Finding the Gap

Schoepe, Thorben; Janotte, Ella; Milde, Moritz B.; Bertrand, Olivier J. N.; Egelhaaf, Martin; Chicca, Elisabetta

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version
Early version, also known as pre-print

Publication date:
2022

[Link to publication in University of Groningen/UMCG research database](#)

Finding the Gap: Neuromorphic Motion Vision in Cluttered Environments

Thorben Schoepe (✉ t.schoepe@rug.nl)

University of Groningen and Bielefeld University

Ella Janotte

Italian Institute of Technology

Moritz Milde

International Centre for Neuromorphic Systems, MARCS Institute, Western Sydney University

Olivier Bertrand

Neurobiology, Faculty of Biology, Bielefeld University

Martin Egelhaaf

Neurobiology, Faculty of Biology, Bielefeld University

Elisabetta Chicca

University of Groningen

Finding the Gap: Neuromorphic Motion Vision in Cluttered Environments

Thorben Schoepe[1,2,3*]**, Ella Janotte**[4]**, Moritz B. Milde**[5]**, Olivier J.N. Bertrand**[6]**, Martin Egelhaaf**[6]**, and Elisabetta Chicca**[1,2,3]

[1]Faculty of Technology and Cognitive Interaction Technology Center of Excellence (CITEC), Bielefeld University, Germany.
[2]Bio-Inspired Circuits and Systems (BICS) Lab. Zernike Institute for Advanced Materials (Zernike Inst Adv Mat), University of Groningen, Netherlands.
[3]CogniGron (Groningen Cognitive Systems and Materials Center), University of Groningen, Netherlands.
[4]Event Driven Perception for Robotics, Italian Institute of Technology, iCub facility, Genoa, Italy.
[5]International Centre for Neuromorphic Systems, MARCS Institute, Western Sydney University, Penrith, Australia.
[6]Neurobiology, Faculty of Biology, Bielefeld University, Bielefeld, Germany.
*t.schoepe@rug.nl

ABSTRACT

Many animals meander in environments and avoid collisions. How the underlying neuronal machinery can yield robust behaviour in a variety of environments remains unclear. In the fly brain, motion-sensitive neurons indicate the presence of nearby objects and directional cues are integrated within an area known as the central complex. Such neuronal machinery, in contrast with the traditional stream-based approach to signal processing, uses an event-based approach, with events occurring when changes are sensed by the animal. Contrary to classical von Neumann computing architectures, event-based neuromorphic hardware is designed to process information asynchronously and in a distributed manner. Inspired by the fly brain, we model, for the first time, a neuromorphic closed-loop system mimicking essential behaviours observed in flying insects, such as meandering in clutter and crossing of gaps, both of which are also highly relevant for autonomous vehicles. We implemented our system both in software and on neuromorphic hardware. While moving through an environment, our agent perceives changes in its surroundings and uses this information for collision avoidance. The agent's manoeuvres result from a closed action-perception loop implementing probabilistic decision-making processes. This loop-closure is thought to have driven the development of neural circuitry in biological agents since the Cambrian explosion. In the fundamental quest to understand neural computation in artificial agents, we come closer to understanding and modelling biological intelligence by closing the loop also in neuromorphic systems. As a closed-loop system, our system deepens our understanding of processing in neural networks and their computations in both biological and artificial systems. With these investigations, we aim to set the foundations for neuromorphic intelligence in the future, moving towards leveraging the full potential of neuromorphic systems.

1 Introduction

While navigating through the environment, our proprioception informs us about our posture, our eyes look for a familiar direction or goal, and our ears watch-out for dangers. The brain deals with multiple data-streams in a continuous and parallel manner. Autonomous vehicles requiring to safely manoeuvre in their environment also have to deal with such high-dimensional data-streams which are conventionally acquired and analysed at a fixed sampling frequency. A fixed sampling frequency limits the temporal resolution of data-processing and the amount of data which can be processed. To address these limitations, two approaches can be combined. First, data-streams can be sparsified by sending only information when an observed quantity changes, i.e. when it is required. Second, the data-stream can be processed in a parallel and asynchronous fashion. This calls for an alternative approach to sensing and computing which, much like the brain, acquires and processes information completely asynchronously and in a distributed network of computing elements, e.g. neurons and synapses. To fully demonstrate the advantages of this approach we use the example of autonomous navigation as it is well studied and algorithmically understood in a variety of environments be they water[1], ground[2], air[3], or space[4]. In the last decades, part of the engineering community has sought inspiration from animals[2,5,6]. For example, flying insects such as bees and flies share the same requirements as light-weight flying vehicles manoeuvring in various habitats from almost object-free terrains[7] to overly cluttered forests[8] via human-made landscapes. They need to avoid collisions to prevent damaging their wings[9] and they accomplish this task by using limited neuronal resources (less than 1M[10] and 100k[11] neurons for honeybees and fruit-flies respectively). At the core of this machinery is a well-described subset of neurons responding to the apparent motion of surrounding objects[12,13]. While the

animal translates in its environment, the responses of such neurons provide estimates to the time-to-contact to nearby objects by approximating the apparent motion of the objects on the retina (i.e. the optic flow[14]). These neurons are thought to steer the animal away from obstacles[15–18] or toward gaps[19–21] resulting in a collision-free path.

The collision avoidance machinery in insects is thought to be driven by a large array of motion-sensitive neurons, distributed in an omnidirectional visual field. These neurons operate asynchronously. Hence, biology has found an asynchronous and distributed solution to the problem of collision avoidance. We seek to emulate such a solution in bio-inspired neuromorphic hardware which has the advantage of being low-volume and low-power. More importantly, it also requires an asynchronous and parallel information processing implementation yielding a better understanding of neural computation.

To date, most of the mimics of the collision avoidance machinery rely on traditional cameras from which every pixel at every time point (i.e. at a fixed sampling frequency) needs to be processed[6, 15, 16, 22–24]. The processing occurs even when nothing is changing in the agent's surroundings. This constant processing leads to a dense stream of data and consequently a high energy consumption. To reduce this, an efficient means of communication can be employed, such as action potentials observed in biological neural circuits. Action potentials or spikes enable to transmit information only when necessary, i.e. event-driven. In an analogous way, event-based cameras send events asynchronously only when a change in luminance over time is observed[25–29]. This sampling scheme is referred to as Lebesgue sampling[30]. Contrary to frame-based cameras, which employ Riemann sampling[30], bandwidth and power demands are significantly reduced (see Section Event-Based Cameras in Gazebo for more details).

Open-loop collision avoidance based on optic-flow can use event-streams[31–39](for more detailed comparison of mentioned approaches refer to[39]) and an insect-inspired motion pathway has been suggested for collision avoidance[39]. Closed-loop collision avoidance behaviour have been demonstrated previously using fully conventional sensory-processing (frame-based sensor and CPUs/GPUs) approaches[23, 24] (for extensive review please refer to[40, 41]). These insect-inspired approaches reduce the computational demands for collision avoidance by reducing the bandwidth of the visual input. This reduction is achieved by collapsing the visual field into a left and right components. Later processing only needs to compare left versus right signals. These approaches, however, are hardwired processing of visual features. The hard-coded features may not be relevant in other environments. Mixed-system (event-based camera and conventional processing) approaches[36, 42], on the other hand, do not reduce the visual input by separating left-right signal pathways, but utilise event-based cameras which only transmit changes. In contrast to biological systems, they do not, however, leverage the advantages of event-based processing until the actuation of the motors. Finally, fully neuromorphic (event-based camera and parallel, asynchronous processing) approaches[43, 44] rely on spike-based information processing from sensing to actuation of motors. To date, these approaches rely on hardwired, deterministic decision making processing. The hard-coded decisions, i.e. creating a reflex-like machine, may lead to sub-optimal decisions when multiple directions to avoid collisions are viable. Here, we aim for the first time at closing the action-perception loop[40, 41, 45], while explicitly extracting insect-inspired visual features, making active decisions, and using neuromorphic spike-based computation from sensing to actuation. Inspired by the collision avoidance algorithm proposed for flies and bees, we developed a spiking neural network (SNN)[1] that profits from the parsimony of event-based cameras and is compatible with state-of-the-art digital and mixed-signal neuromorphic processing systems. The response of the visual motion pathway of our network resembles the activity of motion-sensitive neurons in the visual system of flies. We ran closed-loop experiments with an autonomous agent in a variety of conditions to assess the collision avoidance and gap finding capabilities of our network. These conditions were chosen from the biological evidence for collision avoidance obtained for flying insects (empty box[46], corridors[47–50], gap crossing[8, 20, 21], and cluttered environments[9]). Our agent, utilising its underlying neural network, manages to stay away from walls in a box, centres in corridors, crosses gaps and meanders in cluttered environments. Therefore, it may find applications for autonomous vehicles. Besides, it may serve as a theoretical playground to understand biological systems by using neuromorphic principles replicating an entire action-perception loop.

2 Results

The SNN model proposed in this work consists of two main components, namely a retinotopical map of insect-inspired motion detectors, i.e. spiking Elementary Motion Detectors (sEMDs)[39], and an inverse soft Winner-Take-All (WTA) network (see Figure 1d and Methods Figure 4). The former extracts optic flow (OF) which, during a translation, is anti-proportionally related to the agent's relative distance to objects in the environment. The latter searches for a region of low apparent motion, hence an obstacle free direction (see Figure 1a-c). After the detection of such a path in the environment the agent executes a turn towards the new movement course. We characterised the network in two steps. First we evaluated the sEMD's response and discussed similarities to its biological counterpart, i.e. T4/T5 neurons, which are thought to be at the core of elementary motion processing in fruit flies[51, 52]. Second, to further prove the real-world applicability of sEMD based gap finding in an SNN, we performed closed-loop experiments. We simulated an agent seeing the world through an event-based camera in the Neurorobotics physical

---

[1]Spiking Neural Network: Massively parallel network consisting of populations of spike-based artificial neurons and synapses.

simulation platform[53]. The camera output was processed by the SNN resulting in a steering command. We selected a set of parameters that yield the agent to keep at least a mean clearance of ~6 a.u.[2] to objects in a box and to enter corridors only with a width greater than 10 a.u. (see Appendix section The Motion-Vision Network). We tested the performance of this simulated agent with these parameters in all reported experimental conditions hereafter. These experimental conditions were inspired by previous experiments with flying insects.

2.1 Spiking Elementary Motion Detector

The sEMD represents an event-driven adaptation for neuromorphic sensory-processing systems of the well established correlation-based elementary motion detector[54]. To evaluate the response of the sEMD in the Nest simulator[55], we compared the normalised velocity tuning curves of its ON-Pathway (with recorded event-based camera's input) to the corresponding normalised tuning curve of *Drosophila's* T4 and T5 neurons[56]. Both velocity tuning curves are determined in response to square-wave gratings with 100 % contrast and a wavelength of 20° moving at a range of constant velocities (with temporal frequencies from 0.1 Hz to 10 Hz). The sEMD preferred direction exhibits a bell-shaped velocity tuning curve (see Figure 1 e), which has the maximum response (mean population activity) at 5 Hz. The null direction response is much lower than the preferred direction.

The sEMD model, which is composed of an event-based camera, a Spatio-Temporal Correlation (SPTC) population and the Time Difference Encoder (TDE) (see Figure 4), exhibits a drop in its output response when the temporal frequency exceeds 5 Hz. This drop is, however, not anticipated from the TDE's response (see Figure 3). We would expect the response to saturate at high temporal frequencies since the TDE produces interspike intervals and firing rates proportional to the time difference between the two inputs of the TDE. The drop in response being rather a consequence of the motion detector model itself, we suggest it to be a consequence of the spatio-temporal band-pass filtering installed by the SPTC layer. While low temporal frequencies lead to unambiguous spatio-temporally correlated and causal SPTC spikes from adjacent neurons, high temporal frequencies lead to anti-correlated and non-causal spikes. Thus, the TDE can no longer (spatially) match the spikes unambiguously, which results in a bell-shaped velocity tuning curve of the preferred direction response.

A similar bell-shaped velocity tuning curve can be observed in *Drosophila's* T4 cells[12,51,56]. While *Drosophila's* velocity tuning curves peak at 3 Hz in a drug induced flying state, the sEMD's preferred direction velocity tuning curve peaks at 5 Hz. This suggests that based on the reported parameter set of the sEMD, it is tuned to higher relative velocities. The model performs in a robust way for a wide range of illuminations (from 5 lux to 5000 lux) and relative contrasts (50 % response reached at approximately 35 % relative contrast), as shown in Figure A.2. The sEMD approximates the elementary motion processing in the fly brain. This processing is part of the input to the flight control and collision avoidance machinery, hence it can be used as an input for determining a collision-free path.

2.2 Agent's Behaviour

The robot's collision avoidance performance was evaluated in an experiment with the agent moving through environments with varying obstacle density. To further understand the mechanisms underlying the robot's movement performance two more experiments were designed. The agent's gap crossing behaviour and tunnel centering behaviour were investigated. These behaviour were analysed in insects in a plane, therefore little is known about the effect of flying altitude in most behaviour. We limited our agent to a 2D motion due to this limited understanding.

2.2.1 Densely Cluttered Environments

We evaluated the agent's collision avoidance performance in an arena with an obstacle density[3] between 0 and 38 %(0.05 objects per square a.u.). The simulation stops either when the robot collides with an obstacle[4], when it leaves the arena, or when the simulation real-world-time of six hours is over (see Figure 2f). At low obstacle densities ($< 5\%$) there exist several collision-free paths. The robot exhibits a random walk as the decision making inverse WTA neuron population is receiving background spiking activity sampled from a Poisson process (see Figure 2a,f). In the absence of OF input the Poisson distributed background spikes dominates the inverse WTA output which results in a probabilistic decision process. The decisions made by the network become less probabilistic with increasing obstacle density since the robot starts to follow the locally low object-density paths forming in the environment (see Figure 2b,f). At obstacle densities higher than 20 % most of the gaps in the environment are smaller than the robot's minimum mean obstacle clearance[5] of 7 a.u. (see Figure A.5 left) so that the agent stays close to its start location (see Figure A.5 right and Figure 2c,f). In this range the robot starts to crash into obstacles reaching a minimum success rate of around 60 % at 22 % obstacle density. For higher obstacle densities the success rate increases again (see Figure 2i). A collision of the robot is generally caused by the robot's too long reaction time in

---

[2]A.u.: Arbitrary unit, distance divided by robot size, see section Closed-loop simulation in environments

[3]Obstacle density: Percentage of total area covered with objects.

