2025 IEEE 7th International Conference on Artificial Intelligence Circuits and Systems (AICAS) | 979-8-3315-2424-1/25/$31.00 ©2025 IEEE | DOI: 10.1109/AICAS64808.2025.11173128

Enhancing Ant-Inspired Visual Compass with Focused Visual Scan in a Compact Robot

Gattaux Gabriel[1], Antoine Wystrach[2], Franck Ruffier[1], Julien Serres[1,3]
Email: gabriel.gattaux@univ-amu.fr
[1]Aix Marseille Univ, CNRS, ISM, Marseille, France
[2]Univ Toulouse, CRCA, CBI, UMR CNRS-UPS 5169, Toulouse, France
[3]Institut Universitaire de France, IUF, Paris, France

***Abstract*—As the demand for autonomous robots grows, from self-driving cars to factory automation, so does the need for resource-efficient navigation algorithms. This challenge has inspired the development of sparse neural networks for visual navigation, drawing particularly from bio-inspired, ant-based neuromorphic artificial intelligence algorithms. Recent approaches leverage the neural architecture of the mushroom body in ants to learn and follow routes by recognizing familiar visual patterns. However, such models have not yet been implemented in closed-loop, resource-constrained robotic systems with satisfying performance. In this work, we present a mushroom body-inspired model embedded in the Antcar, a compact, car-like robot that processes panoramic images at low resolution. Our experiments reveal that reducing the scanning range and resolution improves visual compass performance for robot's route-following. These findings highlight the trade-off between increased refresh rates and the benefits of selective perception, paving the way for more efficient real-time decision-making in resource-constrained systems.**

***Index Terms*—Bio-inspired robotics, neuromorphic, neural network, computer vision, navigation.**

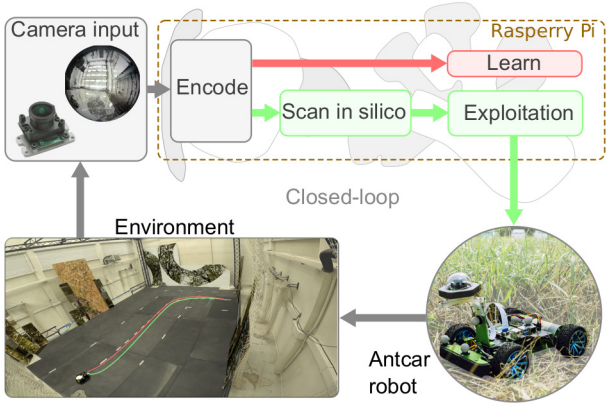

Fig. 1. The Antcar robot encodes and learns panoramic images online along a predefined route (red) in the Mediterranean Flight Arena using a mushroom body-inspired neural structure. During autonomous operation, it retraces the route by performing in silico scans of the images at various orientations. All computations are performed onboard the Antcar robot using a Raspberry Pi.

I. Introduction

In a world moving toward automation, the demand for low-energy, cost-effective, and easily deployable navigation systems with minimal memory footprints is growing, especially in applications ranging from search-and-rescue missions to space exploration and beyond [1], [2]. To address the dependency on external systems like satellites for Global Navigation Satellite System (GNSS)-based navigation and the drift issues in odometry-based systems, visual teach-and-repeat methods are emerging as a promising approach for autonomous mobile robots [3]. Similarly, insect-inspired models of visual navigation converge in this direction, highlighting the parsimonious nature of learning-based visual navigation [4].

Solitary foraging ants, in particular, exhibit remarkable learning and route-following abilities relative to their small size [5]. They use the mushroom body (MB) part of their brain to learn low-resolution, quasi-panoramic views to navigate on familiar routes, without relying on path integration [6], [7]. While several MB neural models have been developed as associative network, the precise way ants use this brain region to control motor output remains a topic of discussion [8]–[10]. Zeil et al. [11] first proposed that ants may move in the direction that maximizes a familiarity index relative to stored images along a route. This familiarity index was originally computed using the pixelwise root mean square difference (RMS) between the current view and stored raw images by scanning in all orientations. The process of moving in the direction of maximum familiarity has been tested in robots using either RMS or Infomax networks, forming the basis of the visual compass (VC) model [12]–[14].

