ROBOT LOCALIZATION

A compact neuromorphic system for ultra–energy-efficient, on-device robot localization

Adam D. Hines*, Michael Milford, Tobias Fischer

Neuromorphic computing offers a transformative pathway to overcome the computational and energy challenges faced in deploying robotic localization and navigation systems at the edge. Visual place recognition, a critical component for navigation, is often hampered by the high resource demands of conventional systems, making them unsuitable for small-scale robotic platforms, which still require accurate long-endurance localization. Although neuromorphic approaches offer potential for greater efficiency, real-time edge deployment remains constrained by the complexity of biorealistic networks. To overcome this challenge, fusion of hardware and algorithms is critical when using this specialized computing paradigm. Here, we demonstrate a neuromorphic localization system that performs competitive place recognition in up to 8 kilometers of traversal using models as small as 180 kilobytes with 44,000 parameters while consuming less than 8% of the energy required by conventional methods. Our system, locational encoding with neuromorphic systems (LENS), integrates spiking neural networks, an event-based dynamic vision sensor, and a neuromorphic processor within a single SynSense Speck chip, enabling real-time, energy-efficient localization on a hexapod robot. When compared with a benchmark place recognition method, sum of absolute differences, LENS performs comparably in overall precision. LENS represents an accurate fully neuromorphic localization system capable of large-scale, on-device deployment for energy-efficient robotic place recognition. Neuromorphic computing enables resource-constrained robots to perform energy-efficient, accurate localization.

INTRODUCTION

Robot localization is a critical component of many autonomous navigation systems, enabling robots to determine their location while supporting the ability to understand and interact with their surroundings. A central challenge in robot localization is visual place recognition (VPR), which requires robots to identify and classify previously visited locations under varying conditions (*1*–*3*). Conventional VPR strategies often rely on deep convolutional neural networks or transformer-based architectures to robustly extract features from the environment (*4*–*7*). For resource-constrained robotic platforms that nonetheless require the ability to keep track of where they are located over vast distances, using such conventional VPR methods is often not feasible because of their computational demands. An open challenge for real-world deployment of VPR is to find models that are computationally and energy efficient and can be deployed at the edge.

Neuromorphic computing, which takes inspiration by the brain, has emerged as a promising solution for addressing the energy and computational challenges associated with robot localization and navigation (*8*–*12*). The human brain's ability to learn and navigate complex environments efficiently (*13*–*16*) has inspired roboticists to create more computationally efficient localization systems (*17*–*21*). In particular, spiking neural networks (SNNs) have been widely explored because of their ability to perform tasks through biorealistic simulations of neuron activity (*22*, *23*). Although several SNN models have been proposed for robotic localization (*10*, *12*, *24*–*32*), they ultimately have not yet fully lived up to the promise of computationally efficient deployment because of the complexity of continually modeling neurons in real time (*19*, *24*, *26*).

To advance the practicality of neuromorphic computing in VPR, the focus must shift toward developing models that can be effectively deployed on size-, weight-, and power-constrained robotic platforms. One approach to overcoming deployment barriers is to trade the intricate biorealism of traditional SNNs for simpler, more efficient networks that still deliver robust performance (*33*). However, small-scale systems can often be limited in their ability to map large environments, which restricts the downstream use cases (*11*). To improve efficiency, the full potential of SNNs for robotic deployment can be realized by integrating them with neuromorphic hardware (*10*–*12*, *19*, *34*–*37*) such as Intel's Loihi 2, TrueNorth, and Tianjic (*8*, *38*, *39*), which draw inspiration from neuroscience for their design, enabling them to transmit and receive physical spikes from neurons within the processor cores (*40*). Neuromorphic computing has been successfully used in tasks other than localization, such as control systems for drone flight (*41*), route following (*11*), autonomous driving controllers (*42*), and motor controllers (*43*). In contrast, nonspiking algorithms that are brain inspired are widely used in localization tasks but have limited ability to be deployed on neuromorphic hardware. Although several neuromorphic localization systems have been developed and tested, they are limited by environment scale (*11*), require complex multimodal fusion of large network models (*12*), or have restricted real-time capability (*10*).

Event cameras, including dynamic vision sensors (DVSs), offer further computational advantages when paired with neuromorphic processors, given that they only transmit information based on pixelwise changes in light intensity exceeding a threshold, thereby reducing unnecessary data processing (*35*). Spiking activity can be easily triggered and propagated on the basis of incoming input event streams, with SNNs being well suited to process them. Fusing neuromorphic algorithms, sensors, and processors to perform robotic localization tasks therefore provides a promising avenue to overcome the computational limitations inherent in SNN-based VPR (*10*, *12*).

QUT Centre for Robotics, Queensland University of Technology, 2 George St, Brisbane, QLD 4001, Australia.
*Corresponding author. Email: adam.hines@qut.edu.au

Here, we introduce a neuromorphic pipeline for robotic localization that is compact, real-time capable, and able to map environments up to 8 km in length. Our system, locational encoding with neuromorphic systems (LENS), is designed to deliver high accuracy and efficiency with a model size of less than 180 kilobytes and 44,000 parameters and is capable of processing up to 8 km of traversal data while consuming less than 8% of the energy required by conventional platforms. We trained an SNN using a temporal time–to–first spike encoding scheme designed to perform VPR (*27*) and deployed our model on the SynSense Speck (*44*), which combines a DVS and system-on-chip neuromorphic processor. The LENS neuromorphic algorithm was developed in tandem with neuromorphic hardware deployment to deliver ultra–energy-efficient place recognition for robotic localization. By deploying on Speck, we eliminated the need for any external sensory modalities or computational resources, given that all localization is performed on chip. By abstracting complex biorealism in favor of enhanced performance, LENS represents a notable advancement in neuromorphic localization, as a fully event-driven platform designed for VPR. We validated LENS on a Hiwonder JetHexa hexapod robot (Fig. 1), demonstrating its effectiveness in both indoor and outdoor small-scale traversals (≈25 to 40 m, 50 to 80 places) and large-scale datasets (≈8 km, 600 places) (*45*), showcasing its potential for online VPR.

RESULTS

Locational encoding with neuromorphic systems

An overview of our LENS network is presented in Fig. 2. We used event streams from the Speck DAVIS128 sensor in two different ways for the mapping and localization phases of the robot navigation. A place was represented as 1 s of movement, translating to a distance of ≈0.15 m (Fig. 2A). For the mapping phase, the number of events was counted per pixel and then flattened into a two-dimensional (2D) temporal representation image of the place (Fig. 2B). Using a 2D convolutional kernel with the center weight set to 1 and all other weights set to 0 (see the "Model training" section under Materials and Methods), we selected the center pixel from the 2D image to downsample to 10 pixels by 10 pixels to fit in our LENS (Fig. 2B). Following Hines *et al.* (*27*), we trained our model by first normalizing pixel intensities between 0 and 1 to use a temporally coded spike scheme, such that higher pixel intensities result in earlier spiking, with lower intensities activating later (Fig. 2C and see the "Model training" section under Materials and Methods). Connection weights were trained via spike timing–dependent plasticity (STDP) learning, with enhanced connectivity when postsynaptic activity occurred after presynaptic spikes and diminished connectivity if postsynaptic neurons spike beforehand (Fig. 2C and see the "Model training" section under Materials and Methods). For our three-layer architecture, we used unsupervised learning between the