[4]Collision: Simulated robot's outline overlaps with area occupied by object

[5]Obstacle clearance: Robot's distance to the center of the closest object.
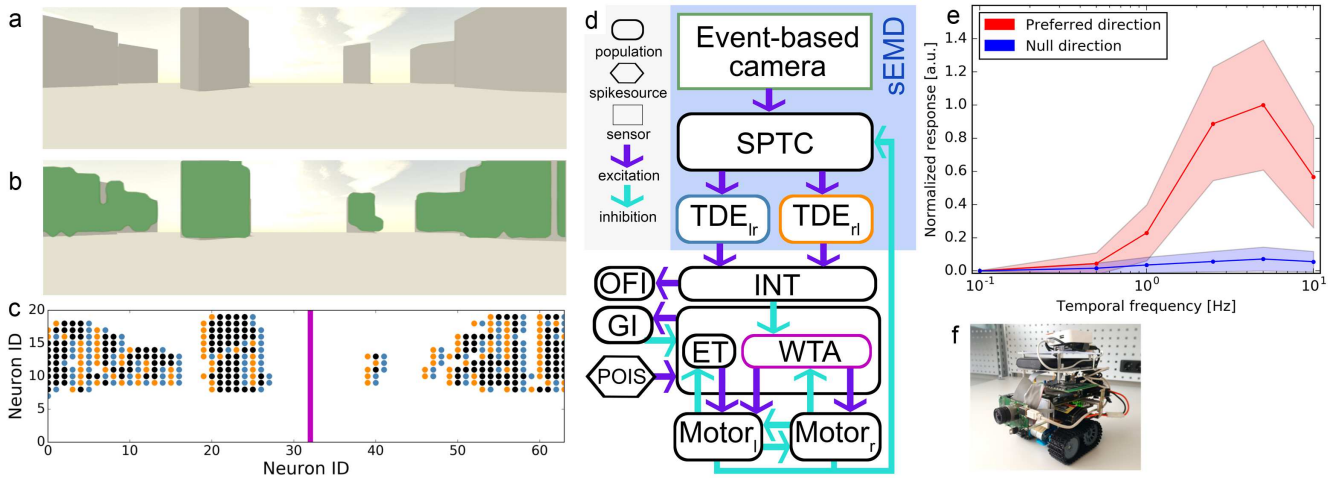
Figure 1. (a-c) Network response in a cluttered environment, (d) collision avoidance network, (e) normalised sEMD mean response to a square wave grating and (f) robot used in real-world experiment. a) Cluttered neurorobotics platform environment. The obstacle walls are covered with vertical square-wave-gratings only visible for the event-based camera b) Green areas: simulated event-based camera events directly extracted from the Neurorobotics visual front-end while the agent is slowly moving through the scene in a). c) Bright blue, orange and black dots: Spike response of left-right Time Difference Encoder (TDE), right-left TDE and both TDEs together to the scene in a) binned over ~0.5 seconds of simulation time, pink stripe: inverse WTA spike which indicates an obstacle free direction. This spike activates the agent's next turning movement (saccade) by exciting one of the two motor populations in d). The location of the winning neuron defines the direction and duration of the upcoming turn. d) Collision avoidance network with spiking Elementary Motion Detectors (sEMDs) which consist of an event-based camera, a Spatio-Temporal Correlation (SPTC) population and two Time Difference Encoder (TDE) populations. Event-based camera (sensory input), SPTC population (noise filter and downsampling), TDE populations (time-to-travel translation to spike rate and Inter-Spike-Interval (ISI)), Integrator (INT) population (reduces 2D retinotopical map to 1D), inverse Winner-Take-All (WTA) population (detects minimum of OF, hence obstacle free direction), Escape Turn (ET) population (triggers turn when inverse WTA can not find direction), Motor (MOT) populations (control turn direction and duration), Optic Flow Integrator (OFI) population (modulates robot velocity), Poisson Spike Generators (POIS) (drive decision process with Poisson spike process) and Global Inhibition (GI) population (suppresses loosing neurons in inverse WTA population and ET population). e) Normalised preferred direction and null direction mean response of two Nest sEMD populations to a square wave grating moving in one cardinal direction with a wavelength of 20° and 100 % relative contrast recorded with an event-based camera at 5000 lx illumination. The standard deviation was calculated on the response of the sEMD population. f) Robot used for real-world experiment in Figure 2d. An embedded event-based camera serves as input to a SpiNN-5 board which drives the motor controllers through an Odroid mini-computer.

an environment with low mean obstacle clearance, hence with high obstacle density (see Figure A.5). Since the robot only senses visual stimuli in a 140 degrees horizontal visual field, symmetrically centered around its direction of motion, there is a blind-spot behind the agent. After a strong turn the simulated robot might be confronted with a previously not seen object and directly crash into it. Nevertheless, the agent shows a very robust gap centering behaviour in a large range of different environments with obstacle densities between 0 and 38 %. The robot's mean success rate amounts to 81 %.

While local OF is instrumental in finding gaps, global OF provides information about the clutteredness of the environment. Flies and bees decrease their flying speed when the clutteredness of the environment increases[48,49]. Our agent regulates its speed based on the global OF and, consequently, moves slower in denser regions of the environment (see Figure A.7). To examine the effect of the velocity dependency, we ran a second experiment with the robot moving with constant velocity (see Figure 2i and Figure A.6). With velocity control collisions were encountered only in few runs, however, for obstacle densities higher than 24 percent the number of collisions significantly increased when the velocity was kept constant.

2.2.2 Gaps

When presented with a choice between two gaps of different size bees prefer to pass the larger gap[8,57]. This behaviour decreases the insect's collision probability significantly. While bees might choose the gap in a complex decision process[21] our agent's preference underlies a simple probabilistic integration mechanism. The simulated robot's upcoming movement direction is determined by an inverse WTA spike occurring in an obstacle-free direction as shown in Figure 1a-c. When confronted with a
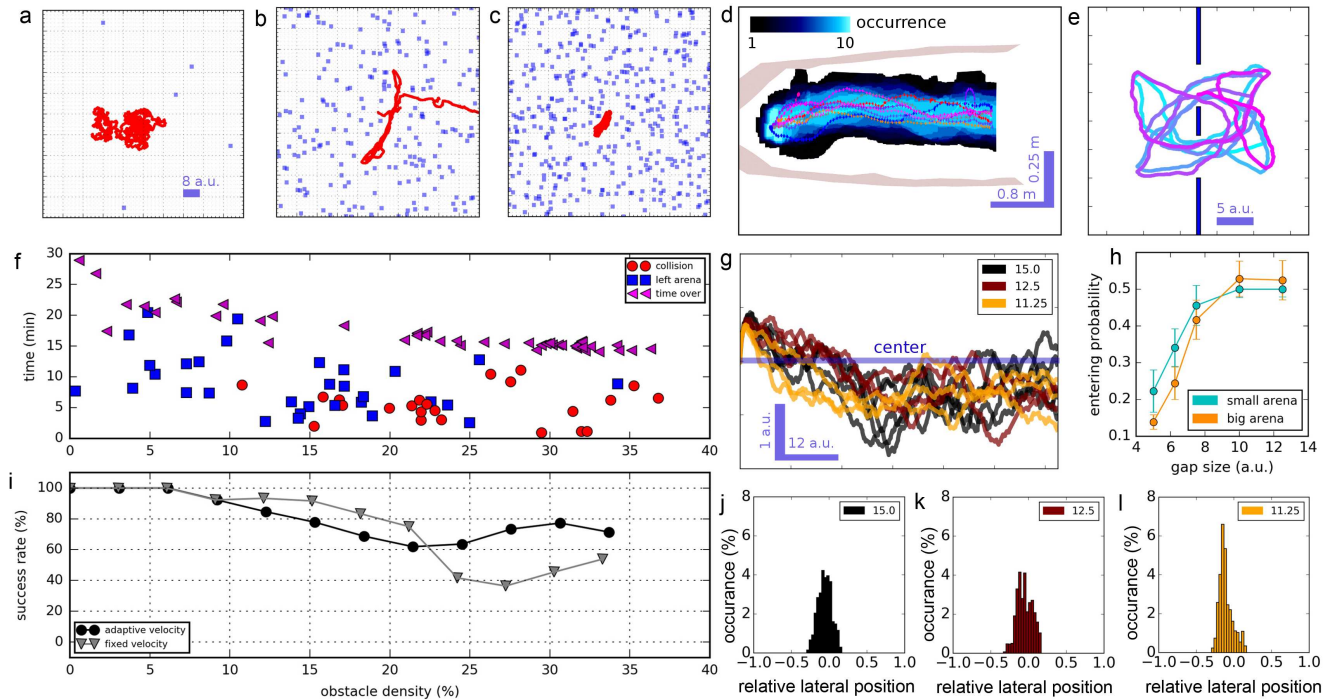
Figure 2. Agent's behaviour in different environments. a-c) Trajectories recorded in arenas with increasing obstacle densities. d) Real-world centering behaviour of the robot shown in Figure 1f. Grey areas mark the corridor walls. Small dots indicate the agent's center of mass while the blue area shows the frequency of occurrence of the whole robot. For more information see appendix section Corridor-centering in Real-World. e) Simulated robot's trajectory in the gap crossing experiment in a large arena. Colour represents time (t_0: light blue, t_{end}: magenta). f) Simulated robot's performance in different environments as shown in a-c with modulated velocity. Simulation time at which the simulated robot leaves the arena, collides or the time is over. g) Trajectories in tunnels with a tunnel width of 15, 12.5 and 11.25 a.u.. h) Gap crossing probability in dependency of the gap width for a large and a small arena. i) Simulated robot's performance in cluttered environments as shown in a-c with modulated velocity (black, calculated from data in f) and fixed velocity (grey). Agent's success rate, hence number of runs without collisions. j-l) Agent's variance from tunnel center for different tunnels.

small and a large gap the probability of an inverse WTA spike appearing in the greater gap is higher. Hence, we assume that the robot automatically follows pathways with a larger gap size. To evaluate this assumption we observed the robot's gap crossing in an arena with two alternative gaps (see Figure 2e). The robot can decide to cross any of the two gaps or stay in one half of the arena. There is a competition between staying in the open-space and crossing a gap. The larger the gap size is, the more likely the robot will cross a gap. We investigated the probability to cross gaps by having two gaps, one with a fixed gap size (10 times the agent width), the other with a gap size between 5 a.u and 13 a.u. We calculated the gap entering probability by comparing the number of passes through both gaps. As expected the entering probability increases with gap size until a width of 10 a.u. (see Figure 2h). For a larger gap width the entering probability does not change significantly. However, for smaller gap sizes the probability of a spike pointing towards open space in the inverse WTA becomes significantly higher. Therefore, the robot prefers to pass through gaps of larger size. Besides the gap width the arena size changes the passing probability. In a smaller arena the simulated robot stays closer to the gap entry which increases the relative gap size sensed by the agent. Therefore, a larger part of the vehicle's visual field is occupied by the gap entry which increases the probability of a spike occurring in the gap area. In a smaller arena we observed that the robot's gap entering probability is higher for gaps smaller then 10 a.u. than in a big arena (see Figure 2h). A decrease in arena size can be compared to an increase in obstacle density since both parameters reduce the robot's obstacle mean clearance (see Figure A.5, left). Therefore, the agent tends to enter gaps of smaller size in densely cluttered environments. This automatic scaling mechanism keeps the agent's collision probability very low in sparsely cluttered environments by staying away from small gaps. In environments with high obstacle density the robot still keeps its mobility by passing through smaller gaps. Finally, when the obstacle density exceeds 20 %, most gaps fall below the gap entering threshold so that the robot can not leave the arena anymore (see Figure A.5, right and Figure 2c,f).

2.2.3 Corridors

One common experiment to characterise an agent's motion vision response is to observe its centering behaviour in a tunnel equipped with vertical stripes on the walls. The simple geometry of the environment enables the observer to directly relate the received visual input with the agent's actions. In bees and flies an increase in flight velocity proportionally to the tunnel width has been observed[17,48,49]. In very narrow tunnels insects show a pronounced centering behaviour which declines with increasing tunnel width. We evaluated the robot's performance in three tunnels with different tunnel widths. Similar to the biological role model the robot's velocity stands in a positive linear relationship with the tunnel width. The measured velocity in a.u. per second is ~0.79, ~0.75 and ~0.72 for a tunnel width of 15, 12.5 and 11.25 a.u. respectively. Furthermore, the robot always stays in the center of the tunnel, especially in very narrow tunnels (see Figure 2g). The deviation from the tunnel center is proportional to the tunnel width (for the simulated robot, see Figure 2j–l, physical robot see 2d). Therefore, similar to observations in blowflies, the robot's lateral position in the tunnel changes linearly with the tunnel width[49].

3 Discussion

Autonomous agents need to successfully avoid obstacles in a variety of different environments, be they human made or of natural origin. Our investigations present a closed-loop proof of concept of how obstacle avoidance could be performed in a parsimonious, asynchronous and fully distributed fashion. While most results reported here are based on computer simulations, the implementation on digital or mixed-signal neuromorphic hardware of each building block of the simulated SNN have been demonstrated for event-based cameras[26], the sEMD[39,58] (see Figure A.2), artificial neurons[59] and synapses[60], as well as the inverse WTA[61]. We demonstrated for the first time a simulation of a neuromorphic system that takes informed decisions while moving in its environment by closing the action-perception loop. We emulated this system on neuromorphic sensory-processing hardware carried by a physical robot (see Figure 1f, 2d, A.8 and A.9), tested it in a corridor centering experiment, and obtained similar results to the simulation. These real-world experiments suggest that the underlying computational primitives lead to robust decision making in operational real-time. Due to the physical simulation with the engine Gazebo that capture the physics of the movements and our real-world proof of implementation, our simulations are likely to translate to real-world situations. While producing relatively simple, yet crucial decisions, the proposed model represents a critical milestone towards enabling parallel, asynchronous and purely event-driven neuromorphic systems.

Our proposed SNN architecture comprises of ~300k synapses and ~4k neurons which yields a low-power, lightweight and robust neural algorithm. When implemented on mixed-signal neuromorphic processing hardware, e.g.[62–64], the payload required to perform on-board processing will be drastically reduced. This reduction stems from the low volume and lower power requirements of neuromorphic hardware. In addition such hardware implementation would ensure operational real-time decision making capabilities. The features outlined above are quite desirable in the context of highly constrained autonomous systems such as drones or other unmanned vehicles.

