BIOMIMETICS

Neuromorphic sequence learning with an event camera on routes through vegetation

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

For many robotics applications, it is desirable to have relatively low-power and efficient onboard solutions. We took 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 and motion over uneven terrain, and the effects of wind on leaves increase the variability of the input. We used a bioinspired event camera on a terrestrial robot to collect visual sequences along routes in natural outdoor environments and applied a neural algorithm for spatiotemporal memory that is closely based on a known neural circuit in the insect brain. We show that 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.

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 behavior 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 that drew directly on recent insights from insect neuroscience.

Even "simple" animals such as ants can excel at navigation under natural outdoor conditions, including through dense vegetation, that still challenge current robots. One problem in such environments is to recognize 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 viewpoint changes (*8*–*12*). In addition, 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 meters 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 recognize familiar locations after arbitrary displacements (solving 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 recognizing which place they are in, provided they are capable of recognizing 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 memories (*25*–*29*) and recognizing scenes while in continuous motion. Insect visual systems are highly sensitive to motion; that is, they fundamentally experience a spatiotemporal 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 improve the ability to localize 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 (SNN) 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 (MBs) (*41*). Recent results showing that lesioning the MB in ants specifically affects performance on tasks requiring learned (but not innate) visual orientation have supported this

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

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 performances in somewhat simplified visual environments with little of the variability that occurs in the real world. Our current model, by contrast, addresses the problem of learning and recognizing, on repeated traversal, the pattern of input spikes produced from visual change detection using an event camera (Fig. 1) on a robot moving distances comparable to those of an ant in a real outdoor environment with a low-level view of nearby natural vegetation.

Event cameras [also called dynamic vision sensors (DVSs)] 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 photoreceptor responses that encode intensity change in natural visual systems (*51*). Compared with 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 combined the address event representation of event-based cameras with SNNs have been developed (*58*), demonstrating their utility for this task. The SpiNNaker neuromorphic computing system used in our work can simulate a large-scale SNN in real time using massively parallel low-power ARM processors incorporating a globally asynchronous locally synchronous system (*59*, *60*).

As described in more detail below, we used this hardware combination (event camera and neuromorphic computing platform) to implement a biologically plausible model for sequence memory. The model includes a previously unexplored form of SNN learning, in which neurons make axo-axonic inhibitory connections that adapt to the spatiotemporal pattern of spikes. Such axonic lateral interaction has been found not only in insect MBs (*61*–*64*) but also in many other biological neural systems (*65*–*67*) where the output of one neuron can effectively shut down the output of other neurons (Fig. 2) (*68*). 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 trajectory.

RESULTS

Mushroom body SNN

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 (PNs) that then fan out, with random connectivity to a much larger number of Kenyon cells (KCs). This produced a relatively unique sparse pattern of KC activity for any visual scene. This pattern was learned, for selected visual scenes along a homeward route, by reducing the excitatory weights from KCs to a single MB output neuron (MBON) when a reward signal is given. Subsequently, a familiar (learned) view produced a low MBON output, which could be used to guide directional decisions.

The current neural model (Fig. 3), first presented in (*40*), has a similar PN layer, this time activated by the event camera output and down-sampled in space and time. There is a similar fan-out to KCs, which again converge to a single MBON that should signal familiarity by reduced activity. However, the learning mechanism was

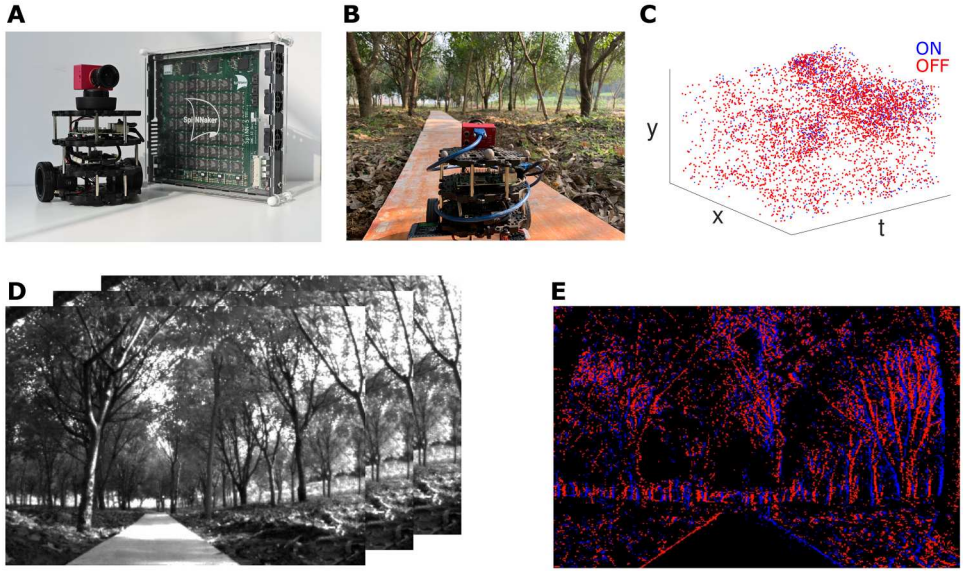

Fig. 1. Robot hardware and event-based video. (**A**) The mobile robot was constructed from a TurtleBot3 Burger with a DAVIS346 event camera. A SpiNN-5 board (*59*) housing 48 SpiNNaker chips was used for simulating our SNN model. (**B**) The model was trained/tested on data recorded from the robot as 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 visualized in the movement "frame" from the event camera. Red and blue colors are polarities of events as shown in (C).

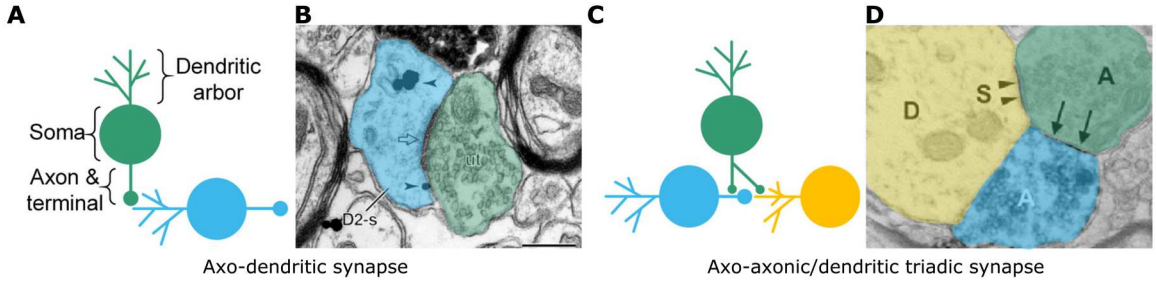

Fig. 2. Axo-axonic connection. Canonical axodendritic synapse and non-canonical axo-axonic synapse depicted through schematic illustrations (**A** and **C**) and electron micrographs of axon fiber slices (**B** and **D**). The axo-axonic triad describes a configuration in which a neuron (green) synapses on both the presynaptic element (blue) and postsynaptic target (yellow) of an axodendritic or -somatic synapse. In this work, we modeled 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 (*68*)() with permission granted (copyright: 2020 Wiley Periodicals LLC). Original micrographs (B) and (D) were from (106) and (107), respectively.

