
Article:
Zhu, L. orcid.org/0000-0003-3118-2882, Mangan, M. orcid.org/0000-0002-0293-8874 and Webb, B. orcid.org/0000-0002-8336-6926 (2023) Neuromorphic sequence learning with an event camera on routes through vegetation. Science Robotics, 8 (82). ISSN 2470-9476

https://doi.org/10.1126/scirobotics.adg3679

Neuromorphic sequence learning with an event camera on routes through vegetation

Le Zhu,[1*] Michael Mangan,[2] Barbara Webb[1]

[1]School of Informatics, University of Edinburgh, UK, EH8 9AB
[2]Sheffield Robotics, Department of Computer Science, University of Sheffield, UK, S1 4DP
[*]Corresponding author. E-mail: Le.zhu@ed.ac.uk.

For many robotics applications it is desirable to have relatively low-power and efficient on-board solutions. We take inspiration from insects such as ants that are capable of learning and following routes in complex natural environments using relatively constrained sensory and neural systems. Such capabilities are particularly relevant to applications such as agricultural robotics, where visual navigation through dense vegetation remains a challenging task. In this scenario, a route is likely to have high self-similarity and be subject to changing lighting conditions, while motion over uneven terrain, and the effects of wind on leaves increase the variability of the input. We use a bio-inspired event camera on a terrestrial robot to collect visual sequences along routes in natural outdoor environments, and apply a novel neural algorithm for spatio-temporal memory that is closely based on a known neural circuit in the insect brain. We show this method is plausible to support route recognition for visual navigation, and more robust than SeqSLAM when evaluated on repeated runs on the same route or routes with small lateral offsets. By encoding memory in a spiking neural network running on a neuromorphic computer, our model can evaluate visual familiarity in real-time from event camera footage.

1 INTRODUCTION

The challenges of robot autonomy have sparked increasing interest in understanding the efficient and low-power neural controllers that allow animals to perform robust, adaptive behaviour in complex environments. For example, it has been recently argued (*1, 2*) that unlocking the potential of new neuromorphic hardware for robotics requires a better understanding of the computing principles of real biological brains. Insect brains in particular (*3–7*) provide a powerful combination of efficiency and effectiveness, as well as tractability for understanding and emulating the details of their functional architecture. Here we provide an exemplar of such an approach, implementing a network for visual route memory on neuromorphic hardware, drawing directly on recent insights from insect neuroscience.

Even 'simple' animals such as ants can excel at navigation under conditions that still challenge current robots, such as through natural outdoor conditions including dense vegetation. One problem posed in such environments is to recognise previously visited places or traversed routes, as the basis for a navigation system. Vision is frequently used for this purpose, but faces a number of problems including changing appearances, lack of any distinctive landmarks, moving vegetation and highly similar scenes irrespective of the distance traversed. Visual place recognition (VPR) research has proposed solutions to tackle the appearance variance caused by lighting, weather, and also viewpoint changes (*8–12*). Additionally, there has been work on improving computational efficiency so as to run VPR on resource-constrained platforms (*13, 14*). Nevertheless for applications such as agriculture, forestry and environmental monitoring, the current state of the art (*15–19*) falls short of the abilities shown by insects. Specifically, desert ants follow precise routes to feeder locations many metres away in their desert habitat composed of dense scrub (*20, 21*) using visual cues alone (pheromones are unusable in the desert heat). Ants learn routes after a single trial and can recognise familiar locations after arbitrary displacements (i.e. solve the kidnapped robot problem). These abilities are robust across weather and light conditions (*22*) and support route-following at a higher precision than GPS would allow

for a robot.

One key insight from previous algorithmic models of ant navigation is that they do not necessarily need to solve the VPR problem explicitly by recognising *which* place they are in, provided they are capable of recognising whether or not a place is *familiar*. By modulating their actions in response to the current familiarity of the view, they can successfully recapitulate previously experienced routes leading to desired locations (*23, 24*). One limitation of models based on this principle (including robotic instantiations) has been the assumption that the ant stores static 'snapshots' of places along its route. Yet ants appear capable of creating visual memorie (*25–29*) and recognising scenes while in continuous motion. Insect visual systems are highly sensitive to motion, i.e., they fundamentally experience a spatio-temporal input rather than static frames (*30–32*). Some previous models have explored the use of optic flow images as a basis for recognition of locations (*33, 34*). Work in robotics has alternatively shown that matching the sequence information in (static) video frames during repeated traversals of a route can significantly improve the ability to localise the current position on the route, even with low-resolution images and changing light conditions (*12, 35–37*). There is also some evidence that ants' navigational decisions are influenced by the sequence of views they experience, and not just the current view (*38, 39*). Here we explore the effectiveness of a route memory based on dynamic visual input and using event timing in a spiking neural network to learn sequences.

This network model, first described in (*40*), is closely based on biology. It follows our earlier work in assuming that the key neural circuit in the insect brain for learning visual patterns is the mushroom bodies (MB) (*41*). Recent results showing that lesioning the MB in ants specifically affects performance on tasks requiring learnt (but not innate) visual orientation have supported this assumption (*42–44*). However, computational models based on the MB (for both olfactory and visual learning) have also mostly used static input patterns (*28, 45–49*), and for route following, have evaluated the performance in somewhat simplified visual environments with little of the variability that occurs in the real world. Our new model, by

contrast, addresses the problem of learning and recognising, on repeated traversal, the pattern of input spikes produced from visual change detection using an event camera (figure 1) on a robot moving comparable distances to an ant in a real outdoor environment with a low-level view of nearby natural vegetation.

Event cameras (also called dynamic vision sensors, DVS) are inspired by the processing of light in the animal's retina (*50*): local intensity changes are output by each pixel asynchronously in continuous time in a manner resembling the transient photo-receptor responses that encode intensity change in natural visual systems (*51*). Compared to conventional global shutter cameras, event-based systems offer high dynamic range, high temporal resolution, and low latency leading to their adoption for many computer vision and robotics tasks (for a recent review see (*52*)). VPR algorithms have benefited from improved images reconstructed from event streams (*53, 54*), or by using the event stream directly in conjunction with standard sequence matching algorithms (*55*) or deep-learning-based place recognition models (*56, 57*). More recently, the first VPR algorithms that combine the address event representation (AER) of event-based cameras with spiking neural networks (SNN) have been developed (*58*), demonstrating their utility for this task. The SpiNNaker neuromorphic computing system used in our work can simulate large-scale SNN in real-time using massively parallel low-power Arm processors incorporating a globally asynchronous locally synchronous (GALS) system (*59, 60*).

As described in more detail below, we use this hardware combination (event camera and neuromorphic computing platform) to implement a biologically plausible model for sequence memory. The model includes a new form of SNN learning, in which neurons make axo-axonic inhibitory connections that adapt to the spatio-temporal pattern of spikes. This novel axonic lateral interaction has been found not only in insect MBs (*64–67*) but also many other biological neural systems (*68–70*) where the output of one neuron and can effectively shut down the output of other neurons (*61*) (figure 2). We show that using this model it is possible to detect the familiarity of a visual stream, which could be used to maintain a course along a familiar

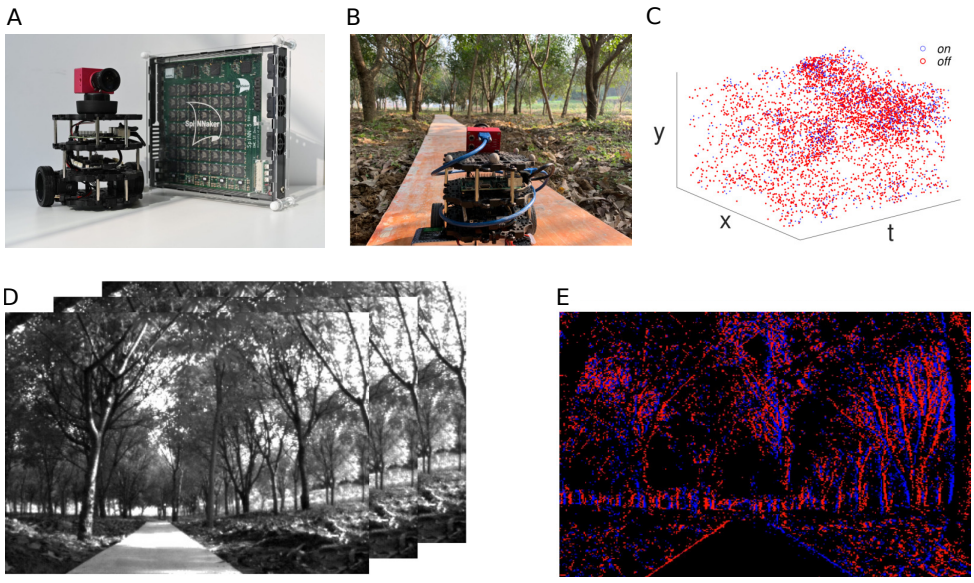

Figure 1: Robot hardware and event-based video. (**A**) The mobile robot is constructed from a TurtleBot3 Burger with a DAVIS346 event camera. A Spinn-5 board (*59*) housing 48 SpiNNaker chips was used for simulating our SNN model. (**B**) The model was trained/tested on data recorded from the robot as it was driven through natural environments with different levels of visual clutter. (**C**) The camera produces 'events' in continuous time whenever a pixel changes intensity.'x' and 'y': pixel address, 't': time (nanosecond time resolution from raw DAVIS output), 'on': dark to bright changes, 'off': bright to dark changes. (**D**) Conventional video has static intensity frames at a fixed rate. (**E**) Integrating 'events' over a period of forward motion, a scene can be visualised in the movement 'frame' from the event camera. Red and blue colours are polarities of events as shown in (C).

trajectory.

2 RESULTS

2.1 Mushroom Body Spiking Neural Network

In the previous image-matching model based on the MB circuit structure (*41*), visual patterns (image frames) formed a retinotopic pattern on a layer of visual projection neurons (PN) which then fan-out, with random connectivity to a much larger number of Kenyon cells (KC). This produces a relatively unique sparse pattern of KC activity for any visual scene. This pattern is learnt, for selected visual scenes along a homeward route, by reducing the excitatory weights

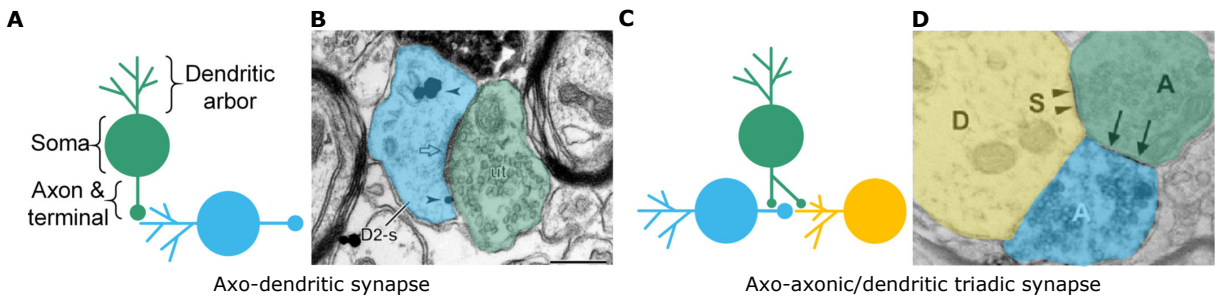

Figure 2: Canonical axodendritic synapse and non-canonical axo-axonic synapse depicted through schematic illustrations (**A, C**) and electron micrographs of axon fiber slices (**B, D**). The axo-axonic triad describes a configuration in which a neuron (green) synapses on both the presynaptic element (blue) and postsynaptic target (yellow) of an axodendritic or -somatic synapse. In this work, we modelled how a KC (green) can learn to inhibit another KC's (blue) excitatory output to MBON (yellow) so as to generate a lower output when the network comes across a familiar visual motion pattern. Figure reproduced from (*61*) with copyright permission granted. Original micrographs (B) and (D) were from (*62*) and (*63*) respectively.

from KCs to a single MB output neuron (MBON) when a reward signal is given. Subsequently, a familiar (learned) view will produce a low MBON output, which can be used to guide directional decisions.

Our new neural model (figure 3), first presented in (*40*), has a similar PN layer, this time activated by the event camera output, down-sampled in space and time. There is a similar fan-out to KCs, which again converge to a single MBON which should signal familiarity by reduced activity. However, the learning mechanism is altered. Instead of depressing the KC-MBON connection weights, we introduce KC-KC inhibition and adapt the strength of the inhibition using spike-timing dependent plasticity (STDP) (*71*). This allows the circuit to learn the spatio-temporal pattern of KC spikes produced when the robot moves along a route segment. If the same pattern occurs (e.g. when re-traversing the route) the increased KC-KC inhibition reduces the overall input from the KCs to the MBON, signalling familiarity.