Despite advancements in visual teach-and-repeat algorithms, many remain computationally intensive [15], relying on GPUs and limiting their use on resource-constrained platforms. To address this, we build on the MB neural model [10] and propose an enhanced visual compass for Antcar, a compact 25×20 cm car-like robot, inspired by a recent preprint [16]. his preliminary study introduced the first real-time integration of an MB model in a mobile robot, demonstrating its potential but also revealing challenges, such as oscillations during autonomous trajectories.

While spiking MB models, neuromorphic hardware, and event-based cameras show promise, widely accessible platforms like the Raspberry Pi and omnidirectional RGB cameras still offer significant optimization potential due to their maturity and accessibility. Studies using SpiNNaker have shown improvements in spiking MB models through offline analysis [17], but these solutions remain complex for real-

time robotics [18], even when implemented onboard [19]. To meet the demands of online, one-shot learning and real-time operation, our approach focuses on improving the quality of visual information for navigation without increasing system complexity.

This paper presents a parsimonious mushroom body neural network (Fig. 1) designed for visual teach-and-repeat navigation in a closed-loop system on a car-like robot, enabling it to follow an indoor L-shaped route without stop. Furthermore, we demonstrate that narrowing the angular scan range while increasing resolution significantly reduces unwanted oscillations and enhances Antcar's performance compared to previous work [16].

In Section II, we present the Antcar robot, its components, and the MB neural network, detailing the encoding, learning, and exploitation phases. Section III provides a vector-field representation of familiar terrain and analyzes real-time results, comparing Antcar 1.0 [16] and Antcar 2.0. The advantages of our approach are contrasted with a recent familiarity-based navigation model. Finally, Section IV summarizes our findings and outlines future directions for robotic navigation.

II. Methods

A. The Antcar robot

To implement our frugal visual learning strategy, we used the Antcar robot, a 25-cm mobile platform with DC-powered rear wheels and a servo-steered front. A Raspberry Pi 4B (Broadcom BCM2711, CPU quad-core Cortex-A72, 1.5GHz) handled all processing, efficiently managing tasks with low power consumption. The Pi communicated via I2C with a motor driver hat using a PCA9685 module. The robot was tested in the Mediterranean Flight Arena[1] (Fig. 1), where ground truth data was captured by a VICON™ motion capture system with 17 infrared cameras covering a $8 \times 6 \times 6$ m area. Communication between the base station, robot, and motion capture system was managed through a ROS network. Power is supplied by three rechargeable 18650 batteries (2600mAh) with a combined output voltage of 12.6V.

Image acquisition was handled by an Entaniya™ fisheye camera, mounted on the robot's head, providing a 220° vertical and 360° horizontal field of view. Operating at 30Hz, the images were processed directly without panoramic expansion to reduce computational load. All code, from image acquisition and neural network (Fig. 2) to motor control, was written in Python ROS node for simplicity and ease of implementation.

B. MB-like neural network architecture

The navigation algorithm, inspired by the biological mushroom body and implemented in the MB-like network (Fig. 2), has three phases: encoding, learning, and exploitation.

Encoding phase: The encoding phase operated during both learning and exploitation, transforming raw images into a compact hash suitable for further processing. First, the green channel of each 160×160-pixel image captured by the camera