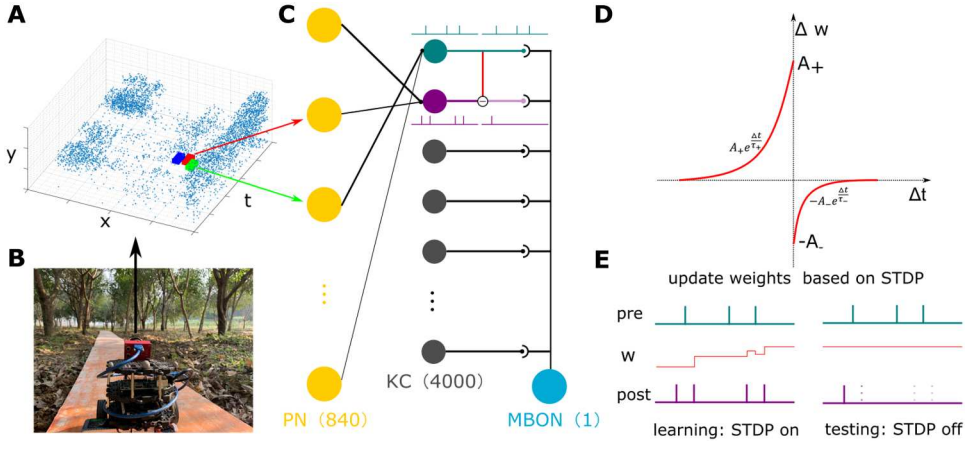

Fig. 3. Mushroom body SNN (MBSNN) encodes spatiotemporal memory as KC-KC inhibition. (**A** and **B**) Event stream input from the camera is firstly down-sampled in both spatial and temporal resolution. Down-sampled megapixels one-to-one map to the PNs in the MB network. (**C**) MB network structure. Eight hundred and forty PNs make sparse random connections to 4000 KCs, which converge on a single MBON. (**D**) Modified 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 postsynaptic 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 because of 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.

altered. Instead of depressing the KC-MBON connection weights, we introduced KC-KC inhibition and adapted the strength of the inhibition using spike timing–dependent plasticity (STDP) (*69*). This allowed the circuit to learn the spatiotemporal pattern of KC spikes produced when the robot moved along a route segment. If the same pattern occured (for example, when retraversing the route), the increased KC-KC inhibition reduced the overall input from the KCs to the MBON, signaling familiarity.

In the work published in (*40*), we simulated the MB model using Python-based simulation tool Brian2 (*70*). With 10,000 KCs, the whole network needed more than 20 min of run time to process a 2-s event stream recording, making real-time closed-loop robot control impossible. Here, to speed up the SNN simulation, we used the neuromorphic computing platform SpiNNaker (*59*, *60*). To run the model on SpiNNaker, it had to be completely reimplemented using the sPyNNaker software package (*71*), and this necessitated substantial 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 was limited by the local memory on each core and time constraints in simulation. Consequently, the network was reduced from 10,000 to 4000 KCs.

An important additional modification was to introduce adaptation to the PN layer. This addressed the variability that 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 parameters for each environment. We used an adaptive leaky integrate-and-fire (LIF) neuron model (*72*) to model PN responses. We found a trade-off in using adaptive neural responses: On one hand, improved generalizability from normalized network activity against varied input in different environments; on the other hand, temporal dynamics caused the network activity to be less deterministic on its current

Table 1. Parameters of neuron models. Names of parameters are consistent with the default variable names in sPyNNaker wherever possible.

Parameter	Neuron			Units	Description
	PN	KC	MBON		
V_{rest}	−60.0	−80.0	−56.7	mV	Resting membrane potential
c_m(= τ_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

input. This is because the adaptation accumulates over time, and 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 on the basis of calculations and electrophysiological data (see Table 1) (*73*–*77*).

Some effort has been made to interface event-based cameras with SpiNNaker boards using either an Ethernet port (with some delay) (*78*) or a field programmable gate array–based SpiNN-link port (in real time) (*79*). However, because of some ongoing software compatibility issues, we have not solved the real-time interfacing problem to close the control loop. Thus, in this paper, we demonstrated 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 visual input in 1-ms 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.

Test environment

Outdoor tests were carried out in three different environments with varied vegetation heights (Fig. 4A). In each environment, we ran the robot on straight routes of 6 m between rows of plants. We note that 4 to 12 m is a typical range over which ant visual route navigation has been observed and models tested. Experiments (*80*) show that ants rely more on path integration for longer distances (>7 m) 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 https://doi.org/10.5281/zenodo.8289547. To ensure smooth running of our wheeled robot, we placed wooden boards on grassy and muddy ground. Note that in the preprocessing stage, the lower part of the frames was chopped to get rid of the wooden board in the camera view. Compared with the indoor environment that was used in our previous work (*40*), outdoor visual surroundings became more cluttered, which meant denser visual input for the same length of robot movement. Apart from the visual environment change, the uneven grassy outdoor ground challenged our wheeled robot. Even running on paved wooden boards, the unavoidable shake of the robot and camera introduced more noise motion and thus more noisy event 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 added potential noise and increased the difficulty of route recognition.

Recognizing routes

Although our MB model supported online learning, because of the insufficient computing power of the robot's onboard computer (and unexpected technical hurdles for real-time integration of the event camera to the SpiNNaker board), we decided to collect the event video and perform all learning and testing offline. In offline learning, we replayed one event video while the plasticity of KC-KC connections was turned on to generate KC-KC inhibition that adapts to the visual motion input. In testing, the KC-KC plasticity was turned off, and the MBON output change due to learned KC-KC connections was 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 Fig. 4, we learned different parts of the route (the first 2 or 3 m or 2 m in the middle) and compared the response of the MBON in this learned segment to the remainder of the route, which had not been learned. Our model was able to recognize 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 generalizes to this simulated change in robot speed.

Recognition after offset

Successful route following requires that the same route traversed again is still recognized. In practice, a robot running on the same trajectory will not do so precisely, and the 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 toward the route because it creates a "valley" for the robot to follow [see (*47*)]. In this offset test, we ran the robot on the same route multiple times and introduced parallel displacements to test the spatial extent of familiarity recognition (Fig. 5). To evaluate for multiple trials and displacements and compare different algorithms, we defined a familiarity threshold and assessed the rate of true and false positives.

Benchmarking against SeqSLAM and perfect memory

SeqSLAM (*35*, *81*) is a well-known VPR algorithm that calculates a locally enhanced difference matrix and selects the best matches only in a short sequence of images. To compare performance, we ran