In the work published in (*40*), we simulated the MB model using Python based simulation tool Brian2 (*72*). With 10,000 KCs, the whole network needs more than 20 minutes of run-time to process a two-second event stream recording, making real-time closed-loop robot control

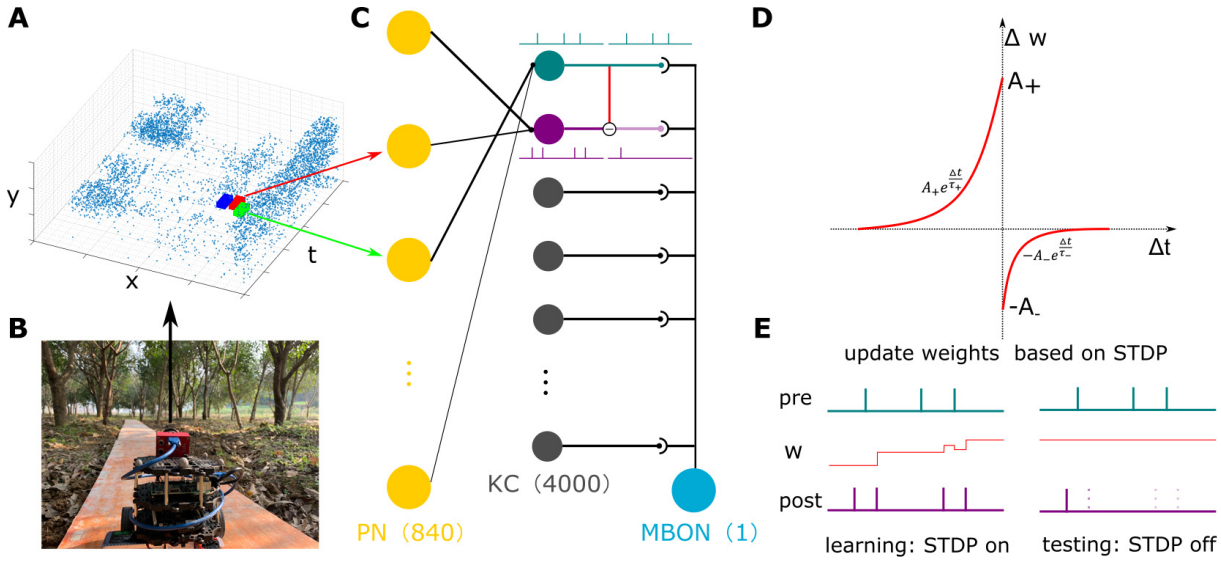

Figure 3: Mushroom body spiking neural network (MBSNN) encodes spatio-temporal memory as KC-KC inhibition. **(A, B)** Event stream input from the camera is firstly down-sampled both in spatial and temporal resolution. Down-sampled megapixels one-to-one map to the projection neurons in the MB network. **(C)** MB network structure. 840 projection neurons (PN) make sparse random connections to 4000 Kenyon cells (KC), which converge on a single mushroom body output neuron (MBON)). **(D)** Modified spike-timing-dependent plasticity (STDP) learning rule for KC-KC inhibition. **(E)** Example of a KC spiking sequence and how the weights are updated to learn this sequence pattern based on STDP. The nearer in time (within 50 ms) the KC pair fires together, the stronger the inhibitory connection from the first to the second in the firing sequence will become. Learned inhibitory weights will also be weakened if the pre- and post-synaptic neurons fire in reversed order. After learning, when the same visual flow occurs, the KC firing sequence will be the same. Excitation from KCs to MBON will be shut down, due to the inhibition generated precisely for this sequence. The MBON output can thus be interpreted as the (un)familiarity between current visual flow and stored memories.

impossible. To speed up the SNN simulation, we here use the neuromorphic computing platform - SpiNNaker (*59, 60*).In order to run the model on SpiNNaker, it had to be completely reimplemented using the sPyNNaker software package (*73*), and this necessitated significant changes. Brian2 has the flexibility of defining neuron models and synapse models using any user-proposed mathematical equations, which is not supported by sPyNNaker. As a result, the neuron model, synapse model, and the STDP learning rule had to be adapted from our previous work to be compatible with the default models offered by the SpyNNaker package. On the 48-chip board (SpiNN-5), the network capacity is limited by the local memory on each core and time constraints in simulation. Consequently, the network was reduced from 10,000 to 4,000 KCs.

An important additional modification was to introduce adaptation to the PN layer. This addresses the variability we observed in the input for different environments (and sometimes different areas in the same environment) which could produce very different rates of input events. Initially this required hand-tuning of parameters for each environment. We used a new adaptive leaky integrate-and-fire (LIF) neuron model (*74*) to model PN responses. We found a trade-off in using adaptive neural responses: on the one hand, improved generalisability from normalised network activity against varied input in different environments; on the other hand, temporal dynamics causing the network activity to be less deterministic on its current input, i.e., the adaptation accumulates over time then the network generates different spike train patterns when seeing the same visual flow but with different preceding visual experiences. Overall, the advantage of being able to use the same parameters across environments was considered more crucial to demonstrate the robustness of the approach. Neuron parameters were set based on calculations and electrophysiological data (see table 1) (*75–79*).

Some effort has been made to interface event-based cameras with SpiNNaker boards using either Ethernet port (with some delay) (*80*) or FPGA based SpiNN-link port (in real-time) (*81*). However, due to some ongoing software compatibility issues, we have not solved the real-

time interfacing problem to close the control loop. Thus in this paper, we demonstrate real-time offline processing in which the simulation time is the same as the input time length, in other words, the MB model processes 1 ms of visual input in 1 ms of simulation time on the SpiNNaker board. In principle, this means that a robot with this hardware could react in real-time according to the instantaneous familiarity signal output produced in response to the event camera input.

2.2 Test Environment

Outdoor tests were carried out in three different environments with varied vegetation heights (figure 4A). In each environment, we ran the robot on straight routes of 6 m between rows of plants. We note that 4-12 meters is a typical range over which ant visual route navigation has been observed and models tested. Experiments (*82*) show that ants rely on path integration more for longer distance (>7 m) travel especially when the two cues are competing. In the middle height environment, we also ran shorter offset routes (1 m). The dataset is available at Github page. For a smoother running of our wheeled robot, wooden boards were placed on the grassy and muddy ground. Note that in pre-processing stage the lower part of the frames was chopped so to get rid of the wooden board in the camera view. Compared with the indoor environment which was used in our previous work (*40*), outdoor visual surroundings become more cluttered which means denser visual input for the same length of robot movement. Apart from the visual environmental change, the uneven grassy outdoor ground challenged our wheeled robot. Even running on paved wooden boards, the unavoidable shake of the robot and camera introduces more noise motion and thus more noisy events input. The noise motion from the camera shaking altered the visual flow input for repeated runs in the same location. Other factors such as leaves moving in the wind and lighting changes during the day also add potential noise and increase the difficulty of route recognition.

2.3 Recognising routes

Although our MB model supports online learning, due to the insufficient computing power of the onboard robot PC (and unexpected technical hurdles for real-time integration of the event camera to the SpiNNaker board) we decided to collect the event video and performed all learning and testing offline. In offline learning, we playback one event video while the plasticity of KC-KC connections is turned on to generate KC-KC inhibition that adapts to the visual motion input. In testing, the KC-KC plasticity is turned off and the MBON output change due to learned KC-KC connections is used to indicate the similarity of the test input (either the same or a different event video) to the learned input. In the examples shown in figure 4, we learn different parts of the route (the first 2 or 3 metres, or 2 metres in the middle and compare the response of the MBON in this learned segment to the remainder of the route, which has not been learned. Our model is able to recognise learned motion patterns (shown as strongly reduced MBON activity in the pink segment) from unlearned ones, in various environments. Stretching or compressing the input time dimension without changing the dynamics of the spiking network shows that recognition generalises to this simulated change in robot speed.

2.4 Recognition after offset

Successful route following requires that the same route traversed again is still recognised. In practice, a robot running on the same trajectory will not do so precisely, and noise from camera shake and changing environmental conditions will contribute to input variation between runs. Also, small lateral displacements should not result in complete unfamiliarity, and indeed a gradient of increasing unfamiliarity for further displacement from the route can be used as a signal to guide the robot back towards the route, as it creates a 'valley' for the robot to follow (e.g. (*47*)). In this offset test, we ran the robot on the same route multiple times, and introduce parallel displacements, to test the spatial extent of familiarity recognition (figure 5). To evaluate for multiple trials and displacements, and compare different algorithms, we define a familiarity

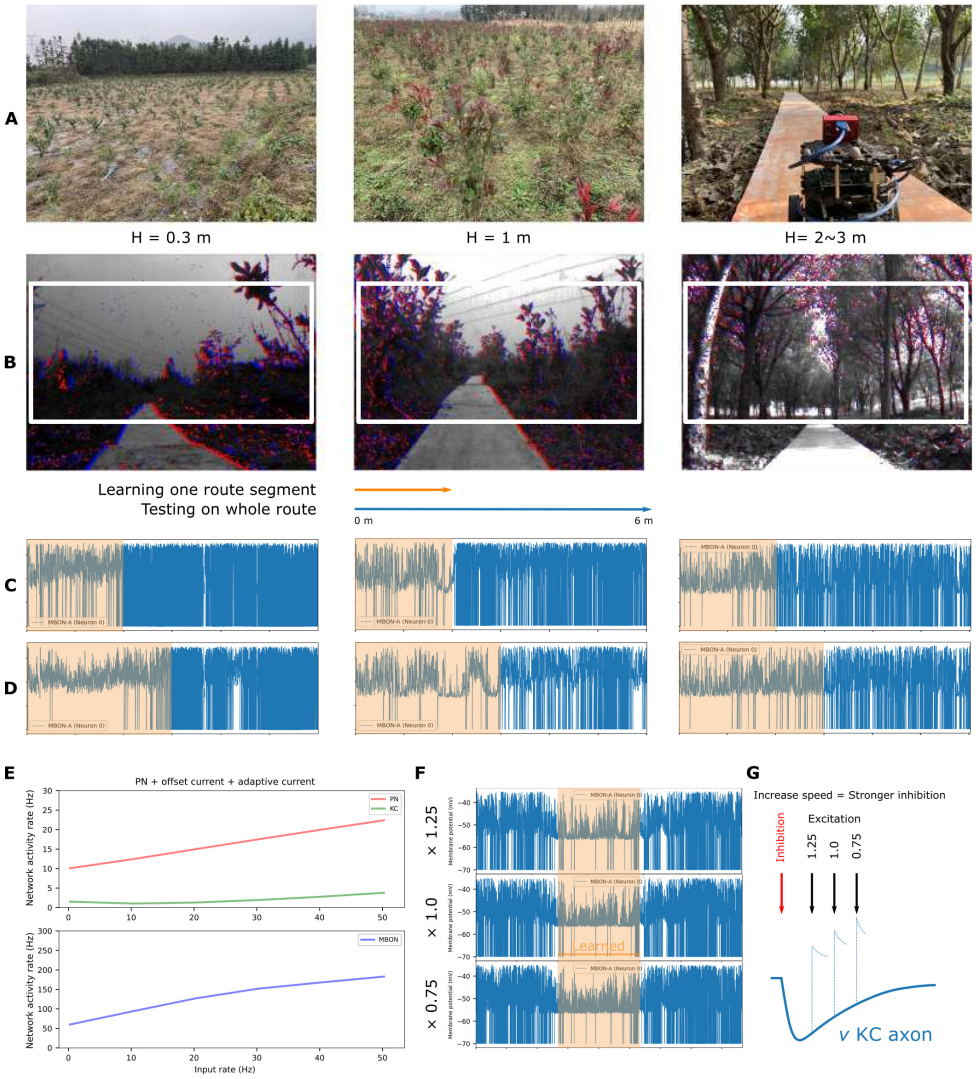

Figure 4: Route segment learning in three outdoor environments (**A**) where vegetation has varied height (H) and planting density. (**B**) In the DVS camera view, red and blue dots are ON and OFF events. Frames are chopped to get rid of some parts of the sky and ground. (**C**) and (**D**) In each test environment, a visual flow pattern was collected while the robot travelled through the vegetation on a straight route (6 m). After learning the visual flow pattern from a route segment (2 m in (C) and 3 m in (D)) the whole route pattern is replayed to the trained network. The MBON activity (membrane potential) is lower for learned segments (yellow) than for unlearned ones, signalling familiarity. (**E**) The adaptive PN response keeps similar generalised activity against input variation so as to maintain robustness in all three environments. (**F, G**) By distorting the time axis of the visual input, speed variation can be simulated. After learning the original speed ($\times 1.0 = 0.2\ m/s$), increased or decreased speed tests also detect the learned segment as familiar. This works because in our two-compartmental model, KC axonic inhibition brings the post-synaptic axon membrane potential down so cancels the excitation from PN. Increased speed (compressed time axis) leaves a shorter time for the KC axon membrane potential to climb up to resting potential, causing stronger inhibition.