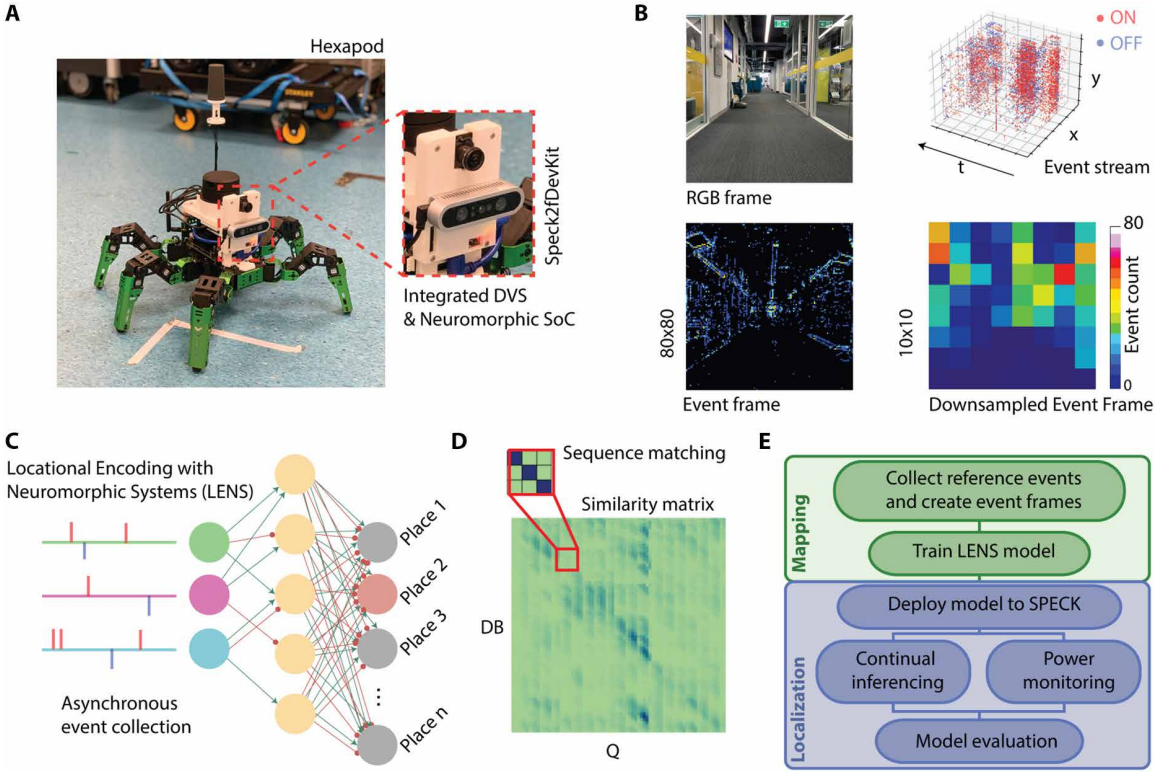

Fig. 1. Neuromorphic place recognition system on a hexapod robot. (**A**) Our system was deployed on a Hiwonder JetHexa hexapod robotic platform equipped with a custom 3D-printed mount housing a SynSense Speck2fDevKit. Event streams captured by the onboard DVS were processed into event frames by counting on and off events during the hexapod's traversal (**B**). The initial 128 pixel–by–128 pixel input was cropped to an 80 pixel–by–80 pixel ROI, which was then downsampled to 10 pixels by 10 pixels by selecting central pixels via a 2D convolutional layer (see the "Deployment time" section under Materials and Methods). (**C**) The SNN LENS learned reference event frames for on-chip deployment, enabling real-time localization using asynchronously collected events. (**D**) Sequence matching techniques enhanced the similarity between reference and query inputs, improving overall precision. (**E**) Schematic of the fully neuromorphic VPR pipeline.

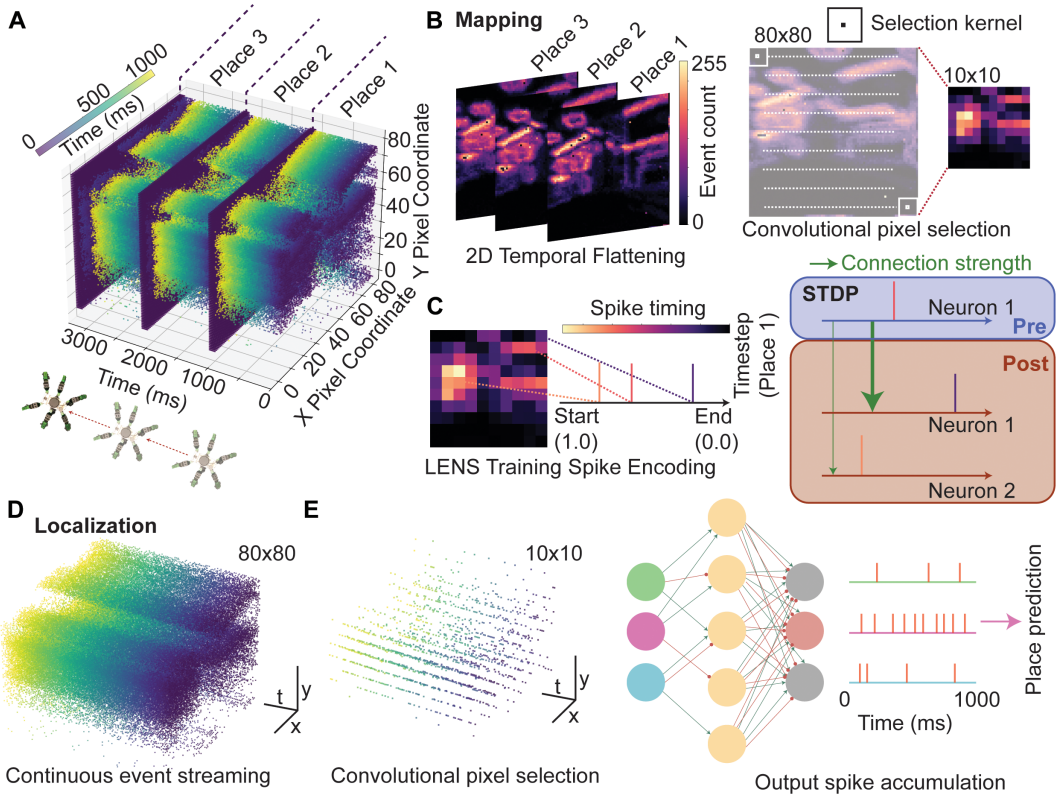

Fig. 2. Event stream processing using LENS. (**A**) Event streams were collected from the Spec2fDevKit board from an 80 pixel–by–80 pixel ROI using a DAVIS128 sensor as our hexapod robot moves through the environment. One place represents 1 s or ≈0.15 m of movement. The event stream was then used in two unique ways during mapping and localization phases. (**B**) During mapping, events over the 1-s time period were counted and flattened into a 2D temporal representation of the place. We used a convolutional selection kernel to pick the center pixel from the 2D image (see the "Model training" section under Materials and Methods), resulting in a 10 pixel–by–10 pixel downsampled image. (**C**) Downsampled images were normalized in the range between 0 and 1 with pixel intensity determining spike timing in our LENS training regime, following (*27*). Using STDP across the network layers, we strengthened or weakened both excitatory and inhibitory connections if the spike occurred after or before presynaptic activity, respectively. (**D**) During localization, we forwent the temporal representation of places and used the raw event streams to continuously activate and bias the LENS network on-chip. (**E**) We applied the same convolutional pixel selection as the first layer of our neural network, deployed on the Speck directly. Over the time collection period of 1 s, we accumulated output spikes from our model and made a place prediction on the basis of the neuron with the highest spiking activity.

input and feature layers and a supervised delta learning rule between the feature and output layer (see the "Model training" section under Materials and Methods).

During the localization phase, we used the raw event streams collected from the Speck directly in our model instead of 2D temporal representations of places (Fig. 2D and see the "Deployment time" section under Materials and Methods). The first layer of our LENS was the convolutional pixel selection filter applied earlier to reduce the input dimensionality to 10 pixels by 10 pixels (Fig. 2E and see the "Deployment time" section under Materials and Methods). We performed place recognition with asynchronous events over 1-s collection windows and picked the location with the highest spiking activity for the neuron in the output layer (Fig. 2E and see the "Online place matching" under Materials and Methods).

Power and energy efficiency

Neuromorphic computing platforms offer an advantage in terms of power and energy consumption when compared with traditional von Neumann hardware, particularly in tasks such as VPR where energy efficiency is crucial for long-term operation on battery-powered robots. We measured the power consumed by various components of the Speck neuromorphic processor while performing VPR (Fig. 3A). When the robot was stationary, the DVS generated very few events, which resulted in further reductions in power consumption. To quantify the energy savings of LENS deployed on the Speck processor relative to von Neumann hardware, we performed VPR using the place matching method of sum of absolute differences (SAD) (*46*) on an Intel i7-9700K central processing unit (CPU) and a NVIDIA Jetson Nano, which ran the hexapod (see the "Power and energy measurements" section under Materials and Methods). After subtracting baseline CPU power use, we found that the Speck consumed an average of 2.7 mW, which is only 8.8 and 0.5% of the power required by the Jetson Nano and Intel CPU to run SAD, respectively (Fig. 3B).