We investigated the performance of the sEMDs, the apparent motion encoders in our SNN, in detail. The sEMDs show a similar velocity response curve to motion-sensitive neurons (e.g. T4 and T5 neurons in the fruitfly's brain[12,18]) when presented with a grating of 20° spatial frequency and temporal frequencies between 0.1 and 10 Hz. Due to the logarithmic illumination sensitivity of the event-based cameras the motion vision apparatus is very robust against absolute brightness levels in the range of 5 up to 5000 lux. The sEMD model shows a much higher sensitivity regarding contrast changes than its biological role model. Current research suggest that *Drosophila*'s optical lobe performs contrast normalisation through inhibitory recurrent feedback to evoke a contrast independent response[52]. In a next step we will implement contrast normalisation in our motion vision network to improve its performance in natural environments.

Besides the similarities in neural response, the agent showed many similarities to flying insects in its behaviour in spatially constrained environments. It meandered in cluttered terrain (Section Densely Cluttered Environments), modulated its speed as a function of object proximity (Section Corridors), selected wider gaps (Section Gaps), centered in tunnels (Section Corridors), while using an active gaze strategy known as saccadic flight control (Section Collision Avoidance Network)[17,48,49,57,65]. The agent moved collision-free through cluttered environments with an obstacle density between 0 and 38 % with a mean success rate of 81 %[6]. We further examined the simulated robot's performance to understand the essential behavioural components which led to a low collision rate. The most significant ingredient in that regard was the implementation of an OF strength dependent locomotion velocity. This insect inspired control mechanism improved the collision avoidance performance of the agent from a mean success rate of 76 % to 81 % (Compare Figure 2i and Figure A.6). We propose that this velocity adaptation mechanism could be regulated in insects by a simple feedback control loop. This loop changes the agent's velocity anti-proportionally to the global OF integrated by a subset of neurons (For further explanations see Section Collision Avoidance Network).

---

[6]Several closed-loop, insect-inspired approaches have been demonstrated[40,41], however, due to a missing unifying benchmark and evaluation metric, to compare insect-inspired collision avoidance algorithms, we cannot provide a quantitative comparison

An OF-dependent control of locomotion velocity is only one of at least three mechanisms which decreased the agent's rate of collision. When moving in environments of high obstacle density the simulated robot follows locally low obstacle density paths. We suggest that a probabilistic decision process in the network model automatically keeps the agent's collision probability low by following these pathways. We further investigated this path choice mechanism in a second experiment. Here, the agent had to cross two gaps of different size. The dependence of the agent' probability to cross the gap resembled that of bees[57]. Similar to insects the agent preferred gaps of larger size. Bees cross gaps with a gap-size as small as 1.5 times their wingspan[21]. In contrast our agent crossed gaps of 5 times its body width. This discrepancy in performance may be due to the absence of a goal. A goal can be understood as providing an incentive to cross a gap despite a risk of collision. Indeed in behavioural experiments, bees had to cross the gap to return to their home. Combining different directions, such as a collision-free path and a goal, require an integration of the two signal representations. Such networks have been proposed for navigating insects[66]. Integration of similar streams of information have been demonstrated to work in neuromorphic systems[58,67,68], however, we envision that a dynamic competition between collision avoidance and goal reaching neural representations could allow our robot to cross gaps 1.5 times its width.

The findings reported here indicate an alternative point of view how flies and bees could use motion-vision input to move through the environment, not by collision avoidance but by gap finding. As also stated by Baird and Dacke[8], flies and bees might not actively avoid obstacles but fly towards open space, i.e. gaps. Looking at our network, we suggest that WTA alike structures in flying insect brains might integrate different sensory inhibitory and excitatory inputs with previously acquired knowledge to take navigational decisions. One could think of the central complex as such a structure which has been described recently in several insect species[69].

The third mechanism is the agent's centering behaviour. By staying in the middle of a tunnel with similar patterns on both walls the simulated robot minimises its risk of colliding with a wall. The agent's deviation from the tunnel center changes approximately linearly with the tunnel width. These results show a very strong resemblance with experimental data from blowflies (see Figure 2j–l)[49]. So far centering behaviour was suggested to result from balancing the OF on both eyes. Centering in a tunnel can be seen as crossing elongated gaps. Our agent is also able to cross gaps. Two hypothesis have been suggested to cross gaps in flying insects, using the OF contrast[20] and the brightness[8]. Our results suggest that collision avoidance could be mediated by identifying minimum optic flow to center in tunnel, cross gaps, or meander in cluttered environment. This strategy has so far not been investigated in flying insects. The main hypothesis to control flight in clutter is to balance either an average or the maximum OF on both eyes[17]. Further behavioural experiments are required to disentangle between the different strategies and their potential interaction. Building on the work of[8], the different hypothesis could be placed into conflict by creating a point-symmetric OF around the gap center (leading to centering), a brightest point away from the gap center, and a minimum OF away from the center (e.g. by using an OF amplitude following a Mexican hat function of the radius from the geometric center).

Our model shares several similarities with the neural correlate of visually-guided behaviour in insects, including motion-sensitive neurons[18], an integration of direction[70], efference copy to motion-sensitive neurons[71], and neurons controlling the saccade amplitude[65]. Our agent was able to adopt an active gaze strategy thanks to a saccadic suppression mechanism (due to an inhibitory efference copy from the motor neurons to the inverse WTA and motion-sensitive neurons). When the inverse WTA layer did not "find" a collision-free path (i.e. a solution to the gap finding task), an alternative response (here a U-turn) was triggered thanks to global inhibitory neurons and excitatory-inhibitory networks (GI-WTA-ET, for more details see Section Collision Avoidance Network). The neuronal correlate of such a switch, to our knowledge, has not been described in flying insects. Our model, thus, serves as a working hypothesis for such a neuronal correlate. Furthermore, by varying the connection between sEMD-inverse WTA, we could allow the agent to cross smaller gaps. We hypothesise that differences in clearance or centering behaviour observed between insect species[8,20] could be due to different wiring or modulation between motion-sensitivity neurons and direction selection layer, likely located in the central complex.

In this study we demonstrated a system-level analysis of a distributed, parallel and asynchronous neural algorithm to enable neuromorphic hardware to perform insect-inspired collision avoidance. To perform a wide variety of biological-relevant behaviour the network comprised approximately 4k neurons and 300k synapses. The agent guided by the algorithm robustly avoided collision in a variety of situations and environments, from centering in a tunnel to crossing densely cluttered terrain and even gap finding, solved by flying insects. These behaviour were accomplished with a single set of parameters, which have not been optimised for any of those. From the investigation of the agent and its underlying behaviour, we hypothesise that insects control their flight by identifying regions of low apparent motion, and that excitatory-inhibitory neural structures drive switches between different behaviours. With these investigations we hope to advance our understanding of closed-loop artificial neural computation and start to bridge the gap between biological intelligence and its neuromorphic aspiration.

4 Methods

Most experiments in this article were conducted in simulation using either the Nest spiking neural network (SNN) simulator[55] or the Neurorobotics Platform environment[53]. A corridor centering experiment was conducted in a real-world corridor centering experiment using a robotic platform equipped with the Dynamic Vision Sensor as visual input and a SpiNN-3[63] board for SNN simulation in operational real-time. Sensory data for the sEMD characterisation were recorded with an event-based camera in a real world environment. The hardware, software, SNN models and methodologies used in this article are explained in the following.

4.1 Spiking Neural Networks

In contrast to conventional processing as postulated by von Neumann[72] which is characterised by synchronous and inherently sequential processing, neural networks, whether rate-based or spike-based, feature parallel and distributed processing. Artificial neural networks, the rate-based counterpart of SNNs, perform synchronous and clock-driven processing, SNNs, additionally, feature an asynchronous and event-driven processing style. SNNs represent a promising alternative to conventional von Neumann processing and hence computing which potentially feature low-latency, low-power, distributed and parallel computation. Neuromorphic hardware present a solution to the aforementioned limitations of conventional von Neumann architectures including parallel, distributed processing[73] in the absence of a central clock[74,75], as well as co-localisation of memory and computation[76,77]. Moreover, neuromorphic processors benefit from the underlying algorithm to be implemented in a SNN. Emulating a SNN on a neuromorphic processor (especially a mixed-signal one) enables the network to operate in continuous time[7] as time represents itself[74]. SNNs consist of massively parallel connected networks of artificial synapses and spiking neurons[78]. SNNs, as any processing algorithm, aim to structure and represent incoming information (e.g. measurements) in a stable, robust and compressed manner (e.g. memory). Measurements sampled at fixed time intervals have the disadvantage that collected data is highly redundant and prone to aliasing if the signal of interest varies faster than half the sampling frequency. Event-driven approaches to sampling alleviate these limitations. As incoming measurements shouldn't be sampled at fixed temporal intervals (i.e. Riemann sampling), they need to be taken based on fixed or relative amplitude changes (i.e. Lebesque sampling) to take full advantage of the time-continuous nature of SNNs and neuromorphic hardware[30]. Such measurements can be obtained from different sensory domains (e.g. touch[79], smell[80], auditory[81,82] and vision[83,84]), with vision being the most studied and well understood sensory pathway (but see[85] for a critical review) both in the brain and its artificial aspiration. While images taken with conventional cameras can be converted to spike trains which are proportional to the pixel intensity[8], event-based cameras directly sample only relative changes of log intensity and transmit these changes as events. A variety of event-based cameras have been proposed in the last two decades[25–27,84] that all feature an asynchronous, parallel sampling scheme[9] in which changes are reported at the time of occurrence in complete time-continuous manner. The output of event-based cameras is hence ideally suited to be processed by an SNN implemented on a neuromorphic processor. We collected real-world data using the DVS128 event-based camera[26] to characterise the sEMD response (see Figure 1e). The event-based camera comprises 128×128 independently operating pixels which respond to relative changes in log-intensity, i.e. in temporal contrast. When the change in light intensity exceeds an adaptive threshold the corresponding pixel produces an event. The address and polarity of the pixel are communicated through an Address Event Representation bus[83]. Light increments lead to ON-events, whereas light decrements lead to OFF-events. The sensor reaches a dynamic range of more than 120 dB and is highly invariant to the absolute level of illumination due to the logarithmic nature of the switched-capacitor differencing circuit[26,84].

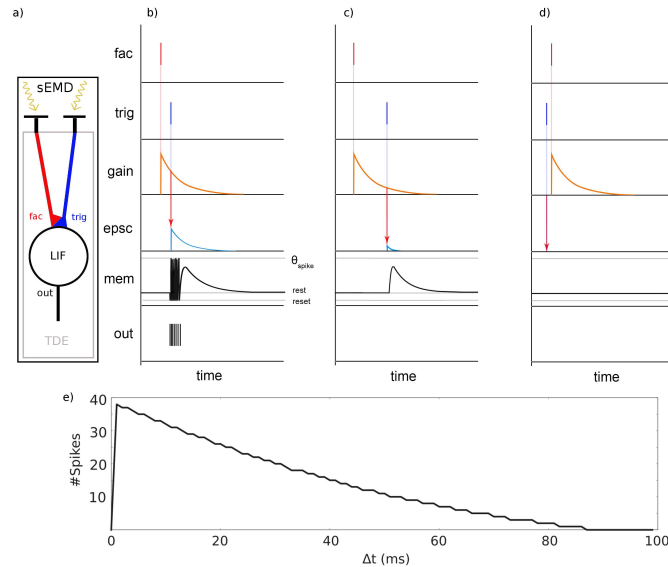### 4.2 Spiking Elementary Motion Detector

In 2018 we proposed a new insect-inspired building block for motion vision in the framework of SNNs designed to operate on the output event-stream of event-based cameras, the spiking Elementary Motion Detector (sEMD)[39]. The sEMD is inspired by the computation of apparent motion, i.e. optic flow (OF), in flying insects. In contrast to its correlation-based role model the sEMD is spike-based. It translates the time-to-travel of a spatio-temporally correlated pairs of events into direction dependent, output burst of spikes. While the sEMD provides OF estimates with higher precision when the entire burst is considered (rate-code), the interspike interval distribution (temporal-code) within the burst provides low-latency estimates. The sEMD consists of two building blocks, a retina to extract visual information from the environment, and the TDE which translates the temporal difference into output spikes (see Figure 3a). When the sEMD receives an input spike at its facilitatory pathway an exponentially decaying gain variable is generated. The magnitude of the synaptic gain variable during the arrival of a spike at

---

[7]A time-continuous mode of operation, in contrast to a time-varying one, is characterised by the absence of a fixed sampling frequency

[8]To perform this conversion one can use a different encoding schemes including rank-order code[86], timing-code[87,88] or Poisson rate-code.

[9]Level or Lebesque sampling means that a given time-continuous signal is sampled when the level changes by fixed (relative) amount ε, whereas time sampling, i.e. Riemann sampling, means that the signal is sampled when the time has changed by fixed amount ε[30]

the trigger synapse defines the amplitude of the excitatory post-synaptic current generated. This current is integrated onto the sEMD's membrane potential which generates a short burst of output spikes. Therefore, the number of output spikes encodes direction sensitive and anti-proportionally the stimulus' time-to-travel (see Figure 3e) between two adjacent input pixels. We implemented and evaluated the motion detector model in various software applications (Brian2, Nengo, Nest), in neuromorphic digital hardware (SpiNNaker, Loihi) and also as analog CMOS circuit[39,58].