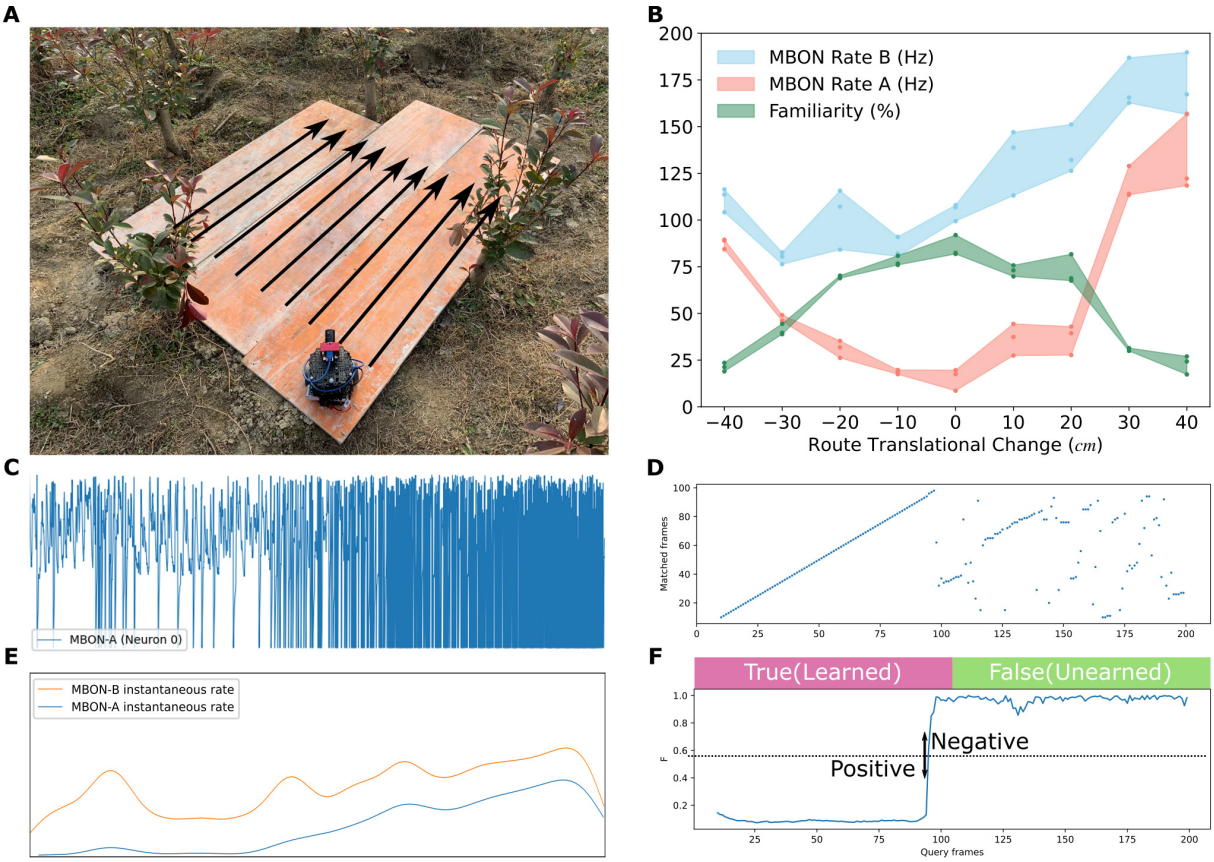

Figure 5: Offset tests. (**A**) In the $H = 1\ m$ environment, we record parallel offset routes at $0.1\ m$ intervals, each $1\ m$ long). The robot recorded each route three times. (**B**) After learning on the most central route, tests were conducted on all 27 displacement routes. The averaged MBON spiking rates over each run before (MBON Rate B) and after learning (MBON Rate A) are plotted. Familiarity (%) were calculated using $(MBONRateB - MBONRateA)/MBONRateB$, showing how much the MBON spiking rate has dropped because of learning. (**C-F**) Instead of learning the whole short route, the first half of the route is learned and tested by replaying the complete route ($1\ m$). (**C**) MBON membrane potential after learning shows the different responses for learned and unlearned segments. (**E**) The orange MBON spiking rate shows network output before learning (MBON without KC lateral inhibition). The blue (MBON-A) rate is the after-learning MBON spiking rate, used as the familiarity score, F. (**D**) SeqSLAM can correctly match the query to an identical reference image set (note only half of the images are learned). We adapted SeqSLAM (see figure 7) to obtain a similar F score. (**F**) After getting the familiarity index for both algorithms, a variable threshold is applied to calculate the true positive rate (TPR) and false positive rate (FPR), to further plot the receiver operating characteristic (ROC) curves in figure 6.

threshold, and assess the rate of true and false positives.

2.5 Benchmarking against SeqSLAM and Perfect Memory

SeqSLAM (*35, 83*) is a well-known VPR algorithm which calculates a locally enhanced difference matrix and selects the best matches only in a short sequence of images. To compare performance, we ran SeqSLAM on our data set, using standard grey scale images collected simultaneously with event streams (for results using the event data for SeqSLAM, see supplementary materials). SeqSLAM aims to match each query image to the learned reference images to get the image index (or sequential information in a video) of each query, while our model only identifies the familiarity of the current input based on learned memory. Thus we adapted the algorithm of SeqSLAM to calculate the familiarity of each query image, by ignoring the location of the reference image at the final match step, but only using the match score as a familiarity index (see figure 7). As an alternative comparison, we use the 'Perfect memory' benchmark to represent the potential performance of previous MB models based on static snapshot memories (*24*). This also compares a query image to all learned reference images (by direct pixel-pixel differencing) and takes the minimal difference as the familiarity score. Essentially, this is equivalent to SeqSLAM with the local sequence search distance set to one. For these benchmark tests, we plotted receiver operating characteristic (ROC) curves to visualise the performance of the algorithms. We calculated the area under the ROC curve (AUC-ROC) to assess the performance across a range of conditions and repeated trials.

An AUC-ROC score of 0.5 means the recognition is no better than a random guess. Although SeqSLAM can perform well in recognising a previously traversed location, regardless of lighting or weather change (*35, 83*), in our test it only gave good results when the reference and query data are the same, that is, the identical sequence is replayed. Input changes caused by re-traversal of the same route degrade the performance significantly (AUC-ROC of SeqSLAM is already near 0.5 with 0 cm offset), and small spatially offset routes (10cm or more) are un-

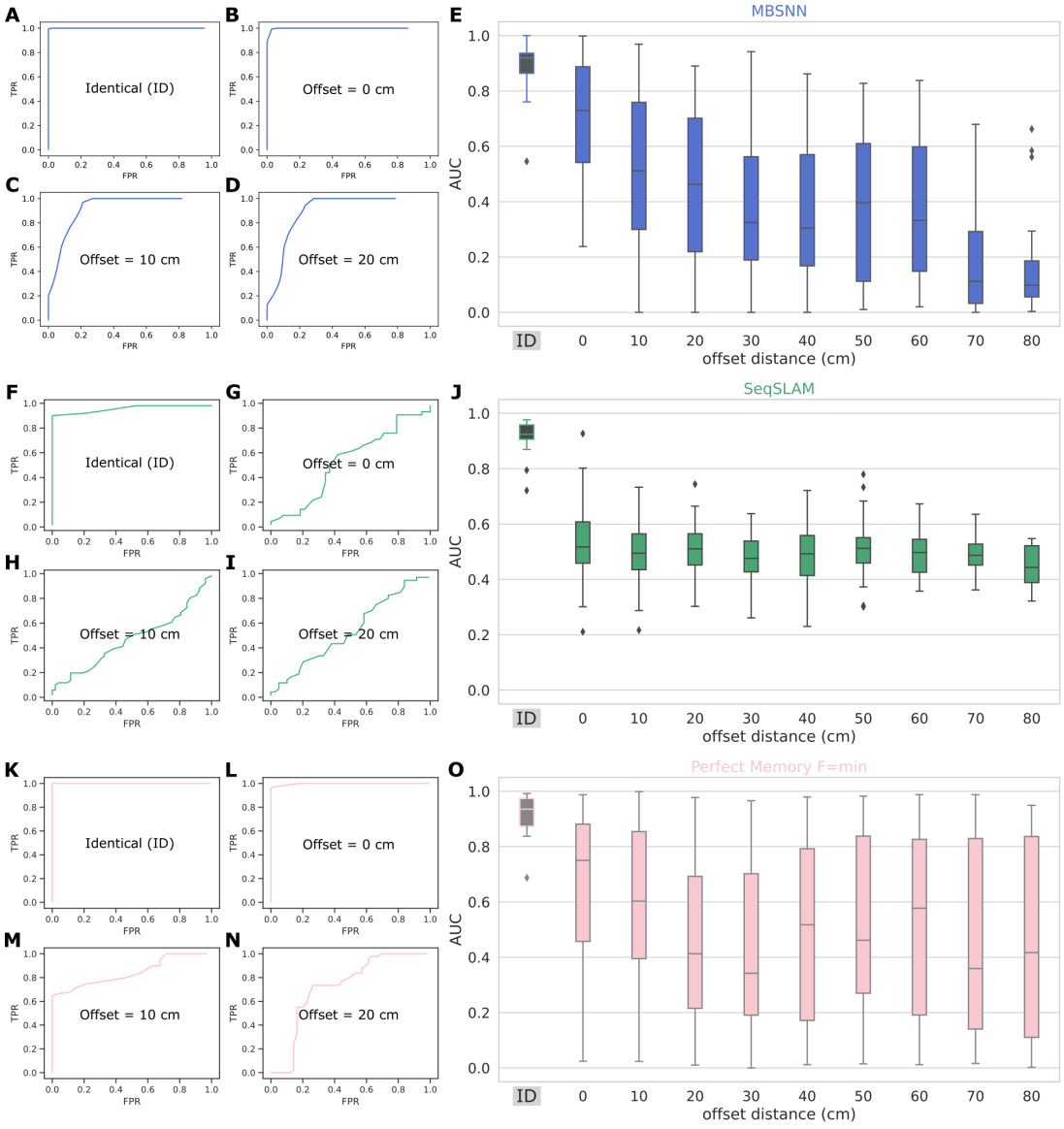

Figure 6: Offset routes recognition using MBSNN, SeqSLAM and Perfect Memory (PM). (**A-D, F-I, K-N**) Receiver operating characteristic (ROC) curves plotting true positive rate (TPR) against the false positive rate (FPR) from MBSNN, SeqSLAM and PM respectively. (**E, J, O**) The area under ROC curve (AUC-ROC) plots for all traverses, using MBSNN, SeqSLAM and PM respectively. When the query and the reference images are identical (the same recording from the same route, as in **A**, **F** and **K**, and boxplots labelled 'ID'), all three approaches can easily classify the query input as familiar. SeqSLAM is not able to recognise the other traverses from the same route (**G**: offset = 0 cm), or for any spatial offsets. PM can perform accurate but not precise detection of familiarity within 20 cm of offset. PM fails to detect familiarity in all cases after normalisation and this is again caused by the high self-similarity of the scenes (see Supplementary Materials for analysis). Our MB spiking neural network model (MBSNN) shows a gradual familiarity drop with spatial offset. Note the drop to below 0.5 at the highest displacements is due to the unbalanced distribution of vegetation along the routes (see SM).

recognised by SeqSLAM. The fall in recognition score is more gradual for MBSNN. After 30 cm offset, none of the algorithms can recognise the route as learned or familiar. The drop of AUC-ROC to below 0.5 for large offsets (70 or 80 cm) in MBSNN is caused by the unbalanced distribution of vegetation (see SM for more details). An additional minor drawback in SeqSLAM is that the very beginning and the very end of the video can not be put into the reference image set, i.e., in figure 5 D when the local sequence search is over 20 query frames, the first 10 frames have no chance to be matched. This is not a big issue when the video is long enough but will be when the local sequence search distance is close to the whole video length. Our model does nearly immediate recognition from the beginning of the input, thus suits better for short-route detection.

3 Discussion

We have presented an ant-inspired neural mechanism by which robot route memory based on continuous visual motion can be stored. The spatio-temporal pattern of spikes produced by an event camera on a robot moving through outdoor environments can be learnt, and recognised as familiar when presented again. Re-traversing the route produces a familiarity signal that decreases smoothly with the extent of lateral displacement, and hence in principle could be used to keep the robot on the route.

The model uses a spiking neural network with $4,841$ neurons which we demonstrate can run in real time on neuromorphic hardware. It is based on new neuroscientific data concerning the MB region of the insect brain, which is known to be responsible for associative memories. More specifically, it postulates a role for the recently observed, but functionally unexplained, KC interconnections (*64*). We assume that these KC-KC connections provide axo-axonic inhibition, such that activity in one neuron can block the output of another. Although observed in several circuits in both vertebrate and invertebrate brains (*68–70*), this form of neural interaction is not often considered in standard neural network models or even in spiking neuron simulators and

hardware. As we show, using an STDP learning rule that increases the inhibitory strength when one KC fires shortly before another, this architecture is well suited to learn spatio-temporal patterns such as those produced by dynamic visual input. The result is that familiar (previously experienced) visual sequences will produce a low output.

To clearly distinguish learned and unlearned patterns, the key is to generate an appropriate amount of KC-KC inhibition. Too few altered KC-KC connections will limit how much the MBON activity drops for trained patterns, making them hard to distinguish, against a noisy background, from untrained patterns. Too many altered KC-KC connections is even worse as the network has so much inhibition that any input pattern produces a low response, and again the pattern used for training can not be distinguished - the network has exceeded its capacity to hold distinguishable memories. Instead of using machine learning approaches to train the network, we calculated the network parameters based on electrophysiology data. The adaptive PN neuron generalised the network spiking rate in various environments, providing better robustness and generalisability. The size of our MB model (only 4,000 KCs) is smaller than for insect navigators (e.g. the honeybee MB has about 368,000 KCs (*84*)). At present the model attempts to learn a completely unfiltered stream of events when we turn on the KC-KC plasticity, but learning in the insect MB is likely to be more sophisticated and efficient in dealing with redundancies in the sensory stream. Introducing additional smoothing and motion processing layers between the camera output and the MB input is also likely to be helpful. For example, edge detection or edge-motion pre-processing might reduce the variability due to lighting. In insects, there are multiple neural layers of visual processing. Modelling these and other properties of the visual pathway is one of the obvious next steps for this work (*85–88*). In short, the memory capacity of the network can be extended by: 1) tuning the learning parameters so the weights grow slower; 2) increasing the number of KCs; 3) selectively learning only useful features; 4) introducing memory modulation and forgetting. It would then be interesting to test the memory capacity tuning using a different DVS dataset collected from longer routes with

omini-directional frames (*89*).