The overall energy consumption (the integral of power over time) of the Speck was 327 mJ, with the Jetson using 2968 mJ and the Intel CPU requiring 61,427 mJ, meaning that, overall, our system required only 8.9 and 0.5% of the energy needed to perform the same task on von Neumann hardware (Fig. 3C). Fewer pixel representations of places still resulted in energy consumption far greater than what was required for LENS on Speck (see table S2). We also

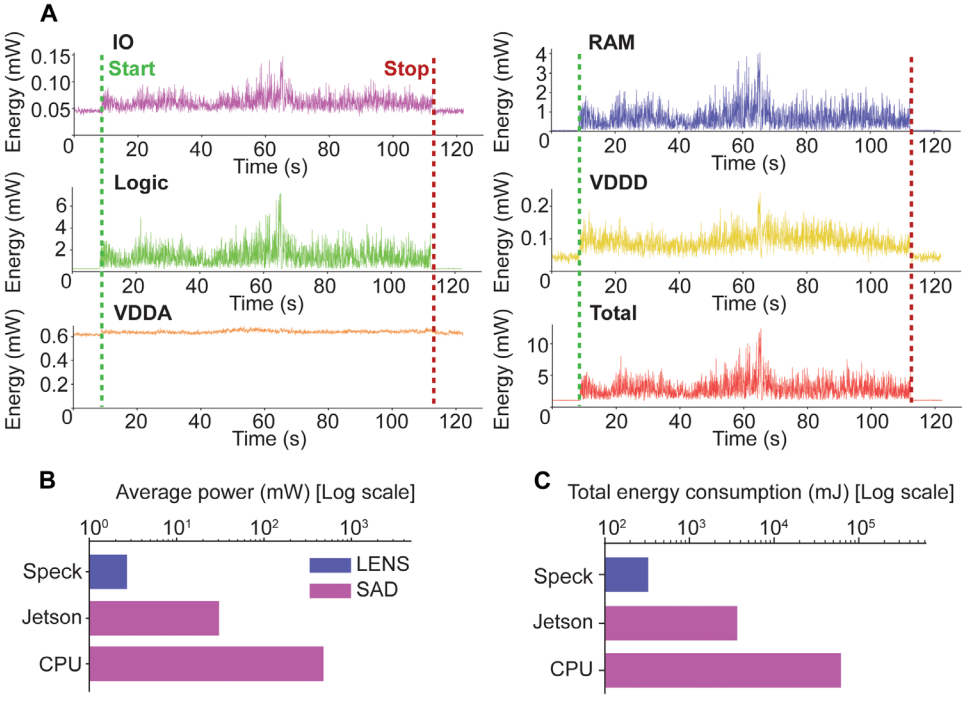

Fig. 3. Energy efficiency of real-time VPR on a neuromorphic processor versus von Neumann hardware. (**A**) Power consumption was recorded across five components of the Speck neuromorphic processor, namely, IO, RAM, logic, VDDD, and VDDA, and the total. Start indicates when the robot began to move until the navigation task had finished, indicated by stop. (**B**) The Speck processor consumed substantially less total energy compared with the Jetson Nano and Intel CPU (**C**). The milliwatt and millijoule values are presented on a logarithmic scale for display purposes.

observed a prominent reduction in power and energy consumption deploying to Speck when running our LENS on von Neumann hardware (fig. S1).

Large-scale place recognition with compact models

To assess our system's suitability for real-world scenarios, we evaluated its performance in large-scale environments using the Brisbane Event VPR dataset containing event streams from an ≈8-km route in Brisbane, Australia (Fig. 4) (*45*). This dataset is representative of a stable, four-wheeled robotic platform that our model could be deployed on. We generated 641 event frames of unique places by sampling over 1-s intervals from the dataset (Fig. 4B). To fit the model onto the Speck chip, we used a network architecture of 49 input neurons, 63 feature layer neurons, and 641 output neurons (Fig. 4B). This was the largest and highest performing model architecture we could use to run our model on chip, balancing the numbers of input pixels and feature representations (fig. S6). We used the "sunset2" traverse for mapping (reference dataset) and the "sunset1" traverse for localization (query dataset). We also used the SAD method (*46*, *47*) as a baseline comparison method, calculating pixelwise similarity between mapping (training) and localization (test) data.

We used spikes from the model's output layer and the Euclidean distance calculated in SAD to generate similarity matrices and used sequence matching, which preserves the distances of sequence matches while nullifying cross-distances, improving localization precision during deployment time (Fig. 4C and see the "Online place matching" under Materials and Methods). The similarity matrix was used to compute Recall@*N* (the proportion of true place matches retrieved within the top *N* predicted matches) and precision and recall metrics, which evaluated how well the robot is correctly identifying its location as it moves through the environment. Although other methods for event-based place recognition exist, they require the reconstruction of images from event streams for deep feature extraction (*48*, *49*). Hence, we elected not to compare our system with these methods because they require higher complexity and computational time than what is feasible to run on our robotic platform.

Both LENS and SAD generated similarity matrices that closely aligned with the ground truth (Fig. 4C). LENS performed best overall for Recall@*N*, achieving a Recall@1 of 0.88 compared with SAD's 0.81 (Fig. 4D). LENS also demonstrated better precision at lower recall, indicating higher confidence in its place recognition (Fig. 4E). The nonmonotonic behavior in some precision-recall curves, where precision temporarily improves with increasing recall, occurred when the rate of true-positive matches increased faster than false positives at certain threshold ranges. An SNN to achieve this degree of accuracy with an architecture of only 753 neurons across three layers showed promise for large-scale place recognition applications. We required the same amount of storage space capacity to store our model and the downsampled images for SAD, which was ≈179 kilobytes.

Localization performance during a navigation task

To evaluate our system's real-time performance at the edge, we performed VPR on our hexapod in both indoor and outdoor environments (Fig. 5). The model architecture was designed with 100 by 200 by 75 for the input, feature, and output neurons, respectively. As illustrated in Fig. 5 (A and F), the generated event frames corresponded to the environment being traversed, particularly by detecting edges in the scene. The robot dealt with viewpoint and timing differences between the mapping and localization phases, given that the route paths differed because of imperfect teleoperation (Fig. 5, B and G). In comparison with the Brisbane Event dataset, the hexapod represented a more unstable and challenging platform to perform place recognition.

For the indoor task, LENS achieved a Recall@1 of 0.64, outperforming SAD's performance of 0.36 (Fig. 5, C and D). The precision-recall curve indicated that LENS achieved higher precision at lower recall values compared with SAD (Fig. 5E), suggesting that LENS was highly accurate in its identification of places. In the outdoor task, LENS performed comparably to SAD with a Recall@1 of 0.33 and 0.29, respectively (Fig. 5, H and I). LENS also achieved a higher precision at lower recall, shown in the precision-recall curve (Fig. 5J). The reduced performance in outdoor environments was likely because of the increased number of