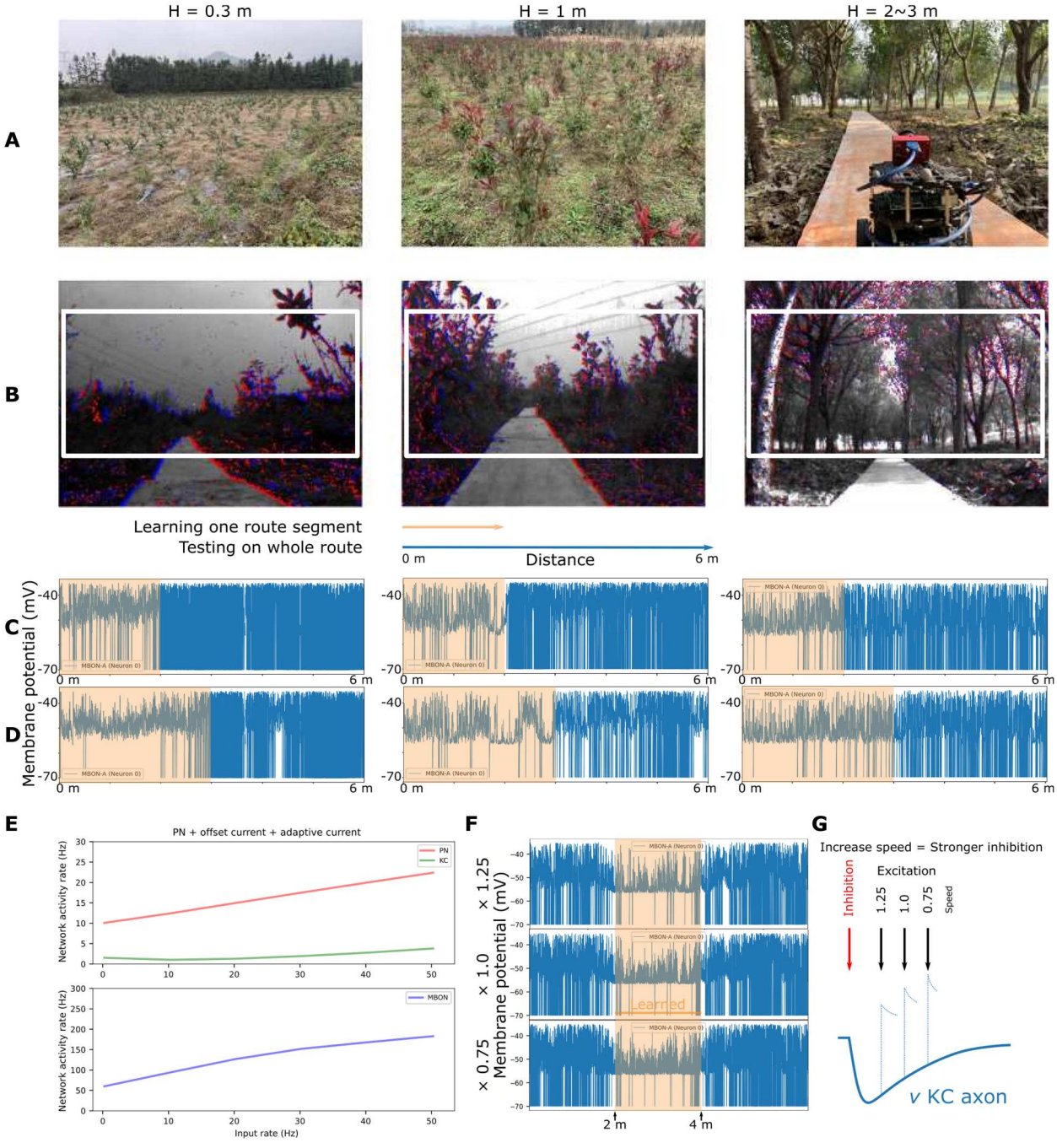

Fig. 4. Route segment learning in three outdoor environments. (**A**) Photographs showing vegetation with varied height (*H*) and planting density. (**B**) In the DVS camera view, red and blue dots are on and off events. Frames were cropped to get rid of some parts of the sky and ground. (**C** and **D**) In each test environment, a visual flow pattern was collected while the robot traveled 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 was replayed to the trained network. The MBON activity (membrane potential) was lower for learned segments (yellow) than for unlearned ones, signaling familiarity. (**E**) The adaptive PN response kept similar generalized activity against input variation so as to maintain robustness in all three environments. (**F** and **G**) By distorting the time axis of the visual input, speed variation can be simulated. After learning the original speed (×1.0 = 0.2 m/s), increased or decreased speed tests also detected the learned segment as familiar. This works because in our two-compartmental model, KC axonic inhibition brings the postsynaptic axon membrane potential down, which cancels the excitation from PNs. Increased speed (compressed time axis) leaves a shorter time for the KC axon membrane potential to climb up to resting potential, causing stronger inhibition.

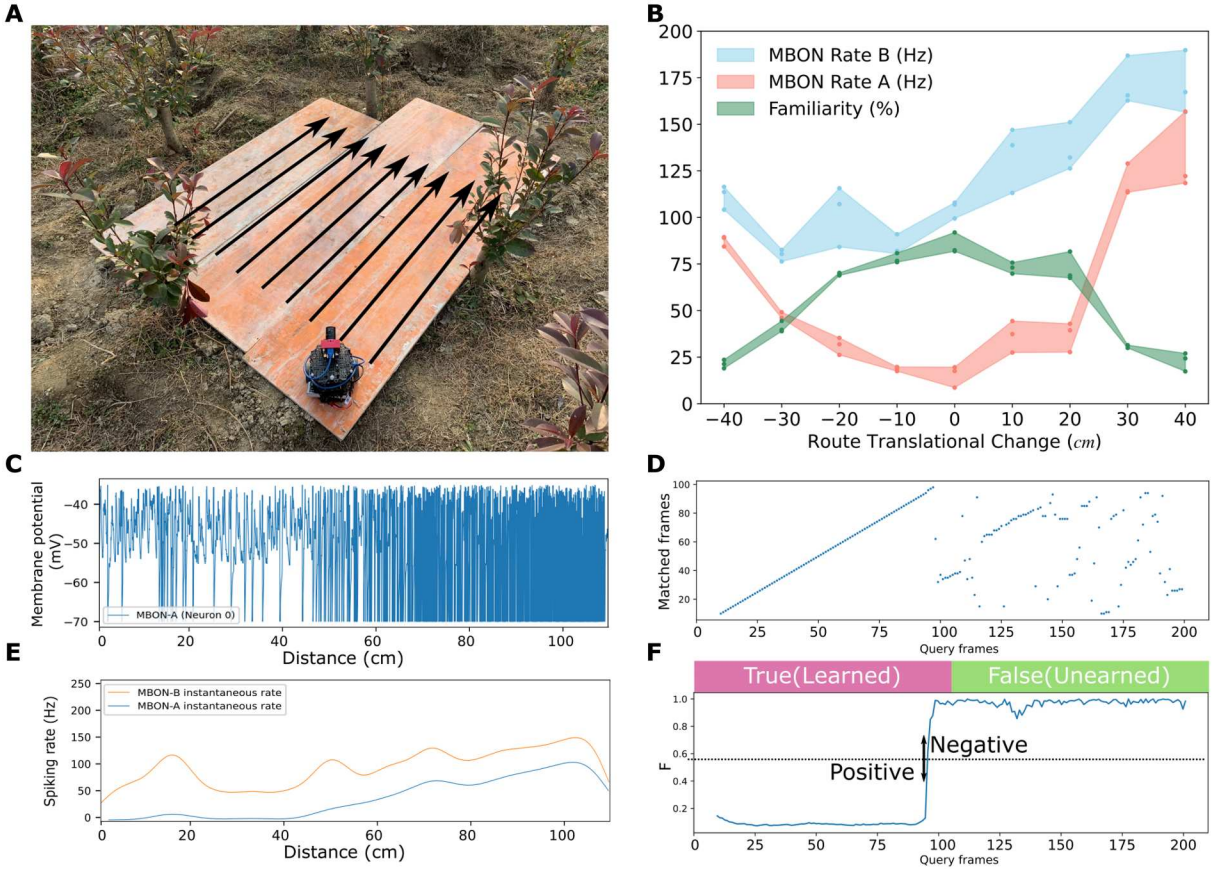

Fig. 5. Offset tests. (**A**) In the $H = 1$ m environment, we recorded parallel offset routes at 0.1-m intervals, each 1 m long. The robot recorded each route three times. (**B**) After learning on the most central route, tests were conducted on all 27 displacement routes. The averaged MBON spiking rates over each run before (MBON Rate B) and after learning (MBON Rate A) are plotted. Familiarity (%) was calculated using (MBON Rate B − MBON Rate A)/MBON Rate B, showing how much the MBON spiking rate dropped because of learning. (C to F) Instead of learning the whole short route, the first half of the route was 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 Materials and Methods) to obtain a similar *F* score. (**F**) After getting the familiarity index for both algorithms, a variable threshold was applied to calculate the TPR and FPR to further plot the ROC curves in Fig. 7.

SeqSLAM on our dataset using standard greyscale images collected simultaneously with event streams (for results using the event data for SeqSLAM, see the 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, whereas our model only identified 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 and only using the match score as a familiarity index. As an alternative comparison, we used the "perfect memory" (PM) 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 visualize the performance of the algorithms. We calculated the area under the curve ROC (AUC-ROC) to assess the performance across a range of conditions and repeated trials.