We evaluated the model in a setting that we believe is relevant, and still challenging, for robot applications such as agriculture. This is to perform fine-scale route following outdoors through corridors of dense vegetation. Such a scenario raises many issues, including high similarity of images along the route, changes induced by small scale robot deviations that are inevitable on uneven terrain, and environmental changes that influence the input. We collected a new data set from an event camera (which also recorded conventional video) on a mobile robot travelling short distances through several different heights of vegetation. Our model showed successful recognition of learned route segments between different instances of the same route and a gradual decrease in recognition as parallel displacement from the route increased. By contrast, testing SeqSLAM in these conditions, it was only able to recognise the route if the same video recording was used for training and testing. One reason for the poor generalisation might be the relatively high rate of image capture of our short outdoor routes. The data set used in the original SeqSLAM papers were either collected from a car or a train over long-distance travel, moving relatively fast compared to the frame rate. The image frames are consequently rather distinct in appearance. While in our short routes, the similarity between frames could be too high for SeqSLAM. The image difference from the start frame of the video to the end frame (1 m long) is no larger than the input change caused by parallel displacement (even in a 10 cm offset). This illustrates that different scenarios can offer different challenges, and that perhaps different algorithms need to be combined to utilise their complementary strengths.

We note that in principle, SeqSLAM (and other VPR methods) have the advantage of potentially returning the specific memory location that matches the current location (if they are linked during the learning phase), whereas our method only produces a general familiarity signal. In our application, we don't need the localisation of each query image; rather it is necessary for our route-following robot to perform a fine-scale familiarity assessment so the robot agent can stay in the valley of familiarity. However, one way to expand the network towards VPR applica-

tions would be to introduce multiple MBONs (as known to exist in the insect MB) which could represent distinct sections, or salient locations, on the route. Alternatively, the low-power and instant recognition model presented could be use to guide a robot between sparse waypoints provided by VPR algorithms or GPS (*90, 91*).

The first step of our future work is exploring the connectivity and pattern between PN-KC and KC-KC. Both biological findings and modelling work have argued that the connection between neuron groups is not globally random but rather functionally patterned (*92–97*). Here we construct the network connectivity pattern using a flexible coding function and can easily adjust the pattern in future work. Our future work also includes designing a navigation strategy which can utilise the output of the MB model to generate proper route-following patterns. Our model explains how an insect would know it is on the route, but not how it would re-find the route. Our results suggested that getting closer to the route should provide some increase in familiarity, so an oscillation strategy as suggested in (*98*) could be effective. Also, the event camera provides continuous motion sensing which should be complemented by continuous motor control, which requires the model to run in real-time. Work is ongoing to bridge fast sensing and real-time neuromorphic computing. In the end, we aim to model insect visual motion navigation behaviour using neuromorphic hardware on a robot that directly interacts with a world of comparable complexity to that of the ant.

4 MATERIALS AND METHODS

4.1 Objective and study design

The objectives of our study were (i) evaluate the biological hypothesis that MB can encode spatial-temporal memory into its KC-KC axo-axonic connections; (ii) implement the MB model onto a hardware robot system with a neuromorphic computer SpiNNaker and learn outdoor routes sensed by an event camera; (iii) Test the learning of sequential visual information and benchmark with SeqSLAM and PM.

4.2 Robot Platform, Bio-inspired Sensor and Neuromorphic Computer

To evaluate our model in a realistic way, we built a hardware robot system to interact with the real world and tested the system in outdoor natural environments. Our robot (figure 1 a) is a Turtlebot3 Burger (*99*) robot platform augmented with a novel visual sensor DAVIS346 (*100*). The Turtlebot has a single board computer (SBC) - Raspberry Pi 3 Model B+ (1.4GHz 64-bit quad-core processor), installed with Ubuntu Mate 18.04. Related ROS packages control the robot movement via an OpenCR board (STM32F746ZGT6 / 32-bit ARM Cortex-M7). Testing revealed that the SBC has the insufficient computational power to run our model online. Rather, camera data was recorded as the robot was manually driven along a predefined path and model learning and testing were processed offline. In offline processing, we simulated our spiking neural network model on a neuromorphic computer SpiNNaker (*59, 60*). The SpiNNaker neuromorphic computer utilizes a massively parallel computing design. Each SpiNNaker chip consists of 18 low-power ARM cores, featuring local instruction and data memory on each core. Given the local memory and the requirement for real-time simulations, each core can simulate up to 250 neurons, contingent on the complexity of neuron and synapse models, as well as the neural activity. During simulations, SpiNNaker cores are predominantly idle, becoming active only when interrupted by incoming spikes to update neural activity. Power consumption per chip ranges from 0.25 W to 0.9 W (*101*), leading to the SpiNNaker 5 board (48 chips, 864 cores) consuming 12-43 W during real-time simulations which is hundreds of times faster than our previous Brian2 (*72*) implementation (*40*). It is important to note that "real-time" simulation speed here implies that the computation time associated with the simulation does not exceed the time that has elapsed in the model. Despite the challenges of procuring neuromorphic computing hardware and overcoming software compatibility issues in this developing field, the power efficiency of neuromorphic computing when running large SNNs in real-time is worth noting.

To better embed the DVS into our model, some pre-processing of the raw input is necessary. The spatial and temporal acuity of insect visual systems vary among species, and also depend

on environmental lighting conditions. The spatial acuity of insect eyes commonly falls between 2-5 degrees of their visual field (*102*); their temporal resolution, measured by photoreceptor flickering response, is usually no higher than 300 Hz (3ms) (*103–105*). Therefore, before feeding the camera output to the neural network, the event flow is re-sampled. The whole image frame is down sampled by 8 pixels $\times$ 8 pixels. In each down-sampled mega pixel, an input event will trigger an event counter lasting for 1 ms. In this 1 ms, when the number of events happening in the mega pixels outnumbers a noise threshold (three events), the mega pixel will output a valid spike to the spatially mapped PN. After this event counter, the next input event will trigger the next counter on this mega pixel. After re-sampling, the effective spatial acuity is about 5 degrees and the temporal resolution is 1ms. Note that there are additional processing steps in the insect visual motion pathway (*51*), but we have not included these steps in our modelling so far.

4.3 Mushroom Body Network and Implementation

4.3.1 Neuron Models

When implementing our model, we used the software package *sPyNNaker* (*73*) to simulate PyNN (*106*) defined network on SpiNNaker hardware. The KC and MBON were modelled as standard LIF neurons with fixed threshold and decaying-exponential post-synaptic current (named as *'IF_curr_exp'* in PyNN and *sPyNNaker*) described by equation 1, 2 and 3.

$$\frac{\mathrm{d}V}{\mathrm{d}t} = -\frac{V - (V_{rest} + R_m I(t))}{\tau_m} \tag{1}$$

Equation 1 models the dynamics of sub-threshold membrane potential V. I is the current combining synaptic (I_{syn}), intrinsic and background input. R_m is the membrane resistance, τ_m is the membrane leak time constant, and V_{rest} is the resting membrane potential. When V reaches a threshold voltage (V_{thresh}), the neuron will generate a spike and the the membrane potential resets:

$$if V > V_{thresh}, V = V_{reset} \tag{2}$$

For the KC and MBON LIF neuron (*'IF_curr_exp'*), its synaptic input current I_{syn} is modelled as equation 3:

$$\frac{\mathrm{d}I_{syn}}{\mathrm{d}t} = -\frac{I_{syn}}{\tau_{syn}} + \delta(t - t^j) \tag{3}$$

This *'IF_curr_exp'* model has separate synaptic currents for excitatory and inhibitory synapses with independent time constants τ_{syn}. The delta function represents addition of a step change in input from the weight of an incoming spike.

For PN we used an adaptive LIF neuron model (*74*) (named as *'IFCurrExpCa2Adaptive'* in *sPyNNaker*). Compared to the KC and MBON model, the PN model has one more Ca^{2+} activated K^+ current (I_{AHP}) which adapts the membrane potential according to its firing activity. At each spike , the adaptive current increases by:

$$I_{AHP} = I_{AHP} + \alpha \tag{4}$$

where α is set to 0.2 mA in our model. When the input side is overactive, the increased adaptive current will bring down the PN membrane potential to lower its firing rate. The I_{AHP} also decays to an offset values (I_{offset} = 0.02 mA) and keeps the PN firing activity at a baseline when the input is silent (see figure 4 E and SM S7).

$$\frac{\mathrm{d}I_{AHP}}{\mathrm{d}t} = -\frac{I_{AHP} - I_{offset}}{\tau_{AHP}} \tag{5}$$

We set the parameters for neuron models and synapses models based on biological data found in (*75–79*) and the neuron parameters shown in table 1.

4.3.2 Connections

Each core in the SpiNNaker system updates its neuron states using a fixed simulation timestep (Δt). When a neuron fires, spikes are transmitted to all postsynaptic neurons for real-time

evaluation of synaptic contribution. Although cores operate asynchronously, it is preferable for neurons on all cores to advance roughly in parallel for a coherent simulation progression. Thus, all cores in a simulation start synchronised. The synapse state is updated during the periodic neuron update using exact integration, with step changes based on synaptic input buffer contributions, as described by equations 6 and 7:

$$I_{t+1} = I_t e^{-\frac{\Delta t}{\tau_{syn}}} + \Sigma_j w_{ij} \delta(t - t_j) \tag{6}$$

$$V_{t+1} = V_{rest} + R_m I_{t+\Delta t} - e^{-\frac{\Delta t}{\tau_m}} (V_{rest} + R_m I_{t+\Delta t} - V_t) \tag{7}$$

For a static synapse construction, the only parameter we set is the w_{ij}, apart from defining the pre-synaptic and post-synaptic neuron index.

Although the connections between PNs and KCs were modelled as globally random and weighted equally in many experimental studies, some recent anatomical and electrophysiological findings revealed the feasibility of patterned structure and variability of this connection (*92–94*), and some modelling work further evaluated how functional patterns can impact the performance in learning and classification (*95–97*). By changing the ratio between presynaptic olfactory PN and postsynaptic KC, (*107*) found that connection density between PNs and KCs is set by KC: KC claw number does not vary much as PN number changes while PNs change their boutons (terminal of axons) depending on KC number. Based on the aforementioned findings, we coded both the PN-KC and KC-KC using one framework with the flexibility to easily change the connection pattern and weights distribution. Although we haven't systematically evaluated how the connection structure will affect learning and performance, this will be part of our future work. In our model, each KC randomly connects to around 5 PNs ($n_{PN} \sim \mathcal{N}(\mu, \sigma^2)$, $\mu = 5$, $\sigma = 1$, and rounded to the nearest integer) and the input weights to each KC (w_{ij}) are random distributed ($w_{ij} \sim \mathcal{N}(\mu, \sigma^2)$, $\mu = 0.3$, $\sigma = 0.1$). In the KC-KC connection, to our knowledge, there are few published biological justifications to clarify the connection pattern.

In our model, we limited the number of KCs (4,000) to much fewer than the number observed in navigating insects such as ants and bees. We also limited the density so each KC connects to 500 other KCs and the connection weights are all zero before learning. So the sparseness of KC-KC connections depends on the amount of KC sequences learned. By setting these limitations in KC-KC connections, we not only saved some computational load for the neuromorphic computer but also reserved the potential of expanding our model to a larger capacity.

4.3.3 KC-KC STDP

In practice, we use a two-compartment model for each KC, implemented as two leaky integrate and fire units which get the same PN activation. The second compartment gets inhibitory input from the first compartment of other KCs, and its output excites the MBON. This implementation facilitates the use of the standard STDP module (*AdditiveWeightDependence STDPMechanism*) in *sPyNNaker* (*108–111*).

$$w(\Delta t) = \begin{cases} A_+ e^{\frac{\Delta t}{\tau_+}} & \Delta t < 0 \\ -A_- e^{\frac{-\Delta t}{\tau_-}} & \Delta t \geq 0 \end{cases} \tag{8}$$

In equation 8, Δt ($\Delta t = T_{post} - T_{pre}$) is the time difference between the pre- and post- synaptic spike timing. A_+ and A_- are the maximum synaptic modifications, τ_+ and τ_- determine the time range of spike interval over which the STDP occurs. $w(\Delta t)$ is the weight modification during one pair of pre- and post-synaptic spikes. Using the *AdditiveWeightDependence*, the weight w will be added by $w(\Delta t)$ and then clipped within w_{max} and w_{min}. The parameters used for constructing KC-KC STDP were shown in table 1. The weights of all KC-KC connections are initialised to zero, and they are altered by STDP as described in figure 3. That is, if one KC fires shortly before another KC, the inhibitory effect from the first KC to the output compartment of the second KC will be increased, reducing the excitation it passes to the MBON.

Parameter	Neuron			Units	Description
	PN	KC	MBON		
V_{rest}	-60.0	-80.0	-56.7	mV	Resting membrane potential
$c_m (= \tau_m / R_m)$	15.0	6.0	12.0	pF	Capacity of the membrane
τ_m	20.0	10.0	16.06	ms	Membrane time constant
τ_{refrac}	20	20	0.1	ms	Duration of refractory period
$\tau_{syn}E$	5.0	5.0	1.0	ms	Decay time of the excitatory conductance
$\tau_{syn}I$	5.0	1.5	1.0	ms	Decay time of the inhibitory conductance
I_{offset}	0.02	0.0	0.0	nA	Offset current
V_{thresh}	-35.0	-40.0	-35.0	mV	Spike threshold
V_{reset}	-70.0	-90.0	-70.0	mV	Reset potential after a spike
w_DVS2PN	1.0			null	Input to PN connection weights
w_PN2KC	0.3			null	PN to KC connection (mean) weights
w_KC2MBON	0.01			null	KC to MBON connection weights
$STDP\tau_+$	1.25			ms	exponential decay factor of potentiation weight
$STDP\tau_-$	0.1			ms	exponential decay factor of depression weight
$STDPA_+$	0.3			null	maximum weight to add during potentiation
$STDPA_-$	0.15			null	maximum weight to subtract during depression
$STDPw_{min}$	0.0			null	minimum weight for KC-KC connection
$STDPw_{max}$	0.5			null	maximum weight for KC-KC connection

Table 1: Parameters used to construct the SNN. Names of parameters are consistent with the default variable names in *sPyNNaker* wherever possible.