[1]https://tirrex.fr/plateforme/avm/

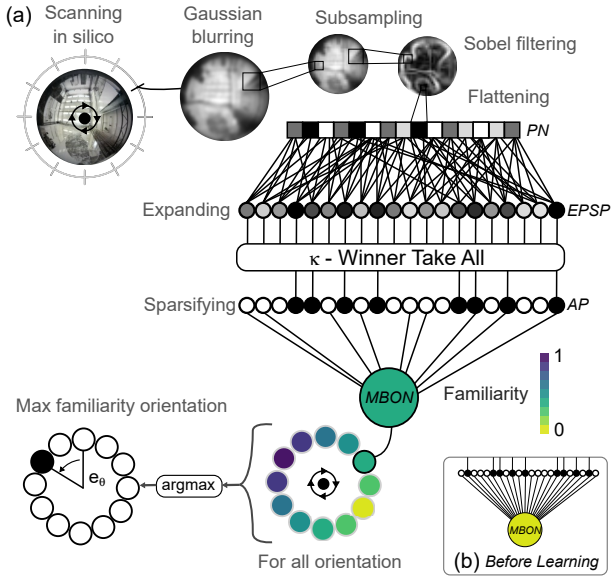

Fig. 2. (a) Architecture of the mushroom body (MB)-like neural network during exploitation (MBON stands for Mushroom Body Output Neurons; EPSP for Excitatory Post Synaptic Projections vector; AP for Action Potential; PN for Projection Neurons). Panoramic images are rotated in silico across all orientations and processed through the network. The familiarity for each orientation is calculated as a weighted average, and the direction with the highest familiarity is selected. This process yields the egocentric angular error (e_θ) between the agent and the predefined route. (b) Synaptic weight matrix before learning, illustrating the initial fully connected state. Learning occurs through long-term synaptic depression, progressively reducing connections.

was extracted to mimic the ant's green-sensitive photoreceptor [20]. The image was then blurred using a Gaussian filter with a standard deviation (σ) of 3 pixels to reduce high-frequency noise. Subsequently, the image was downsampled to 32×32 pixels using nearest-neighbor interpolation, achieving a visual acuity of 0.145 pixel per degree (PPD), similar to that of ants. A Sobel filter was applied to enhance edges, eliminating unnecessary information. The resulting circular image was then flattened to form approximately 785 Projection Neurons (PN). Next, the dimensionality of this PN vector was expanded using a fixed pseudo-random synaptic weight matrix, creating the Excitatory Post Synaptic Projections vector (EPSP). Four PN were connected to each EPSP neuron, resulting in a 10,000 neurons representation, analogous to the Kenyon Cells in the MB. Then, the representation was sparsified using a κ-Winner Take All function, which retained only the most active 1% of EPSP neurons. This produced a high-dimensional hash of the image, with only hundreds of active binary states (1s) out of thousands of otherwise silent Action Potential (AP) neurons, resulting in a highly efficient and sparsified representation.

Learning phase: During the learning phase, the robot navigated a predefined route, which could be accomplished using any standard navigation algorithm. As it encountered new views, the images were encoded. These images hashes were continuously learned via anti-Hebbian synaptic depression in the synaptic weight vector connecting the AP neurons to the Mushroom Body Output Neurons (MBON). Initially, this AP-

to-MBON weight matrix was fully connected. As the robot encountered new image hashes, the corresponding synaptic weights were depressed (set to zero), progressively sparsifying the weight matrix. This reflected the growing diversity of images the robot learned as it explored.

Exploitation phase: In the exploitation (or autonomous) phase, the hash of the current view was compared to the learned memory (AP-to-MBON matrix) using a weighted average between the current sparsified AP neurons and the synaptic weight matrix AP-to-MBON. This computation produced a familiarity index in the MBON, ranging from 0 to 1, where a value of 1 indicated a familiar view, and a value of 0 indicated an unfamiliar view. A key feature of the exploitation process was the internal rotation of the image, simulating a scan across different orientations (Fig. 2). Two parameters controlled this scan: the scanning range and the scanning resolution. For each rotation, the familiarity value was computed and temporarily stored. Finally, the direction that maximized the familiarity value was determined using an argmax function and was employed to control the robot's steering angle. This angle corresponded to the egocentric angular error (e_θ) between the current view and the learned route.

III. Results

A. The visual compass vector field

To evaluate the network's ability to effectively learn and follow a route, we tested offline how well it could identify the direction that maximizes familiarity. The tests were conducted using a 360-degree scanning range and a 1-degree scan resolution, with an L-shaped indoor learning route fitted with artificial visual cues along the scene boundaries (Fig. 3 and video[2]). The route learning dataset spanned approximately 6 meters, during which 134 images were learned (Fig. 3b). The exploitation dataset was composed of images at 450 different positions with 360 orientations (totaling 162,000 poses) both on and around the learned route. For the majority of the tested positions, the familiarity index peaked when aligned with the learned route's orientation (Fig. 3a, circular insets), producing a vector field map with arrows pointing in the most familiar directions (Fig. 3a). Additionally, a gradient of familiarity was observed not only when rotating (Fig. 3a, circular insets) but also when translating, illustrating a canyon-like pattern that reflects both angular and distance variations from the learned route. These results validate the use of this neural network, as the familiarity index behaves similarly to state-of-the-art familiarity assessment functions (Perfect memory and Infomax).

