt} = -(dv_i(t) - v_{\text{rest}}) + I_i(t) \tag{A.5}$$

$$\frac{dI_i(t)}{dt} = -\frac{I_i(t)}{\tau_i} \tag{A.6}$$

where $I_i(t)$ is the neuron's internal current.

When the MBON fires its membrane potential is reset and its current set to 0 mA. An overview of the neuronal and synaptic parameters can be found in Table A.1.

Table A.1: Neuronal and synaptic parameters of L. Zhu et al. (2020)′s Mushroom Body model.

	Neuron Properties				
	PN	KC	MBON	APL	unit
neuron number n	210	2×5000	1	1	-
resting potential v_{rest}	-60	-60	-55	-60	mV
threshold voltage v_{thresh}	-40	-45	15	-45	mV
reset potential v_{reset}	-45	-50	-50	-50	mV
voltage scaling factor α_v	80	20	-	20	-
voltage time constant τ_v	11.5	11.5	20	11.5	ms
trace scaling factor α_t	80	23	20	23	-
trace time constant τ_t	11.5	23	20	23	ms
threshold scaling factor α_{thresh}	20	-	-	-	-
threshold time constant τ_{thresh}	$250 \div \ln(21)$	-	-	-	ms
current scaling factor α_i	-	-	-10	-	-
current time constant τ_i	-	-	20	-	ms

	Synapse properties					
	input to PN	PN to KC	KC to EN	EN to APL	APL to KC	unit
connectivity	one-to-one	10 per KC	fc	fc	fc	-
synaptic weight	1	1	1	1	-0.05	-
synaptic delay (optional)	0	0	0	0	0	ms

Bibliography

Ahmadi, Aria and Ioannis Patras (2016). "Unsupervised convolutional neural networks for motion estimation". In: *2016 IEEE international conference on image processing (ICIP)*. IEEE, pp. 1629–1633. DOI: 10.1109/ICIP.2016.7532634

Akopyan, Filipp, Jun Sawada, Andrew Cassidy, Rodrigo Alvarez-Icaza, John Arthur, Paul Merolla, Nabil Imam, Yutaka Nakamura, Pallab Datta, Gi-Joon Nam, Brian Taba, Michael Beakes, Bernard Brezzo, Jente B. Kuang, Rajit Manohar, William P. Risk, Bryan Jackson, and Dharmendra S. Modha (Oct. 2015). "TrueNorth: Design and Tool Flow of a 65 mW 1 Million Neuron Programmable Neurosynaptic Chip". In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 34.10, pp. 1537–1557. DOI: 10.1109/TCAD.2015.2474396

Aleotti, Filippo, Giulio Zaccaroni, Luca Bartolomei, Matteo Poggi, Fabio Tosi, and Stefano Mattoccia (2020). "Real-time single image depth perception in the wild with handheld devices". In: pp. 1–11. eprint: 2006.05724

Amin, Hoger, Anthi A Apostolopoulou, Raquel Suárez-Grimalt, Eleftheria Vrontou, and Andrew C Lin (Sept. 2020). "Localized inhibition in the Drosophila mushroom body". In: *eLife* 9. DOI: 10.7554/eLife.56954

Amiri, Ali Jahani, Shing Yan Loo, and Hong Zhang (2019). "Semi-supervised monocular depth estimation with left-right consistency using deep neural network". In: *IEEE International Conference on Robotics and Biomimetics, ROBIO 2019* December, pp. 602–607. DOI: 10.1109/ROBIO49542.2019.8961504

Ardin, Paul, Fei Peng, Michael Mangan, Konstantinos Lagogiannis, and Barbara Webb (Feb. 2016). "Using an Insect Mushroom Body Circuit to Encode Route Memory in Complex Natural Environments". In: *PLOS Computational Biology* 12.2. Ed. by Joseph Ayers, e1004683. DOI: 10.1371/journal.pcbi.1004683

Argyros, Antonis A., Kostas E. Bekris, Stelios C. Orphanoudakis, and Lydia E. Kavraki (July 2005). "Robot Homing by Exploiting Panoramic Vision". In: *Autonomous Robots* 19.1, pp. 7–25. DOI: 10.1007/s10514-005-0603-7

Aso, Yoshinori, Daisuke Hattori, Yang Yu, Rebecca M. Johnston, Nirmala A. Iyer, Teri T.B. Ngo, Heather Dionne, L. F. Abbott, Richard Axel, Hiromu Tanimoto, and Gerald M. Rubin (2014). "The neuronal architecture of the mushroom body provides a logic for associative learning". In: *eLife* 3, e04577. DOI: 10.7554/eLife.04577

Avarguès-Weber, Aurore, Theo Mota, and Martin Giurfa (May 2012). "New vistas on honey bee vision". In: *Apidologie* 43.3, pp. 244–268. DOI: 10.1007/s13592-012-0124-2

Baddeley, Bart, Paul Graham, Philip Husbands, and Andrew Philippides (Jan. 2012). "A Model of Ant Route Navigation Driven by Scene Familiarity". In: *PLoS Computational Biology* 8.1. Ed. by Holk Cruse, e1002336. DOI: 10.1371/journal.pcbi.1002336

Bar-Haim, Aviram and Lior Wolf (2020). "ScopeFlow: Dynamic Scene Scoping for Optical Flow". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7998–8007

Bausenwein, B., A. P. M. Dittrich, and K. F. Fischbach (Jan. 1992). "The optic lobe of Drosophila melanogaster". In: *Cell & Tissue Research* 267.1, pp. 17–28. DOI: 10.1007/BF00318687

Bell, Anthony J. and Terrence J. Sejnowski (Nov. 1995). "An Information-Maximization Approach to Blind Separation and Blind Deconvolution". In: *Neural Computation* 7.6, pp. 1129–1159. DOI: 10.1162/neco.1995.7.6.1129

Berry, Richard P., William T. Wcislo, and Eric J. Warrant (Apr. 2011). “Ocellar adaptations for dim light vision in a nocturnal bee”. In: *Journal of Experimental Biology* 214.8, pp. 1283–1293. DOI: 10.1242/jeb.050427

Brandli, Christian, Raphael Berner, Minhao Yang, Shih-Chii Liu, and Tobi Delbruck (Oct. 2014). “A 240 × 180 130 dB 3 µs Latency Global Shutter Spatiotemporal Vision Sensor”. In: *IEEE Journal of Solid-State Circuits* 49.10, pp. 2333–2341. DOI: 10.1109/JSSC.2014.2342715