An AUC-ROC score of 0.5 means that the recognition is no better than a random guess. Although SeqSLAM can perform well in recognizing a previously traversed location regardless of lighting or weather change (*35*, *81*), in our test, it only gave good results when the reference and query data were the same, that is, the identical sequence was replayed. Input changes caused by retraversal of the same route degraded the performance (AUC-ROC of SeqSLAM was already near 0.5 with 0-cm offset), and small spatially offset routes (10 cm or more) were unrecognized by SeqSLAM. The fall in recognition score was more gradual for MBSNN. After a 30-cm offset, none of the algorithms could recognize the route as learned or familiar. The drop of AUC-ROC to below 0.5 for large offsets (70 or 80 cm) in MBSNN was caused by the unbalanced distribution of vegetation (see the Supplementary Materials for more details). An additional minor drawback in SeqSLAM is that the very beginning and the very end of the video could not be put into the reference image set; in Fig. 5D, when the local sequence search is

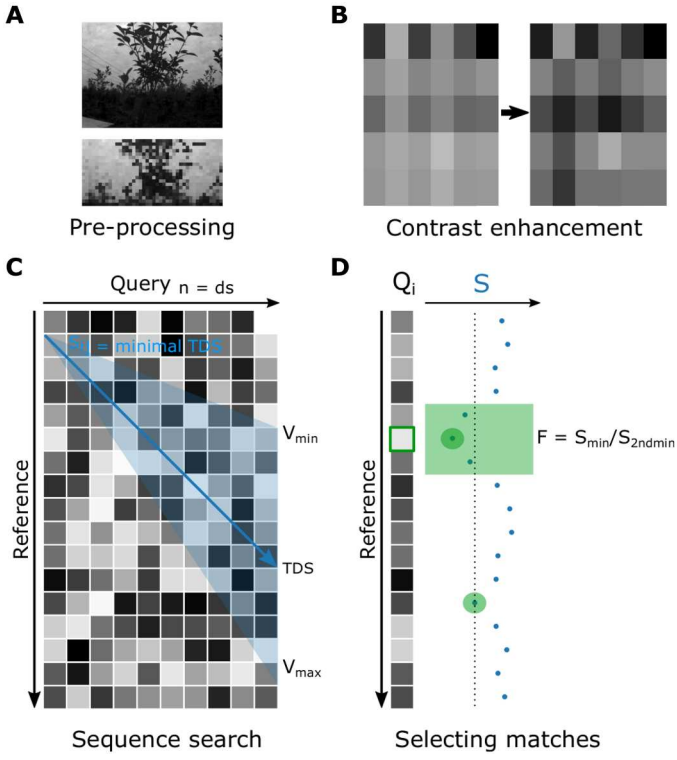

Fig. 6. Adapting SeqSLAM for our benchmark test. (**A**) In preprocessing, each image was cropped and down-sampled. (**B**) Enhance contrast on the absolute difference matrix between the query images and the reference images to facilitate sequence search. Darker color means a bigger difference. (**C**) For the *i*th query image, a local difference matrix is selected from the contrast-enhanced difference matrix, containing d_s query images and all reference images. TDS on each local sequence (blue line) between the range of V_{min} and V_{max} was calculated, and the minimal TDS (as from the best local sequence) was set as the match score (S_{ij}) for the *i*th query image on the j_{th} reference image. (**D**) For the *i*th query image Q_i, the best-matched reference image is where the minimal match score *S* stands and its match index *F* is normalized by the second minimal *S*. The second minimal *S* needs to be out of a selection window to make sure that the global minimum and second minimum are not neighboring. A global threshold F_{th} can be applied to filter out weaker matches. In this work, we ignored the localization of the reference image and only used the score *F* indicating how familiar Q_i is. Figure is based on (*35*).

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 and is thus better suited to short route detection.

DISCUSSION

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

The model used an SNN with 4841 neurons, which we demonstrate can run in real time on neuromorphic hardware. It is based on recent 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 (*61*). 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 (*65*–*67*), 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 learning spatiotemporal patterns such as those produced by dynamic visual input. The result is that familiar (previously experienced) visual sequences will produce a low output.

To 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 because the network has so much inhibition that any input pattern produces a low response, and again the pattern used for training cannot 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 generalized the network spiking rate in various environments, providing better robustness and generalizability. The size of our MB model (only 4000 KCs) is smaller than those of insect navigators [for example, the honeybee MB has about 368,000 KCs (*82*)]. 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 preprocessing might reduce the variability due to lighting. In insects, there are multiple neural layers of visual processing. Modeling these and other properties of the visual pathway is one of the obvious next steps for this work (*83*–*86*). In short, the memory capacity of the network can be extended by tuning the learning parameters so that the weights grow slower, increasing the number of KCs, selectively learning only useful features, and/or 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 omnidirectional frames (*87*).

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 an outdoor dataset from an event camera (which also recorded