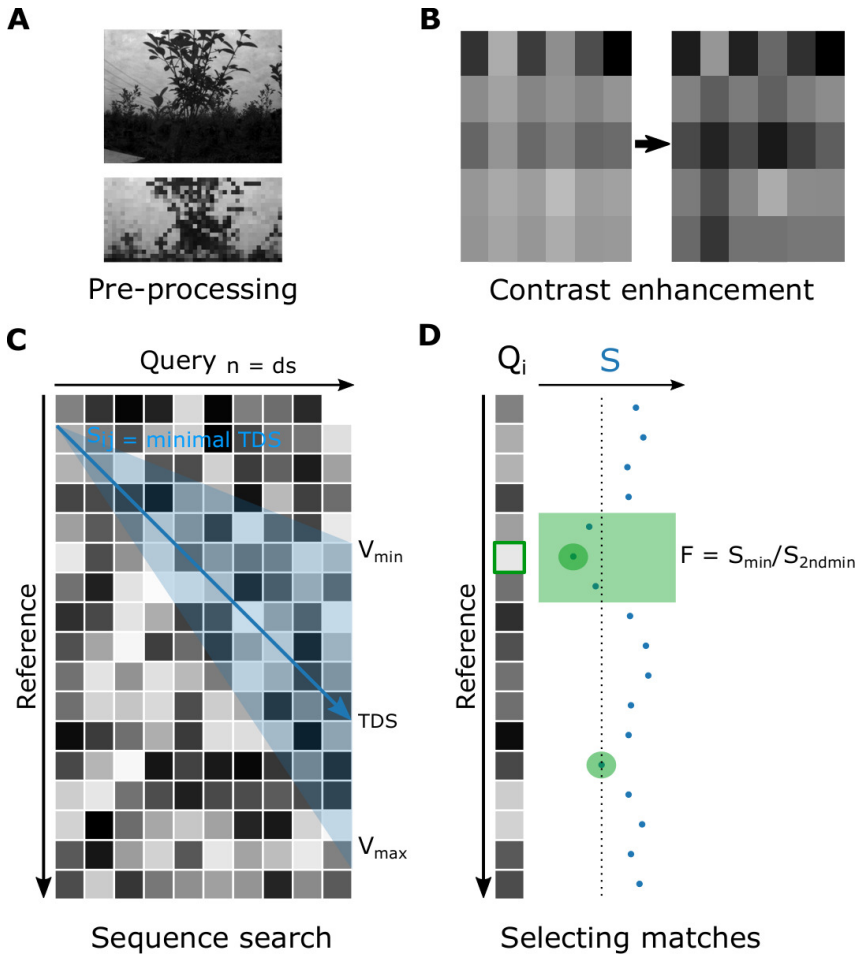

Figure 7: Adapting SeqSLAM for our benchmark test. Figure reproduced from (*35*). (**A**) In pre-processing, each image is cropped and down-sampled. (**B**) Enhance contrast on the absolute difference matrix between the query images and the reference images to facilitate sequence search. Darker colour means a bigger difference. (**C**) For the ith query image, a local difference matrix is selected from the contrast-enhanced difference matrix, containing ds query images and all reference images. Trajectory difference scores (TDS) on each local sequence (blue line) between the range of Vmin and Vmax were calculated, and the minimal TDS (as from the best local sequence) is set as the match score (Sij) for the ith query image on the jth reference image. (**D**) For the ith query image Qi, the best-matched reference image is where the minimal match score S stands and its match index F is normalised by the second minimal S. The second minimal S needs to be out of a selection window so to make sure the global minimum and second minimum are not neighbouring. A global threshold Fth can be applied to filter out weaker matches. In this work, we ignored the localisation of the reference image and only used the score F indicating how familiar Qi is.

4.4 SeqSLAM and Perfect Memory

From the DAVIS346 we can record both frame-based video and event stream. Being consistent with the prepossessing of events, the grey scale frames are resized to the same view as pre-processing in the pre-processing of MBSNN input. The frames were chopped to get rid of the ground and part of the sky and downsampled by 8×8 pixels. Then the absolute differences between query images and reference images were calculated to get a difference matrix which is then enhanced (equation 9) to facilitate finding local best matches.

$$\hat{D}_i = \frac{D_i - \overline{D_l}}{\sigma_l} \tag{9}$$

D is the vector of the differences between an input image. Each element, D_i, in D is normalised using equation 9, where D_l and σ_l denote the mean and standard deviation of D, respectively. This enhancement ensures that even when an input image significantly differs from the reference images due to substantial illumination changes, resulting in large difference values, the true correspondence is anticipated to have a relatively smaller difference compared to the others.

For each pair of query image Q and reference image R, local sequences are searched in space M in which the trajectories travel over ds query images between Vmin and Vmax (blue area in figure 7 C).

$$\mathbf{M} = \left[\hat{\mathbf{D}}^{T-d_s}, \hat{\mathbf{D}}^{T-d_s+1} \ldots \hat{\mathbf{D}}^{T}\right] \tag{10}$$

where T is the current time.

$$TDS = \sum_{t=T-d_S}^{T} D_k^t \tag{11}$$

where TDS is the trajectory difference score of this local sequence and k is the particular difference value the trajectory passes through at time t:

$$k = s + V(d_s - T + t) \tag{12}$$

Each query image Q has a vector of scores representing the perceived likelihood that each reference image R could be a match. A sliding window is used to select the global minimal and the second minimal score. In the original SeqSLAM, the match score (M) is then calculated as the global minimum divided by the second minimum. In our benchmark test, we ignored the localisation of reference image Rj, and directly used (F = M) as the familiarity score of the query image Qi.

The final step of SeqSLAM was setting a threshold to filter some of the weaker matches. In our work, we learned the first half of the route and then changed the threshold on the familiarity index, to get the true positive rate (TPR) and false positive rate (FPR), and plotted the ROC curve and AUC-ROC.

In the PM (*24*) test, the novelty of the current view is the minimum of the sum squared difference in pixel values between the current view and each of the stored views. This is equivalent to setting the local sequence distance to one frame in SeqSLAM (n =1 in figure 7 C). We applied the same contrast enhancement (figure 7 B) and normalisation (divided by the second minimum, figure 7 D) to better compare with SeqSLAM. The results of the PM test before normalisation can be seen in figure 6, while the results after normalization are included in the SM.

5 REFERENCES

References

1. Y. Sandamirskaya, Rethinking computing hardware for robots (2022).

2. Y. Sandamirskaya, M. Kaboli, J. Conradt, T. Celikel, Neuromorphic computing hardware and neural architectures for robotics, *Science Robotics* 7, 67, (2022) eabl8419.

3. B. Webb, Robots in invertebrate neuroscience, *Nature* 417, 6886, (2002) 359–363.

4. B. Webb, Robots with insect brains, *Science* 368, 6488, (2020) 244–245.

5. P. Manoonpong, L. Patanè, X. Xiong, I. Brodoline, J. Dupeyroux, S. Viollet, P. Arena, J. R. Serres, Insect-inspired robots: Bridging biological and artificial systems, *Sensors* 21, 22, (2021) 7609.

6. G. De Croon, J. Dupeyroux, S. Fuller, J. Marshall, Insect-inspired ai for autonomous robots, *Science Robotics* 7, 67, (2022) eabl6334.

7. M. Mangan, D. Floreano, K. Yasui, B. A. Trimmer, N. Gravish, S. Hauert, B. Webb, P. Manoonpong, N. Szczecinski, A virtuous cycle between invertebrate and robotics research: perspective on a decade of living machines research, *Bioinspiration Biomimetics* 18, 3, (2023) 035005, doi:10.1088/1748-3190/acc223, URL https://dx.doi.org/10.1088/1748-3190/acc223.

8. R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, J. Sivic, Netvlad: Cnn architecture for weakly supervised place recognition, in Proceedings of the IEEE conference on computer vision and pattern recognition, 5297–5307 (2016).

9. S. Garg, N. Suenderhauf, M. Milford, Lost? appearance-invariant place recognition for opposite viewpoints using visual semantics, *Proceedings of Robotics: Science and Systems XIV* .

10. F. Warburg, S. Hauberg, M. Lopez-Antequera, P. Gargallo, Y. Kuang, J. Civera, Mapillary street-level sequences: A dataset for lifelong place recognition, in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2626–2635 (2020).

11. M.-A. Tomită, M. Zaffar, M. J. Milford, K. D. McDonald-Maier, S. Ehsan, Convsequential-slam: A sequence-based, training-less visual place recognition technique for changing environments, *IEEE Access* 9, (2021) 118673–118683.

12. A. Özdemir, M. Scerri, A. B. Barron, A. Philippides, M. Mangan, E. Vasilaki, L. Manneschi, Echovpr: Echo state networks for visual place recognition, *IEEE Robotics and Automation Letters* 7, 2, (2022) 4520–4527.

13. M. Zaffar, S. Ehsan, M. Milford, K. McDonald-Maier, Cohog: A light-weight, compute-efficient, and training-free visual place recognition technique for changing environments, *IEEE Robotics and Automation Letters* 5, 2, (2020) 1835–1842.

14. B. Ferrarini, M. Waheed, S. Waheed, S. Ehsan, M. Milford, K. D. McDonald-Maier, Visual place recognition for aerial robotics: Exploring accuracy-computation trade-off for local image descriptors, in 2019 NASA/ESA Conference on Adaptive Hardware and Systems (AHS), 103–108, IEEE (2019).

15. S. Lowry, N. Sünderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, M. J. Milford, Visual place recognition: A survey, *IEEE Transactions on Robotics* 32, 1, (2015) 1–19.

16. C. Masone, B. Caputo, A survey on deep visual place recognition, *IEEE Access* 9, (2021) 19516–19547.

17. A. S. Aguiar, F. N. dos Santos, J. B. Cunha, H. Sobreira, A. J. Sousa, Localization and mapping for robots in agriculture and forestry: A survey, *Robotics* 9, 4, (2020) 97.

18. J. Dong, J. G. Burnham, B. Boots, G. Rains, F. Dellaert, 4d crop monitoring: Spatio-temporal reconstruction for agriculture, in 2017 IEEE International Conference on Robotics and Automation (ICRA), 3878–3885 (2017), doi:10.1109/ICRA.2017.7989447.

19. J. Sarmento, A. Silva Aguiar, F. Neves dos Santos, A. J. Sousa, Autonomous robot visual-only guidance in agriculture using vanishing point estimation, in EPIA Conference on Artificial Intelligence, 3–15, Springer (2021).

20. M. Mangan, B. Webb, Spontaneous formation of multiple routes in individual desert ants (cataglyphis velox), *Behavioral ecology* 23, 5, (2012) 944–954.

21. M. Kohler, R. Wehner, Idiosyncratic route-based memories in desert ants, melophorus bagoti: how do they interact with path-integration vectors?, *Neurobiology of learning and memory* 83, 1, (2005) 1–12.

22. L. Haalck, M. Mangan, A. Wystrach, L. Clement, B. Webb, B. Risse, Cater: Combined animal tracking & environment reconstruction, *Science Advances* 9, 16, (2023) eadg2094.

23. B. Baddeley, P. Graham, A. Philippides, P. Husbands, Holistic visual encoding of ant-like routes: Navigation without waypoints, *Adaptive Behavior* 19, 1, (2011) 3–15, doi:10.1177/1059712310395410, https://doi.org/10.1177/1059712310395410, URL https://doi.org/10.1177/1059712310395410.

24. B. Baddeley, P. Graham, P. Husbands, A. Philippides, A model of ant route navigation driven by scene familiarity, *PLOS Computational Biology* 8, 1, (2012) 1–16, doi:10.1371/journal.pcbi.1002336, URL https://doi.org/10.1371/journal.pcbi.1002336.

25. T. S. Collett, M. Collett, Memory use in insect visual navigation, *Nature Reviews Neuroscience* 3, 7, (2002) 542.

26. M. Collett, L. Chittka, T. Collett, Spatial memory in insect navigation, *Current Biology* 23, 17, (2013) R789–R800.

27. J. Zeil, Visual homing: an insect perspective, *Current opinion in neurobiology* 22, 2, (2012) 285–293.

28. B. Webb, The internal maps of insects, *The Journal of Experimental Biology* 222, Suppl 1, (2019) jeb188094, doi:10.1242/jeb.188094, URL http://jeb.biologists.org/content/jexbio/222/Suppl{_}1/jeb188094.full.pdf.

29. N. H. de Ibarra, P. Graham, T. S. Collett, R. A. Harris, Ant navigation priming of visual route memories, *Nature* 438, 7066, (2005) 302–302.

30. A. Arenz, M. S. Drews, F. G. Richter, G. Ammer, A. Borst, The temporal tuning of the drosophila motion detectors is determined by the dynamics of their input elements, *Current Biology* 27, 7, (2017) 929–944.

31. K. Nagel, Motion vision: Pinning down motion computation in an ever-changing circuit, *Current Biology* 31, 23, (2021) R1523–R1525.

32. A. D. Gonzalez-Suarez, J. A. Zavatone-Veth, J. Chen, C. A. Matulis, B. A. Badwan, D. A. Clark, Excitatory and inhibitory neural dynamics jointly tune motion detection, *Current Biology* 32, 17, (2022) 3659–3675.

33. A. Vardy, R. Moller, Biologically plausible visual homing methods based on optical flow techniques, *Connection Science* 17, 1-2, (2005) 47–89.

34. L. Dittmar, W. Stürzl, E. Baird, N. Boeddeker, M. Egelhaaf, Goal seeking in honeybees: matching of optic flow snapshots?, *Journal of Experimental Biology* 213, 17, (2010) 2913–2923.