Cartwright, B. A. and T. S. Collett (1983). “Landmark learning in bees - Experiments and models”. In: *Journal of Comparative Physiology A* 151.4, pp. 521–543. DOI: 10.1007/BF00605469

Caruso, David, Jakob Engel, and Daniel Cremers (Sept. 2015). “Large-scale direct SLAM for omnidirectional cameras”. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Vol. 2015-Decem. IEEE, pp. 141–148. DOI: 10.1109/IROS.2015.7353366

Casser, Vincent, Soeren Pirk, Reza Mahjourian, and Anelia Angelova (2019). “Depth Prediction without the Sensors: Leveraging Structure for Unsupervised Learning from Monocular Videos”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 33, pp. 8001–8008. DOI: 10.1609/aaai.v33i01.33018001. eprint: 1811.06152

Censi, Andrea and Davide Scaramuzza (May 2014). “Low-latency event-based visual odometry”. In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 703–710. DOI: 10.1109/ICRA.2014.6906931

Chen, Yuhua, Cordelia Schmid, and Cristian Sminchisescu (2019). “Self-supervised learning with geometric constraints in monocular video: Connecting flow, depth, and camera”. In: *Proceedings of the IEEE International Conference on Computer Vision* 2019-October, pp. 7062–7071. DOI: 10.1109/ICCV.2019.00716. eprint: 1907.05820

Cheng, Yang, Jie Cao, Yangkun Zhang, and Qun Hao (Feb. 2019). “Review of state-of-the-art artificial compound eye imaging systems”. In: *Bioinspiration & Biomimetics* 14.3, p. 031002. DOI: 10.1088/1748-3190/aaffb5

Cheung, Allen, Matthew Collett, Thomas S. Collett, Alex Dewar, Fred Dyer, Paul Graham, Michael Mangan, Ajay Narendra, Andrew Philippides, Wolfgang Stürzl, Barbara Webb, Antoine Wystrach, and Jochen Zeil (Oct. 2014). “Still no convincing evidence for cognitive map use by honeybees”. In: *Proceedings of the National Academy of Sciences*. Vol. 111. 42. National Academy of Sciences, E4396–E4397. DOI: 10.1073/pnas.1413581111

Cheung, Allen and Robert Vickerstaff (Nov. 2010). “Finding the Way with a Noisy Brain”. In: *PLoS Computational Biology* 6.11. Ed. by Karl J. Friston, e1000992. DOI: 10.1371/journal.pcbi.1000992

Collett, Thomas S. and Jochen Zeil (Sept. 2018). “Insect learning flights and walks”. In: *Current Biology* 28.17, R984–R988. DOI: 10.1016/j.cub.2018.04.050

Cordts, Marius, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele (2016). “The cityscapes dataset for semantic urban scene understanding”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223

Czech, Daniel and Garrick Orchard (June 2016). “Evaluating noise filtering for event-based asynchronous change detection image sensors”. In: *2016 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob)*. Vol. 2016-July. IEEE, pp. 19–24. DOI: 10.1109/BIOROB.2016.7523452

Dalen, Gerald J. J. van, Kimberly N. McGuire, and Guido C. H. E. de Croon (Apr. 2018). “Visual Homing for Micro Aerial Vehicles Using Scene Familiarity”. In: *Unmanned Systems* 06.02, pp. 119–130. DOI: 10.1142/S230138501850005X

Davies, Mike, Narayan Srinivasa, Tsung-Han Lin, Gautham Chinya, Yongqiang Cao, Sri Harsha Choday, Georgios Dimou, Prasad Joshi, Nabil Imam, Shweta Jain, Yuyun Liao, Chit-Kwan Lin, Andrew Lines, Ruokun Liu, Deepak Mathaikutty, Steven McCoy, Arnab Paul, Jonathan Tse, Guruguhanathan Venkataramanan, Yi-Hsin Weng, Andreas Wild, Yoonseok Yang, and Hong Wang (Jan. 2018). “Loihi: A Neuromorphic Manycore Processor with On-Chip Learning”. In: *IEEE Micro* 38.1, pp. 82–99. DOI: 10.1109/MM.2018.112130359

Delmerico, Jeffrey, Stefano Mintchev, Alessandro Giusti, Boris Gromov, Kamilo Melo, Tomislav Horvat, Cesar Cadena, Marco Hutter, Auke Ijspeert, Dario Floreano, Luca M Gambardella, Roland Siegwart, and Davide Scaramuzza (2019). "The current state and future outlook of rescue robotics". In: *Journal of Field Robotics* 36.7, pp. 1171–1191. DOI: 10.1002/rob.21887

Deng, Jia, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei (2009). "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee, pp. 248–255

Denuelle, Aymeric and Mandyam V. Srinivasan (May 2016). "A sparse snapshot-based navigation strategy for UAS guidance in natural environments". In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. Vol. 2016-June. IEEE, pp. 3455–3462. DOI: 10.1109/ICRA.2016.7487524

Differt, Dario and Ralf Möller (Sept. 2015). "Insect models of illumination-invariant skyline extraction from UV and green channels". In: *Journal of Theoretical Biology* 380, pp. 444–462. DOI: 10.1016/j.jtbi.2015.06.020

Dijk, Tom van (2017). "Low-memory Visual Route Following for Micro Aerial Vehicles in Indoor Environments". PhD thesis

Dosovitskiy, Alexey, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox (Dec. 2015). "FlowNet: Learning Optical Flow with Convolutional Networks". In: *2015 IEEE International Conference on Computer Vision (ICCV)*. IEEE, pp. 2758–2766. DOI: 10.1109/ICCV.2015.316. eprint: 1504.06852

Dupeyroux, Julien, Stéphane Viollet, and Julien R. Serres (Jan. 2019). "Polarized skylight-based heading measurements: a bio-inspired approach". In: *Journal of The Royal Society Interface* 16.150, p. 20180878. DOI: 10.1098/rsif.2018.0878

Dyakova, Olga (2017). "The processing of natural images in the visual system". PhD thesis. Acta Universitatis Upsaliensis

Dyer, Adrian G, Angelique C Paulk, and David H Reser (2011). "Colour processing in complex environments: insights from the visual system of bees". In: *Proceedings of the Royal Society B: Biological Sciences* 278.1707, pp. 952–959