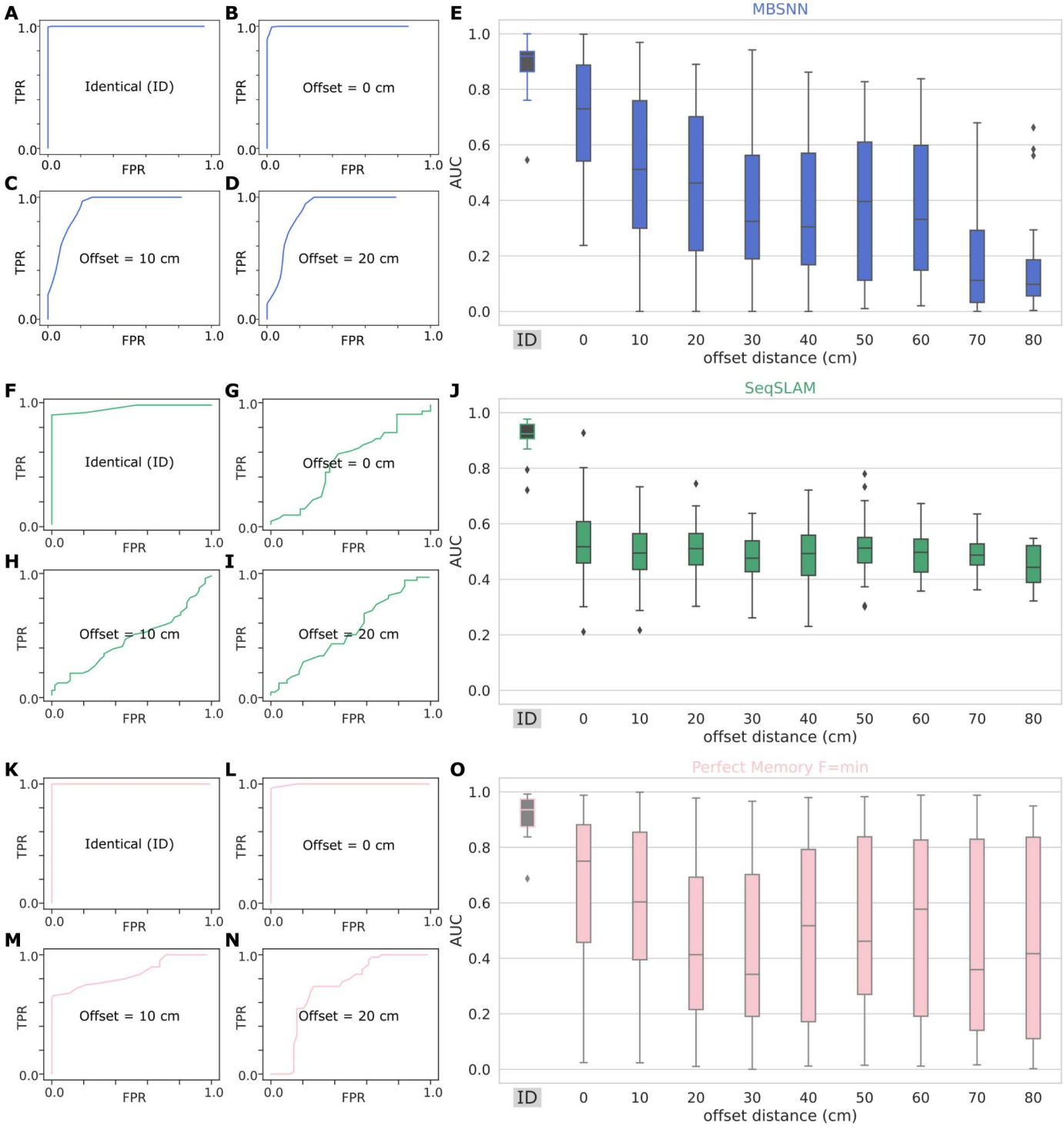

Fig. 7. Offset routes recognition using MBSNN, SeqSLAM, and PM. (**A** to **D**, **F** to **I**, and **K** to **N**) ROC curves plotting TPR against FPR from MBSNN, SeqSLAM, and PM, respectively. (**E**, **J**, and **O**) The AUC-ROC plots for all traverses, using MBSNN, SeqSLAM, and PM, respectively. When the query and the reference images were identical [the same recording from the same route, as in (A), (F), and (K), and box plots labeled "ID"], all three approaches easily classified the query input as familiar. SeqSLAM was not able to recognize 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 offset. PM failed to detect familiarity in all cases after normalization, and this was again caused by the high self-similarity of the scenes (see figs. S1, S2, and S4 for analysis). Our MBSNN model showed a gradual familiarity drop with spatial offset. Note that the drop to below 0.5 at the highest displacements is due to the unbalanced distribution of vegetation along the routes (see fig. S3). Boxplots show median and quartiles, and whiskers show the max and minimum, with * for outlying data points.

conventional video) on a mobile robot traveling 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, SeqSLAM in these conditions was only able to recognize the route if the same video recording was used for training and testing. One reason for the poor generalization might be the relatively high rate of image capture of our short outdoor routes. The datasets used in the original SeqSLAM papers were collected from either a car or a train over long-distance travel, moving relatively fast compared with the frame rate. The image frames were consequently rather distinct in appearance, whereas in our short routes, the similarity between frames may have been 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 use 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 do not need the localization of each query image; rather, it is necessary for our route-following robot to perform a fine-scale familiarity assessment so that the robot agent can stay in the valley of familiarity. However, one way to expand the network toward VPR applications would be to introduce multiple MBONs (as known to exist in insect MB) that could represent distinct sections, or salient locations, on the route. Alternatively, the low-power and instant recognition model presented could be used to guide a robot between sparse waypoints provided by VPR algorithms or GPS (*53*, *88*).

The first step of our future work is exploring the connectivity and pattern between PN-KC and KC-KC. Both biological findings and modeling work have argued that the connection between neuron groups is not globally random but rather functionally patterned (*89*–*94*). Here, we constructed 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 that can use 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 refind the route. Our results suggested that getting closer to the route should provide some increase in familiarity, so an oscillation strategy as suggested in (*95*) could be effective. Also, the event camera provides continuous motion sensing that 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 behavior using neuromorphic hardware on a robot that directly interacts with a world of comparable complexity to that of the ant.

MATERIALS AND METHODS

Objective and study design

The objectives of our study were first to evaluate the biological hypothesis that MB can encode spatial-temporal memory into its KC-KC axo-axonic connections; second, to implement the MB model on a hardware robot system with a neuromorphic computer SpiNNaker and to learn outdoor routes sensed by an event camera; and last, to test the learning of sequential visual information and benchmark with SeqSLAM and PM.

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 (Fig. 1A) is a Turtlebot3 Burger (*96*) robot platform augmented with the visual sensor DAVIS346 (*97*). The Turtlebot has a single-board computer (SBC)—Raspberry Pi 3 Model B+ (1.4-GHz 64-bit quad-core processor) installed with Ubuntu Mate 18.04. Related Robot Operating System (ROS) packages control the robot movement via an OpenCR board (STM32F746ZGT6 / 32-bit ARM Cortex-M7). Testing revealed that the SBC had insufficient computational power to run our model online. Rather, camera data were 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 SNN model on a neuromorphic computer, SpiNNaker (*59*, *60*). The SpiNNaker neuromorphic computer uses 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 and the neural activity. During simulations, SpiNNaker cores were predominantly idle, becoming active only when interrupted by incoming spikes to update neural activity. Power consumption per chip ranged from 0.25 to 0.9 W (*98*), leading to the SpiNNaker 5 board (48 chips, 864 cores) consuming 12 to 43 W during real-time simulations, which is hundreds of times faster than our previous Brian2 (*70*) implementation (*40*). 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 preprocessing of the raw input was 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° and 5° of their visual field (*99*); their temporal resolution, measured by photoreceptor flickering response, is usually no higher than 300 Hz (3 ms) (*100*–*102*). Therefore, before feeding the camera output to the neural network, the event flow was resampled. The whole image frame was down-sampled by 8 pixels by 8 pixels. In each down-sampled megapixel, an input event would trigger an event counter lasting 1 ms. In this 1 ms, when the number of events happening in the mega pixels outnumbered a noise threshold (three events), the megapixel would output a valid spike to the spatially mapped PN. After this event counter, the next input event would trigger the next counter on this megapixel. After resampling, the effective spatial acuity was about 5°, and the temporal resolution was 1 ms. Note that there are additional processing steps in the insect visual motion

Table 2. Parameters of synapses. Names of parameters are consistent with the default variable names in sPyNNaker wherever possible.

Parameter	Weight	Units	Description
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
STDPτ_+	1.25	ms	Exponential decay factor of potentiation weight
STDPτ_-	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

pathway (*51*), but we have not included these steps in our modeling so far.

Mushroom body network and implementation

Neuron models

When implementing our model, we used the software package sPyNNaker (*71*) to simulate PyNN (*103*) defined network on SpiNNaker hardware. The KC and MBON were modeled as standard LIF neurons with fixed threshold and decaying exponential postsynaptic current (named IF_curr_exp in PyNN and sPyNNaker) described by Eqs. 1 to 3

$$\frac{dV}{dt} = -\frac{V - (V_{rest} + R_m I(t))}{\tau_m} \quad (1)$$

Equation 1 models the dynamics of subthreshold 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 generates a spike and then the membrane potential resets

$$\text{if } V > V_{thresh}, V = V_{reset} \quad (2)$$

For the KC and MBON LIF neuron (IF_curr_exp), its synaptic input current I_{syn} is modeled as Eq. 3

$$\frac{dI_{syn}}{dt} = -\frac{I_{syn}}{\tau_{syn}} + \delta(t - t^j) \quad (3)$$

This IF_curr_exp model has separate synaptic currents for excitatory and inhibitory synapses with independent time constant 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 (*72*) (named IFCurrExpCa2Adaptive in sPyNNaker). Compared with the KC and MBON model, the PN model has one more Ca^{2+} activated K^+ current (I_{AHP}) that adapts the membrane potential according to its firing activity. At each spike, the adaptive current increases by

$$I_{AHP} = I_{AHP} + \alpha \quad (4)$$

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

$$\frac{dI_{AHP}}{dt} = -\frac{I_{AHP} - I_{offset}}{\tau_{AHP}} \quad (5)$$

We set the parameters for neuron models and synapse models based on biological data found in (*73*–*77*) and the neuron parameters shown in Table 1.