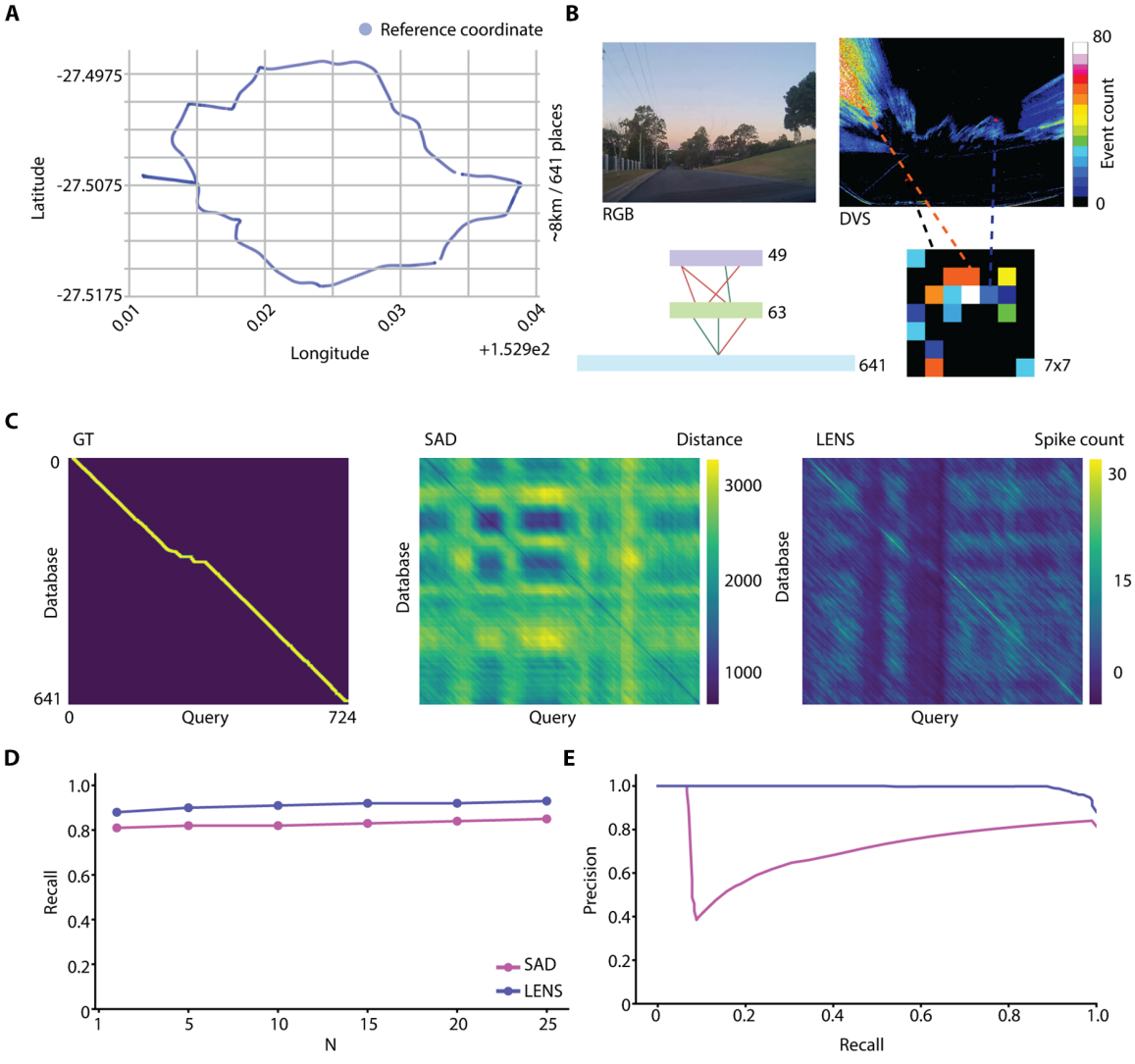

Fig. 4. Large-scale place recognition using a compact neuromorphic ecosystem. (**A**) Coordinate map of the Brisbane Event VPR dataset route (*45*), which represents an ≈8-km traversal and 641 unique place locations around the suburbs of Brisbane, Australia. (**B**) Event frames were generated in the same way as in robot experiments (Fig. 5). Pixels were selected randomly rather than by applying a 2D convolutional filter because of a larger event camera resolution used in (*45*). We selected pixels to generate event frames that are 7 pixels by 7 pixels in size. Our model architecture was modified to 49 input neurons, 63 feature layer neurons, and 641 output neurons to allow for on-chip localization. (**C**) Similarity matrices for both SAD and LENS, showing how the similarity for both methods aligns closely with the ground truth (GT). (**D**) Recall@*N* and (**E**) precision-recall curves for our method when compared with SAD. LENS performed best overall compared with SAD and also achieved better precision at lower recall, indicating that our system is more confident at the places it selected. The energy efficiency and compact model size of just 753 neurons with 44,000 parameters highlights the benefit of our system.

turns in the trajectory, which caused widespread neuronal activation across the field of view. As indicated by the similarity matrix (Fig. 5H), LENS showed a higher amount of activity across multiple output neurons. Figure S8 suggests that increased turning in the navigation route introduced noise, distributing spikes across more neurons.

Overall, our compact SNN model (130 to 150 kilobytes in size) demonstrated comparable performance for VPR tasks in multiple environments, with key advantages of real-time deployability while being highly power and energy efficient. By contrast, the SAD method required 388 kilobytes to store the images, ≈2.6 times the storage space needed for our model. Our system offered substantial benefits in terms of model size, making it feasible for deployment on computationally resource-constrained platforms.

Both SAD and LENS performed better on the Brisbane Event VPR dataset than during the on-robot VPR task (Fig. 5). The Brisbane Event dataset used a 346 pixel–by–240 pixel sensor (*45*), compared with the 128 pixel–by–128-pixel sensor (*50*) on our robot. In addition, the more stable platform (i.e., a car) used for dataset collection could reduce errors from viewpoint variance or swaying movement observed with the hexapod (see movie S1). These results indicate that our proposed model can be effectively deployed for both short- and long-range place recognition tasks for robotic localization.

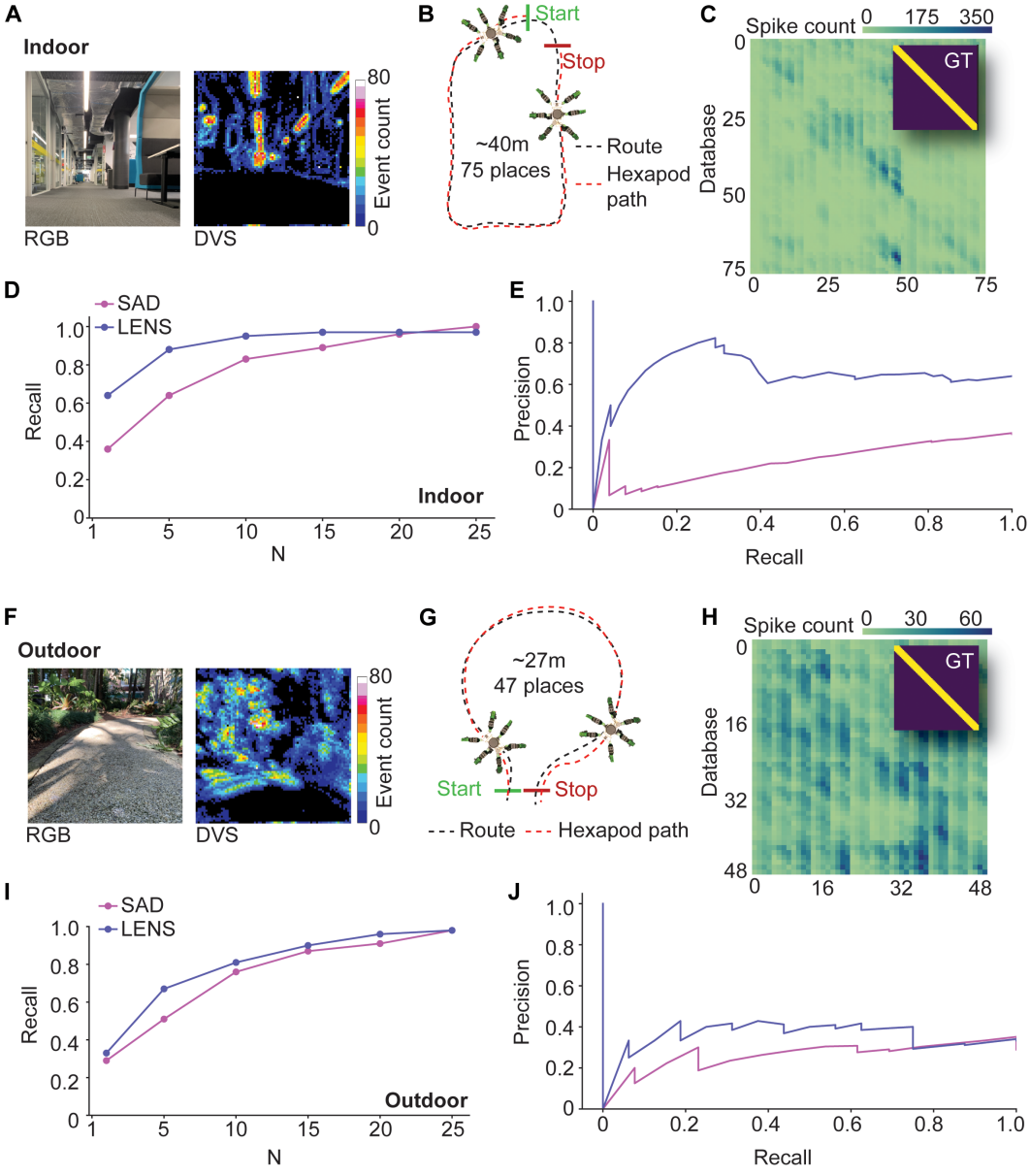

Fig. 5. Precision of a real-time neuromorphic robotic localization platform. We deployed our system on a hexapod robotic platform for on-device localization. (**A**) First, we tested an indoor environment by mapping and localizing along a predetermined route (**B**). Given that the hexapod was teleoperated, there are slight differences in viewpoint between the mapping and localization phases. (**C**) To analyze the precision and recall of our indoor environment, we generated sequentially matched similarity matrices of the output spikes from all output neurons in our model while localizing. (**D**) Our compact system generally performed better than the SAD method for Recall@*N*, measuring accuracy in the top *N* matches, and precision recall (**E**). We then deployed our hexapod in an outdoor environment (**F** and **G**). The similarity matrix for the outdoor environment had more aliased incorrect matches than the indoor one (**H**), most likely because of the presence of multiple turns in the route. (**I** and **J**) In the outdoor task, LENS performed comparably to the SAD method in both the Recall@*N* and precision-recall evaluation.