Ehmer, Birgit and Wulfila Gronenberg (Feb. 2004). "Mushroom body volumes and visual interneurons in ants: Comparison between sexes and castes". In: *The Journal of Comparative Neurology* 469.2, pp. 198–213. DOI: 10.1002/cne.11014

Eigen, David, Christian Puhrsch, and Rob Fergus (2014). "Depth map prediction from a single image using a multi-scale deep network". In: *Advances in neural information processing systems*, pp. 2366–2374

Engel, Jakob, Thomas Schöps, and Daniel Cremers (Jan. 2014). "LSD-SLAM: Large-Scale Direct monocular SLAM". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 8690 LNCS. PART 2, pp. 834–849. DOI: 10.1007/978-3-319-10605-2_54

Fahrbach, Susan E. (Jan. 2006). "Structure Of The Mushroom Bodies Of The Insect Brain". In: *Annual Review of Entomology* 51.1, pp. 209–232. DOI: 10.1146/annurev.ento.51.110104.150954

Felsenberg, Johannes, Pedro F. Jacob, Thomas Walker, Oliver Barnstedt, Amelia J. Edmondson-Stait, Markus W. Pleijzier, Nils Otto, Philipp Schlegel, Nadiya Sharifi, Emmanuel Perisse, Carlas S. Smith, J. Scott Lauritzen, Marta Costa, Gregory S.X.E. Jefferis, Davi D. Bock, and Scott Waddell (Oct. 2018). "Integration of Parallel Opposing Memories Underlies Memory Extinction". In: *Cell* 175.3, 709–722.e15. DOI: 10.1016/j.cell.2018.08.021

Feng, Tuo and Dongbing Gu (2019). "SGANVO: Unsupervised Deep Visual Odometry and Depth Estimation With Stacked Generative Adversarial Networks". In: 4.4, pp. 4431–4437

Field, David J. and Nuala Brady (Dec. 1997). "Visual sensitivity, blur and the sources of variability in the amplitude spectra of natural scenes". In: *Vision Research* 37.23, pp. 3367–3383. DOI: 10.1016/S0042-6989(97)00181-8

Forster, Christian, Matia Pizzoli, and Davide Scaramuzza (May 2014). "SVO: Fast semi-direct monocular visual odometry". In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 15–22. DOI: 10.1109/ICRA.2014.6906584

Franz, Matthias O., Bernhard Schölkopf, Hanspeter A. Mallot, and Heinrich H. Bülthoff (Oct. 1998). "Where did I take that snapshot? Scene-based homing by image matching". In: *Biological Cybernetics* 79.3, pp. 191–202. DOI: 10.1007/s004220050470

Gallego, Guillermo, Tobi Delbruck, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew Davison, Joerg Conradt, Kostas Daniilidis, and Davide Scaramuzza (Apr. 2019). "Event-based Vision: A Survey". In: *CoRR* abs/1904.0, pp. 1–30. eprint: 1904.08405

Gallego, Guillermo, Henri Rebecq, and Davide Scaramuzza (2018). "A Unifying Contrast Maximization Framework for Event Cameras, with Applications to Motion, Depth, and Optical Flow Estimation". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3867–3876. DOI: 10.1109/CVPR.2018.00407. eprint: 1804.01306

Garg, Ravi, Vijay Kumar Bg, Gustavo Carneiro, and Ian Reid (2016). "Unsupervised cnn for single view depth estimation: Geometry to the rescue". In: *European conference on computer vision*. Springer, pp. 740–756

Gehrig, Mathias, Sumit Bam Shrestha, Daniel Mouritzen, and Davide Scaramuzza (2020). "Event-Based Angular Velocity Regression with Spiking Networks". In: eprint: 2003.02790

Geiger, Andreas, Philip Lenz, and Raquel Urtasun (2012). "Are we ready for autonomous driving? the kitti vision benchmark suite". In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 3354–3361

Giurfa, M., M. Vorobyev, P. Kevan, and R. Menzel (1996). "Detection of coloured stimuli by honeybees: Minimum visual angles and receptor specific contrasts". In: *Journal of Comparative Physiology A: Sensory, Neural, and Behavioral Physiology* 178.5, pp. 699–709. DOI: 10.1007/BF00227381

Giurfa, Martin, Misha Vorobyev, Robert Brandt, Britta Posner, and Randolf Menzel (1997). "Discrimination of coloured stimuli by honeybees: Alternative use of achromatic and chromatic signals". In: *Journal of Comparative Physiology - A Sensory, Neural, and Behavioral Physiology* 180.3, pp. 235–243. DOI: 10.1007/s003590050044

Giusti, Alessandro, Jerome Guzzi, Dan C. Ciresan, Fang Lin He, Juan P. Rodriguez, Flavio Fontana, Matthias Faessler, Christian Forster, Jurgen Schmidhuber, Gianni Di Caro, Davide Scaramuzza, and Luca M. Gambardella (2016). "A Machine Learning Approach to Visual Perception of Forest Trails for Mobile Robots". In: *IEEE Robotics and Automation Letters* 1.2, pp. 661–667. DOI: 10.1109/LRA.2015.2509024

Godard, Clément, Oisin Mac Aodha, and Gabriel J Brostow (2017). "Unsupervised monocular depth estimation with left-right consistency". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 270–279

Güney, Fatma and Andreas Geiger (2017). "Deep Discrete Flow". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 10114 LNCS. Springer, pp. 207–224. DOI: 10.1007/978-3-319-54190-7_13

Hagenaars, J. J., F. Paredes-Vallés, S. M. Bohté, and G. C. H. E. de Croon (Mar. 2020). "Evolved Neuromorphic Control for High Speed Divergence-based Landings of MAVs". In: eprint: 2003.03118

Hayakawa, Takashi, Takeshi Kaneko, and Toshio Aoyagi (Nov. 2014). "A biologically plausible learning rule for the Infomax on recurrent neural networks". In: *Frontiers in Computational Neuroscience* 8. DOI: 10.3389/fncom.2014.00143

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016). "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778. DOI: 10.1109/CVPR.2016.90

Heisenberg, Martin (2003). "Mushroom body memoir: From maps to models". In: *Nature Reviews Neuroscience* 4.4, pp. 266–275. DOI: 10.1038/nrn1074

Hertel, B Y Horst and Ulrike Maronde (1987). "The Physiology and Morphology of Visual Commissures in the Honeybee Brain". In: *Journal of Experimental Biology* 133.1, pp. 283–300