Connections

Each core in the SpiNNaker system updates its neuron states using a fixed simulation time step (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 coherent simulation progression. Thus, all cores in a simulation start synchronized. 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 Eqs. 6 and 7

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

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

For a static synapse construction, the only parameter we set is the w_{ij}, apart from defining the presynaptic and postsynaptic neuron index.

Although the connections between PNs and KCs were modeled 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 (*89*–*91*), and some modeling work further evaluated how functional patterns can influence the performance in learning and classification (*92*–*94*). By changing the ratio between presynaptic olfactory PN and postsynaptic KC, Elkahlah *et al.* (*96*) found that the connection density between PNs and KCs is set by KC: KC claw number does not vary much as PN number changes, whereas PNs change their boutons (terminals of axons) depending on KC number. On the basis of 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 have not 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 five PNs [$n_{PN} \sim \mathcal{N}(\mu, \sigma^2)$, μ = 5, σ = 1, and rounded to the nearest integer], and the input weights to each KC (w_{ij}) are randomly distributed [$w_{ij} \sim \mathcal{N}(\mu, \sigma^2)$, μ = 0.3, σ = 0.1] (see fig. S6). 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 (4000) 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 were all zero before learning.

Therefore, the sparseness of KC-KC connections depends on the number 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.

KC-KC STDP

In practice, we used a two-compartment model for each KC, implemented as two LIF units that 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 (*104*–*107*)

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

In Eq. 8, Δt ($\Delta t = T_{post} - T_{pre}$) is the time difference between the pre- and postsynaptic spike timing. A_+ and A_- are the maximum synaptic modifications, and 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 postsynaptic 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 are shown in Table 2. The weights of all KC-KC connections were initialized to zero, and they were altered by STDP as described in Fig. 3. That is, if one KC fires shortly before another KC, then 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.

SeqSLAM and PM

From the DAVIS346, we can record both frame-based video and event stream. Being consistent with the prepossessing of events, the grayscale frames were resized to the same view as in the preprocessing of MBSNN input. The frames were chopped to get rid of the ground and part of the sky and down-sampled by 8 pixels by 8 pixels. Then, the absolute differences between query images and reference images were calculated to get a difference matrix that was then enhanced (Eq. 9) to facilitate finding local best matches

$$\hat{D}_i = \frac{D_i - \overline{D_l}}{\sigma_l} \quad (9)$$

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

For each pair of query image Q and reference image R, local sequences were searched in space M in which the trajectories travel over ds query images between V_{min} and V_{max} (blue area in Fig. 6C)

$$M = [D^{T-d_s}, D^{T-d_s+1} \ldots D^T] \quad (10)$$

where T is the current time

$$\mathrm{TDS} = \sum_{t=T-d_S}^{T} D_k^t \quad (11)$$

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

$$k = s + V(d_s - T + t) \quad (12)$$

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 was 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 localization of reference image R_j and directly used ($F = M$) as the familiarity score of the query image Q_i.

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, 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 Fig. 6C). We applied the same contrast enhancement (Fig. 6B) and normalization (divided by the second minimum; Fig. 6D) to better compare with SeqSLAM. The results of the PM test before normalization can be seen in Fig. 7, whereas the results after normalization are included in the Supplementary Materials.

Supplementary Materials

This PDF file includes:

Figs. S1 to S7

REFERENCES AND NOTES

1. Y. Sandamirskaya, Rethinking computing hardware for robots. *Sci. Robot.* **7**, eabq3909 (2022).
2. Y. Sandamirskaya, M. Kaboli, J. Conradt, T. Celikel, Neuromorphic computing hardware and neural architectures for robotics. *Sci. Robot.* **7**, eabl8419 (2022).
3. B. Webb, Robots in invertebrate neuroscience. *Nature* **417**, 359–363 (2002).
4. B. Webb, Robots with insect brains. *Science* **368**, 244–245 (2020).
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**, 7609 (2021).
6. G. De Croon, J. Dupeyroux, S. Fuller, J. Marshall, Insect-inspired AI for autonomous robots. *Sci. Robot.* **7**, eabl6334 (2022).
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. *Bioinspir. Biomim.* **18**, 035005 (2023).
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* (IEEE, 2016), pp. 5297–5307.
9. S. Garg, N. Suenderhauf, M. Milford, Lost? Appearance-invariant place recognition for opposite viewpoints using visual semantics, in *Proceedings of Robotics: Science and Systems XIV* (2018).
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 (CVPR),* (IEEE, 2020), pp. 2626–2635.
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**, 118673–118683 (2021).