Figure 3. Spiking Elementary Motion Detector model adapted from[89]. a) sEMD model consisting of visual input and TDE unit. Two adjacent retina inputs are connected to the facilitatory synapse (fac) and the trigger synapse (trig), respectively. The facilitatory synapse controls the gain of the trigger synapse's postsynaptic current (epsc) which integrates onto the Leaky Integrate and Fire (LIF) neuron's membrane potential which produces output spikes (out). b) Model behaviour for small positive Δt. c) Behaviour for large positive Δt. d) Behaviour for negative Δt. e) Number of output spikes over Δt.
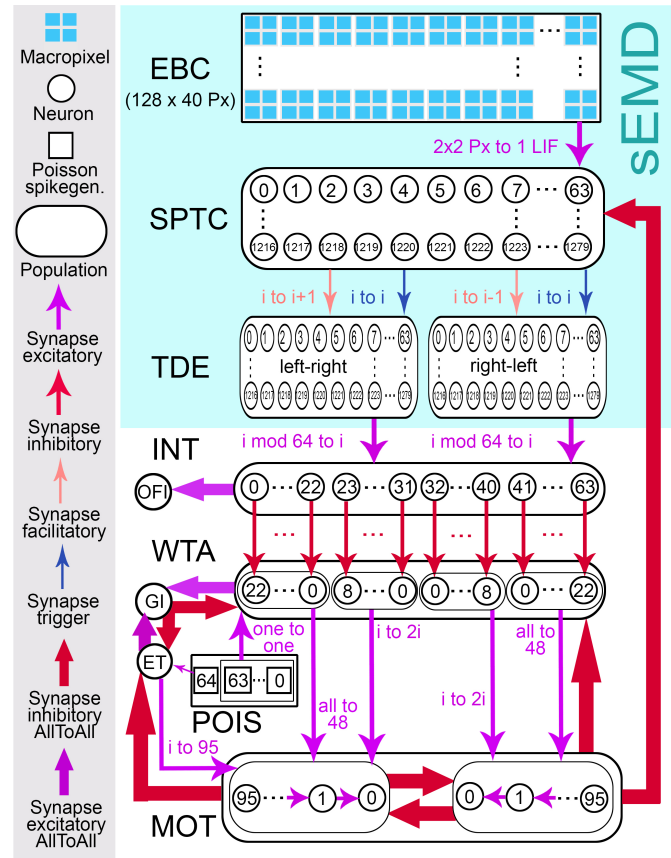
4.3 Collision Avoidance Network

The collision avoidance network (see Figure 4) extracts a collision-free direction from its sEMD outputs and translates this spatial information into a steering command towards open space. The first layer, the event-based camera, generates an event when a relative change in log-illumination, i.e. temporal contrast, is perceived by a pixel. A macropixel consists of 2 x 2 event-based camera pixels. Each macropixel projects onto a single current-based exponential LIF neuron (hereafter referred to as LIF for sake of clarity) in the Spatio-Temporal Correlation (SPTC) layer (in Nest the neuron model used throughout this study is called iaf_psc_exp). Each single SPTC neuron emits a spike only when more than 50% of its receptive field elicit an event within a rolling window of 20 ms. Therefore, the SPTC population removes uncorrelated events, which can be interpreted as noise. Additionally, it decreases the network resolution from 128 times 40 pixels to 64 times 20 neurons. The next layer extracts OF information from the filtered visual stimulus. It consists of two TDE populations sensitive to the two horizontal cardinal directions respectively. Each TDE receives facilitatory input from its adjacent SPTC neuron and trigger input from its corresponding SPTC neuron. The facilitatory input might arise either from the left (left-right population) or from the right (right-left population). The TDE output encodes the OF as number of spikes in a two-dimensional retinotopical map. Since the agent moves on the ground it only estimates the amount of horizontal OF. Hence, the subsequent INT population integrates the spikes of each TDE column in a single LIF neuron. This layer encodes the OF in a one-dimensional retinotopical map. The subsequent population, an inverse soft Winner-Take-All (WTA) determines the agent's movement direction, a minimum of OF in the one-dimensional retinotopical map. Since OF encodes the relative distance to objects during a translational movement this direction represents an object-free pathway, hence the inverse Winner-Take-All (WTA) is inverted by sending feed-forward inhibition into the neural population. A population of POIS injects Poisson distributed background spikes which ensures a neuron within the inverse WTA to win at any moment in time even in the absence of OF. In the absence of INT input the inverse WTA neuron with the strongest POIS input wins and suppresses through the GI neuron the activity of all others. Local lateral connections in the inverse WTA population strengthen the winning neuron due to excitatory feedback (for the sake of clarity recurrent excitation is not shown in Figure 4).

Due to the consistently changing nature of the POIS spike trains the winner changes frequently and the agent executes a random walk (see Figure 2a). When the agent approaches an object the position and relative distance of the obstacle is indicated by a number of spikes in the INT population. These spikes strongly inhibit the inverse WTA at the corresponding position and its closest neighbours so that this direction cannot win. Therefore, the active neurons in the inverse WTA always represent an obstacle-free direction. In case no object-free direction has been found for ~700 milliseconds since the start of an intersaccade the ET neuron emits a spike. This neuron is only weakly excited by the POIS population and connected to the GI neuron similarly to the inverse WTA population. Only when the ET has not been inhibited for a long time, hence the inverse WTA was not able to generate a spike due to strong over all inhibition, the ET neuron wins. The final layer called MOT population translates the inverse WTA population and ET neuron activity into a turn direction and duration using pulse-width modulation to control the motors. The left turn MOT population becomes activated by inverse WTA neurons on the left side and the right turn population by inverse WTA neurons on the right side. Since the turning velocity is always constant the angle of rotation is defined by the duration of the turn. This duration of the excitatory wave in the MOT population relates proportionally to the distance of the inverse WTA neuron from the center of the horizontal visual field. The duration saturates for neuron distances higher than nine neurons. Since a left turn and a right turn are mutually exclusive events, strong inhibition between the two MOT populations assures to disambiguate the MOT layer outputs. In case the ET neuron emits a spike the excitatory wave passes through most neurons of the left MOT population. Hence, the turning duration is slightly higher than for any turn induced by the inverse WTA population. The agent turns completely away from the faced scene since no collision free path was found in that direction. During the execution of a turn the gap finding network receives mainly rotational OF. This type of apparent motion does not contain any depth information and therefore no new movement direction should be chosen during or shortly after a turn. Because of that the MOT layer strongly inhibits the inverse WTA and SPTC populations as well as the ET neuron. After a turn has finished and none of the MOT populations is spiking anymore the agent moves purely translatory. The movement speed during this phase v_{ints} is defined in equation 1 where $\bar{f}_{OFI}$ is the mean firing rate of the OFI population. During this movement phase, called intersaccade, the agent integrates translational OF information in its INT population. The inverse WTA population slowly depolarizes from its strongly inhibited state and releases a spike indicating the new movement direction. This spike triggers the next saccadic turn of the robot while the id

Figure 4. Collision avoidance network. The macropixels (2x2 pixels) of the Event-Based Camera (EBC) project onto single neurons of the Spatio-Temporal Correlation (SPTC) population removing spatio-temporal uncorrelated events, i.e. noise. Two adjacent SPTC neurons are connected to one Time Difference Encoder (TDE) in the left-right sub-population and the right-left sub-population respectively. Trigger and facilitator connection are opposite in the two populations. The Integrator (INT) population reduces the two dimensional retinotopical map to a one-dimensional map by integrating the spikes of each TDE column onto a single LIF neuron. The inverse Winner-Take-All (WTA) population and Escape Turn (ET) population become excited by Poisson spike sources. The winner-take-all mechanism is driven by recurrent suppression through the Global Inhibition (GI) neuron. The two Motor (MOT) populations are activated by a spike in the inverse WTA population. The id of the spiking inverse WTA neuron defines which MOT becomes activated and for how long. When the ET neuron spikes the left MOT population becomes activated for the maximal time duration. When the MOT population is inactive the robot moves straight forward collecting apparent motion information. When one MOT population is active the robot turns. All-to-all inhibition between the MOT sub-populations guarantees to disambiguate the steering commands. Inhibition from the MOT to the SPTC population suppresses rotational OF input which contains no relative depth information. Inhibition from MOT to inverse WTA hinders the network from taking any new decision during a turn.

of the winning neuron defines the direction and duration of the movement.

$$v_{ints}(\frac{m}{s}) = 1 - \bar{f}_{OFI} \times 0.001 \tag{1}$$

4.4 Neurorobotics Platform

To perform our behavioural experiments we decided to simulate the entire system, from visual input to actions, using the Neurorobotics Platform. This platform combines simulated SNNs with physical realistic robot models in a simulated 3D environment[53]. The platform consists of three main parts, the world simulator Gazebo, the SNN simulator Nest and the Transfer Function Manager Brain Interface and Body Integrator (BIBI). The BIBI middleware consists of a set of transfer functions which enables the communication between Gazebo and NEST via Robot Operating System (ROS)[90] and PyNN adapters. The Closed Loop Engine (CLE) synchronizes the two simulators Gazebo and Nest and controls the data exchange through transfer functions. The simulation front-end virtual coach is useful to control the whole simulation procedure through a single python script. Furthermore, the State Machines Manager of the SMACH framework can be used to write State Machines which manipulate the robot or world environment during the experiment.

4.5 Real World Robot

The robot receives visual input from a Dynamic Vision Sensor with a horizontal viewing angle of 110 degrees. The event-based camera sends its events to a SpiNN-3 board which simulates a simplified version of the collision avoidance network described in the section Collision Avoidance Network. The network does not contain any OFI neuron and the agent moves with a constant velocity of around 0.1 m/s. No ET population is included. The motor control is regulated by an FPGA-based AERnode board. The board receives input from one SpiNNaker output neuron population. It translates the spiking input into a pulse-width-modulation signal to control the motors. The pulse-width of the signal depends on the id of the output neuron spiking on the SpiNNaker board. The motor controller drives the robot in a differential manner.

4.6 Event-Based Cameras in Gazebo

Kaiser et al. 2016[91] developed a Neurorobotics Platform implementation of an event-based camera based on the world simulator Gazebo. This model samples the environment with a fixed update rate and produces an event when the brightness change between old and new frame exceeds a threshold. We used this camera model in our closed-loop simulations as visual input to the collision avoidance network. Even though Gazebo produces an event-stream from regularly sampled synchronous frame-difference, our sEMD characterisation and open-loop experiments (see Section sEMD characterisation and[39]) confirmed the working principle of the motion detector model with real-world event-based camera data. We could further demonstrate the real-world fully-neuromorphic applicability in closed-loop of most parts of the simulated agent including the apparent motion computation by the sEMDs and the saccadic suppression[92]. We set the resolution of the Gazebo event-based camera model to 128 times 40 pixels. The reduction of the vertical resolution from 128 to 40 pixels was done to speed up the simulation time and to make the model fit onto a SpiNN-3 board[63]. To further accelerate the simulation we limited the number of events per update-cycle to 1000 and set the refresh rate to 200 Hz. Therefore, the sEMD can only detect time differences with a resolution of 5 ms. We decided for a large horizontal visual angle of 140 degrees so that the robot does not crash into unforeseen objects after a strong turn. At the same time the uniform distribution of 128 pixels over a 140 degrees horizontal visual field leads to an inter-pixel angle of approximately 1.1 degrees. This visual acuity lies in a biologically plausible range of inter-ommatidial angles measured in Diptera and Hymneoptera which varies between 0.4 and 5.8 degrees[93].

4.7 Driving Agent

We designed a four-wheeled simulated robot Gazebo model. The robot's dimensions are $20 \times 20 \times 10$ cm and it is equipped with an event-based camera (see Section Event-Based Cameras in Gazebo) and the husky differential motor controller plugin. The BIBI[53] connects the robot with the collision avoidance network implemented in NEST (see Section Collision Avoidance Network). The connections consist of one transfer function from the vision sensor to the SPTC population and another one from the MOT population to the differential motor controller as well as two Poisson input spike sources. The first transfer function sends visual input events. The second transfer function controls the agent's insect-inspired movement pattern. During inactivity of the MOT populations the robot drives purely translatory with a maximum speed of 2.5 a.u/s. The movement velocity changes anti-proportionally to the environment's obstacle density as explained in the section Densely Cluttered Environments. When one of the two MOT populations spikes the robot fixes its forward velocity to 0.38 a.u/s and turns either to the left or to the right with an angular velocity of $4 \, °/s$. The two Poisson spike source populations send spikes with a medium spike rate of 100 Hz to the inverse, soft WTA population and the ET neuron (For more details see Table 7 and Table 5).

4.8 sEMD characterisation

For the sEMD characterisation we stimulated an event-based camera with a 79° lens (see Section Event-Based Cameras in Gazebo) using square-wave gratings with a wavelength of 20° and various constant velocities (from 0.1 to 10 Hz). These recordings were performed in a controlled environment containing an event-based camera, an LED light ring and a moving screen which projects exchangeable stimuli (see Figure A.1). The controllable light ring illuminates the screen. The camera's lens is positioned in the light ring's centre to ensure a homogeneous illumination of the pattern. The screen itself is moved by an Arduino controlled motor. During recordings, the box can be closed and thus be isolated from interfering light sources. The contrast refers to absolute grey-scale values printed on white paper to form the screen. However, given the printed contrast we calculated the Michelson contrast as follows:

$$\frac{I_{max} - I_{min}}{I_{max} + I_{min}} = \frac{I_{max} - I_{max}(1 - C_{printed})}{I_{max} + I_{max}(1 - C_{printed})} = \frac{C_{printed}}{2 - C_{printed}} \tag{2}$$

To show the model's robustness to a wide range of environments, we varied the following three parameters in the recordings: The illumination, the grating velocity and the grating's contrast (see Table 1). Each possible parameter combination was recorded three times, with a recording duration of four seconds, to allow statistical evaluation of the results. The event-based camera was biased for slow velocities.

The model (see Figure 4 the first three populations) was simulated in Nest with the connections and neuron parameters defined in Table 7 and Table 5 respectively. The network was simulated for four seconds, receiving the events as emitted by the event-based camera as spike-source array input. To define a response to the various stimuli, from the simulation results, the mean population activity of the preferred direction and null direction population were calculated (see Figure 1e). For the closest comparability to the biologically imposed environment parameters, we chose to compare and discuss the sEMD's velocity tuning curve for a grating contrast of 100 % and an illumination of 5000 lux.

4.9 Closed-loop simulation in environments

Five different environments were designed to evaluate the agent's performance, a cluttered environment with randomly distributed obstacles sizing 1×1 m, an environment with two arenas connected by two gaps of variable size, a tunnel with varying width, an empty box environment and a narrowing tunnel. No obstacles were placed in a radius of two meters around the agent's start point so that the system can reach a stable state of activity before confronted with the first object. At obstacle densities higher than 35 percent the agent stays at its start point since no obstacle free direction can be detected anymore. Therefore, we limited the tested obstacle density range to 0 up to 38 percent. All obstacles placed in the environment including walls were covered with vertical black-and-white square-wave-gratings.

A state-machine was written within the Neurorobotics Platform environment to automatise the experiments. The state-machine consists of eight states as shown in Figure 5
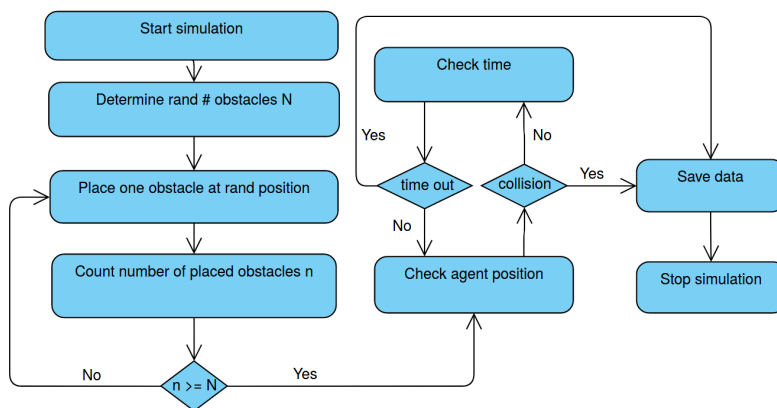
Figure 5. State machine to create the cluttered environment and check the agent's collision avoidance performance.