Hoinville, Thierry and Rüdiger Wehner (2018). "Optimal multiguidance integration in insect navigation". In: *Proceedings of the National Academy of Sciences of the United States of America* 115.11, pp. 2824–2829. doi: `10.1073/pnas.1721668115`

Horn, Berthold KP and Brian G Schunck (1981). "Determining optical flow". In: *Techniques and Applications of Image Understanding*. Vol. 281. International Society for Optics and Photonics, pp. 319–331

Howard, Andrew G, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam (2017). "Mobilenets: Efficient convolutional neural networks for mobile vision applications". In: *arXiv preprint arXiv:1704.04861*

Huang, Guoquan (May 2019). "Visual-Inertial Navigation: A Concise Review". In: *2019 International Conference on Robotics and Automation (ICRA)*. Vol. 2019-May. IEEE, pp. 9572–9582. doi: `10.1109/ICRA.2019.8793604`. eprint: `1906.02650`

Hui, Tak Wai, Xiaoou Tang, and Chen Change Loy (2018). "LiteFlowNet: A Lightweight Convolutional Neural Network for Optical Flow Estimation". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 8981–8989. doi: `10.1109/CVPR.2018.00936`. eprint: `1805.07036`

Hur, Junhwa and Stefan Roth (2019). "Iterative residual refinement for joint optical flow and occlusion estimation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5754–5763

— (2020). "Optical Flow Estimation in the Deep Learning Age". In: pp. 1–23. eprint: `2004.02853`

Ibbotson, M. R., T. Maddess, and R. DuBois (1991). "A system of insect neurons sensitive to horizontal and vertical image motion connects the medulla and midbrain". In: *Journal of Comparative Physiology A* 169.3, pp. 355–367. doi: `10.1007/BF00207000`

Ilg, Eddy, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox (2017). "Flownet 2.0: Evolution of optical flow estimation with deep networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2462–2470

Izhikevich, E.M. (Nov. 2003). "Simple model of spiking neurons". In: *IEEE Transactions on Neural Networks* 14.6, pp. 1569–1572. doi: `10.1109/TNN.2003.820440`

Janai, Joel, Fatma Guney, Anurag Ranjan, Michael Black, and Andreas Geiger (2018). "Unsupervised learning of multi-frame optical flow with occlusions". In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 690–706

Jason, J Yu, Adam W Harley, and Konstantinos G Derpanis (2016). "Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness". In: *European Conference on Computer Vision*. Springer, pp. 3–10

Jayatilaka, Piyankarie, Trevor Murray, Ajay Narendra, and Jochen Zeil (Oct. 2018). "The choreography of learning walks in the Australian jack jumper ant Myrmecia croslandi". In: *The Journal of Experimental Biology* 221.20, jeb185306. doi: `10.1242/jeb.185306`

Jung, Sunggoo, Sunyou Hwang, Heemin Shin, and David Hyunchul Shim (July 2018). "Perception, Guidance, and Navigation for Indoor Autonomous Drone Racing Using Deep Learning". In: *IEEE Robotics and Automation Letters* 3.3, pp. 2539–2544. doi: `10.1109/LRA.2018.2808368`

Kaiser, Jacques, J. Camilo Vasquez Tieck, Christian Hubschneider, Peter Wolf, Michael Weber, Michael Hoff, Alexander Friedrich, Konrad Wojtasik, Arne Roennau, Ralf Kohlhaas, Rudiger Dillmann, and J. Marius Zollner (Dec. 2016). "Towards a framework for end-to-end control of a simulated vehicle with spiking neural networks". In: *2016 IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR)*. IEEE, pp. 127–134. doi: `10.1109/SIMPAR.2016.7862386`

Karásek, Matěj, Florian T Muijres, Christophe De Wagter, Bart DW Remes, and Guido CHE de Croon (2018). "A tailless aerial robotic flapper reveals that flies use torque coupling in rapid banked turns". In: *Science* 361.6407, pp. 1089–1094

Kaufmann, Elia, Mathias Gehrig, Philipp Foehn, René Ranftl, Alexey Dosovitskiy, Vladlen Koltun, and Davide Scaramuzza (Oct. 2018). "Beauty and the Beast: Optimal Methods Meet Learning for Drone

Racing". In: *2019 International Conference on Robotics and Automation (ICRA)* 2019-May, pp. 690–696. DOI: 10.1109/ICRA.2019.8793631. eprint: 1810.06224

Kelber, Almut and Hema Somanathan (Nov. 2019). "Spatial Vision and Visually Guided Behavior in Apidae". In: *Insects* 10.12, p. 418. DOI: 10.3390/insects10120418

Knight, James C., Daniil Sakhapov, Norbert Domcsek, Alex D.M. Dewar, Paul Graham, Thomas Nowotny, and Andrew Philippides (2019). "Insect-Inspired Visual Navigation On-Board an Autonomous Robot: Real-World Routes Encoded in a Single Layer Network". In: *The 2019 Conference on Artificial Life*. Cambridge, MA: MIT Press, pp. 60–67. DOI: 10.1162/isal_a_00141.xml

Kreiser, Raphaela, Alpha Renner, Yulia Sandamirskaya, and Panin Pienroj (2018). "Pose estimation and map formation with spiking neural networks: towards neuromorphic slam". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 2159–2166

Krig, Scott and Scott Krig (2014). "Interest Point Detector and Feature Descriptor Survey". In: *Computer Vision Metrics*. 1. Berkeley, CA: Apress, pp. 217–282. DOI: 10.1007/978-1-4302-5930-5_6

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton (2012). "ImageNet classification with deep convolutional neural networks". In: *Advances in Neural Information Processing Systems*

Kuznietsov, Yevhen, Jorg Stuckler, and Bastian Leibe (2017). "Semi-supervised deep learning for monocular depth map prediction". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6647–6655

Labhart, Thomas and Eric P. Meyer (Dec. 1999). "Detectors for polarized skylight in insects: a survey of ommatidial specializations in the dorsal rim area of the compound eye". In: *Microscopy Research and Technique* 47.6, pp. 368–379. DOI: 10.1002/(SICI)1097-0029(19991215)47:6<368::AID-JEMT2>3.0.CO;2-Q