12. A. Ozdemir, M. Scerri, A. Barron, A. Philippides, M. Mangan, E. Vasilaki, L. Manneschi, Echovpr: Echo state networks for visual place recognition. *IEEE Robot. Autom. Lett.* **7**, 4520–4527 (2022).
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 Robot. Autom. Lett.* **5**, 1835–1842 (2020).
14. B. Ferrarini, M. Waheed, S. Waheed, S. Ehsan, M. Milford, Klaus 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)* (IEEE, 2019), pp. 103–108.
15. S. Lowry, N. Sunderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, M. J. Milford, Visual place recognition: A survey. *IEEE Trans. Robot.* **32**, 1, 19 (2015).
16. C. Masone, B. Caputo, A survey on deep visual place recognition. *IEEE Access* **9**, 19516–19547 (2021).
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**, 97 (2020).
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)* (IEEE, 2017), pp. 3878–3885.
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* (Springer, 2021), pp. 3–15.
20. M. Mangan, B. Webb, Spontaneous formation of multiple routes in individual desert ants (*Cataglyphis velox*). *Behav. Ecol.* **23**, 944–954 (2012).
21. M. Kohler, R. Wehner, Idiosyncratic route-based memories in desert ants, *Melophorus bagoti*: How do they interact with path-integration vectors? *Neurobiol. Learn. Mem.* **83**, 1–12 (2005).
22. L. Haalck, M. Mangan, A. Wystrach, L. Clement, B. Webb, B. Risse, Cater: Combined animal tracking & environment reconstruction. *Sci. Adv.* **9**, eadg2094 (2023).
23. B. Baddeley, P. Graham, A. Philippides, P. Husbands, Holistic visual encoding of ant-like routes: Navigation without waypoints. *Adapt. Behav.* **19**, 3–15 (2011).
24. B. Baddeley, P. Graham, P. Husbands, A. Philippides, A model of ant route navigation driven by scene familiarity. *PLOS Comput. Biol.* **8**, e1002336 (2012).
25. T. S. Collett, M. Collett, Memory use in insect visual navigation. *Nat. Rev. Neurosci.* **3**, 542–552 (2002).
26. M. Collett, L. Chittka, T. Collett, Spatial memory in insect navigation. *Curr. Biol.* **23**, R789–R800 (2013).
27. J. Zeil, Visual homing: An insect perspective. *Curr. Opin. Neurobiol.* **22**, 285–293 (2012).
28. B. Webb, The internal maps of insects. *J. Exp. Biol.* **222**, jeb188094 (2019).
29. R. A. Harris, N. H. de Ibarra, P. Graham, T. S. Collett, Priming of visual route memories. *Nature* **438**, 302 (2005).
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. *Curr. Biol.* **27**, 929–944 (2017).
31. K. Nagel, Motion vision: Pinning down motion computation in an ever-changing circuit. *Curr. Biol.* **31**, R1523–R1525 (2021).
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. *Curr. Biol.* **32**, 3659–3675.e8 (2022).
33. A. Vardy, R. Moller, Biologically plausible visual homing methods based on optical flow techniques. *Connect. Sci.* **17**, 47–89 (2005).
34. L. Dittmar, W. Stürzl, E. Baird, N. Boeddeker, M. Egelhaaf, Goal seeking in honeybees: Matching of optic flow snapshots? *J. Exp. Biol.* **213**, 2913–2923 (2010).
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* (IEEE, 2012), pp. 1643–1649.
36. M. Milford, Vision-based place recognition: How low can you go? *Int. J. Rob. Res.* **32**, 766–789 (2013).
37. M. Chancán, L. Hernandez-Nunez, A. Narendra, A. B. Barron, M. Milford, A hybrid compact neural architecture for visual place recognition. *IEEE Robot. Autom. Lett.* **5**, 993–1000 (2020).
38. P. Graham, M. Mangan, Insect navigation: Do ants live in the now? *J. Exp. Biol.* **218**, 819–823 (2015).
39. S. Schwarz, M. Mangan, B. Webb, A. Wystrach, Route-following ants respond to alterations of the view sequence. *J. Exp. Biol.* **223**, jeb.218701 (2020).
40. L. Zhu, M. Mangan, B. Webb, Spatio-temporal memory for navigation in a mushroom body model, in *Conference on Biomimetic and Biohybrid Systems,* V. Vouloutsi, A. Mura, F. Tauber, T. Speck, T. J. Prescott, P. F. M. J. Verschure, Eds (Springer, Cham, 2020), vol. 12413, pp. 415–426.
41. P. Ardin, F. Peng, M. Mangan, K. Lagogiannis, B. Webb, Using an insect mushroom body circuit to encode route memory in complex natural environments. *PLOS Comput. Biol.* **12**, e1004683 (2016).
42. C. Buehlmann, B. Wozniak, R. Goulard, B. Webb, P. Graham, J. E. Niven, Mushroom bodies are required for learned visual navigation, but not for innate visual behavior, in ants. *Curr. Biol.* **30**, 3438–3443.e2 (2020).
43. J. F. Kamhi, A. B. Barron, A. Narendra, Vertical lobes of the mushroom bodies are essential for view-based navigation in australian myrmecia ants. *Curr. Biol.* **30**, 3432–3437.e3 (2020).
44. S. Heinze, Visual navigation: Ants lose track without mushroom bodies. *Curr. Biol.* **30**, R984–R986 (2020).
45. E. Kagioulis, A. Philippides, P. Graham, J. C. Knight, T. Nowotny, Insect inspired view based navigation exploiting temporal information, in *Biomimetic and Biohybrid Systems*, A. Mura, F. Tauber, T. Speck, T. J. Prescott, P. F. M. J. Verschure, Eds. (Springer International Publishing, 2020), pp. 204–216.
46. F. Le Möel, A. Wystrach, Opponent processes in visual memories: A model of attraction and repulsion in navigating insects' mushroom bodies. *PLOS Comput. Biol.* **16**, e1007631 (2020).
47. X. Sun, S. Yue, M. Mangan, A decentralised neural model explaining optimal integration of navigational strategies in insects. *eLife* **9**, e54026 (2020).
48. A. Wystrach, C. Buehlmann, S. Schwarz, K. Cheng, P. Graham, Rapid aversive and memory trace learning during route navigation in desert ants. *Curr. Biol.* **30**, 1927–1933.e2 (2020).
49. C. A. Freas, A. Wystrach, S. Schwarz, M. L. Spetch, Aversive view memories and risk perception in navigating ants. *Sci. Rep.* **12**, 2899 (2022).
50. C. Posch, T. Serrano-Gotarredona, B. Linares-Barranco, T. Delbruck, Retinomorphic event-based vision sensors: Bioinspired cameras with spiking output. *Proc. IEEE* **102**, 1470–1484 (2014).
51. G. Ramos-Traslosheros, M. Henning, M. Silies, Motion detection: Cells, circuits and algorithms. *Neuroforum* **24**, A61–A72 (2018).
52. G. Gallego, T. Delbrück, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. Davison, J. Conradt, K. Daniilidis, D. Scaramuzza, Event-based vision: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **44**, 154–180 (2020).
53. T. Fischer, M. Milford, Event-based visual place recognition with ensembles of temporal windows. *IEEE Robot. Autom. Lett.* **5**, 6924–6931 (2020).
54. D. Kong, Z. Fang, H. Li, K. Hou, S. Coleman, D. Kerr, Event-vpr: End-to-end weakly supervised deep network architecture for visual place recognition using event-based vision sensor. *IEEE Trans. Instrum. Meas.* **71**, 5011418 (2022).
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. 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)* (IEEE, 2021), pp. 2247–2252.
57. T. Fischer, M. Milford, How many events do you need? event-based visual place recognition using sparse but varying pixels. *IEEE Robot. Autom. Lett.* **7**, 12275–12282 (2022).
58. S. Hussaini, M. Milford, T. Fischer, Spiking neural networks for visual place recognition via weighted neuronal assignments. *IEEE Robot. Autom. Lett.* **7**, 4094–4101 (2022).
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 Trans. Comput.* **62**, 2454–2467 (2012).
60. S. Furber, P. Bogdan, *Spinnaker-A Spiking Neural Network Architecture* (Now Publishers, 2020).
61. K. Eichler, F. Li, A. Litwin-Kumar, Y. Park, I. Andrade, C. M. Schneider-Mizell, T. Saumweber, A. Huser, C. Eschbach, B. Gerber, R. D. Fetter, J. W. Truman, C. E. Priebe, L. F. Abbott, A. S. Thum, M. Zlatic, A. Cardona, The complete connectome of a learning and memory centre in an insect brain. *Nature* **548**, 175–182 (2017).
62. 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**, e48264 (2019).
63. J. E. Manoim, A. M. Davidson, S. Weiss, T. Hige, M. Parnas, Lateral axonal modulation is required for stimulus-specific olfactory conditioning in *Drosophila*. *Curr. Biol.* **32**, 4438–4450.e5 (2022).
64. 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**, eadd9330 (2023).
65. X. Wang, Q.-Q. Sun, Characterization of axo-axonic synapses in the piriform cortex of mus musculus. *J. Comp. Neurol.* **520**, 832–847 (2012).
66. P. Somogyi, T. Freund, A. Cowey, The axo-axonic interneuron in the cerebral cortex of the rat, cat and monkey. *Neuroscience* **7**, 2577–2607 (1982).