35. M. J. Milford, G. F. Wyeth, Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights, in 2012 IEEE International Conference on Robotics and Automation, 1643–1649, IEEE (2012).

563 36. M. Milford, Vision-based place recognition: how low can you go?, *The International*
564 *Journal of Robotics Research* 32, 7, (2013) 766–789.

565 37. M. Chancán, L. Hernandez-Nunez, A. Narendra, A. B. Barron, M. Milford, A hybrid
566 compact neural architecture for visual place recognition, *IEEE Robotics and Automation*
567 *Letters* 5, 2, (2020) 993–1000.

568 38. P. Graham, M. Mangan, Insect navigation: do ants live in the now?, *Journal*
569 *of Experimental Biology* 218, 6, (2015) 819–823, doi:10.1242/jeb.065409, https:
570 //jeb.biologists.org/content/218/6/819.full.pdf, URL https://
571 jeb.biologists.org/content/218/6/819.

572 39. S. Schwarz, M. Mangan, B. Webb, A. Wystrach, Route-following ants respond to alter-
573 ations of the view sequence, *The Journal of Experimental Biology* 223, (2020) jeb.218701,
574 doi:10.1242/jeb.218701.

575 40. L. Zhu, M. Mangan, B. Webb, Spatio-temporal memory for navigation in a mushroom
576 body model, in Conference on Biomimetic and Biohybrid Systems, 415–426, Springer
577 (2020).

578 41. P. Ardin, F. Peng, M. Mangan, K. Lagogiannis, B. Webb, Using an insect mushroom body
579 circuit to encode route memory in complex natural environments, *PLOS Computational*
580 *Biology* 12, 2, (2016) 1–22, doi:10.1371/journal.pcbi.1004683, URL https://doi.
581 org/10.1371/journal.pcbi.1004683.

582 42. C. Buehlmann, B. Wozniak, R. Goulard, B. Webb, P. Graham, J. E. Niven, Mushroom
583 bodies are required for learned visual navigation, but not for innate visual behavior, in
584 ants, *Current Biology* 30, 17, (2020) 3438–3443.e2, doi:https://doi.org/10.1016/j.cub.
585 2020.07.013, URL https://www.sciencedirect.com/science/article/
586 pii/S0960982220310058.

587 43. J. F. Kamhi, A. B. Barron, A. Narendra, Vertical lobes of the mushroom bodies are es-
588 sential for view-based navigation in australian myrmecia ants, *Current Biology* 30, 17,
589 (2020) 3432–3437.e3, doi:https://doi.org/10.1016/j.cub.2020.06.030, URL https://
590 www.sciencedirect.com/science/article/pii/S0960982220308460.

591 44. S. Heinze, Visual navigation: Ants lose track without mushroom bodies, *Cur-*
592 *rent Biology* 30, 17, (2020) R984–R986, doi:https://doi.org/10.1016/j.cub.2020.
593 07.038, URL https://www.sciencedirect.com/science/article/pii/
594 S0960982220310307.

595 45. E. Kagioulis, A. Philippides, P. Graham, J. C. Knight, T. Nowotny, Insect inspired view
596 based navigation exploiting temporal information, in V. Vouloutsi, A. Mura, F. Tauber,
597 T. Speck, T. J. Prescott, P. F. M. J. Verschure, editors, Biomimetic and Biohybrid Systems,
598 204–216, Springer International Publishing, Cham (2020).

599 46. F. Le Möel, A. Wystrach, Opponent processes in visual memories: A model of attraction
600 and repulsion in navigating insects' mushroom bodies, *PLoS computational biology* 16,
601 2, (2020) e1007631.

602 47. X. Sun, S. Yue, M. Mangan, A decentralised neural model explaining optimal integration
603 of navigational strategies in insects, *Elife* 9, (2020) e54026.

604 48. A. Wystrach, C. Buehlmann, S. Schwarz, K. Cheng, P. Graham, Rapid aversive and mem-
605 ory trace learning during route navigation in desert ants, *Current Biology* 30, 10, (2020)
606 1927–1933.

607 49. C. A. Freas, A. Wystrach, S. Schwarz, M. L. Spetch, Aversive view memories and risk
608 perception in navigating ants, *Scientific reports* 12, 1, (2022) 1–15.

50. C. Posch, T. Serrano-Gotarredona, B. Linares-Barranco, T. Delbruck, Retinomorphic event-based vision sensors: Bioinspired cameras with spiking output, *Proceedings of the IEEE* 102, 10, (2014) 1470–1484, doi:10.1109/JPROC.2014.2346153.

51. G. Ramos-Traslosheros, M. Henning, M. Silies, Motion detection: cells, circuits and algorithms, *Neuroforum* 24, 2, (2018) A61 – A72, URL https://www.degruyter.com/view/journals/nf/24/2/article-pA61.xml.

52. G. Gallego, T. Delbrück, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. J. Davison, J. Conradt, K. Daniilidis, *et al.*, Event-based vision: A survey, *IEEE transactions on pattern analysis and machine intelligence* 44, 1, (2020) 154–180.

53. T. Fischer, M. Milford, Event-based visual place recognition with ensembles of temporal windows, *IEEE Robotics and Automation Letters* 5, 4, (2020) 6924–6931.

54. D. Kong, Z. Fang, K. Hou, H. Li, J. Jiang, S. Coleman, D. Kerr, Event-vpr: End-to-end weakly supervised deep network architecture for visual place recognition using event-based vision sensor, *IEEE Transactions on Instrumentation and Measurement* 71, (2022) 1–18.

55. M. Milford, H. Kim, S. Leutenegger, A. Davison, Towards visual slam with event-based cameras, in The problem of mobile sensors workshop in conjunction with RSS (2015).

56. A. J. Lee, A. Kim, Eventvlad: Visual place recognition with reconstructed edges from event cameras, in 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2247–2252, IEEE (2021).

57. T. Fischer, M. Milford, How many events do you need? event-based visual place recognition using sparse but varying pixels, *IEEE Robotics and Automation Letters* 7, 4, (2022) 12275–12282.

58. S. Hussaini, M. Milford, T. Fischer, Spiking neural networks for visual place recognition via weighted neuronal assignments, *IEEE Robotics and Automation Letters* 7, 2, (2022) 4094–4101.

59. S. B. Furber, D. R. Lester, L. A. Plana, J. D. Garside, E. Painkras, S. Temple, A. D. Brown, Overview of the spinnaker system architecture, *IEEE transactions on computers* 62, 12, (2012) 2454–2467.

60. S. Furber, P. Bogdan, Spinnaker-a spiking neural network architecture (2020).

61. K. K. Cover, B. N. Mathur, Axo-axonic synapses: Diversity in neural circuit function, *Journal of Comparative Neurology* 529, 9, (2021) 2391–2401.

62. H. Wang, V. M. Pickel, Dopamine d2 receptors are present in prefrontal cortical afferents and their targets in patches of the rat caudate-putamen nucleus, *Journal of Comparative Neurology* 442, 4, (2002) 392–404.

63. M. Soiza-Reilly, W. B. Anderson, C. W. Vaughan, K. G. Commons, Presynaptic gating of excitation in the dorsal raphe nucleus by gaba, *Proceedings of the National Academy of Sciences* 110, 39, (2013) 15800–15805.

64. K. Eichler, F. Li, A. Litwin-Kumar, Y. Park, I. Andrade, C. M. Schneider-Mizell, T. Saumweber, A. Huser, C. Eschbach, B. Gerber, *et al.*, The complete connectome of a learning and memory centre in an insect brain, *Nature* 548, 7666, (2017) 175.

65. N. Bielopolski, H. Amin, A. A. Apostolopoulou, E. Rozenfeld, H. Lerner, W. Huetteroth, A. C. Lin, M. Parnas, Inhibitory muscarinic acetylcholine receptors enhance aversive olfactory learning in adult drosophila, *Elife* 8, (2019) e48264.

66. J. E. Manoim, A. M. Davidson, S. Weiss, T. Hige, M. Parnas, Lateral axonal modulation is required for stimulus-specific olfactory conditioning in drosophila, *Current Biology* 32, 20, (2022) 4438–4450.

67. M. Winding, B. D. Pedigo, C. L. Barnes, H. G. Patsolic, Y. Park, T. Kazimiers, A. Fushiki, I. V. Andrade, A. Khandelwal, J. Valdes-Aleman, F. Li, N. Randel, E. Barsotti, A. Correia, R. D. Fetter, V. Hartenstein, C. E. Priebe, J. T. Vogelstein, A. Cardona, M. Zlatic, The connectome of an insect brain, *Science* 379, 6636, (2023) eadd9330, doi:10.1126/science.add9330, URL https://www.science.org/doi/abs/10.1126/science.add9330.

68. X. Wang, Q.-Q. Sun, Characterization of axo-axonic synapses in the piriform cortex of mus musculus, *Journal of Comparative Neurology* 520, 4, (2012) 832–847.

69. P. Somogyi, T. Freund, A. Cowey, The axo-axonic interneuron in the cerebral cortex of the rat, cat and monkey, *Neuroscience* 7, 11, (1982) 2577–2607.

70. J. Szabadics, C. Varga, G. Molnár, S. Oláh, P. Barzó, G. Tamás, Excitatory effect of gabaergic axo-axonic cells in cortical microcircuits, *Science* 311, 5758, (2006) 233–235, doi:10.1126/science.1121325, https://science.sciencemag.org/content/311/5758/233.full.pdf, URL https://science.sciencemag.org/content/311/5758/233.

71. A. Pan-Vazquez, W. Wefelmeyer, V. G. Sabater, G. Neves, J. Burrone, Activity-dependent plasticity of axo-axonic synapses at the axon initial segment, *Neuron* 106, 2, (2020) 265–276.

72. M. Stimberg, R. Brette, D. F. Goodman, Brian 2, an intuitive and efficient neural simulator, *Elife* 8, (2019) e47314.

73. O. Rhodes, P. A. Bogdan, C. Brenninkmeijer, S. Davidson, D. Fellows, A. Gait, D. R. Lester, M. Mikaitis, L. A. Plana, A. G. Rowley, *et al.*, spynnaker: a software package for running pynn simulations on spinnaker, *Frontiers in neuroscience* 12, (2018) 816.

74. Y.-H. Liu, X.-J. Wang, Spike-frequency adaptation of a generalized leaky integrate-and-fire model neuron, *Journal of computational neuroscience* 10, 1, (2001) 25–45.

75. J. Kropf, W. Rössler, In-situ recording of ionic currents in projection neurons and kenyon cells in the olfactory pathway of the honeybee, *PloS one* 13, 1.

76. M. Tabuchi, S. Inoue, R. Kanzaki, K. Nakatani, Whole-cell recording from kenyon cells in silkmoths, *Neuroscience letters* 528, 1, (2012) 61–66.

77. I. Ito, R. C.-y. Ong, B. Raman, M. Stopfer, Sparse odor representation and olfactory learning, *Nature neuroscience* 11, 10, (2008) 1177.

78. J. Perez-Orive, O. Mazor, G. C. Turner, S. Cassenaer, R. I. Wilson, G. Laurent, Oscillations and sparsening of odor representations in the mushroom body, *Science* 297, 5580, (2002) 359–365.

79. O. A. Hafez, B. Escribano, R. L. Ziegler, J. J. Hirtz, E. Niebur, J. Pielage, The cellular architecture of memory modules in drosophila supports stochastic input integration, *bioRxiv* 2020–12.

80. S. Davies, C. Patterson, F. Galluppi, A. Rast, D. Lester, S. B. Furber, Interfacing real-time spiking i/o with the spinnaker neuromimetic architecture, in Proceedings of the 17th International Conference on Neural Information Processing: Australian Journal of Intelligent Information Processing Systems, 7–11 (2010).

81. L. A. Plana, J. Garside, J. Heathcote, J. Pepper, S. Temple, S. Davidson, M. Luján, S. Furber, spinnlink: Fpga-based interconnect for the million-core spinnaker system, *IEEE Access* 8, (2020) 84918–84928.

82. A. Wystrach, M. Mangan, B. Webb, Optimal cue integration in ants, *Proceedings of the Royal Society B: Biological Sciences* 282, doi:10.1098/rspb.2015.1484.

83. B. Talbot, S. Garg, M. Milford, Openseqslam2. 0: An open source toolbox for visual place recognition under changing conditions, in 2018 IEEE/RSJ international conference on intelligent robots and systems (IROS), 7758–7765, IEEE (2018).

84. C. Groh, W. Rössler, Analysis of synaptic microcircuits in the mushroom bodies of the honeybee, *Insects* 11, 1, (2020) 43.

85. T. Schoepe, E. Janotte, M. B. Milde, O. J. Bertrand, M. Egelhaaf, E. Chicca, Finding the gap: Neuromorphic motion vision in cluttered environments, *arXiv preprint arXiv:2102.08417* .

86. G. Haessig, A. Cassidy, R. Alvarez, R. Benosman, G. Orchard, Spiking optical flow for event-based sensors using ibm's truenorth neurosynaptic system, *IEEE Transactions on Biomedical Circuits and Systems* 12, 4, (2018) 860–870, doi:10.1109/TBCAS.2018.2834558.

87. K. Chaney, A. Panagopoulou, C. Lee, K. Roy, K. Daniilidis, Self-supervised optical flow with spiking neural networks and event based cameras, in 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 5892–5899 (2021), doi:10.1109/IROS51168.2021.9635975.

88. F. Paredes-Vallés, K. Y. W. Scheper, G. C. H. E. de Croon, Unsupervised learning of a hierarchical spiking neural network for optical flow estimation: From events to global

motion perception, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42, 8, (2020) 2051–2064, doi:10.1109/TPAMI.2019.2903179.