Laidlow, Tristan, Jan Czarnowski, and Stefan Leutenegger (2019). "DeepFusion: Real-time dense 3D reconstruction for monocular SLAM using single-view depth and gradient predictions". In: *Proceedings - IEEE International Conference on Robotics and Automation* 2019-May, pp. 4068–4074. DOI: 10.1109/ICRA.2019.8793527

Laina, Iro, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab (2016). "Deeper depth prediction with fully convolutional residual networks". In: *2016 Fourth international conference on 3D vision (3DV)*. IEEE, pp. 239–248

Lambrinos, D, Ralf Möller, R Pfeifer, and R Wehner (1998). "Landmark Navigation without Snapshots: the Average Landmark Vector Model". In: *Proc. Neurobiol. Conf. Göttingen*. Ed. by N Elsner and R Wehner. Georg Thieme Verlag, 30a

Lambrinos, Dimitrios, Ralf Möller, Thomas Labhart, Rolf Pfeifer, and Rüdiger Wehner (Jan. 2000). "A mobile robot employing insect strategies for navigation". In: *Robotics and Autonomous Systems* 30.1-2, pp. 39–64. DOI: 10.1016/S0921-8890(99)00064-0

Le Möel, Florent and Antoine Wystrach (Feb. 2020). "Opponent processes in visual memories: A model of attraction and repulsion in navigating insects' mushroom bodies". In: *PLOS Computational Biology* 16.2. Ed. by Joseph Ayers, e1007631. DOI: 10.1371/journal.pcbi.1007631

Lebhardt, Fleur and Claude Desplan (Dec. 2017). "Retinal perception and ecological significance of color vision in insects". In: *Current Opinion in Insect Science* 24, pp. 75–83. DOI: 10.1016/j.cois.2017.09.007

Lee, Chankyu, Adarsh Kosta, Alex Zihao Zhu, Kenneth Chaney, Kostas Daniilidis, and Kaushik Roy (2020). "Spike-FlowNet: Event-based Optical Flow Estimation with Energy-Efficient Hybrid Neural Networks". In: pp. 1–16. eprint: 2003.06696

Lee, Luke P. and Robert Szema (Nov. 2005). *Inspirations from biological optics for advanced photonic systems*. DOI: 10.1126/science.1115248

Lehrer, M. (1999). "Dorsoventral asymmetry of colour discrimination in bees". In: *Journal of Comparative Physiology - A Sensory, Neural, and Behavioral Physiology* 184.2, pp. 195–206. DOI: 10.1007/s003590050318

Leutenegger, Stefan, Simon Lynen, Michael Bosse, Roland Siegwart, and Paul Furgale (Mar. 2015). "Keyframe-based visual–inertial odometry using nonlinear optimization". In: *The International Journal of Robotics Research* 34.3, pp. 314–334. DOI: 10.1177/0278364914554813

Li, Ruihao, Sen Wang, Zhiqiang Long, and Dongbing Gu (May 2018). "UnDeepVO: Monocular Visual Odometry Through Unsupervised Deep Learning". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 7286–7291. DOI: 10.1109/ICRA.2018.8461251. eprint: 1709.06841

Lichtsteiner, Patrick, Christoph Posch, and Tobi Delbruck (2008). "A 128×128 120 dB 15 μs Latency Asynchronous Temporal Contrast Vision Sensor". In: *IEEE Journal of Solid-State Circuits* 43.2, pp. 566–576. DOI: 10.1109/JSSC.2007.914337

Liu, Pengpeng, Michael Lyu, Irwin King, and Jia Xu (2019). "Selflow: Self-supervised learning of optical flow". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2019-June, pp. 4566–4575. DOI: 10.1109/CVPR.2019.00470

Liu, Wei, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg (2016). "Ssd: Single shot multibox detector". In: *European conference on computer vision*. Springer, pp. 21–37

Longuet-higgins, H. C. (1981). "A computer algorithm for reconstructing a scene from two projections". In: *Nature* 293.5828, pp. 133–135. DOI: 10.1038/293133a0

Loquercio, Antonio, Elia Kaufmann, Rene Ranftl, Alexey Dosovitskiy, Vladlen Koltun, and Davide Scaramuzza (Feb. 2020). "Deep Drone Racing: From Simulation to Reality With Domain Randomization". In: *IEEE Transactions on Robotics* 36.1, pp. 1–14. DOI: 10.1109/TRO.2019.2942989. eprint: 1905.09727

Loquercio, Antonio, Ana I. Maqueda, Carlos R. Del-Blanco, and Davide Scaramuzza (2018). "DroNet: Learning to Fly by Driving". In: *IEEE Robotics and Automation Letters* 3.2, pp. 1088–1095. DOI: 10.1109/LRA.2018.2795643

Menzel, Randolf (Apr. 2014). "The insect mushroom body, an experience-dependent recoding device". In: *Journal of Physiology-Paris* 108.2-3, pp. 84–95. DOI: 10.1016/j.jphysparis.2014.07.004

Menzel, Randolf and Martin Giurfa (Feb. 2001). "Cognitive architecture of a mini-brain: the honeybee". In: *Trends in Cognitive Sciences* 5.2, pp. 62–71. DOI: 10.1016/S1364-6613(00)01601-6

Mitrokhin, Anton, Chengxi Ye, Cornelia Fermuller, Yiannis Aloimonos, and Tobi Delbruck (2019). "EV-IMO: Motion Segmentation Dataset and Learning Pipeline for Event Cameras". In: *IEEE International Conference on Intelligent Robots and Systems*, pp. 6105–6112. DOI: 10.1109/IROS40897.2019.8968520. eprint: 1903.07520

Mizunami, Makoto (Feb. 1995). "Functional diversity of neural organization in insect ocellar systems". In: *Vision Research* 35.4, pp. 443–452. DOI: 10.1016/0042-6989(94)00192-O

Mohanty, Vikram, Shubh Agrawal, Shaswat Datta, Arna Ghosh, Vishnu Dutt Sharma, and Debashish Chakravarty (Nov. 2016). "DeepVO: A Deep Learning approach for Monocular Visual Odometry". In: eprint: 1611.06069

Möller, Ralf (July 2012). "A model of ant navigation based on visual prediction". In: *Journal of Theoretical Biology* 305, pp. 118–130. DOI: 10.1016/j.jtbi.2012.04.022