67. J. Szabadics, C. Varga, G. Molnár, S. Oláh, P. Barzó, G. Tamás, Excitatory effect of gabaergic axo-axonic cells in cortical microcircuits. *Science* **311**, 233–235 (2006).
68. K. K. Cover, B. N. Mathur, Axo-axonic synapses: Diversity in neural circuit function. *J. Compar. Neurol.* **529**, 2391–2401 (2021). 10.1002/cne.v529.9.
69. 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**, 265–276.e6 (2020).
70. M. Stimberg, R. Brette, D. F. Goodman, Brian 2, an intuitive and efficient neural simulator. *eLife* **8**, e47314 (2019).
71. O. Rhodes, P. A. Bogdan, C. Brenninkmeijer, S. Davidson, D. Fellows, A. Gait, D. R. Lester, M. Mikaitis, L. A. Plana, A. G. D. Rowley, A. B. Stokes, S. B. Furber, Spynnaker: A software package for running pynn simulations on spinnaker. *Front. Neurosci.* **12**, 816 (2018).
72. Y.-H. Liu, X.-J. Wang, Spike-frequency adaptation of a generalized leaky integrate-and-fire model neuron. *J. Comput. Neurosci.* **10**, 25–45 (2001).
73. 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**, e0191425 (2018).
74. M. Tabuchi, S. Inoue, R. Kanzaki, K. Nakatani, Whole-cell recording from Kenyon cells in silkmoths. *Neurosci. Lett.* **528**, 61–66 (2012).
75. I. Ito, R. C.-Y. Ong, B. Raman, M. Stopfer, Sparse odor representation and olfactory learning. *Nat. Neurosci.* **11**, 1177–1184 (2008).
76. 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**, 359–365 (2002).
77. 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. *eLife* **12**, e77578 (2023).
78. 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* (NeurIPS, 2010), pp. 7–11.
79. L. A. Plana, J. Garside, J. Heathcote, J. Pepper, S. Temple, S. Davidson, M. Lujan, S. Furber, spinnlink: Fpga-based interconnect for the million-core spinnaker system. *IEEE Access* **8**, 84918–84928 (2020).
80. A. Wystrach, M. Mangan, B. Webb, Optimal cue integration in ants. *Proc. Biol. Sci.* **282**, 20151484 (2015).
81. 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)* (IEEE, 2018), pp. 7758–7765.
82. C. Groh, W. Rössler, Analysis of synaptic microcircuits in the mushroom bodies of the honeybee. *Insects* **11**, 43 (2020).
83. T. Schoepe, E. Janotte, M. B. Milde, O. J. N. Bertrand, M. Egelhaaf, E. Chicca, Finding the gap: Neuromorphic motion vision in cluttered environments. arXiv:2102.08417 [cs.NE] (2021). https://doi.org/10.48550/arXiv.2102.08417.
84. G. Haessig, A. Cassidy, R. Alvarez, R. Benosman, G. Orchard, Spiking optical flow for event-based sensors using IBM's truenorth neurosynaptic system. *IEEE Trans. Biomed. Circuits Syst.* **12**, 860–870 (2018).
85. 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)* (IEEE, 2021), pp. 5892–5899.
86. 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 Trans. Pattern Anal. Mach. Intell.* **42**, 2051–2064 (2020).
87. J. K. N. Verheyen, J. Dupeyroux, G. C. H. E. de. Croon, A novel multi-vision sensor dataset for insect-inspired outdoor autonomous navigation, in *Biomimetic and Biohybrid Systems*, A. Hunt, V. Vouloutsi, K. Moses, R. Quinn, A. Mura, T. Prescott, & P. F. Verschure Eds. (Springer-Verlag, 2022), pp. 279–291.
88. S. Hausler, A. Jacobson, M. Milford, Multi-process fusion: Visual place recognition using multiple image processing methods. *IEEE Robot. Autom. Lett.* **4**, 1924–1931 (2019).
89. W. F. Tobin, R. I. Wilson, W.-C. A. Lee, Wiring variations that enable and constrain neural computation in a sensory microcircuit. *eLife* **6**, e24838 (2017).
90. C. L. Barnes, D. Bonnéry, A. Cardona, Synaptic counts approximate synaptic contact area in *Drosophila*. *PLOS ONE* **17**, e0266064 (2022).
91. S. Holler, G. Köstinger, K. A. Martin, G. F. Schuhknecht, K. J. Stratford, Structure and function of a neocortical synapse. *Nature* **591**, 111–116 (2021).
92. Z. Zheng, F. Li, C. Fisher, I. J. Ali, N. Sharifi, S. Calle-Schuler, J. Hsu, N. Masoodpanah, L. Kmecova, T. Kazimiers, E. Perlman, M. Nichols, P. H. Li, V. Jain, D. D. Bock, Structured sampling of olfactory input by the fly mushroom body. *Curr. Biol.* **32**, 3334–3349.e6 (2022).
93. 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**, e52278 (2020).
94. D. Zavitz, E. A. Amematsro, A. Borisyuk, S. J. Caron, Connectivity patterns shape sensory representation in a cerebellum-like network. bioRxiv 430647 [Preprint] (16 August 2021). https://doi.org/10.1101/2021.02.10.430647.
95. A. Kodzhabashev, M. Mangan, Route following without scanning, in *Biomimetic and Biohybrid Systems*, S. P. Wilson, P. F. Verschure, A. Mura, T. J. Prescott, Eds. (Springer International Publishing, 2015), pp. 199–210.
96. TurtleBot, Robotis (2021), https://emanual.robotis.com/docs/en/platform/turtlebot3/overview/.
97. 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)* (IEEE, 2014), pp. 686–689.
98. 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)* (IEEE, 2013), pp. 1–8.
99. M. F. Land, Visual acuity in insects. *Annu. Rev. Entomol.* **42**, 147–177 (1997).
100. J. Howard, A. Dubs, R. Payne, The dynamics of phototransduction in insects. *J. Comp. Physiol. A* **154**, 707–718 (1984).
101. A. Borst, *Drosophila*'s view on insect vision. *Curr. Biol.* **19**, R36–R47 (2009).
102. D. O'Carroll, N. Bidweii, S. Laughlin, E. Warrant, Insect motion detectors matched to visual ecology. *Nature* **382**, 63–66 (1996).
103. A. P. Davison, D. Brüderle, J. Eppler, J. Kremkow, E. Muller, D. Pecevski, L. Perrinet, P. Yger, Pynn: A common interface for neuronal network simulators. *Front. Neuroinform.* **2**, 11 (2009).
104. S. Song, K. D. Miller, L. F. Abbott, Competitive hebbian learning through spike-timing-dependent synaptic plasticity. *Nat. Neurosci.* **3**, 919–926 (2000).
105. A. Morrison, M. Diesmann, W. Gerstner, Phenomenological models of synaptic plasticity based on spike timing. *Biol. Cybern.* **98**, 459–478 (2008).
106. M. Mikaitis, G. Pineda García, J. C. Knight, S. B. Furber, Neuromodulated synaptic plasticity on the spinnaker neuromorphic system. *Front. Neurosci.* **12**, 105 (2018).
107. S. Davies, F. Galluppi, A. Rast, S. Furber, A forecast-based stdp rule suitable for neuromorphic implementation. *Neural Netw.* **32**, 3–14 (2012).
108. S. Skrede, Nordland dataset (2013).
109. H. Rebecq, R. Ranftl, V. Koltun, D. Scaramuzza, High speed and high dynamic range video with an event camera. *IEEE Trans. Pattern Anal. Mach. Intell.* **43**, 1964 (2019).

Acknowledgments: We thank S. Davidson (University of Manchester), L. P. Cabrera (University of Manchester), K. Stratford (University of Edinburgh), A. Rowley (University of Manchester), M. Hopkins (University of Manchester), and T. Schoepe (University of Groningen) for the support in SpiNNaker software and hardware. We also thank J. Conradt (KTH Royal Institute of Technology) and J. P. Romero Bermudez (KTH Royal Institute of Technology) for the support in using their SpiNNaker-DVS interface board. **Funding:** This work was supported by the China Scholarship Council (201801060165) and a research grant from Huawei Technologies (to B.W.)
Author contributions: Conceptualization: B.W. and L.Z. Methodology: L.Z., B.W., and M.M. Investigation: L.Z. Visualization: L.Z. Funding acquisition: L.Z. Project administration: L.Z. and B.W. Supervision: B.W. and M.M. Writing—original draft: L.Z. and B.W. Writing—review and editing: L.Z., B.W., and M.M. **Competing interests:** M. M. is the research director at Opteran Technologies (the work and funding are completely separate). The other authors declare that they have no competing interests. **Data and materials availability:** All data needed to evaluate the conclusions in the paper are present in the paper. The codebase and dataset of the work can be found at https://doi.org/10.5281/zenodo.8289547.

Submitted 30 December 2022
Accepted 29 August 2023
Published 27 September 2023
10.1126/scirobotics.adg3679

Science Robotics

Neuromorphic sequence learning with an event camera on routes through vegetation

Le Zhu, Michael Mangan, and Barbara Webb

Sci. Robot. **8** (82), eadg3679. DOI: 10.1126/scirobotics.adg3679

Science Robotics (ISSN 2470-9476) is published by the American Association for the Advancement of Science. 1200 New York Avenue NW, Washington, DC 20005. The title *Science Robotics* is a registered trademark of AAAS.