89. J. K. N. Verheyen, J. Dupeyroux, G. C. H. E. d. Croon, A novel multi-vision sensor dataset for insect-inspired outdoor autonomous navigation, 279–291, Springer-Verlag, Berlin, Heidelberg (2022), doi:10.1007/978-3-031-20470-8_28, URL https://doi.org/10.1007/978-3-031-20470-8_28.

90. S. Hausler, A. Jacobson, M. Milford, Multi-process fusion: Visual place recognition using multiple image processing methods, *IEEE Robotics and Automation Letters* 4, 2, (2019) 1924–1931, doi:10.1109/LRA.2019.2898427.

91. T. Fischer, M. Milford, Event-based visual place recognition with ensembles of temporal windows, *IEEE Robotics and Automation Letters* 5, 4, (2020) 6924–6931, doi:10.1109/LRA.2020.3025505.

92. W. F. Tobin, R. I. Wilson, W.-C. A. Lee, Wiring variations that enable and constrain neural computation in a sensory microcircuit, *Elife* 6, (2017) e24838.

93. C. L. Barnes, D. Bonnéry, A. Cardona, Synaptic counts approximate synaptic contact area in drosophila, *PloS one* 17, 4, (2022) e0266064.

94. S. Holler, G. Köstinger, K. A. Martin, G. F. Schuhknecht, K. J. Stratford, Structure and function of a neocortical synapse, *Nature* 591, 7848, (2021) 111–116.

95. Z. Zheng, F. Li, C. Fisher, I. J. Ali, N. Sharifi, S. Calle-Schuler, J. Hsu, N. Masoodpanah, L. Kmecova, T. Kazimiers, *et al.*, Structured sampling of olfactory input by the fly mushroom body, *Current Biology* 32, 15, (2022) 3334–3349.

96. N. A. Elkahlah, J. A. Rogow, M. Ahmed, E. J. Clowney, Presynaptic developmental plasticity allows robust sparse wiring of the *Drosophila* mushroom body, *eLife*

9, (2020) e52278, doi:10.7554/eLife.52278, URL https://doi.org/10.7554/eLife.52278.

97. D. Zavitz, E. A. Amematsro, A. Borisyuk, S. J. Caron, Connectivity patterns shape sensory representation in a cerebellum-like network, *bioRxiv* .

98. A. Kodzhabashev, M. Mangan, Route following without scanning, in S. P. Wilson, P. F. Verschure, A. Mura, T. J. Prescott, editors, Biomimetic and Biohybrid Systems, 199–210, Springer International Publishing, Cham (2015).

99. (2021), URL https://www.turtlebot.com/.

100. C. Brandli, L. Muller, T. Delbruck, Real-time, high-speed video decompression using a frame-and event-based davis sensor, in 2014 IEEE International Symposium on Circuits and Systems (ISCAS), 686–689, IEEE (2014).

101. E. Stromatias, F. Galluppi, C. Patterson, S. Furber, Power analysis of large-scale, real-time neural networks on spinnaker, in The 2013 International Joint Conference on Neural Networks (IJCNN), 1–8 (2013), doi:10.1109/IJCNN.2013.6706927.

102. M. F. Land, Visual acuity in insects, *Annual Review of Entomology* 42, 1, (1997) 147–177, doi:10.1146/annurev.ento.42.1.147, pMID: 15012311, https://doi.org/10.1146/annurev.ento.42.1.147, URL https://doi.org/10.1146/annurev.ento.42.1.147.

103. J. Howard, A. Dubs, R. Payne, The dynamics of phototransduction in insects, *Journal of Comparative Physiology A* 154, 5, (1984) 707–718.

104. A. Borst, Drosophila's view on insect vision, *Current Biology* 19, 1, (2009) R36–R47, doi: https://doi.org/10.1016/j.cub.2008.11.001, URL https://www.sciencedirect.com/science/article/pii/S0960982208014322.

105. D. O'Carroll, N. Bidweii, S. Laughlin, E. Warrant, Insect motion detectors matched to visual ecology, *Nature* 382, 6586, (1996) 63–66.

106. A. P. Davison, D. Brüderle, J. M. Eppler, J. Kremkow, E. Muller, D. Pecevski, L. Perrinet, P. Yger, Pynn: a common interface for neuronal network simulators, *Frontiers in neuroinformatics* 2, (2009) 11.

107. N. A. Elkahlah, J. A. Rogow, M. Ahmed, E. J. Clowney, Presynaptic developmental plasticity allows robust sparse wiring of the *Drosophila* mushroom body, *eLife* 9, (2020) e52278, doi:10.7554/eLife.52278, URL https://doi.org/10.7554/eLife.52278.

108. S. Song, K. D. Miller, L. F. Abbott, Competitive hebbian learning through spike-timing-dependent synaptic plasticity, *Nature neuroscience* 3, 9, (2000) 919–926.

109. A. Morrison, M. Diesmann, W. Gerstner, Phenomenological models of synaptic plasticity based on spike timing, *Biological cybernetics* 98, 6, (2008) 459–478.

110. M. Mikaitis, G. Pineda García, J. C. Knight, S. B. Furber, Neuromodulated synaptic plasticity on the spinnaker neuromorphic system, *Frontiers in neuroscience* 12, (2018) 105.

111. S. Davies, F. Galluppi, A. Rast, S. Furber, A forecast-based stdp rule suitable for neuromorphic implementation, *Neural Networks* 32, (2012) 3–14, doi:https://doi.org/10.1016/j.neunet.2012.02.018, selected Papers from IJCNN 2011, URL https://www.sciencedirect.com/science/article/pii/S0893608012000470.

112. S. Skrede, Nordland dataset (2013), URL https://bit.ly/2QVBOym.

113. H. Rebecq, R. Ranftl, V. Koltun, D. Scaramuzza, High speed and high dynamic range video with an event camera, *IEEE transactions on pattern analysis and machine intelligence* 43, 6, (2019) 1964–1980.

6 ACKNOWLEDGEMENTS

We thank Simon Davidson (University of Manchester), Luis Plana Cabrera (University of Manchester), Kevin Stratford (University of Edinburgh), Andrew Rowley (University of Manchester) and Thorben Schoepe (University of Groningen) for their support in SpiNNaker software and hardware. We also thank Jörg Conradt (KTH Royal Institute of Technology) and Juan Pablo Romero Bermudez (KTH Royal Institute of Technology) for their support in using their SpiNNaker-DVS interface board. **Funding:** China Scholarship Council (201801060165) **Author contributions:** Conceptualization: B.W., L.Z.; Methodology: L.Z., B.W., M.M.; Investigation: L.Z.; Visualization: L.Z.; Funding acquisition: L.Z.; Project administration: L.Z., B.W.; Supervision: B.W., M.M.; Writing – original draft: L.Z., B.W.; Writing – review and editing: L.Z., B.W., M.M.; **Competing interests:** Authors declare that they have no competing interests. **Data and materials availability:** All data are available in the main text or the supplementary materials.

7 SUPPLEMENTARY MATERIALS

1. Views in the Nordland dataset (*112*) and our dataset.

2. Calculate familiarity from SeqSLAM using different metrics.

3. Unbalanced vegetation distribution causes AUC to drop under 0.5 in some cases.

4. Perfect memory after normalisation.

5. SeqSLAM on events constructed frames.

6.PN-KC and KC-KC connection matrix.

7. Adaptive current buffers PN membrane potential and generalises the network activity.

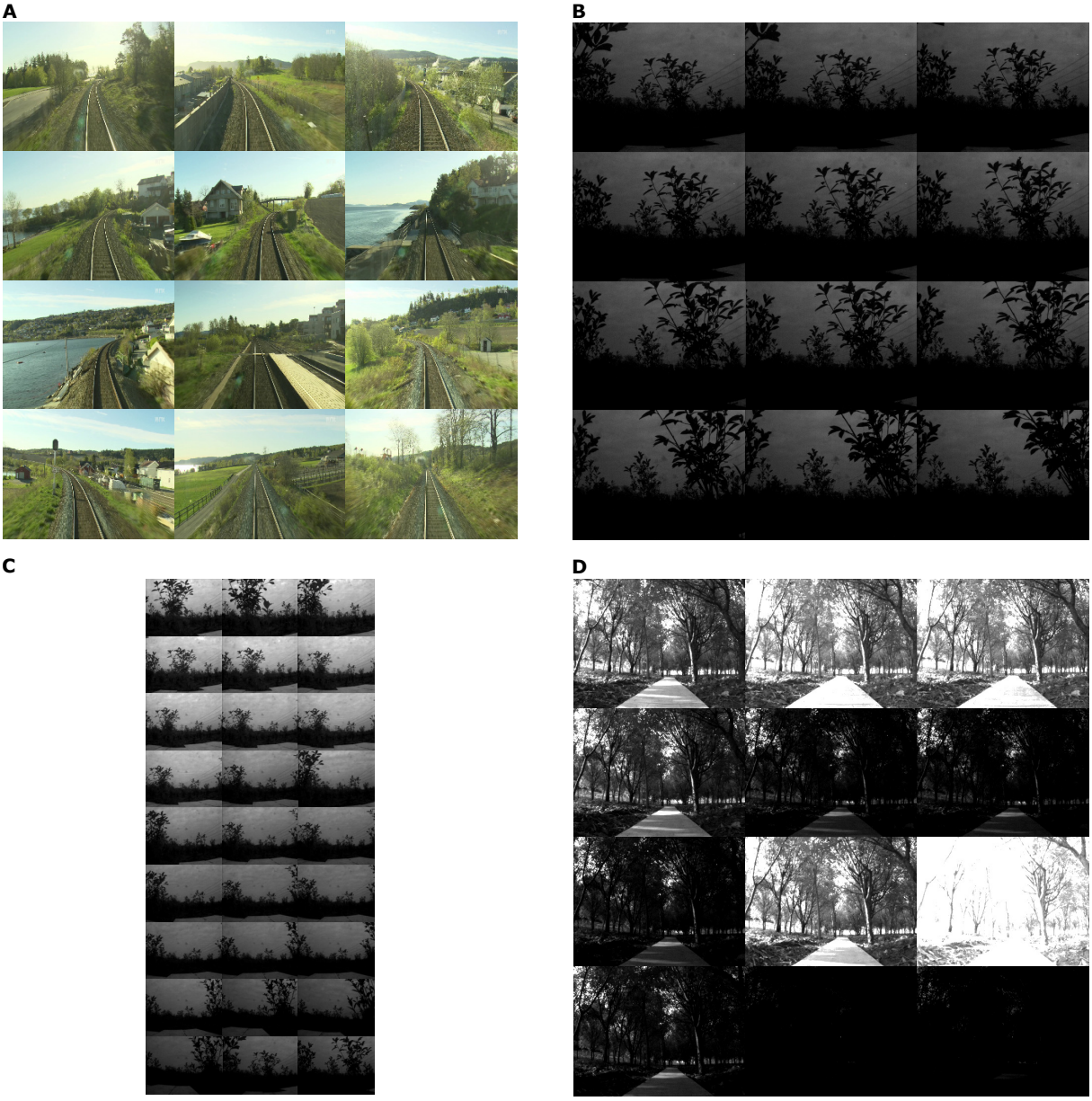

Figure 8: Views in the Nordland dataset (*112*) and our dataset. (**A**) The visual appearance change over every 50 frames in the Nordland dataset. The Nordland dataset was recorded from a camera mounted on a train running over 729km. The recording was done in four different seasons and the video in each season is about 10 hours long.(**B**) The high self-similarity views from our outdoor test environment in a tree nursery. The video recorded from a 6-meter route is about 30 seconds long. Here we show the view change over every 30 frames. (**C**) One frame from each of the short route recordings in our offset test showing the lighting change over time. (**D**) Lighting change when the robot was running in and out of shadows under the trees.

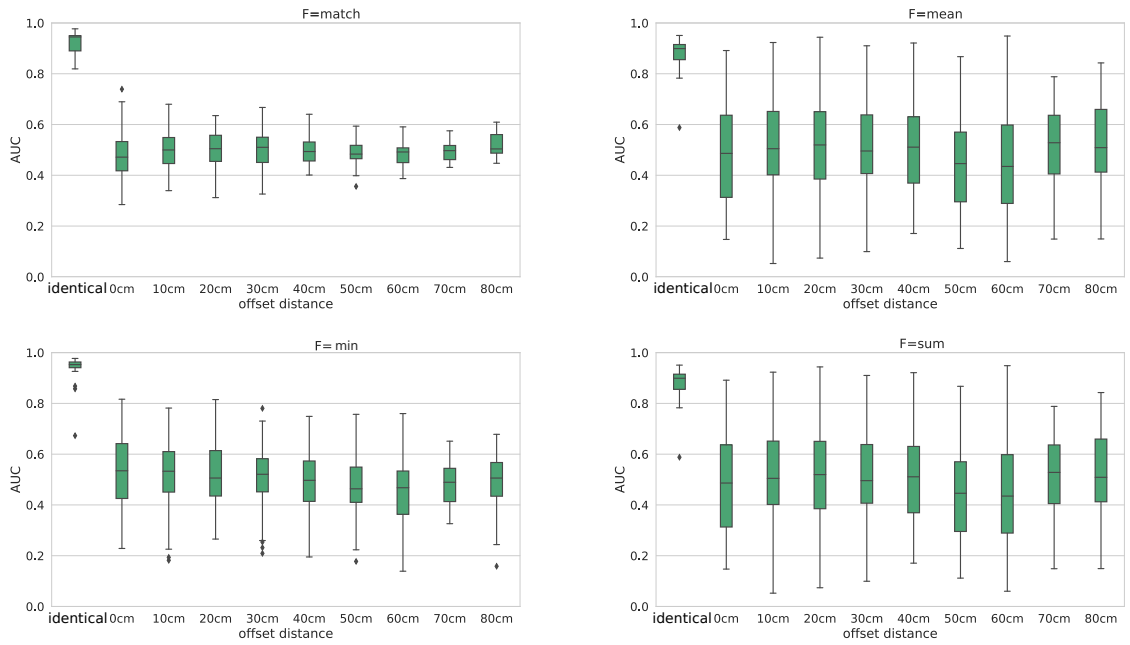

Figure 9: Calculate familiarity from SeqSLAM using different metrics. In the original SeqSLAM, for each query image Qi, the matched reference image Rj will be localised and the match score for Qi and Rj was calculated (as in figure 7). In our benchmark tests, after getting the matching score S in figure7 (D), we here tested using (a) match score (which is the minimal S normalised by second minimal S), (b) mean of S, (c) minimal S and (d) sum of S as the familiarity. For all the metrics, as long as the query and reference are not identical image sets, SeqSLAM fails to detect the familiarity, showing as AUC distributed around 0.5, which means the recognition performance is no better than a random guess.