Möller, Ralf, Martin Krzykawski, and Lorenz Gerstmayr (2010). "Three 2D-warping schemes for visual robot navigation". In: *Autonomous Robots* 29.3-4, pp. 253–291. DOI: 10.1007/s10514-010-9195-y

Möller, Ralf and Andrew Vardy (Oct. 2006). "Local visual homing by matched-filter descent in image distances". In: *Biological Cybernetics* 95.5, pp. 413–430. DOI: 10.1007/s00422-006-0095-3

Moon, Hyungpil, Yu Sun, Jacky Baltes, and Si Jung Kim (2017). "The IROS 2016 competitions [competitions]". In: *IEEE Robotics and Automation Magazine* 24.1, pp. 20–29

Müller, Jurek, Martin Nawrot, Randolf Menzel, and Tim Landgraf (Apr. 2018). "A neural network model for familiarity and context learning during honeybee foraging flights". In: *Biological Cybernetics* 112.1-2, pp. 113–126. DOI: 10.1007/s00422-017-0732-z

Müller, Matthias, Vincent Casser, Neil Smith, Dominik L. Michels, and Bernard Ghanem (2019). "Teaching UAVs to Race: End-to-End Regression of Agile Controls in Simulation". In: *Lecture Notes in Com-*

puter Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Vol. 11130 LNCS, pp. 11–29. DOI: 10.1007/978-3-030-11012-3_2. eprint: 1708.05884

Mur-Artal, Raul, J. M. M. Montiel, and Juan D. Tardos (Oct. 2015). "ORB-SLAM: A Versatile and Accurate Monocular SLAM System". In: *IEEE Transactions on Robotics* 31.5, pp. 1147–1163. DOI: 10.1109/TRO.2015.2463671. eprint: 1502.00956

Mur-Artal, Raul and Juan D. Tardos (2017). "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras". In: *IEEE Transactions on Robotics* 33.5, pp. 1255–1262. DOI: 10.1109/TRO.2017.2705103. eprint: 1610.06475

Murray, Trevor, Zoltán Kócsi, Hansjürgen Dahmen, Ajay Narendra, Florent Le Möel, Antoine Wystrach, and Jochen Zeil (Feb. 2020). "The role of attractive and repellent scene memories in ant homing (Myrmecia croslandi)". In: *The Journal of Experimental Biology* 223.3, jeb210021. DOI: 10.1242/jeb.210021

Murray, Trevor and Jochen Zeil (Oct. 2017). "Quantifying navigational information: The catchment volumes of panoramic snapshots in outdoor scenes". In: *PLOS ONE* 12.10. Ed. by Paul Graham, e0187226. DOI: 10.1371/journal.pone.0187226

Nowak, Przemyslaw and Terrence C. Stewart (2019). *A spiking model of desert ant navigation along a habitual route*. Vol. 837. Springer International Publishing, pp. 211–222. DOI: 10.1007/978-3-319-97888-8_18

Paredes-Valles, Federico, Kirk Yannick Willehm Scheper, and Guido Cornelis Henricus Eugene De Croon (2019). "Unsupervised Learning of a Hierarchical Spiking Neural Network for Optical Flow Estimation: From Events to Global Motion Perception". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 8828.c, pp. 1–1. DOI: 10.1109/TPAMI.2019.2903179. eprint: 1807.10936

Paulk, Angelique C., Andrew M. Dacks, James Phillips-Portillo, Jean Marc Fellous, and Wulfila Gronenberg (2009). "Visual processing in the central bee brain". In: *Journal of Neuroscience* 29.32, pp. 9987–9999. DOI: 10.1523/JNEUROSCI.1325-09.2009

Peluso, Valentino, Antonio Cipolletta, Andrea Calimera, Matteo Poggi, Fabio Tosi, and Stefano Mattoccia (2019). "Enabling energy-efficient unsupervised monocular depth estimation on armv7-based platforms". In: *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, pp. 1703–1708

Pfeiffer, Michael and Thomas Pfeil (Oct. 2018). "Deep Learning With Spiking Neurons: Opportunities and Challenges". In: *Frontiers in Neuroscience* 12.October. DOI: 10.3389/fnins.2018.00774

Pink, Oliver, Frank Moosmann, and Alexander Bachmann (June 2009). "Visual features for vehicle localization and ego-motion estimation". In: *2009 IEEE Intelligent Vehicles Symposium*. IEEE, pp. 254–260. DOI: 10.1109/IVS.2009.5164287

Posch, Christoph, Daniel Matolin, and Rainer Wohlgenannt (Jan. 2011). "A QVGA 143 dB Dynamic Range Frame-Free PWM Image Sensor With Lossless Pixel-Level Video Compression and Time-Domain CDS". In: *IEEE Journal of Solid-State Circuits* 46.1, pp. 259–275. DOI: 10.1109/JSSC.2010.2085952

Pye, J. David (July 2018). "The Eye of the Beeholder: Comparing Honey Bee and Human Vision". In: *Bee World* 95.3, pp. 95–98. DOI: 10.1080/0005772X.2018.1467372

Qin, Tong, Peiliang Li, and Shaojie Shen (2018). "VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator". In: *IEEE Transactions on Robotics* 34.4, pp. 1004–1020. DOI: 10.1109/TRO.2018.2853729. eprint: 1708.03852

Ramirez, Pierluigi Zama, Matteo Poggi, Fabio Tosi, Stefano Mattoccia, and Luigi Di Stefano (2018). "Geometry meets semantics for semi-supervised monocular depth estimation". In: *Asian Conference on Computer Vision*. Springer, pp. 298–313

Ranjan, Anurag and Michael J Black (2017). "Optical flow estimation using a spatial pyramid network". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4161–4170

Ren, Zhe, Junchi Yan, Bingbing Ni, Bin Liu, Xiaokang Yang, and Hongyuan Zha (2017). "Unsupervised deep learning for optical flow estimation". In: *Thirty-First AAAI Conference on Artificial Intelligence*

Ribi, WilliA. and Marlies Scheel (Nov. 1981). "The second and third optic ganglia of the worker bee". In: *Cell And Tissue Research* 221.1, pp. 17–43. DOI: 10.1007/BF00216567

Robert, Théo, Elisa Frasnelli, Natalie Hempel de Ibarra, and Thomas S. Collett (Feb. 2018). "Variations on a theme: bumblebee learning flights from the nest and from flowers". In: *The Journal of Experimental Biology* 221.4, jeb172601. DOI: 10.1242/jeb.172601