DISCUSSION

This work demonstrated a fully neuromorphic VPR system capable of real-time localization, which was successfully deployed on a hexapod robotic platform without any external sensing or computational resources. The LENS SNN model, characterized by its simplicity and compactness, performed accurately across multiple environments and scales, demonstrating its versatility for a range of navigation platforms and tasks. Our system integrated with the Speck chip operated with one to two orders of magnitude lower energy than conventional von Neumann hardware, underscoring its viability for deployment on resource-constrained platforms such as unmanned aerial vehicles (*51*) and underwater (*52*) and locomotive robotics (*53*).

Our models were less than 180 kilobytes in size, with the largest containing only 44,000 parameters. Accurate localization with our compact model enables portable deployment on neuromorphic processors with higher memory capability for larger models to improve overall performance (fig. S6). Previous research has shown

that accurate place recognition with event cameras can be achieved with as few as 25 pixels (*47*). Our system capitalized on this by balancing the number of pixels with the size of the feature layer to achieve the optimal performance (fig. S6). A limiting factor in this work was the memory capacity of the neurocores for Speck, which cap at 64 kilobytes per core. Overcoming this limitation with neuromorphic hardware offering higher memory capabilities could further enhance the performance of large-scale place recognition tasks (fig. S6). Nonetheless, we have shown comparable performance for place recognition in highly compact spiking network models across multiple environments.

Our neuromorphic place recognition system uses real event streams directly for localization in a unique way, establishing a previously unreported methodology for event-driven place recognition. This work highlights the potential for substantial efficiency gains when using event-based DVS over conventional cameras, especially in scenarios where energy-efficient processing is critical. Unlike frame-based cameras, which operate continuously, the event-driven DVS captures only essential changes, reducing data redundancy and energy consumption. Our experiments encompass both a conventional benchmark dataset widely used in the robotics community (*45*) and a more challenging legged robot navigation task, ensuring that the localization accuracy of our method is evaluated across a diverse range of environments and baseline performance levels. The accuracy achieved by LENS demonstrates its effectiveness within broader systems such as simultaneous localization and mapping (SLAM), where geometric constraints and filtering methods can mitigate occasional inaccuracies (*54*–*56*). By using a more stable platform, such as a four-wheeled robot, our network is able to achieve high precision comparable to a benchmark technique. This demonstrates that our compact spiking network models perform comparably to conventional nonevent-driven methods, highlighting the potential for further improvements with future developments in neuromorphic hardware and algorithmic enhancements.

As observed in the similarity matrices for the localization tasks (Fig. 5), introducing many turns into shorter trajectories proved challenging for LENS in the outdoor hexapod experiment. Dynamically adjusting these biases and parameters of DVS event streams to better handle these sudden motions could enhance accuracy across challenging environments (*57*). Continued development of this system will focus on detecting and handling turning motions to improve localization performance.

Another limitation of our system is the use of static, temporal representations of places generated from event streams using time-window binning for model training (Fig. 2). Training our model directly on DVS event streams could help overcome the localization challenges observed in dynamic environments. Developing this would also facilitate online learning capabilities, where event streams are used directly to train network models in real time. This would enable the system to adapt to unseen environments on the go, potentially incorporating introspective capabilities that allow the robot to recognize when it is uncertain about its location and learn accordingly (*58*–*61*). In addition, using more information from the DVS sensor, such as including convolutional layers and self-supervised learning, could be useful to extract more features for enhanced performance (*62*).

Integration of our neuromorphic system with other methods for navigation could be used to enhance and improve robotic localization performance. Dupeyroux *et al.* (*63*) developed a bioinspired path integration system for a hexapod robotic platform that would serve as a complementary method for VPR. In addition, other insect-inspired systems use methods such as goal approaching and collision avoidance, which allow for autonomous navigation in unseen environments, typically for homing tasks (*11*, *63*, *64*). Integration of these mixed-model paradigms is a promising method to be used in more comprehensive navigation systems such as SLAM, of which there are many neuroscience-inspired systems available (*18*, *65*–*67*).

In conclusion, we present a fully end-to-end neuromorphic VPR system that performs accurate, low-power localization on robotic platforms. Our work has the potential to enable robots in the field to navigate further and conduct longer missions by reducing the computational and energy demands of current vision systems (*68*). An important aspect of this work was to ensure that neuromorphic localization could be achieved on multiple robotic platforms, including for resource-constrained scenarios. Our approach moves VPR toward efficient, versatile neuromorphic localization, unlocking previously unknown possibilities for autonomous robotic navigation at the edge.

MATERIALS AND METHODS

Model training

Our network models were trained on CPU hardware or a Jetson Nano using static temporal representations of event streams with pixel intensity values relating to the number of events collected over a 1-s time window, subsequently normalized in the range between 0 and 1 (see Fig. 2). The network was trained using spike timing from an abstracted spiking method, where the pixel intensity value represents temporal activity (*27*, *69*, *70*). The network's three-layer architecture (input, feature, and output) used unsupervised learning between the input and feature layers and a supervised delta learning rule for the feature-to-output layer.

Unsupervised learning applied STDP to update weights W, encouraging or pruning connections between neurons. Initial weights and connections were randomly generated from predetermined probabilities, with connections either being positive (excitatory connection) or negative (inhibitory connection). From the input-to-feature layer, there was a 35 and 75% excitatory and inhibitory connection probability, respectively. The feature-to-output layer was fully connected with an equal distribution of excitatory and inhibitory connections. As spikes were propagated through the network, the layer weights were updated in the following way

$$\Delta W_{ji}^{nm}(t) = \frac{\eta_{\mathrm{STDP}}(t)}{f_j^n} \cdot \left[\Theta\left(x_i^m(t-1)\right) \cdot \Theta\left(x_j^n(t)\right) \cdot \left(0.5 - x_j^n(t)\right)\right]_{(1)}$$

where W_{ij}^{nm} is the connection weight between neuron j in layer n to neuron i in layer m, η_{STDP} is the STDP learning rate, f_j^n is the target firing rate of neuron j in layer n, $\Theta(\cdot)$ is the Heaviside step function, x_i^m and x_j^n are the neuron states x in the network layer i and j with connections m and n, and t is the time step. During weight updates throughout training, we did not specify any rule that says that weights must remain excitatory or inhibitory; however, we did not observe weights switching sign. That is, initially seeded connection types remained robust throughout training.

For supervised learning between the feature and output layer, we used a delta learning rule to force each output neuron to learn the feature representation of a single reference place (*27*, *69*)

$$\Delta W_{ji}^{nm}(t) = \eta_{\mathrm{STDP}}(t)/f_j^n \cdot \left[x_i^m(t-1)\left(x_{\mathrm{force},j}^n(t) - x_j^n(t)\right)\right] \quad (2)$$

where x_{force} was the forced spike in the selected output neuron during training. The hyperparameters used for our models can be found in table S1, which were obtained by performing a hyperparameter sweep.

Deployment time

The LENS framework leveraged SNNs on neuromorphic processors for computationally and energy-efficient place recognition. We deployed our pretrained SNN model on a SynSense Speck2fDevKit, which incorporates the Speck neuromorphic processor with a 128 pixel–by–128 pixel DVS (*44*) for event-driven, real-time localization. This was in contrast with the training methodology, where event frames were generated by counting the number of events over a specified time window. Our localization system used truly asynchronous event streams to continually bias and activate neurons as the robot navigated through the environment.