Additionally, a virtual coach script was composed which starts and stops the single runs in a for-loop. After creating the simulation environment the virtual coach script starts the simulation for 10 seconds so that the state machine becomes activated. After that the simulation stops for five minutes which are long enough for the state machine to place all objects in the environment. When five minutes have passed the simulation is activated again and the agent starts moving through the environment. CSV files containing the spiking data of the network, the robot position and angular alignment as well as

the placement of the objects in the arena were saved for all experiments. 100 data points were collected for the collision avoidance experiment in a cluttered environment with adaptive velocity, 70 data points were collected for the experiment with fixed velocity (see Figure 2f,i A.5,A.7). The tunnel centering experiment, gap entering experiment and all other simulation experiments in the appendix were repeated three times for each individual configuration (see Table 2).

Obstacle densities were calculated by plotting the cluttered environment and counting the number of pixels occupied by the objects. The occurrence of collisions was also measured visually by plotting the cluttered environment with the robot's trajectory while considering the agent size and angular alignment. Since the *can_collide* feature of the objects in the cluttered environment was turned off the agent moves through the obstacles when colliding. Therefore, an overlap of obstacle and robot can be interpreted as a collision. The collision avoidance run was marked as failed when such an overlap occurred and the first time of overlap was noted as collision time. Since there is no physical collision the robot's size can be varied during the analysis to evaluate the effect of agent size on the performance. To enhance the comparability of the robotic system to the biological role model, flying insects, we normalised all distance measures by dividing them by the chosen robot's size of 40x40 centimeters. The normalised distance measures were complemented with an arbitrary unit (a.u.).

4.10 Data Availability Statement

The data generated during this study will be available at dataverse.nl/dataset.xhtml?persistentId=doi:10.34894/QTOJJP.

4.11 Code Availability Statement

The code generated during this study will be made available at dataverse.nl/dataset.xhtml?persistentId=doi:10.34894/QTOJJP.

5 Acknowledgements

6 Author contribution

T.S., E.C. and E.J. conceived and designed the experiments. M.B.M., T.S. and E.J. designed and optimised the tested algorithm. T.S. and E.J. carried out and analyzed the experiments. M.B.M. and E.C. developed the original sEMD model. O.J.N.B., M.B.M, T.S., E.J., E.C. and M.E. wrote the manuscript.

7 Competing Interests

The authors declare no competing interests.

References

1. Kelasidi, E. *et al.* Path Following, Obstacle Detection and Obstacle Avoidance for Thrusted Underwater Snake Robots. *Front. Robotics AI* **6**, 57, DOI: 10.3389/frobt.2019.00057 (2019).

2. Floreano, D., Ijspeert, A. & Schaal, S. Robotics and Neuroscience. *Curr. Biol.* **24**, R910–R920, DOI: 10.1016/j.cub.2014. 07.058 (2014).

3. Barca, J. C. & Sekercioglu, Y. A. Swarm robotics reviewed. *Robotica* **31**, 345–359, DOI: 10.1017/S026357471200032X (2013).

4. Obstacle avoidance in space robotics: Review of major challenges and proposed solutions, DOI: 10.1016/j.paerosci.2018. 07.001 (2018).

5. Pandey, A. Mobile Robot Navigation and Obstacle Avoidance Techniques: A Review. *Int. Robotics & Autom. J.* **2**, DOI: 10.15406/iratj.2017.02.00023 (2017).

6. Serres, J. R. & Viollet, S. Insect-inspired vision for autonomous vehicles, DOI: 10.1016/j.cois.2018.09.005 (2018).

7. Dickinson, M. H. Death Valley, <i>Drosophila</i> , and the Devonian Toolkit. *Annu. Rev. Entomol.* **59**, 51–72, DOI: 10.1146/annurev-ento-011613-162041 (2014).

8. Baird, E. & Dacke, M. Finding the gap: a brightness-based strategy for guidance in cluttered environments. *Proceedings. Biol. sciences / The Royal Soc.* **283**, 1794–1799, DOI: 10.1098/rspb.2015.2988 (2016).

9. Mountcastle, A. M., Alexander, T. M., Switzer, C. M. & Combes, S. A. Wing wear reduces bumblebee flight performance in a dynamic obstacle course. *Biol. letters* **12**, 20160294, DOI: 10.1098/rsbl.2016.0294 (2016).

10. Witthöft, W. Absolute anzahl und verteilung der zellen im him der honigbiene. *Zeitschrift für Morphol. der Tiere* **61**, 160–184, DOI: 10.1007/BF00298776 (1967).

11. Zheng, Z. *et al.* A Complete Electron Microscopy Volume of the Brain of Adult Drosophila melanogaster. *Cell* **174**, 730–743, DOI: 10.1016/j.cell.2018.06.019 (2018).

12. Borst, A., Haag, J. & Mauss, A. S. How fly neurons compute the direction of visual motion, DOI: 10.1007/s00359-019-01375-9 (2019).

13. Fu, Q., Wang, H., Hu, C. & Yue, S. Towards computational models and applications of insect visual systems for motion perception: A review. *Artif. Life* **25**, 263–311, DOI: 10.1162/artl_a_00297 (2019). PMID: 31397604, https://doi.org/10.1162/artl_a_00297.

14. Egelhaaf, M. & Lindemann, J. P. Texture dependence of motion sensing and free flight behavior in blowflies. (2012).

15. Bertrand, O. J. N., Lindemann, J. P. & Egelhaaf, M. A bio-inspired collision avoidance model based on spatial information derived from motion detectors leads to common routes. *PLOS Comput. Biol.* **11**, e1004339, DOI: 10.1371/journal.pcbi.1004339 (2015).

16. A Model for an Angular Velocity-Tuned Motion Detector Accounting for Deviations in the Corridor-Centering Response of the Bee. *PLOS Comput. Biol.* **12**, e1004887, DOI: 10.1371/journal.pcbi.1004887 (2016).

17. Lecoeur, J., Dacke, M., Floreano, D. & Baird, E. The role of optic flow pooling in insect flight control in cluttered environments. *Sci. Reports* **9** (2019).

18. Mauss, A. S. & Borst, A. Optic flow-based course control in insects. *Curr. opinion neurobiology* **60**, 21—27, DOI: 10.1016/j.conb.2019.10.007 (2020).

19. Baird, E. & Dacke, M. Finding the gap: a brightness-based strategy for guidance in cluttered environments. *Proc. Royal Soc. B: Biol. Sci.* **283**, 20152988, DOI: 10.1098/rspb.2015.2988 (2016). https://royalsocietypublishing.org/doi/pdf/10.1098/rspb.2015.2988.

20. Ravi, S. *et al.* Gap perception in bumblebees. *J. Exp. Biol.* **222**, jeb184135, DOI: 10.1242/JEB.184135 (2019).

21. Ravi, S. *et al.* Bumblebees perceive the spatial layout of their environment in relation to their body size and form to minimize inflight collisions. *Proc. Natl. Acad. Sci.* 202016872, DOI: 10.1073/pnas.2016872117 (2020).

22. Li, J., Lindemann, J. P. & Egelhaaf, M. Local motion adaptation enhances the representation of spatial structure at emd arrays. *PLOS Comput. Biol.* **13**, e1005919, DOI: 10.1371/journal.pcbi.1005919 (2017).

23. Zingg, S., Scaramuzza, D., Weiss, S. & Siegwart, R. Mav navigation through indoor corridors using optical flow. In *2010 IEEE International Conference on Robotics and Automation*, 3361–3368 (IEEE, 2010).

24. Blösch, M., Weiss, S., Scaramuzza, D. & Siegwart, R. Vision based mav navigation in unknown and unstructured environments. In *2010 IEEE International Conference on Robotics and Automation*, 21–28 (IEEE, 2010).

25. Posch, C., Matolin, D. & Wohlgenannt, R. A qvga 143db dynamic range asynchronous address-event pwm dynamic image sensor with lossless pixel-level video compression. In *Digest of Technical Papers - IEEE International Solid-State Circuits Conference*, vol. 53, 400–401, DOI: 10.1109/ISSCC.2010.5433973 (2010).

26. Lichtsteiner, P., Posch, C. & Delbruck, T. A 128 × 128 120 db 15 μs latency asynchronous temporal contrast vision sensor. *IEEE J. Solid-State Circuits* **43**, 566–576, DOI: 10.1109/JSSC.2007.914337 (2008).

27. Brandli, C., Berner, R., Yang, M., Liu, S. C. & Delbruck, T. A 240 × 180 130 db 3 μs latency global shutter spatiotemporal vision sensor. *IEEE J. Solid-State Circuits* **49**, 2333–2341, DOI: 10.1109/JSSC.2014.2342715 (2014).

28. Posch, C., Serrano-Gotarredona, T., Linares-Barranco, B. & Delbruck, T. Retinomorphic event-based vision sensors: Bioinspired cameras with spiking output. *Proc. IEEE* **102**, 1470–1484, DOI: 10.1109/JPROC.2014.2346153 (2014).

29. Son, B. *et al.* A 640×480 dynamic vision sensor with a 9μm pixel and 300meps address-event representation. In *Digest of Technical Papers - IEEE International Solid-State Circuits Conference*, vol. 60, 66–67, DOI: 10.1109/ISSCC.2017.7870263 (Institute of Electrical and Electronics Engineers Inc., 2017).

30. Astrom, K. J. & Bernhardsson, B. M. Comparison of riemann and lebesgue sampling for first order stochastic systems. In *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, vol. 2, 2011–2016 (IEEE, 2002).

31. Benosman, R., Clercq, C., Lagorce, X., Ieng, S.-H. & Bartolozzi, C. Event-based visual flow. *IEEE transactions on neural networks learning systems* **25**, 407–417 (2013).

32. Conradt, J. On-board real-time optic-flow for miniature event-based vision sensors. In *2015 IEEE International Conference on Robotics and Biomimetics, IEEE-ROBIO 2015*, 1858–1863, DOI: 10.1109/ROBIO.2015.7419043 (Institute of Electrical and Electronics Engineers Inc., 2015).

33. Milde, M. B., Bertrand, O. J., Benosmanz, R., Egelhaaf, M. & Chicca, E. Bioinspired event-driven collision avoidance algorithm based on optic flow. In *2015 International Conference on Event-based Control, Communication, and Signal Processing (EBCCSP)*, 1–7, DOI: 10.1109/EBCCSP.2015.7300673 (IEEE, 2015).

34. Liu, M. & Delbruck, T. Block-matching optical flow for dynamic vision sensors: Algorithm and fpga implementation. In *Proceedings - IEEE International Symposium on Circuits and Systems*, DOI: 10.1109/ISCAS.2017.8050295 (Institute of Electrical and Electronics Engineers Inc., 2017).

35. Rueckauer, B. & Delbruck, T. Evaluation of event-based algorithms for optical flow with ground-truth from inertial measurement sensor. *Front. Neurosci.* **10**, 176, DOI: 10.3389/fnins.2016.00176 (2016).

36. Gallego, G., Rebecq, H. & Scaramuzza, D. A unifying contrast maximization framework for event cameras, with applications to motion, depth, and optical flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2018).

37. Haessig, G., Cassidy, A., Alvarez, R., Benosman, R. & Orchard, G. Spiking optical flow for event-based sensors using ibm's truenorth neurosynaptic system. *IEEE transactions on biomedical circuits systems* **12**, 860–870 (2018).

38. Martel, J. N., Chau, M., Dudek, P. & Cook, M. Toward joint approximate inference of visual quantities on cellular processor arrays. In *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2061–2064 (IEEE, 2015).

39. Milde, M. B., Bertrand, O. J. N., Ramachandran, H., Egelhaaf, M. & Chicca, E. Spiking elementary motion detector in neuromorphic systems. *Neural Comput.* **30**, 2384–2417, DOI: 10.1162/neco_a_01112 (2018). PMID: 30021082.

40. Serres, J. R. & Ruffier, F. Optic flow-based collision-free strategies: From insects to robots. *Arthropod structure & development* **46**, 703–717 (2017).

41. Fu, Q., Wang, H., Hu, C. & Yue, S. Towards computational models and applications of insect visual systems for motion perception: A review. *Artif. life* **25**, 263–311 (2019).

42. Müller, G. R. & Conradt, J. A miniature low-power sensor system for real time 2d visual tracking of led markers. In *2011 IEEE International Conference on Robotics and Biomimetics*, 2429–2434, DOI: 10.1109/ROBIO.2011.6181669 (2011).

43. Milde, M. B., Dietmuller, A., Blum, H., Indiveri, G. & Sandamirskaya, Y. Obstacle avoidance and target acquisition in mobile robots equipped with neuromorphic sensory-processing systems. In *Proceedings - IEEE International Symposium on Circuits and Systems*, DOI: 10.1109/ISCAS.2017.8050984 (Institute of Electrical and Electronics Engineers Inc., 2017).

44. Kreiser, R., Renner, A., Sandamirskaya, Y. & Pienroj, P. Pose estimation and map formation with spiking neural networks: towards neuromorphic slam. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2159–2166 (IEEE, 2018).

45. Indiveri, G. & Sandamirskaya, Y. The importance of space and time for signal processing in neuromorphic agents: the challenge of developing low-power, autonomous agents that interact with the environment. *IEEE Signal Process. Mag.* **36**, 16–28 (2019).

46. Schilstra, C. & van Hateren, J. H. Blowfly flight: kinematics of the thorax. **1490**, 1481–1490 (1999).

47. Serres, J. R., Masson, G. P., Ruffier, F. & Franceschini, N. A bee in the corridor: centering and wall-following. *Naturwissenschaften* **95**, 1181–1187, DOI: 10.1007/s00114-008-0440-6 (2008).

48. Baird, E., Srinivasan, M. V., Zhang, S. & Cowling, A. Visual control of flight speed in honeybees. *J. Exp. Biol.* **208**, 3895–3905 (2005).

49. Kern, R., Boeddeker, N., Dittmar, L. & Egelhaaf, M. Blowfly flight characteristics are shaped by environmental features and controlled by optic flow information. *J. Exp. Biol.* **215**, 2501–2514, DOI: 10.1242/jeb.061713 (2012). https://jeb.biologists.org/content/215/14/2501.full.pdf.

50. Linander, N., Baird, E. & Dacke, M. Bumblebee flight performance in environments of different proximity. *J. Comp. Physiol. A: Neuroethol. Sensory, Neural, Behav. Physiol.* **202**, 97–103, DOI: 10.1007/s00359-015-1055-y (2016).

51. Arenz, A., Drews, M. S., Richter, F. G., Ammer, G. & Borst, A. The temporal tuning of the drosophila motion detectors is determined by the dynamics of their input elements. *Curr. Biol.* **27**, 929–944 (2017).