Ross, Stéphane, Geoffrey Gordon, and Drew Bagnell (2011). "A reduction of imitation learning and structured prediction to no-regret online learning". In: *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635

Rössler, Wolfgang (2019). "Neuroplasticity in desert ants (Hymenoptera: Formicidae) – importance for the ontogeny of navigation". In: *Myrmecological News* 29, pp. 1–20. DOI: 10.25849/myrmecol.news_029:001

Ruck, Philip (Dec. 1958). "A comparison of the electrical responses of compound eyes and dorsal ocelli in four insect species". In: *Journal of Insect Physiology* 2.4, pp. 261–274. DOI: 10.1016/0022-1910(58)90012-X

Saxena, Ashutosh, Min Sun, and Andrew Y Ng (2008). "Make3d: Learning 3d scene structure from a single still image". In: *IEEE transactions on pattern analysis and machine intelligence* 31.5, pp. 824–840

Silberman, Nathan, Derek Hoiem, Pushmeet Kohli, and Rob Fergus (2012). "Indoor segmentation and support inference from rgbd images". In: *European conference on computer vision*. Springer, pp. 746–760

Simonyan, Karen and Andrew Zisserman (2014). "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556*

Skorupski, Peter and Lars Chittka (Mar. 2010). "Differences in Photoreceptor Processing Speed for Chromatic and Achromatic Vision in the Bumblebee, Bombus terrestris". In: *Journal of Neuroscience* 30.11, pp. 3896–3903. DOI: 10.1523/JNEUROSCI.5700-09.2010

Smith, Lincoln, Andrew Philippides, Paul Graham, Bart Baddeley, and Philip Husbands (Sept. 2007). "Linked Local Navigation for Visual Route Guidance". In: *Adaptive Behavior* 15.3, pp. 257–271. DOI: 10.1177/1059712307082091

Smolyanskiy, Nikolai, Alexey Kamenev, Jeffrey Smith, and Stan Birchfield (Sept. 2017). "Toward low-flying autonomous MAV trail navigation using deep neural networks for environmental awareness". In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Vol. 2017-Septe. IEEE, pp. 4241–4247. DOI: 10.1109/IROS.2017.8206285. eprint: 1705.02550

Sommer, Ernst W. and Rüdiger Wehner (1975). "The retina-lamina projection in the visual system of the bee, Apis mellifera". In: *Cell and Tissue Research* 163.1, pp. 45–61. DOI: 10.1007/BF00218590

Son, Bongki, Yunjae Suh, Sungho Kim, Heejae Jung, Jun-Seok Kim, Changwoo Shin, Keunju Park, Kyoobin Lee, Jinman Park, Jooyeon Woo, Yohan Roh, Hyunku Lee, Yibing Wang, Ilia Ovsiannikov, and Hyunsurk Ryu (Feb. 2017). "4.1 A 640×480 dynamic vision sensor with a 9μm pixel and 300Meps address-event representation". In: *2017 IEEE International Solid-State Circuits Conference (ISSCC)*. Vol. 60. IEEE, pp. 66–67. DOI: 10.1109/ISSCC.2017.7870263

Stone, Thomas, Michael Mangan, Antoine Wystrach, and Barbara Webb (Aug. 2018). "Rotation invariant visual processing for spatial memory in insects". In: *Interface Focus* 8.4, p. 20180010. DOI: 10.1098/rsfs.2018.0010

Strausfeld, Nicholas James (1976). "Atlas of an Insect Brain". In: *Springer Berlin Heidelberg*

Stürzl, W. and H.A. Mallot (Apr. 2006). "Efficient visual homing based on Fourier transformed panoramic images". In: *Robotics and Autonomous Systems* 54.4, pp. 300–313. DOI: 10.1016/j.robot.2005.12.001

Stürzl, Wolfgang, Iris Grixa, Elmar Mair, Ajay Narendra, and Jochen Zeil (June 2015). "Three-dimensional models of natural environments and the mapping of navigational information". In: *Journal of Comparative Physiology A* 201.6, pp. 563–584. DOI: 10.1007/s00359-015-1002-y

Stürzl, Wolfgang and Jochen Zeil (2007). "Depth, contrast and view-based homing in outdoor scenes". In: *Biological Cybernetics* 96.5, pp. 519–531. DOI: 10.1007/s00422-007-0147-3

Sun, Deqing, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz (2018). "Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8934–8943

Sun, Xuelong, Shigang Yue, and Michael Mangan (Nov. 2019). "Modelling the Insect Navigation Toolkit: How the Mushroom Bodies and Central Complex Coordinate Guidance Strategies". In: *bioRxiv* Vm, p. 856153. DOI: 10.1101/856153

Szyszka, Paul, Mathias Ditzen, Alexander Galkin, C. Giovanni Galizia, and Randolf Menzel (Nov. 2005). "Sparsening and Temporal Sharpening of Olfactory Representations in the Honeybee Mushroom Bodies". In: *Journal of Neurophysiology* 94.5, pp. 3303–3313. DOI: 10.1152/jn.00397.2005

Tavanaei, Amirhossein, Masoud Ghodrati, Saeed Reza Kheradpisheh, Timothée Masquelier, and Anthony Maida (Mar. 2019). "Deep learning in spiking neural networks". In: *Neural Networks* 111, pp. 47–63. DOI: 10.1016/j.neunet.2018.12.002. eprint: 1804.08150

Tolhurst, D. J., Y. Tadmor, and Tang Chao (Dec. 1992). "Amplitude spectra of natural images". In: *Ophthalmic and Physiological Optics* 12.2, pp. 229–232. DOI: 10.1111/j.1475-1313.1992.tb00296.x

Vardy, Andrew and Ralf Moller (Mar. 2005). "Biologically plausible visual homing methods based on optical flow techniques". In: *Connection Science* 17.1-2, pp. 47–89. DOI: 10.1080/09540090500140958

Wakakuwa, Motohiro, Masumi Kurasawa, Martin Giurfa, and Kentaro Arikawa (Oct. 2005). "Spectral heterogeneity of honeybee ommatidia". In: *Naturwissenschaften* 92.10, pp. 464–467. DOI: 10.1007/s00114-005-0018-5

Webb, Barbara (Feb. 2019). "The internal maps of insects". In: *The Journal of Experimental Biology* 222.Suppl 1, jeb188094. DOI: 10.1242/jeb.188094