This method demonstrated significant potential for efficiently learning visual cues, particularly in terms of parsimonious memory usage. For example, the AP-to-MBON synaptic weight matrix can be stored in a Compressed Sparse Row (CSR) format, which reduces the memory footprint by 99.2% compared to storing raw images (perfect memory), shrinking from 13.38 Mb to just 103 kb. Even when compared to saving the raw memory vectors directly (321 kb), the CSR format still achieves a 32% memory reduction. Additionally, the system's ability to avoid overtraining was evaluated, ensuring it did not relearn identical or highly similar images. Out of 134 images learned, 50 were captured while the robot remained stationary (Fig. 3b). Crucially, the AP-to-MBON sparsity remained unchanged when the network encountered identical images. As the robot moved and encountered distinct images, the sparsity increased without plateauing, reflecting the learning process (Fig. 3b).

[2]https://youtu.be/iRewX9KLJDQ

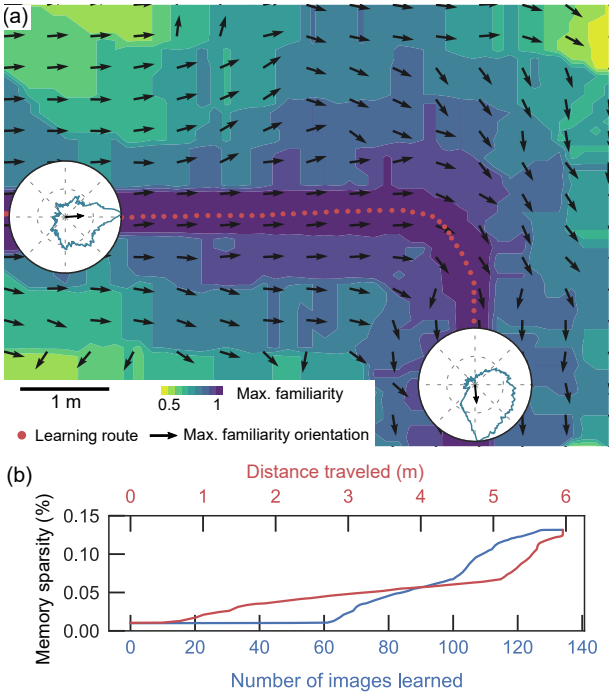

Fig. 3. (a) Post-processed familiarity vector field representation after learning an L-shaped route. The arrows represent the directions that maximize familiarity, with the corresponding familiarity values shown in the background. The insets display complete 360° familiarity scans at specific positions. (b) Sparsity of the synaptic weights during learning, plotted against the number of images learned and the distance traveled.

B. Real-time experiments

To validate our approach in real-world conditions, we conducted a series of real-time experiments that demonstrated the system's ability to perform online learning and autonomous route following (Fig. 4). These tests were carried out in the Mediterranean Flight Arena under similar conditions, with consistent network hyper-parameters and a constant speed of 0.3 m/s during both the learning and exploitation phases. In each experiment, the robot followed a predefined 7.5-meter route, processing and learning images at a rate of 16Hz (one image every 62ms). The key difference between the two experiments lay in the scanning strategies employed (Figs. 4c-d), underscoring the importance of effective visual sampling management to enhance performance. The following paragraph details the differences in performance between Antcar

1.0 and Antcar 2.0. A visualization of the experiments and environment can be viewed in the video.

Antcar 1.0: This first experiment involved 11 autonomous trajectories, covering 96 meters along an L-shaped learning route (Fig. 4a), with artificial visual cues placed along the walls. Visual acuity for this experiment was 0.2 PPD, using a thumbnail size of 44×44 pixels. The scanning range was fixed at 200 degrees and the scanning resolution at 20 degrees, leading to a computational load of 10 computations per decision (Fig. 4c). This resulted in a refresh rate of 3.4Hz during autonomous routes. The robot demonstrated good accuracy in following the learned route, as reflected by the lateral error (Median: 0.06 m ± 0.31 m, median absolute deviation (MAD)). However, its trajectories were characterized by unwanted oscillations, as confirmed by the high magnitude observed in the power spectrum density of the robot's angular speed (Figs. 4a,e). The angular deviation (Median: 0.50 degrees ± 22.16 degrees, MAD) further underlines the oscillatory movement throughout experiments.