52. Drews, M. S. *et al.* Dynamic signal compression for robust motion vision in flies. *Curr. Biol.* **30**, 209–221 (2020).

53. Falotico, E. *et al.* Connecting artificial brains to robots in a comprehensive simulation framework: The neurorobotics platform. *Front. Neurorobotics* **11**, 2, DOI: 10.3389/fnbot.2017.00002 (2017).

54. Hassentstein, B. & Reichardt, W. Systemtheoretische analyse der zeit-, reihenfolgen- und vorzeichenauswertung bei der bewegungsperzeption des rüsselkäfers chlorophanus. *Z. Naturforsch.* **11b**, 513–524, DOI: 10.1515/znb-1956-9-1004 (1956).

55. Diesmann, M. & Gewaltig, M.-O. Nest: An environment for neural systems simulations (2003).

56. Haag J, S. E. G. F. B. A., Arenz A. Complementary mechanisms create direction selectivity in the fly. *Elife* (2016).

57. Ong M, G. J. S. M., Bulmer M. Obstacle traversal and route choice in flying honeybees: Evidence for individual handedness. *PLoS One* **12**, DOI: 10.1371/journal.pone.0184343 (2017).

58. Schoepe, T. *et al.* Neuromorphic sensory integration for combining sound source localization and collision avoidance. 1–4 (2019).

59. Indiveri, G., Chicca, E. & Douglas, R. A vlsi array of low-power spiking neurons and bistable synapses with spike-timing dependent plasticity. *IEEE transactions on neural networks* **17**, 211–221 (2006).

60. Bartolozzi, C. & Indiveri, G. Synaptic dynamics in analog vlsi. *Neural computation* **19**, 2581–2603 (2007).

61. Horiuchi, T. K. A spike-latency model for sonar-based navigation in obstacle fields. *IEEE Transactions on Circuits Syst. I: Regul. Pap.* **56**, 2393–2401, DOI: 10.1109/TCSI.2009.2015597 (2009).

62. Moradi, S., Qiao, N., Stefanini, F. & Indiveri, G. A scalable multicore architecture with heterogeneous memory structures for dynamic neuromorphic asynchronous processors (dynaps). *IEEE transactions on biomedical circuits systems* **12**, 106–122 (2017).

63. Painkras, E. *et al.* Spinnaker: A 1-w 18-core system-on-chip for massively-parallel neural network simulation. *IEEE J. Solid-State Circuits* **48**, 1943–1953 (2013).

64. Wang, R. & van Schaik, A. Breaking liebig's law: an advanced multipurpose neuromorphic engine. *Front. neuroscience* **12**, 593 (2018).

65. Schnell, B., Ros, I. G. & Dickinson, M. A descending neuron correlated with the rapid steering maneuvers of flying drosophila. *Curr. Biol.* **27**, 1200–1205 (2017).

66. Sun, X., Yue, S. & Mangan, M. A decentralised neural model explaining optimal integration of navigational strategies in insects. *bioRxiv* 856153 (2020).

67. Kreiser, R., Cartiglia, M., Martel, J. N., Conradt, J. & Sandamirskaya, Y. A neuromorphic approach to path integration: a head-direction spiking neural network with vision-driven reset. In *2018 IEEE international symposium on circuits and systems (ISCAS)*, 1–5 (IEEE, 2018).

68. Blum, H. *et al.* A neuromorphic controller for a robotic vehicle equipped with a dynamic vision sensor. *Robotics Sci. Syst. RSS 2017* (2017).

69. Honkanen, A., Adden, A., da Silva Freitas, J. & Heinze, S. The insect central complex and the neural basis of navigational strategies. *J. Exp. Biol.* **222**, DOI: 10.1242/jeb.188854 (2019). https://jeb.biologists.org/content/222/Suppl_1/jeb188854.full.pdf.

70. Sun, X., Yue, S. & Mangan, M. A decentralised neural model explaining optimal integration of navigational strategies in insects. *bioRxiv* DOI: 10.1101/856153 (2020). https://www.biorxiv.org/content/early/2020/05/14/856153.full.pdf.

71. Kim, A. J., Fitzgerald, J. K. & Maimon, G. Cellular evidence for efference copy in drosophila visuomotor processing. *Nat. Neurosci.* **18**, 1247–1255 (2015).

72. Alex, L. The Von Neumann architecture topic paper #3. *Comput. Sience* **319**, 360–8771 (2009).

73. Thakur, C. S. *et al.* Large-scale neuromorphic spiking array processors: A quest to mimic the brain. *Front. Neurosci.* **12**, 891, DOI: 10.3389/fnins.2018.00891 (2018).

74. Mead, C. *Analog VLSI and Neural Systems* (Addison Wesley Publishing Company, 1989).

75. Shih-Chii Liu, G. I. A. W. R. D., Tobi Delbruck. *Event-Based Neuromorphic Systems* (John Wiley Sons, 2015).

76. Payvand, M., Nair, M. V., Müller, L. K. & Indiveri, G. A neuromorphic systems approach to in-memory computing with non-ideal memristive devices: From mitigation to exploitation. *Faraday Discuss.* **213**, 487–510 (2019).

77. Serb, A. *et al.* Memristive synapses connect brain and silicon spiking neurons. *Sci. reports* **10**, 1–7 (2020).

78. Gerstner, W. & Kistler, W. M. *Spiking Neuron Models: Single Neurons, Populations, Plasticity* (Cambridge University Press, 2002).

79. Khalil, A. A., Valle, M., Chible, H. & Bartolozzi, C. Cmos dynamic tactile sensor. In *2017 New Generation of CAS (NGCAS)*, 269–272, DOI: 10.1109/NGCAS.2017.48 (2017).

80. Drix, D. & Schmuker, M. Resolving fast transients with metal-oxide gas sensors (2020). 2010.01903.

81. Liu, S.-C., van Schaik, A., Minch, B. A. & Delbruck, T. Asynchronous binaural spatial audition sensor with $2 \times 64 \times 4$ channel output. *IEEE Transactions on Biomed. Circuits Syst.* **8**, 453–464 (2014).

82. Chan, V., Liu, S. & van Schaik, A. Aer ear: A matched silicon cochlea pair with address event representation interface. *IEEE Transactions on Circuits Syst. I: Regul. Pap.* **54**, 48–59, DOI: 10.1109/TCSI.2006.887979 (2007).

83. Mahowald, M. Vlsi analogs of neural visual processing: a synthesis of form and function. *Ph.D. dissertation* (1992).

84. Lichtsteiner, P., Posch, C. & Delbruck, T. A 128 x 128 120db 30mw asynchronous vision sensor that responds to relative intensity change. 2060–2069 (2006).

85. Olshausen, B. A. & Field, D. J. How close are we to understanding v1? *Neural Comput.* **17**, 1665–1699, DOI: 10.1162/0899766054026639 (2005). https://doi.org/10.1162/0899766054026639.

86. S., T. & J., G. Rank order coding. *Bower: j.M. (eds) Comput. Neurosci. Springer* DOI: 10.1007/978-1-4615-4831-7_19 (1998).

87. Thorpe, S. Spike arrival times: A highly efficient coding scheme for neural networks (1990).

88. Masquelier, T. Relative spike time coding and stdp-based orientation selectivity in the early visual system in natural continuous and saccadic vision: a computational model. *J. computational neuroscience* **32**, 425–441 (2012).

89. D'Angelo, G. *et al.* Event-based eccentric motion detection exploiting time difference encoding. *Front. Neurosci.* **14** (2020).

90. Quigley, M. Ros: an open-source robot operating system. In *ICRA 2009* (2009).

91. Kaiser, J. *et al.* 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)*, 127–134, DOI: 10.1109/SIMPAR.2016.7862386 (2016).

92. Schoepe, T. *et al.* Live demonstration: Neuromorphic sensory integration for combining sound source localization and collision avoidance. In *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, 1–1, DOI: 10.1109/ISCAS45731.2020.9181257 (2020).

93. Land, M. F. Visual acuity in insects. *Annu. Rev. Entomol.* **42**, 147–177, DOI: 10.1146/annurev.ento.42.1.147 (1997). PMID: 15012311, https://doi.org/10.1146/annurev.ento.42.1.147.

A Appendix

A.1 sEMD Characterization Setup

To ensure repeatability and reproducibility we recorded the grating in a controlled environment, see Figure A.1. The Dynamic Vision Sensor (DVS) is mounted in a light sealed box, with a variable distance to the screen. An LED-ring (with 32 LEDs) homogeneously illuminates the DVS's field of view. The LEDs them self are controlled by an external power-source. The moving screen consists of a thick paper tube, glued together at the ends with double-sided adhesive tape. This tube is clamped over two horizontally mounted cylinders. The lower cylinder is mounted with a floating bearing in the y-direction. The upper cylinder is driven by a stepper motor controlled by an Arduino Uno and translates its movement to the screen. The possible velocities of the screen range from $23\,\mathrm{mm\,s^{-1}}$ to $210\,\mathrm{mm\,s^{-1}}$. The grating itself is printed on dull thick paper forming the paper-tube and stored in the dark to avoid fading out.
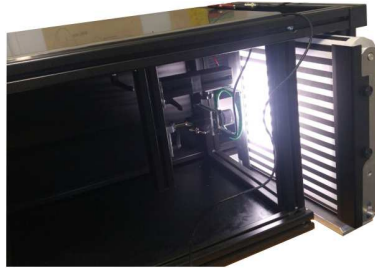
Figure A.1. Controlled environment for the recordings of the grating. The screen can move either from bottom to top or top to bottom. The upper roll of the screen contraption is driven by an Arduino controlled stepper-motor. The LED - ring illuminates the screen and the event driven camera is located in its center.

A.2 sEMD Implementation on SpiNNaker

To demonstrate the sEMD's wide range of operation and applicability on multiple platforms, we characterised the model's behaviour on SpiNNaker. We further investigated the sEMD's robustness regarding contrast and illumination. Figure A.2 a) shows that the model operates well in a wide range of illuminations at 100 % contrast and produces similar velocity tuning curves on SpiNNaker and NEST (see Figure 1e for comparison). Regarding the contrast sensitivity, we found that with the given parameter set, the model reaches half activity at a relative contrast of 45.9 % (see Figure A.2 b) at 5000 lux illumination and temporal frequency of 5 Hz. Thus the applicability of the model is limited by the occurring contrast but the offered range is still high and can possibly be improved by the implementation of contrast normalisation.
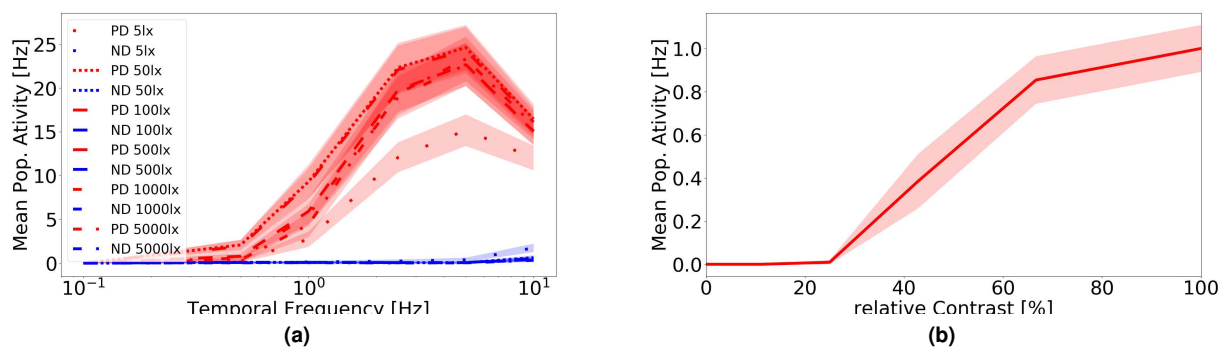
Figure A.2. sEMD population response on SpiNNaker for varying illuminations and contrasts. a) Normalised sEMD population preferred direction and null direction response for 100 % contrast and all illuminations from 5 lux to 5000 lux. b) Normalised preferred direction response for 5000 lux illumination over contrasts varying from 0 % to 100 % at a temporal frequency of 5 Hz. For further information on the model parameters see Table 4.

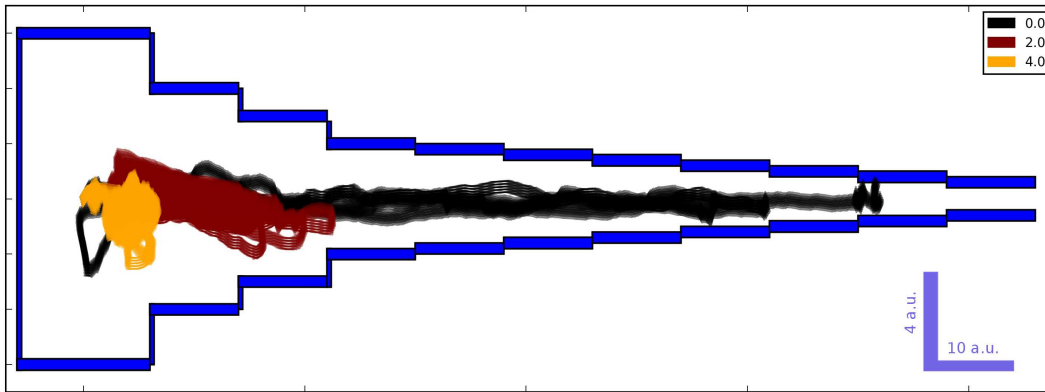### A.3 The Motion-Vision Network

One very important parameter for collision avoidance is the knowledge of the own body size. Orchid bees with a wingspan of approximately 20 mm avoid to pass circular apertures smaller than 25 mm because of a too high collision risk. Some kind

of self-representation in the bee's brain has to drive the insect's decision that the gap is too small for it.[19]. Similarly, we can tune the connectivity of our SNN to indirectly include relevant body size information. Our neural network model needs to consider its own body measures when moving through a gap. This decision process to move or not to move through a gap can be purely driven by the agent's relative perception of the gap. In our collision avoidance network this perception is modifiable by a change of the synaptic connections between the integrator neuron population and the inverse WTA population. OF is encoded in a retinotopical map of the integrator neuron population. This neuron population is initially one-to-one connected to the inverse WTA network. By connecting the integrator neuron to its accordant inverse WTA neuron and its closest neighbours the size of the perceived OF caused by an object increases. Therefore, small gaps between objects are closed with increasing number of neighbouring INT to inverse WTA connections which leads to an increase of a perceived gap's minimum size. The angle occupied by a gap has to be bigger than gap_{min} to be considered a movement direction as shown in Equation 3. α_{INT}, the angle of perception of a single INT neuron, amounts to ~2.2° while $n_{connect}$ represents the number of neighbouring connections.

$$gap_{min} = (2 \times n_{connect} + 1) \times \alpha_{sEMD} \qquad (3)$$

We evaluated how the minimum gap size entered by the robotic agent changes with the OF perception. As expected, with increasing number of neighbouring connections small gaps were not entered anymore (see Figure A.3). By fixing the number of neighbouring connections to 4 nearest neighbours for all following experiments the robot wouldn't enter too small gaps but would still be able to navigate through larger corridors.