To optimize input size and remove low-activity pixels, we first selected an 80 pixel–by–80 pixel region of interest (ROI) in the top center of the sensor. Events from this ROI were processed using a 2D convolution layer with a kernel size and stride of 8, which acted as the first processing layer of our SNN before the pretrained input layer and was deployed on chip. We set the center weight of the kernel to 1, with all other weights at 0, effectively selecting the center pixel from each convolution to reduce the input to 100 neurons. This selection approach followed a previously established method for event-pixel selection (*47*).

The input neurons were sparsely connected to a linear feature layer of 200 neurons, with excitatory and inhibitory connection probabilities set at 35 and 75%, respectively. The feature layer was fully connected to the output layer, with the number of output neurons corresponding to the number of learned places (see the "Model training" section under Materials and Methods). Events were asynchronously collected and passed into our model, where constant-leak integrate-and-fire neurons propagated spikes across the network layers

$$\tau \dot{v} = -v_{\mathrm{leak}} + R \cdot \left(I_{\mathrm{syn}} + I_{\mathrm{bias}}\right) \quad (3)$$

where τ is the membrane time constant, v is the membrane potential, R is the constant resistance value used to match units between membrane potential and currents, I_{syn} is the weighted sum of all input synaptic contributions, and I_{bias} is a constant bias.

Online place matching

Place matching was performed using real-time sequence matching (*46*, *71*) to enhance the precision of our VPR model on the Speck. Spike counts from the output layer were collected in 1-s time bins and then averaged over four time bins (4 s), representing ≈0.5 m of robot movement (average speed of 13.5 cm s^{-1}). All place and sequence matching was performed off-chip on a Jetson Nano during on-robot deployment or on a CPU for simulated experiments. Once four sets of output spike counts were gathered, representing about 2 m of travel, they were compiled into a distance matrix M_{dist} (Fig. 1D), which was convolved (*) with an identity matrix $\mathbb{I}_L$ to generate a sequence-based distance matrix M_{seq}, as previously described (*71*)

$$M_{\mathrm{seq}}(a,b) = \mathbb{I}_L * M_{\mathrm{dist}}(a,b)/L \quad (4)$$

where a and b are the row and column of the matrix, respectively, and L is the sequence length, which we set to 4 matching the number of output spike counts. For the Brisbane Event dataset, we set the sequence length to 30. The choice of sequence length matters for both computational efficiency and evaluation performance, with shorter lengths favoring the former and longer lengths benefiting the latter (fig. S7).

After generating the sequence-based distance matrix M_{seq}, we performed place matching on the processed spike outputs. The matched places $\hat{p}_b$ were identified as the places corresponding to the rows with the highest spike output in M_{seq}

$$\hat{p}_b = \underset{a}{\operatorname{argmax}}\, M_{\mathrm{seq}}(a,b) \text{ for each } b \in 1,2,3,4 \quad (5)$$

where $\hat{p}_b$ is the matched place for the bth sequence, a ranged over all database reference places, and $M_{\mathrm{seq}}(a,b)$ represents the sequence-based distance between the bth spike count set and the ath reference place.

Off-chip simulation of event streams

For power and energy consumption comparisons using LENS on neuromorphic and von Neumann hardware, we simulated event-driven localization using event frames from the Speck energy experiments (see the "Power and energy measurements" section under Materials and Methods). Integrate-and-fire neuron spike rates were generated using a time-based rate code

$$I_{t,h,w} = \begin{cases} 1 & \text{if } R_{t,h,w} < X_{h,w} \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

where $I_{t,h,w}$ is a Boolean tensor with spatial dimensions h and w over t time windows, where 1 = spike and 0 = no spike. $R_{t,h,w} \in [0,1]$ is a uniformly random tensor, and $X_{h,w}$ is the original input. Localization was simulated over 1000 time windows with $dt = 0.001$ s to represent 1 s of output spike collection, similar to the online place matching (see the "Online place matching" under Materials and Methods).

Robotic deployment

We used the commercially available JetHexa hexapod (Hiwonder, Shenzen, China) customized with a 3D-printed mount to house the Speck2fDevKit (see Fig. 1). We also simulated a four-wheeled robotic deployment by evaluating our system on the Brisbane Event dataset (*45*), to highlight that our energy-efficiency performance is robotic platform agnostic. Training data were collected by teleoperating the robot for a reference traversal. Real-time localization was performed on the Speck neuromorphic processor with the pretrained model closely following the reference traversal. Indoor and outdoor datasets represented ~40 and 27 m of traversal, encoding 75 and 48 unique places, respectively. To ensure accurate ground-truth alignment between the mapping and localization phases, experiments were consistently started and ended at the same location.

Power and energy measurements

Intel CPU power was measured using SoC Watch, and NVIDIA Jetson Nano power was recorded with the jetson-stats Python package,

with baseline idle power subtracted in both cases to only measure the energy consumed for the VPR algorithms. Speck power tracks [input/output (IO), random access memory (RAM), memory, analog power supply voltage (VDDA), and digital power supply voltage (VDDD)] were recorded at 20 Hz using SynSense's samna package during testing (see Fig. 3 and movie S1).

Statistical analysis

We evaluated our system using Recall@N and precision-recall curves, two standard metrics for VPR studies (*1*). All statistical analysis was performed on CPU hardware. Our implementation of precision and recall matched methods used in other place recognition systems (*12*) (fig. S2). Recall@N measured the accuracy of the system, such that, for $N = 1$, only the highest match was considered across all reference places, whereas, for $N = 5$, the top five matches were considered when calculating the accuracy, up to a maximum of 25 potential places. Precision-recall curves were used to assess performance by illustrating the relationship between precision and recall across different threshold levels for positive matches, calculated as follows

$$\text{Precision} = \frac{TP}{TP+FP} \quad \text{Recall} = \frac{TP}{GTP} \tag{7}$$

where TP is the number of true positives, FP is the number of false positives, and GTP is the number of ground-truth positives (*1*). Initially, precision and recall are undefined because there are no predictions; however, we initialize precision at 1 and recall at 0 for visualization purposes.

We used the SAD method for comparison, which computed pixelwise similarity across reference and query images and has been used previously for place recognition using event streams (*46*, *47*). Pseudo–ground truth data were collected following (*47*) using Global Positioning System (GPS) correspondence between reference and query images. Pseudo-GPS and manually confirmed ground truth showed high correspondence, with the majority of places being ±1 place label away (fig. S3). Further analysis of the accuracy of the pseudo–ground truth and applied tolerances were confirmed by near-identical results (fig. S4). We allowed a ground-truth tolerance of ≈1 m, which is equivalent to ±2 reference places, and a tolerance of ≈42 m for the Brisbane Event VPR dataset, which is equivalent to ±3 reference places. A visualization of the ground-truth, true-positive, and false-positive matches for each similarity matrix can be found in fig. S5.

Supplementary Materials

The PDF file includes:

Figs. S1 to S8
Tables S1 and S2
Legend for movie S1

Other Supplementary Material for this manuscript includes the following:

Movie S1
MDAR Reproducibility Checklist

REFERENCES AND NOTES

1. S. Schubert, P. Neubert, S. Garg, M. Milford, T. Fischer, Visual place recognition: A tutorial. *IEEE Robot. Autom. Mag.* **31**, 139–153 (2024).
2. X. Zhang, L. Wang, Y. Su, Visual place recognition: A survey from deep learning perspective. *Pattern Recognit.* **113**, 107760 (2021).
3. C. Masone, B. Caputo, A survey on deep visual place recognition. *IEEE Access* **9**, 19516–19547 (2021).
4. G. Berton, C. Masone, B. Caputo, "Rethinking visual geo-localization for large-scale applications," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (IEEE, 2022), pp. 4878–4888.
5. G. Neuhold, T. Ollmann, S. R. Bulò, P. Kontschieder, "The Mapillary Vistas dataset for semantic understanding of street scenes," in *IEEE International Conference on Computer Vision* (IEEE, 2017), pp. 5000–5009.
6. M. Cordts, M, Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, B. Schiele, "The Cityscapes dataset for semantic urban scene understanding," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (IEEE, 2016), pp. 3213–3223.
7. S. Izquierdo, J. Civera, "Optimal transport aggregation for visual place recognition," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (IEEE, 2024), pp. 17658–17668.
8. G. Orchard, E. Paxon Frady, Daniel Ben Dayan Rubin, S. Sanborn, S. B. Shrestha, F. T. Sommer, M. Davies, "Efficient neuromorphic signal processing with Loihi 2," in *IEEE Workshop on Signal Processing Systems* (IEEE, 2021), pp. 254–259.
9. E. Painkras, L. A. Plana, J. Garside, S. Temple, S. Davidson, J. Pepper, D. Clark, C. Patterson, S. Furber, "SpiNNaker: A multi-core system-on-chip for massively-parallel neural net simulation," in *IEEE Custom Integrated Circuits Conference* (IEEE, 2012), pp. 1–4.
10. L. Zhu, M. Mangan, B. Webb, Neuromorphic sequence learning with an event camera on routes through vegetation. *Sci. Robot.* **8**, eadg3679 (2023).
11. T. van Dijk, C. D. Wagter, G. C. H. E. de Croon, Visual route following for tiny autonomous robots. *Sci. Robot.* **9**, eadk0310 (2024).
12. F. Yu, Y. Wu, S. Ma, M. Xu, H. Li, H. Qu, C. Song, T. Wang, R. Zhao, L. Shi, Brain-inspired multimodal hybrid neural network for robot place recognition. *Sci. Robot.* **8**, eabm6996 (2023).
13. K. Friston, The free-energy principle: A unified brain theory? *Nat. Rev. Neurosci.* **11**, 127–138 (2010).
14. D. Chen, N. Axmacher, L. Wang, Grid codes underlie multiple cognitive maps in the human brain. *Prog. Neurobiol.* **233**, 102569 (2024).
15. J. E. Lisman, O. Jensen, The theta-gamma neural code. *Neuron* **77**, 1002–1016 (2013).
16. C. Liu, R. Todorova, W. Tang, A. Oliva, A. Fernandez-Ruiz, Associative and predictive hippocampal codes support memory-guided behaviors. *Science* **382**, eadi8237 (2023).
17. M. J. Milford, G. F. Wyeth, Mapping a suburb with a single camera using a biologically inspired SLAM system. *IEEE Trans. Robot.* **24**, 1038–1053 (2008).
18. F. Yu, J. Shang, Y. Hu, M. Milford, NeuroSLAM: A brain-inspired SLAM system for 3D environments. *Biol. Cybern.* **113**, 515–545 (2019).
19. J. Wang, S. Lin, A. Liu, Bioinspired perception and navigation of service robots in indoor environments: A review. *Biomimetics* **8**, 350 (2023).
20. L. Wang, S. X. Yang, M. Biglarbegian, "Bio-inspired navigation of mobile robots," in *International Conference on Autonomous and Intelligent Systems* (Springer, 2012), pp. 59–68.
21. T. Joseph, T. Fischer, M. Milford, "Trajectory tracking via multiscale continuous attractor networks," in *IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE, 2023), pp. 368–375.
22. A. Tavanaei, M. Ghodrati, S. R. Kheradpisheh, T. Masquelier, A. Maida, Deep learning in spiking neural networks. *Neural Netw.* **111**, 47–63 (2019).
23. J. K. Eshraghian, M. Ward, E. O. Neftci, X. Wang, G. Lenz, G. Dwivedi, M. Bennamoun, D. S. Jeong, W. D. Lu, Training spiking neural networks using lessons from deep learning. *Proc. IEEE* **111**, 1016–1054 (2023).
24. S. Hussaini, M. Milford, T. Fischer, "Ensembles of compact, region-specific and regularized spiking neural networks for scalable place recognition," in *IEEE International Conference on Robotics and Automation* (IEEE, 2023), pp. 4200–4207.
25. S. Hussaini, M. J. Milford, T. Fischer, Spiking neural networks for visual place recognition via weighted neuronal assignments. *IEEE Robot. Autom. Lett.* **7**, 4094–4101 (2022).
26. S. Hussaini, M. Milford, T. Fischer, Applications of spiking neural networks in visual place recognition. *IEEE Trans. Robot.* **41**, 518–537 (2025).
27. A. D. Hines, P. G. Stratton, M. Milford, T. Fischer, "VPRTempo: A fast temporally encoded spiking neural network for visual place recognition," in *IEEE International Conference on Robotics and Automation* (IEEE, 2024), pp. 10200–10207.
28. N. S. Giraldo, S. Isaza, R. A. Velásquez, Sailboat navigation control system based on spiking neural networks. *Control Theory Technol.* **21**, 489–504 (2023).
29. J. Liu, H. Lu, Y. Luo, S. Yang, Spiking neural network-based multi-task autonomous learning for mobile robots. *Eng. Appl. Artif. Intell.* **104**, 104362 (2021).
30. G. Tang, A. Shah, K. P. Michmizos, "Spiking neural network on neuromorphic hardware for energy-efficient unidimensional SLAM," in *IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE, 2019), pp. 4176–4181.
31. F. Galluppi, J. Conradt, T. Stewart, C. Eliasmith, T. Horiuchi, J. Tapson, B. Tripp, S. Furber, R. Etienne-Cummings, "Live demo: Spiking ratSLAM: Rat hippocampus cells in spiking neural hardware," in *IEEE Biomedical Circuits and Systems Conference* (IEEE, 2012), pp. 91–91.