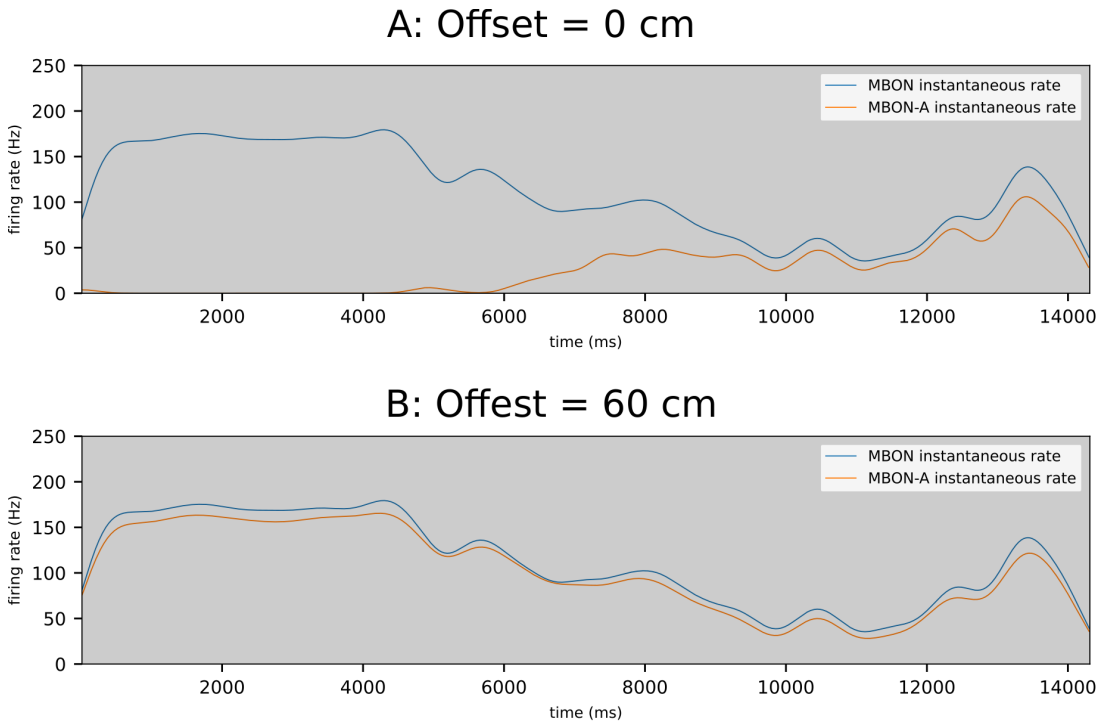

Figure 10: Unbalanced vegetation distribution causes AUC to drop under 0.5 in some cases. MBON instantaneous rate (blue) is the output without learning. MBON-A instantaneous rate (orange) is the MBON activity that takes KC-KC inhibition into account. In some routes, the camera sees more vegetation in the first half and less in the second half. In our offset tests, we learn the first half of the route and test the whole route, which will ideally get a low-high MBON output pattern throughout the route. That is how the orange MBON-A instantaneous rate drops in the first half in A. In B, we learned a less relevant route and the learning can barely change the activity on this unbalanced route. When varying the threshold on the MBON-A rate for calculating AUC, this high-low pattern leads to high false positives and false negatives so that AUC drops under 0.5.

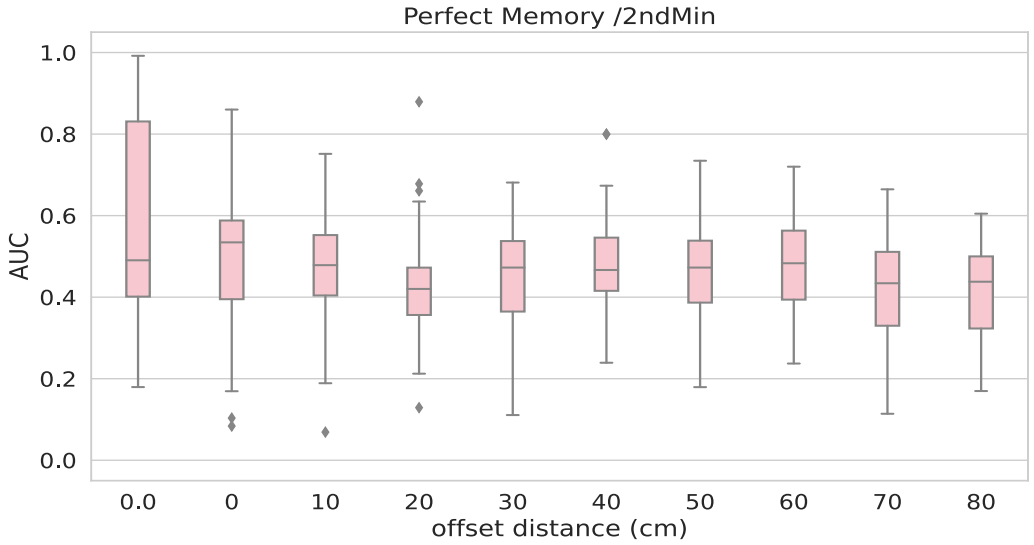

Figure 11: Perfect memory after normalisation. The PM results in figure 6 were selecting the least different reference image as a match for a given query image. The familiarity for this given query image was directly using the sum of pixel-to-pixel differences of the two matched images. Similar to the adaptions we made in SeqSLAM benchmark tests, we also used the second minimal value (not neighbouring to the global minimum) to normalise this global minimum. Before normalisation, the results in figure 6 show PM can detect routes within 10 cm offset as familiar. However, on these familiar routes, the high self-similarity of scenes resulted in many other 'close matches'. The second minimum is very close to the global minimum.

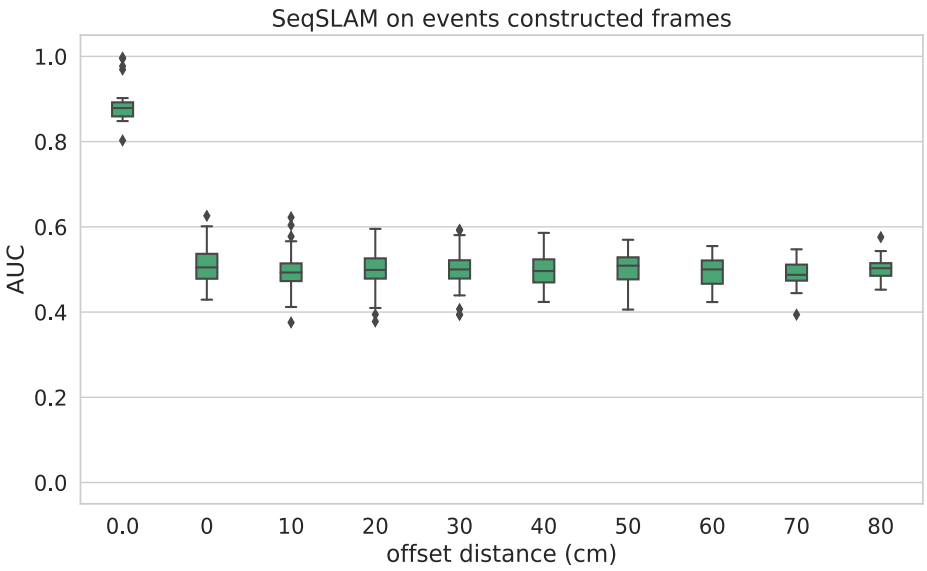

Figure 12: SeqSLAM on events constructed frames. Using a ROS driver, the DAVIS346 camera records both events and conventional frames at the same time. Results running SeqSLAM on conventional frames have been presented in figure 6. We also used the methods from (*91, 113*) to construct frames from events. Then run SeqSLAM in the same way on these event-constructed frames. Again, results show SeqSLAM only detects familiarity when the query and the reference are the same image sets (identical, offset = 0.0 cm).

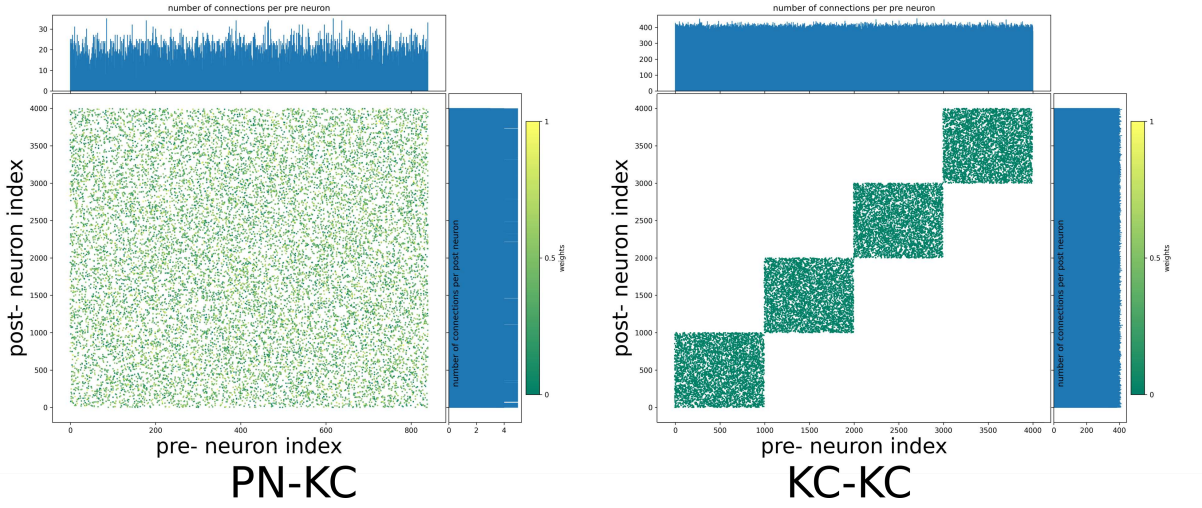

Figure 13: PN-KC and KC-KC connection matrix. In the sparse connection between PN and KC, each KC connects to on average five PNs ($\mathcal{N}(\mu, \sigma^2)$, $\mu = 5$, $\sigma = 1$, and rounded to the nearest integer). In KC-KC axo-axonic connection, KCs are divided into four 1000 KC subgroups. In each subgroup, one single KC randomly connects to other 500 KCs. Before learning, the inhibitory KC-KC weights are all zero.

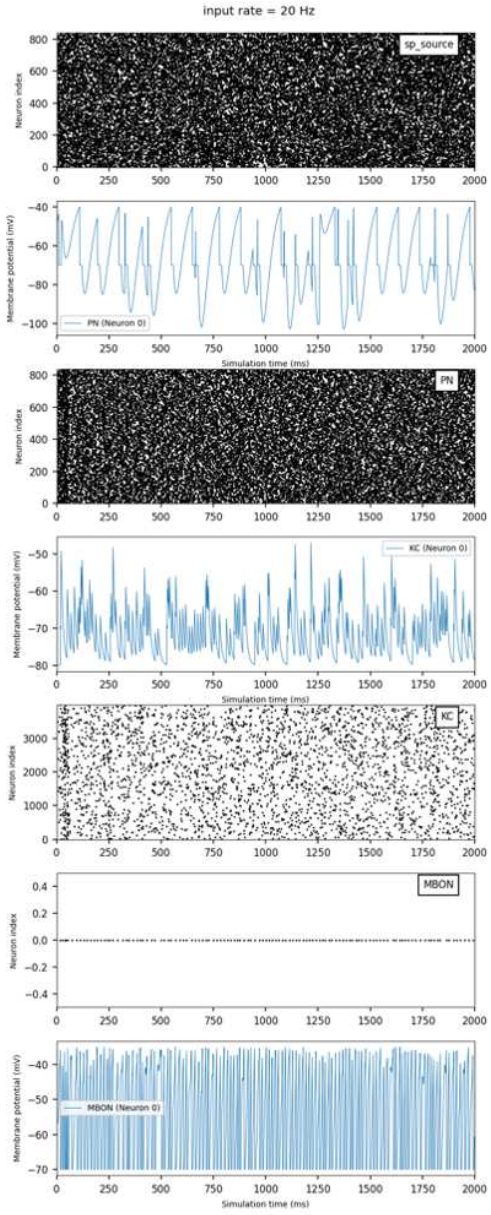

Figure 14: Spikes and membrane potential plot of PN, KC and MBON. The input spikes exhibit a Poisson-like pattern and display a global firing rate of 20 Hz. The reset potential for a PN is -70 mV. Following the reset, the adaptive current lowers the membrane potential to approximately -85 mV. If the PN becomes overly active, the adaptive current increases, causing the membrane potential to decrease further down and reducing the PN firing activity. Note that the KC and MBON models do not incorporate adaptive currents.