Figure A.3. Agent's trajectories in a narrowing corridor with varied connectivity between INT and inverse WTA population as explained in section The Motion-Vision Network. Legend refers to number of neighbouring connections. Simulated robot's start point is on left side.

A.4 The Movement Behaviour

When exposed to a densely cluttered environment, a narrow tunnel or a nearby object flying insects decrease their movement velocity. This mechanisms reduces the agent's collision probability by an increase in the time-of-flight. The agent has more time to react and turn away from the potential threat due to its lower speed. We tested the effect of a change in velocity with the agent in an empty arena. As expected, the simulated robot's minimum wall distance was increasing with lower velocities (see Figure A.4a,b). Therefore, an adaptive obstacle density dependent velocity can be a helpful tool to increase the agents working range towards higher obstacle densities.

A.5 Gap finding behaviour in cluttered environments

Quantifying the relative motion perception and collision avoidance behaviour in controlled environments (see Figure A.1 and A.4) allows us to assess the fundamental capabilities of our agent. However, these tests do not fully capture conditions an agent will encounter in the real-world. These conditions include urban areas, indoors as well as outdoor forest environments. A simple, yet effective test environment thus should be characterised with variable amount of clutter, i.e. obstacle density, of vertical obstacles placed in a random configuration. We introduced the agent in an arena and varied the obstacle density from 0% up to 38% and measured the mean clearance (see Figure A.5 a) and the maximum distance (see Figure A.5 b) as a function of increasing obstacle density. The mean clearance quickly drops from 25 $a.u.$ in roughly exponential fashion to a minimum of 5 $a.u.$. If the obstacle density is greater than $\approx$ 15 %, the mean clearance stays constant. However, the collision rate starts to increase (see Figure A.6). Interestingly, due to the employed adaptive movement strategy the agent's velocity decreases almost
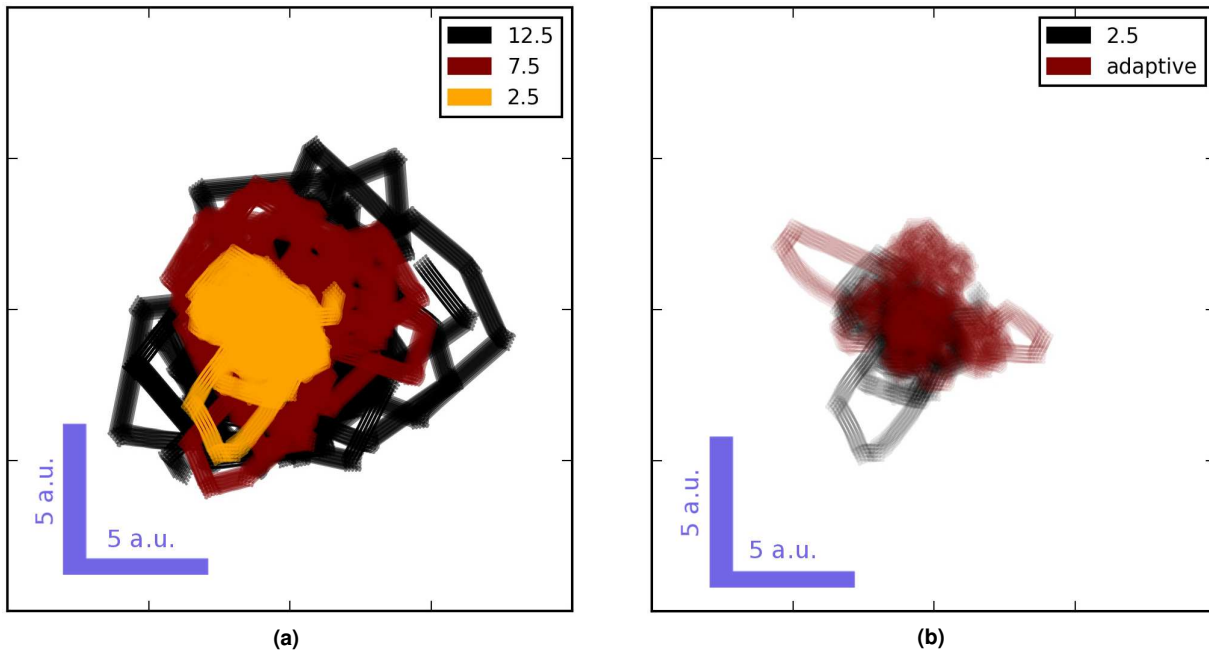
Figure A.4. Agent's trajectories in an empty box. a) Agent's trajectory with different fixed intersaccadic velocities in a.u./s. b) Trajectories with parameters used for cluttered environment experiment in section Densely Cluttered Environments with fixed and adaptive intersaccadic velocity in a.u./s.

linearly with increasing obstacle density (see Figure A.7). This adaptive behaviour ensures that despite high clutter the agent successfully identifies gaps in the environment and steers towards them and consequently avoids collisions with its surrounding.
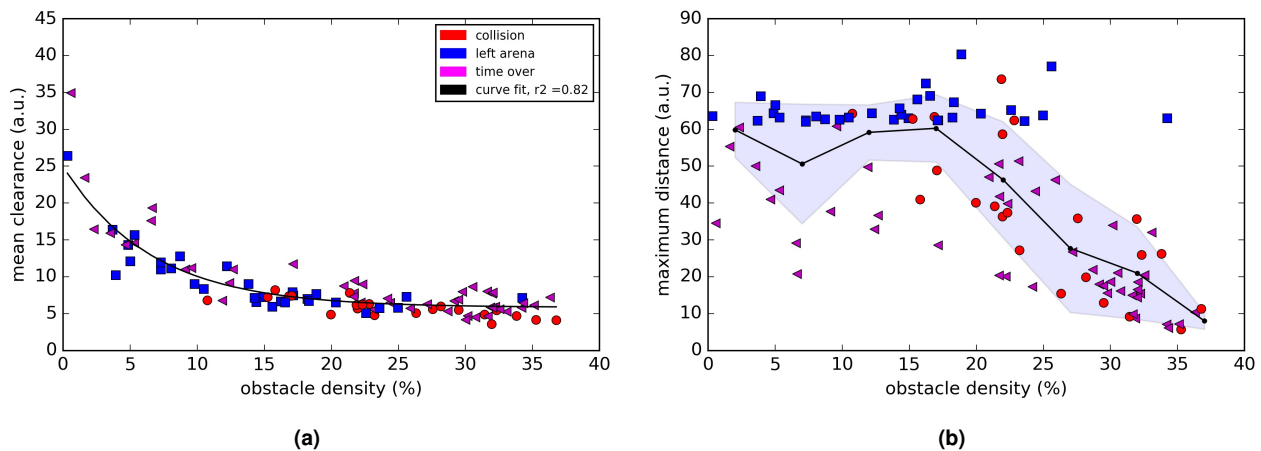
Figure A.5. Agent's mean obstacle clearance and maximum distance to the start location calculated for the data from Figure 2d.
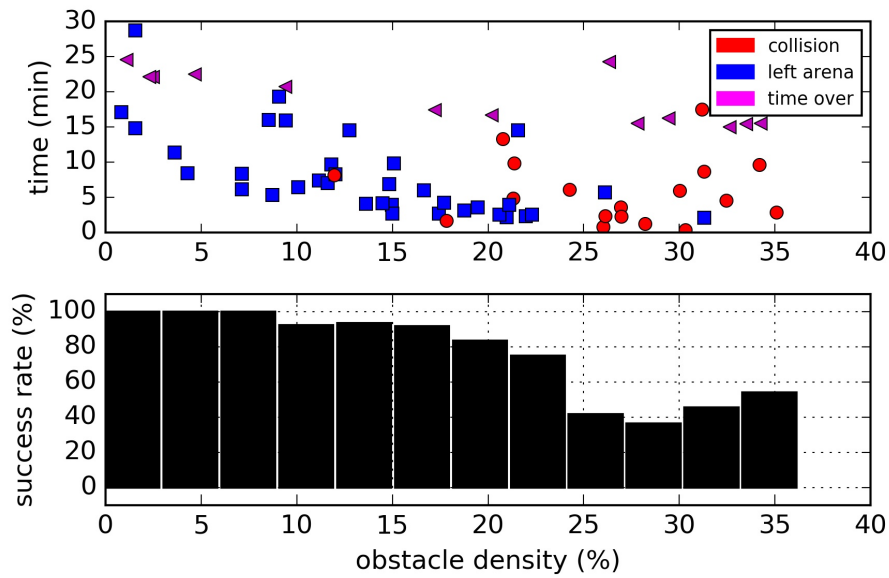
Figure A.6. Agent's behaviour in cluttered environments with the parameters from Table 5 and 7 moving with a fixed intersaccadic velocity of 2.5 a.u./s. Top: Real world time at which the simulated robot leaves the arena, collides or the simulation time is over. Bottom: Agent's success rate, hence number of runs without collisions.
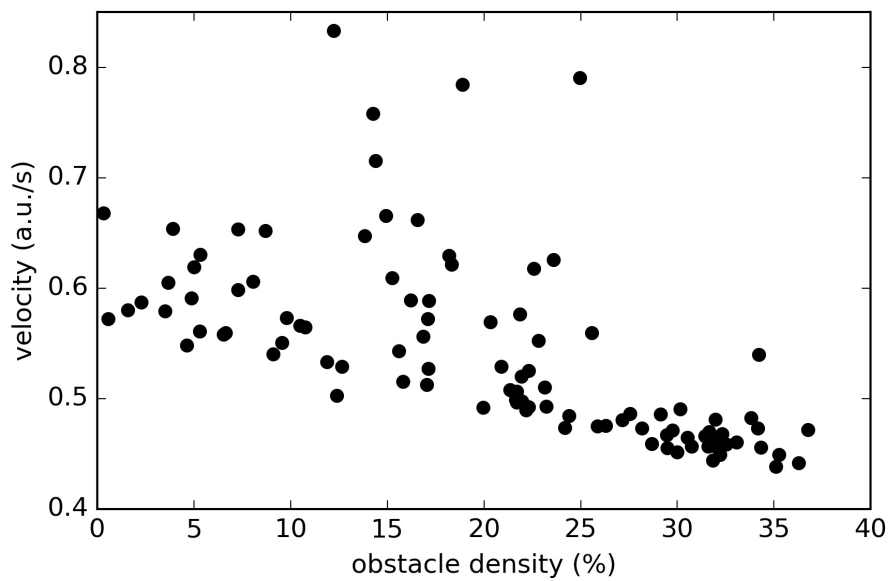
Figure A.7. Agent's mean velocity over obstacle density calculated for the data from Figure 2d

A.6 Corridor-centering in Real-World

To prove the real-time capability and robustness of the SNN on neuromorphic hardware, we evaluated the system in a real-world scenario. A robotic platform described in section Real World Robot was assembled and tested in a narrow (~30 cm wide) and a wide corridor (~50 cm wide, see Figure A.8). The robot itself is approximately 20 cm wide. The robot centred well in nine out of ten runs in the wide corridors(see Figure A.9b). In the remaining run, the robot crashed into the wall at the very beginning. In another run, the robot did a 360 degrees turn close to the end of the corridor. In the narrow corridor, the robot never crashed directly into the wall (see Figure A.9a). In two out of these runs the agent slightly touched the left wall. We observed an overall tendency to the left of the corridor. A slight miss-alignment between the robot's field of view and its movement direction can explain this tendency. In the control experiment in which the collision avoidance population of the SNN did not receive any visual input, the robot turned directly to the left or right. It crashed into a wall in nine out of ten cases (see Figure A.9c). In one case, the robot meandered through half of the tunnel before it collided with a wall. This control experiment showed that the visual input itself drove the robot's centering behaviour.
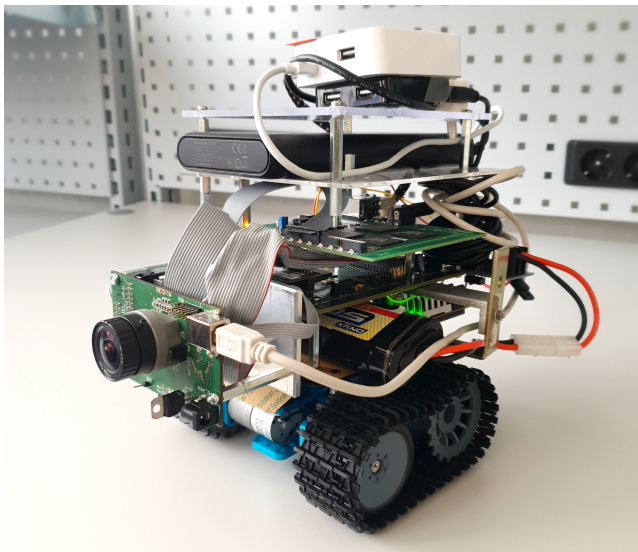
(a)   (b)