Webb, Barbara and Antoine Wystrach (June 2016). "Neural mechanisms of insect navigation". In: *Current Opinion in Insect Science* 15, pp. 27–39. DOI: 10.1016/j.cois.2016.02.011

Wehner, R (2003). "Desert ant navigation: how miniature brains solve complex tasks". In: *Journal of Comparative Physiology A* 189.8, pp. 579–588. DOI: 10.1007/s00359-003-0431-1

Wehner, R. and F. Räber (Dec. 1979). "Visual spatial memory in desert ants,Cataglyphis bicolor (Hymenoptera: Formicidae)". In: *Experientia* 35.12, pp. 1569–1571. DOI: 10.1007/BF01953197

Wehner, Rüdiger (2008). "The desert ant's navigational toolkit: Procedural rather than positional knowledge". In: *Navigation, Journal of the Institute of Navigation* 55.2, pp. 101–114. DOI: 10.1002/j.2161-4296.2008.tb00421.x

Wofk, Diana, Fangchang Ma, Tien-Ju Yang, Sertac Karaman, and Vivienne Sze (May 2019). "FastDepth: Fast Monocular Depth Estimation on Embedded Systems". In: *2019 International Conference on Robotics and Automation (ICRA)*. Vol. 2019-May. IEEE, pp. 6101–6108. DOI: 10.1109/ICRA.2019.8794182. eprint: 1903.03273

Wystrach, Antoine, Cornelia Buehlmann, Sebastian Schwarz, Ken Cheng, and Paul Graham (Apr. 2020). "Rapid Aversive and Memory Trace Learning during Route Navigation in Desert Ants". In: *Current Biology* 30, pp. 1–7. DOI: 10.1016/j.cub.2020.02.082

Wystrach, Antoine, Alex Dewar, Andrew Philippides, and Paul Graham (Feb. 2016). "How do field of view and resolution affect the information content of panoramic scenes for visual navigation? A computational investigation". In: *Journal of Comparative Physiology A* 202.2, pp. 87–95. DOI: 10.1007/s00359-015-1052-1

Wystrach, Antoine, Andrew Philippides, Amandine Aurejac, Ken Cheng, and Paul Graham (July 2014). "Visual scanning behaviours and their role in the navigation of the Australian desert ant Melophorus bagoti". In: *Journal of Comparative Physiology A* 200.7, pp. 615–626. DOI: 10.1007/s00359-014-0900-8

Yang, Gengshan and Deva Ramanan (2019). "Volumetric correspondence networks for optical flow". In: *Advances in neural information processing systems*, pp. 794–805

Yang, Tien-Ju, Andrew Howard, Bo Chen, Xiao Zhang, Alec Go, Mark Sandler, Vivienne Sze, and Hartwig Adam (2018). "Netadapt: Platform-aware neural network adaptation for mobile applications". In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 285–300

Ye, Chengxi, Anton Mitrokhin, Cornelia Fermüller, James A. Yorke, and Yiannis Aloimonos (2018). “Unsupervised Learning of Dense Optical Flow, Depth and Egomotion from Sparse Event Data”. In: eprint: 1809.08625

Yin, Zhichao and Jianping Shi (June 2018). “GeoNet: Unsupervised Learning of Dense Depth, Optical Flow and Camera Pose”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 1983–1992. DOI: 10.1109/CVPR.2018.00212. eprint: 1803.02276

Younes, Georges, Daniel Asmar, Elie Shammas, and John Zelek (2017). “Keyframe-based monocular SLAM: design, survey, and future directions”. In: *Robotics and Autonomous Systems* 98, pp. 67–88. DOI: 10.1016/j.robot.2017.09.010. eprint: 1607.00470

Zeil, Jochen (Apr. 2012). “Visual homing: an insect perspective”. In: *Current Opinion in Neurobiology* 22.2, pp. 285–293. DOI: 10.1016/j.conb.2011.12.008

Zeil, Jochen and Pauline Nikola Fleischmann (2019). “The learning walks of ants (Hymenoptera: Formicidae)”. In: *Myrmecological News* 29, pp. 93–110. DOI: 10.25849/myrmecol.news_029:093

Zeil, Jochen, Martin I. Hofmann, and Javaan S. Chahl (Mar. 2003). “Catchment areas of panoramic snapshots in outdoor scenes”. In: *Journal of the Optical Society of America A* 20.3, p. 450. DOI: 10.1364/JOSAA.20.000450

Zeil, Jochen, Willi A Ribi, and Ajay Narendra (2014). “Polarisation Vision in Ants, Bees and Wasps”. In: *Polarized Light and Polarization Vision in Animal Sciences*. Ed. by Gábor Horváth. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 41–60. DOI: 10.1007/978-3-642-54718-8_3

Zettler, Friedrich and Matti Järvilehto (Sept. 1972). “Lateral inhibition in an insect eye”. In: *Zeitschrift für Vergleichende Physiologie* 76.3, pp. 233–244. DOI: 10.1007/BF00303230

Zhao, ChaoQiang, QiYu Sun, ChongZhen Zhang, Yang Tang, and Feng Qian (Mar. 2020). “Monocular Depth Estimation Based On Deep Learning: An Overview”. In: *Science China Technological Sciences*. DOI: 10.1007/s11431-020-1582-8. eprint: 2003.06620

Zhu, Alex, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis (June 2018). “EV-FlowNet: Self-Supervised Optical Flow Estimation for Event-based Cameras”. In: *Robotics: Science and Systems XIV*. Robotics: Science and Systems Foundation. DOI: 10.15607/RSS.2018.XIV.062. eprint: 1802.06898

Zhu, Alex Zihao, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis (June 2019). “Unsupervised Event-Based Learning of Optical Flow, Depth, and Egomotion”. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 2019-June. IEEE, pp. 989–997. DOI: 10.1109/CVPR.2019.00108. eprint: 1812.08156

Zhu, Le, Michael Mangan, and Barbara Webb (2020). “Spatio-Temporal Memory for Navigation in a Mushroom Body Model”. In: *Biomimetic and Biohybrid Systems. Living Machines 2020. Lecture Notes in Computer Science*. Ed. by Vasiliki Vouloutsi, Anna Mura, Falk Tauber, Thomas Speck, Tony J. Prescott, and Paul F. M. J. Verschure. Vol. 12413. Cham: Springer International Publishing, pp. 415–426. DOI: 10.1007/978-3-030-64313-3_39