32. G. Q. Bi, M. M. Poo, Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type. *J. Neurosci.* **18**, 10464–10472 (1998).
33. 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).
34. C. Bartolozzi, G. Indiveri, E. Donati, Embodied neuromorphic intelligence. *Nat. Commun.* **13**, 1024 (2022).
35. G. Gallego, T. Delbruck, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. J. Davison, J. Conradt, K. Daniilidis, D. Scaramuzza, Event-based vision: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **44**, 154–180 (2022).
36. N. Russo, H. Huang, E. Donati, T. Madsen, K. Nikolic, An interface platform for robotic neuromorphic systems. *Chips* **2**, 20–30 (2023).
37. Y. Yang, C. Bartolozzi, H. H. Zhang, R. A. Nawrocki, Neuromorphic electronics for robotic perception, navigation and control: A survey. *Eng. Appl. Artif. Intell.* **126**, 106838 (2023).
38. F. Akopyan, J. Sawada, A. Cassidy, R. Alvarez-Icaza, J. Arthur, P. Merolla, N. Imam, Y. Nakamura, P. Datta, G. J. Nam, B. Taba, M. Beakes, B. Brezzo, J. B. Kuang, R. Manohar, W. P. Risk, B. Jackson, D. S. Modha, TrueNorth: Design and tool flow of a 65 mW 1 million neuron programmable neurosynaptic chip. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **34**, 1537–1557 (2015).
39. J. Pei, L. Deng, S. Song, M. Zhao, Y. Zhang, S. Wu, G. Wang, Z. Zou, Z. Wu, W. He, F. Chen, N. Deng, S. Wu, Y. Wang, Y. Wu, Z. Yang, C. Ma, G. Li, W. Han, H. Li, H. Wu, R. Zhao, Y. Xie, L. Shi, Towards artificial general intelligence with hybrid Tianjic chip architecture. *Nature* **572**, 106–111 (2019).
40. B. J. Shastri, A. N. Tait, Thomas Ferreira de Lima, M. A. Nahmias, H.-T. Peng, P. R. Prucnal, "Principles of neuromorphic photonics," in *Encyclopedia of Complexity and Systems Science* (Springer, 2018), pp. 83–118.
41. F. Paredes-Vallés, J. J. Hagenaars, J. Dupeyroux, S. Stroobants, Y. Xu, G. C. H. E. de Croon, Fully neuromorphic vision and control for autonomous drone flight. *Sci. Robot.* **9**, eadi0591 (2024).
42. R. Halaly, E. E. Tsur, Autonomous driving controllers with neuromorphic spiking neural networks. *Front. Neurorobot.* **17**, 1234962 (2023).
43. A. Linares-Barranco, F. Perez-Peña, A. Jimenez-Fernandez, E. Chicca, ED-BioRob: A neuromorphic robotic arm with FPGA-based infrastructure for bio-inspired spiking motor controllers. *Front. Neurorobot.* **14**, 590163 (2020).
44. M. Yao, O. Richter, G. Zhao, N. Qiao, Y. Xing, D. Wang, T. Hu, W. Fang, T. Demirci, M. de Marchi, L. Deng, T. Yan, C. Nielsen, S. Sheik, C. Wu, Y. Tian, B. Xu, G. Li, Spike-based dynamic computing with asynchronous sensing-computing neuromorphic chip. *Nat. Commun.* **15**, 4464 (2024).
45. T. Fischer, M. Milford, Event-based visual place recognition with ensembles of temporal windows. *IEEE Robot. Autom. Lett.* **5**, 6924–6931 (2020).
46. M. J. Milford, G. F. Wyeth, "SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights," in *IEEE International Conference on Robotics and Automation* (IEEE, 2012), pp. 1643–1649.
47. 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).
48. 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)* (IEEE, 2021), pp. 2247–2252.
49. H. Lee, H. Hwang, Ev-ReconNet: Visual place recognition using event camera with spiking neural networks. *IEEE Sens. J.* **23**, 20390–20399 (2023).
50. P. Lichtsteiner, C. Posch, T. Delbruck, A 128×128 120 dB 15 μs latency asynchronous temporal contrast vision sensor. *IEEE J. Solid-State Circuits* **43**, 566–576 (2008).
51. S. Bian, L. Schulthess, G. Rutishauser, A. D. Mauro, L. Benini, M. Magno, "ColibriUAV: An ultra-fast, energy-efficient neuromorphic edge processing UAV-platform with event-based and frame-based cameras," in *IEEE International Workshop on Advances in Sensors and Interfaces* (IEEE, 2023), pp. 287–292.
52. E. Kelasidi, K. Y. Pettersen, J. T. Gravdahl, Energy efficiency of underwater robots. *IFAC Conference on Manoeuvring and Control of Marine Craft* **48**, 152–159 (2015).
53. N. Kashiri, A. Abate, S. J. Abram, A. Albu-Schaffer, P. J. Clary, M. Daley, S. Faraji, R. Furnemont, M. Garabini, H. Geyer, A. M. Grabowski, J. Hurst, J. Malzahn, G. Mathijssen, D. Remy, W. Roozing, M. Shahbazi, S. N. Simha, J. B. Song, N. Smit-Anseeuw, S. Stramigioli, B. Vanderborght, Y. Yesilevskiy, N. Tsagarakis, An overview on principles for energy efficient robot locomotion. *Front. Robot. AI* **5**, 129 (2018).
54. R. Islam, H. Habibullah, "A semantically aware place recognition system for loop closure of a visual SLAM system," in *2021 4th International Conference on Mechatronics, Robotics and Automation (ICMRA)* (IEEE, 2021), pp. 117–121.
55. R. Mur-Artal, J. M. M. Montiel, J. D. Tardós, ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Trans. Robot.* **31**, 1147–1163 (2015).
56. S. Yang, C. Zhao, Z. Wu, Y. Wang, G. Wang, D. Li, Visual SLAM based on semantic segmentation and geometric constraints for dynamic indoor environments. *IEEE Access* **10**, 69636–69649 (2022).
57. G. B. Nair, M. Milford, T. Fischer, "Enhancing visual place recognition via fast and slow adaptive biasing in event cameras," in *IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE, 2024), pp. 3356–3363.
58. Y. Wu, R. Zhao, J. Zhu, F. Chen, M. Xu, G. Li, S. Song, L. Deng, G. Wang, H. Zheng, S. Ma, J. Pei, Y. Zhang, M. Zhao, L. Shi, Brain-inspired global-local learning incorporated with neuromorphic computing. *Nat. Commun.* **13**, 65 (2022).
59. R. Zhu, S. Lilak, A. Loeffler, J. Lizier, A. Stieg, J. Gimzewski, Z. Kuncic, Online dynamical learning and sequence memory with neuromorphic nanowire networks. *Nat. Commun.* **14**, 6697 (2023).
60. A. Rostami, B. Vogginger, Y. Yan, C. G. Mayr, E-prop on SpiNNaker 2: Exploring online learning in spiking RNNs on neuromorphic hardware. *Front. Neurosci.* **16**, 1018006 (2022).
61. S. Hwang, Y. Hwang, D. Kim, J. Lee, H. K. Choe, J. Lee, H. Kang, J. Kung, ReplaceNet: Real-time replacement of a biological neural circuit with a hardware-assisted spiking neural network. *Front. Neurosci.* **17**, 1161592 (2023).
62. T. Chen, S. Kornblith, M. Norouzi, G. Hinton, "A simple framework for contrastive learning of visual representations," in *International Conference on Machine Learning* (PMLR, 2020), pp. 1597–1607.
63. J. Dupeyroux, J. R. Serres, S. Viollet, AntBot: A six-legged walking robot able to home like desert ants in outdoor environments. *Sci. Robot.* **4**, eaau0307 (2019).
64. X. Sun, Q. Fu, J. Peng, S. Yue, An insect-inspired model facilitating autonomous navigation by incorporating goal approaching and collision avoidance. *Neural Netw.* **165**, 106–118 (2023).
65. M. Milford, G. Wyeth, D. Prasser, "RatSLAM: A hippocampal model for simultaneous localization and mapping," in *IEEE International Conference on Robotics and Automation* (IEEE, 2004), pp. 403–408.
66. N. S.-Y. Dumont, P. M. Furlong, J. Orchard, C. Eliasmith, Exploiting semantic information in a spiking neural SLAM system. *Front. Neurosci.* **17**, 1190515 (2023).
67. T. Zeng, F. Tang, D. Ji, B. Si, NeuroBayesSLAM: Neurobiologically inspired Bayesian integration of multisensory information for robot navigation. *Neural Netw.* **126**, 21–35 (2020).
68. L. Liu, J. Chen, M. Brocanelli, W. Shi, "E2M: An energy-efficient middleware for computer vision applications on autonomous mobile robots," in *ACM/IEEE Symposium on Edge Computing* (IEEE, 2019), pp. 59–73.
69. P. G. Stratton, A. Wabnitz, C. Essam, A. Cheung, T. J. Hamilton, Making a Spiking Net Work: Robust brain-like unsupervised machine learning. arXiv:2208.01204 [cs.NE] (2022).
70. W. Guo, M. E. Fouda, A. M. Eltawil, K. N. Salama, Neural coding in spiking neural networks: A comparative study for robust neuromorphic systems. *Front. Neurosci.* **15**, 638474 (2021).
71. S. Garg, M. Vankadari, M. Milford, "SeqMatchNet: Contrastive learning with sequence matching for place recognition and relocalization," in *Proceedings of the 5th Conference on Robot Learning* (PMLR, 2022), pp. 429–443.

Acknowledgments: We acknowledge continued support from the Queensland University of Technology (QUT) through the Centre for Robotics. We wish to acknowledge the support of the Research Engineering Facility (REF) team at QUT for the provision of engineering support, expertise, and research infrastructure in enablement of this project. We also thank S. Raine and S. Hussaini for feedback during initial manuscript preparation. Last, we thank the organizers and participants of the 2022 Lifelong Learning at Scale topic area at the Telluride Neuromorphic Workshop for the discussions and inspiring environment, which helped shape some of the ideas explored in this paper. **Funding:** This work received funding from ARC Laureate Fellowship FL210100156 to M.M., AUSMURIB000001 associated with ONR MURI grant N00014191-2571 to M.M. and T.F., and ARC Discovery Early Career Researcher Award DE240100149 to T.F. **Author contributions:** A.D.H. and T.F. conceptualized and designed the experiments. A.D.H. performed and analyzed all experiments. T.F. and M.M. provided general input into the research. T.F. and M.M. provided funding for the project. A.D.H. wrote the paper. A.D.H., M.M., and T.F. edited the paper. **Competing interests:** The authors declare that they have no competing interests. **Data and materials availability:** All data needed to support the conclusions of this manuscript are included in the main text and Supplementary Materials and are also available alongside the code at https://doi.org/10.5281/zenodo.15392412.

Submitted 22 August 2024
Accepted 21 May 2025
Published 18 June 2025
10.1126/scirobotics.ads3968

Science Robotics

A compact neuromorphic system for ultra–energy-efficient, on-device robot localization

Adam D. Hines, Michael Milford, and Tobias Fischer

Sci. Robot. **10** (103), eads3968. DOI: 10.1126/scirobotics.ads3968

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.