Figure A.8. Robot and setup to conduct the real world experiment. a) The robot received visual input from the Dynamic Vision Sensor. The event-based camera sends its events to a SpiNN-3 board which simulates a version of the collision avoidance network described in section Collision Avoidance Network. For more details on the real-world robotic implementation see section Real World Robot. b) Experimental setup for the corridor centering experiment. Results are shown in Figure A.9. The corridor walls were covered with random checkerboard patterns. The ground of the arena consisted of metal plates since the motors of the robot are not strong enough to move the vehicle on the carpet. A webcam and a light-ring mounted on a tripod above the arena were used to film the robot and illuminate the arena.
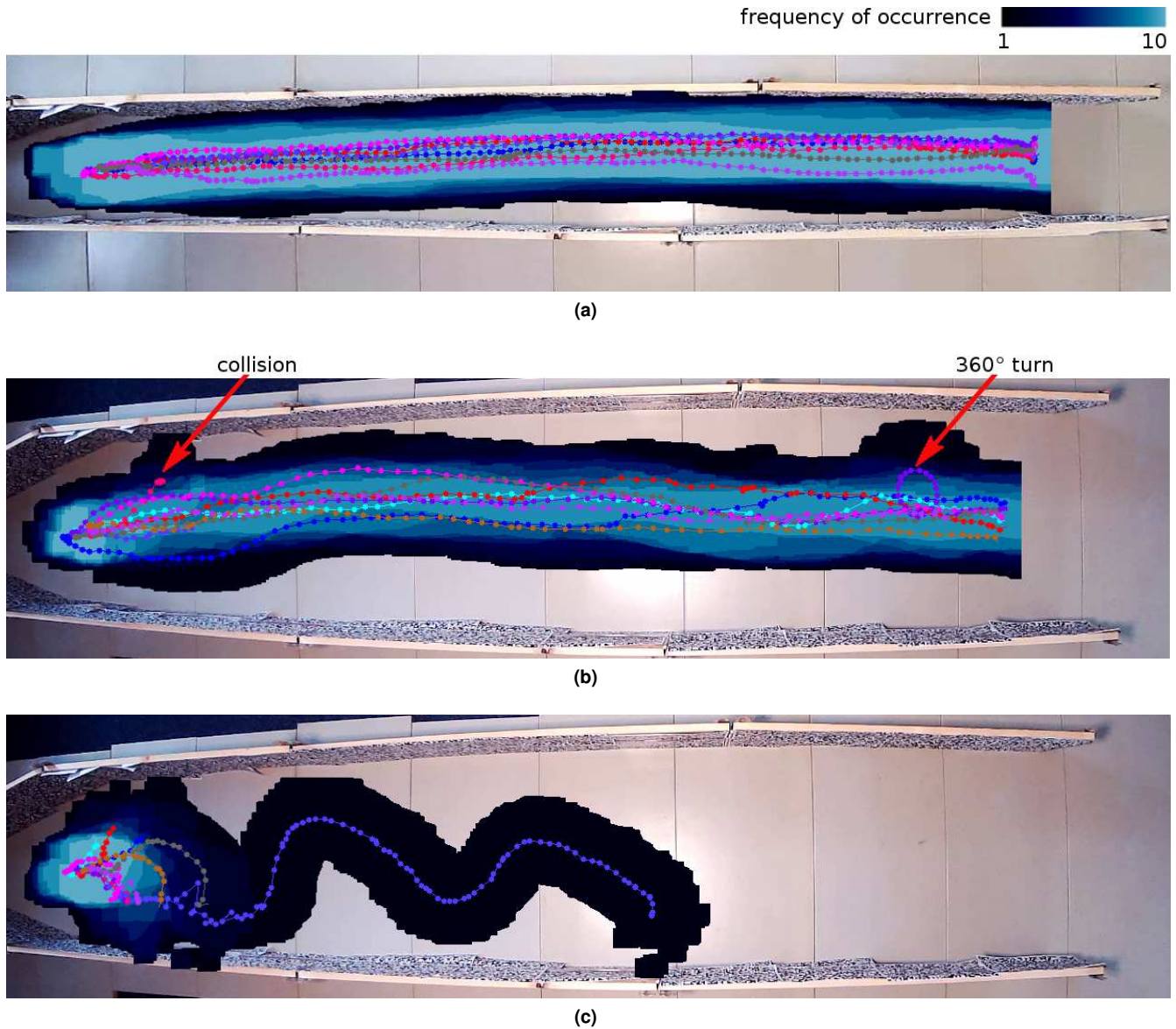
Figure A.9. Real world corridor centering experiment results from the setup shown in figure A.8. a) Robot's movement trajectories for ten runs through a narrow corridor moving from left to right. Dots and lines indicate the center of mass of the robot. Blue area represents the whole area covered by the approximately 20 cm wide robot combined for all ten runs. Frequency of occurrence increases from dark blue to light blue. b) Robot's trajectories for ten runs in a wide corridor. c) Control experiment. Robot's trajectories in a wide corridor with weights from integrator population to inverse WTA set to zero.

B Tables

printed Contrast	Temporal Frequency (Hz)	Illumination (lux)
0	0.1	5
0.2	0.5	50
0.4	1.0	100
0.6	2.5	500
0.8	5.0	1000
1.0	10.0	5000

Table 1. Parameters of grating recordings. Three four second recordings were made for each possible parameter-combination.

Simulation	Figures	Repetitions	Real time duration (min)
Clutter adaptive velocity	2f,i, A.5, A.7a	100	360
Clutter fixed velocity	2i, A.6	70	360
Corridors	2g,j,k,l	3 per corridor width	60
Real World Corridor	2d, A.9.	10 per corridor width	-
Gaps	2e,h	3 per gap size	180
Narrowing Corridor	A.3	3 per configuration	90
Empty Box	A.4, A.7b	3 per configuration	30

Table 2. Parameters of simulations and real world experiment.

Name	Type	C_m (nF)	tau_m (ms)	tau_{ref} (ms)	v_{reset} (mV)	v_{rest} (mV)	v_{thresh} (mV)	tau_{syn_E} (ms)	tau_{syn_I} (ms)	I_{offset} (nA)	Popsize ($col \times row$)	#Pop
DVS	SSA										128×128	1
SPTC	LIF	0.25	20	1	-85	-60	-50	20	20	0	32×32	1
sEMD	TDE	0.25	20	1	-85	-60	-50	20	20	0	32×32	2

From	To	Weight (nA)	Connection type	Synapse type	delay (ms)
DVS	SPTC	0.2	(int(i/(128*4)*32) +int(i % (128*4) / 3) to i)	excitatory	1
SPTC	TDE top-bottom	0.2	one_to_one	facilitator	1
SPTC	TDE top-bottom	0.2	i to i+32	trigger	1
SPTC	TDE bottom-top	0.2	one_to_one	trigger	1
SPTC	TDE bottom-top	0.2	i+32 to i	facilitator	0.1

Table 3. Neuron Parameters and Connections on SpiNNaker for sEMD characterization.

Name	Type	C_m (nF)	tau_m (ms)	tau_{ref} (ms)	v_{reset} (mV)	v_{rest} (mV)	v_{thresh} (mV)	tau_{syn_E} (ms)	tau_{syn_I} (ms)	I_{offset} (nA)	Popsize ($col \times row$)	#Pop
DVS	SSA										128×128	1
SPTC	LIF	0.25	35	1	-70	-65	-40	30	1	0	64×64	1
sEMD	TDE	0.25	30	1	-70	-65	-40	100	50	0	64×20	2
INT	LIF	0.25	20	1	-70	-65	-40	5	5	0	64×1	2
WTA	LIF	0.25	30	1	-70	-65	-40	100	50	0	64×1	1
GI	LIF	0.25	30	2	-68	-65	-50	40	5	0	1×1	1
MOT	LIF	0.25	20	2	-68	-65	-50	5	5	0	96×1	2
OUTPUT	LIF	0.25	20	2	-68	-65	-50	5	5	0	512×1	1
Name	Type	Rate	(Hz)								Popsize	#Pop
POIS1	Spike Source	50									64×1	1

Table 4. Neuron Parameters on SpiNNaker for real-world corridor centering experiment.

Name	Type	E_L (mV)	C_m (pF)	tau_m (ms)	t_{ref} (ms)	tau_{syn_exc} (ms)	tau_{syn_inh} (ms)	V_{th} (mV)	V_{reset} (mV)	V_m (mV)	Popsize ($col \times row$)	#Pop
SPTC	LIF	-60.5	25	20	1	10	10	-60	-60.5	-60.5	64×20	1
sEMD	TDE	-60.0	250	10	1	10	10	-30	-85	-60	64×20	2
INT	LIF	-70	250	20	1	5	5	-40	-70	-65	64×1	2
WTA	LIF	-65	250	20	1	5	80	-50	-68	-65	64×1	1
MOT	LIF	-65	250	20	2	5	5	-50	-68	-65	96×1	2
GI	LIF	-65	250	30	2	40	5	-50	-68	-65	1×1	1
OFI	LIF	-80	250	200	1	100	30	-40	-80	-75	1×1	1
ET	LIF	-65	250	20	1	5	80	-50	-68	-65	1×1	1
Name	Type	Rate	(Hz)								Popsize	#Pop
POIS1	Spike Source	100									64×1	1
POIS2	Spike Source	100									1×1	1

Table 5. Neuron Parameters from Neurorobotics Platform NEST network.

From	To	Weight (nA)	Connection type	Synapse type	delay (ms)
DVS	SPTC	0.5	(i and i+1 and i+128 and i+ 129) to i	excitatory	1
SPTC	TDE left-right	2	34*64+i+1 to i	trigger	1
SPTC	TDE left-right	2	34*64+i to i	facilitator	1
SPTC	TDE right-left	2	34*64+i+1 to i	facilitator	1
SPTC	TDE right-left	2	34*64+i to i	trigger	1
TDE right-left	INT right-left	3	i mod 64 to i	excitatory	1
TDE left-right	INT left-right	3	i mod 64 to i	excitatory	1
INT right-left	WTA	1	one_to_one	inhibitory	1
INT right-left	WTA	0.75	i to $i \pm 1$	inhibitory	1
INT right-left	WTA	0.5	i to $i \pm 2$	inhibitory	1
INT right-left	WTA	0.3	i to $i \pm 3$	inhibitory	1
INT right-left	WTA	0.25	i to $i \pm 4$	inhibitory	1
INT left-right	WTA	1	one_to_one	inhibitory	1
INT left-right	WTA	0.75	i to $i \pm 1$	inhibitory	1
INT left-right	WTA	0.5	i to $i \pm 2$	inhibitory	1
INT left-right	WTA	0.3	i to $i \pm 3$	inhibitory	1
INT left-right	WTA	0.25	i to $i \pm 4$	inhibitory	1
WTA(0-5)	MOT1	10	i to 38	excitatory	1
WTA(6-31)	MOT1	10	i to 2i + 32	excitatory	1
WTA(32-57)	MOT2	10	63 - i to 2i + 32	excitatory	1
WTA(58-63)	MOT2	10	i to 38	excitatory	1
WTA	GI	15	all_to_all	excitatory	1
GI	WTA	15	all_to_all	inhibitory	1
MOT1	WTA	5	all_to_all	inhibitory	1
MOT1	MOT2	10	all_to_all	inhibitory	1
MOT1	SPTC	50	all_to_all	inhibitory	1
MOT1	MOT1	2	i to i + 1	excitatory	4
MOT1	MOT1	10	one_to_one	inhibitory	1
MOT1	OUTPUT	10	4i to 188	excitatory	1
MOT1	OUTPUT	10	4i to 314	excitatory	1
MOT1	OUTPUT	10	95 to 336	excitatory	1
MOT1	OUTPUT	10	95 to 65	excitatory	1
MOT2	WTA	5	all_to_all	inhibitory	1
MOT2	MOT1	10	all_to_all	inhibitory	1
MOT2	SPTC	50	all_to_all	inhibitory	1
MOT2	MOT2	2	i to i + 1	excitatory	4
MOT2	MOT2	10	one_to_one	inhibitory	1
MOT2	OUTPUT	10	4i to 60	excitatory	1
MOT2	OUTPUT	10	4i to 442	excitatory	1
MOT2	OUTPUT	10	95 to 336	excitatory	1
MOT2	OUTPUT	10	95 to 65	excitatory	1
POIS1	WTA	2	one_to_one	excitatory	1

Table 6. Neuron connections from SpiNNaker for real-world corridor experiment. Note: There might be slight differences in the connection scheme when comparing Figure 4 with this table. This is because Figure 4 only serves for demonstration purposes. Always use the connections from this table to rebuild the network for the robot.

From	To	Weight (nA)	Connection type	Synapse type	delay (ms)
DVS NRP	SPTC	default	(i and i+1 and i+128 and i+ 129) to i	excitatory	0.1
DVS real world	SPTC	0.002	(i and i+1 and i+128 and i+ 129) to i	excitatory	0.1
SPTC	TDE left-right	4	one_to_one	trigger	0.1
SPTC	TDE left-right	4	i to i+1	facilitator	0.1
SPTC	TDE right-left	4	one_to_one	facilitator	0.1
SPTC	TDE right-left	4	i+1 to i	trigger	0.1
TDE right-left	INT right-left	1	i mod 64 to i	excitatory	0.1
TDE left-right	INT left-right	1	i mod 64 to i	excitatory	0.1
INT right-left	WTA	-5	one_to_one	inhibitory	0.1
INT right-left	WTA	-3	i to $i \pm 1$	inhibitory	0.1
INT right-left	WTA	-2	i to $i \pm 2$	inhibitory	0.1
INT right-left	WTA	-1.5	i to $i \pm 3$	inhibitory	0.1
INT right-left	OFI	10^{-4}	*all_to_all*	excitatory	0.1
INT left-right	WTA	-5	one_to_one	inhibitory	0.1
INT left-right	WTA	-3	i to $i \pm 1$	inhibitory	0.1
INT left-right	WTA	-2	i to $i \pm 2$	inhibitory	0.1
INT left-right	WTA	-1.5	i to $i \pm 3$	inhibitory	0.1
INT left-right	OFI	10^{-4}	*all_to_all*	excitatory	0.1
WTA(0-8)	MOT1	10	i to 50	excitatory	0.1
WTA(9-31)	MOT1	10	i to 2i + 32	excitatory	0.1
WTA(32-53)	MOT2	10	63 - i to 2i + 32	excitatory	0.1
WTA(54-63)	MOT2	10	i to 50	excitatory	0.1
WTA	GI	10	all_to_all	excitatory	0.1
ET	MOT1	10	0 to 0	excitatory	0.1
ET	GI	10	all_to_all	excitatory	0.1
GI	ET	-10	all_to_all	inhibitory	0.1
GI	WTA	-10	all_to_all	inhibitory	0.1
MOT1	WTA	-30	all_to_all	inhibitory	0.1
MOT1	ET	-30	all_to_all	inhibitory	0.1
MOT1	MOT2	-10	all_to_all	inhibitory	0.1
MOT1	Sensors	-30	all_to_all	inhibitory	0.1
MOT1	MOT1	10	i to i + 1	excitatory	10
MOT1	MOT1	-10	one_to_one	inhibitory	0.1
MOT2	WTA	-30	all_to_all	inhibitory	0.1
MOT2	ET	-30	all_to_all	inhibitory	0.1
MOT2	MOT1	-10	all_to_all	inhibitory	0.1
MOT2	Sensors	-30	all_to_all	inhibitory	0.1
MOT2	MOT2	10	i to i + 1	excitatory	10
MOT2	MOT2	-10	one_to_one	inhibitory	0.1
POIS1	WTA	1	one_to_one	excitatory	0.1
POIS2	ET	0.3	one_to_one	excitatory	0.1

Table 7. Neuron connections from NEST network used in the neurorobotics platform. Note: There might be slight differences in the connection scheme when comparing Figure 4 with this table. This is because Figure 4 only serves for demonstration purposes. Always use the connections from this table to rebuild the network.