Antcar 2.0: The second experiment involved 13 autonomous trajectories, covering 104 meters along a similar L-shaped learning route (Fig. 4b) in a comparable environment. The computational load increased to 18 due to a reduced scan range of 90° and a scan resolution of 5° during the exploitation phase (Fig. 4d). To balance the increased computational demand, visual acuity was lowered to 0.14 PPD (32×32 pixels), highlighting a trade-off between reduced visual acuity and a higher number of scans per action. Despite a slower refresh rate of 2.4Hz, these adjustments led to significant improvements. The narrower scan resolution and range fully eliminated the oscillations observed in Antcar 1.0 (Fig. 4b, 4e). Antcar 2.0 achieved greater route-following precision, with a median lateral deviation of -0.01 m ± 0.18 m (Median±MAD). Improved angular accuracy (Median: 3.36° ± 4.88°, MAD) and the elimination of oscillations demonstrate the effectiveness of selective perception in enhancing precision and navigation performance, even with increased computational demands.

C. *Efficiency of familiarity-based network*

Antcar 1.0 [16] introduced a mushroom body (MB) model for parsimonious route-following in robots. With Antcar 2.0, we demonstrated that selective perception significantly enhances performance, eliminating unwanted oscillations. Since the Antcar 1.0, a recent study implemented a spiking MB model in a car-like robot using a GPU [21], achieving a mean lateral error of 0.14 m ± 0.08 m and a mean angular deviation of 12°, though with a slower refresh rate compared to infomax or perfect memory methods. Antcar 2.0 redefines affordability and accessibility in robotic navigation. Powered by a Raspberry Pi CPU and a Python-implemented MB model, it achieved a lateral error of -0.01 m ± 0.18 m and an angular deviation of 3.36° ± 4.88° at a refresh rate of 2.4Hz. The Raspberry Pi consumed just 3W while running the camera and neural network, with the latter using 500mW, 200MB of RAM, and full utilization of all four CPU cores. These results showcase an exceptional balance between cost and efficiency,

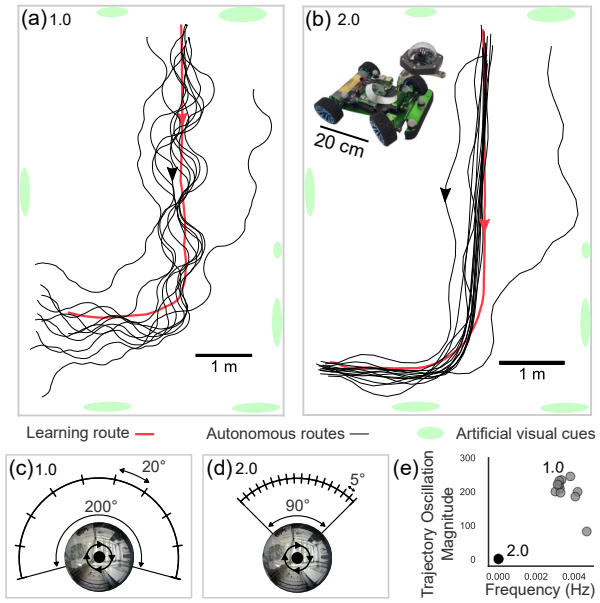

Fig. 4. (a) Real-time experiment on an L-shaped route with 11 autonomous trajectories and a 300 ms latency, showing oscillatory movements using Antcar 1.0 parameters (c). (b) Real-time experiment with 13 trajectories and a 400 ms latency, demonstrating smooth route following using Antcar 2.0 parameters (d). Both employed online learning and exploitation. (c) Scanning parameters for Antcar 1.0. (d) Scanning parameters for Antcar 2.0. (e) Trajectory oscillations are nearly eliminated with a more focused visual scan.

achieving performance comparable to GPU-powered spiking networks.

IV. Conclusion

In this paper, we demonstrated a closed-loop visual learning network and showed that focusing perception in an ant-inspired neural network significantly improves visual compass performance for route-following. Antcar 2.0 eliminated oscillations, achieved precise navigation, and delivered competitive performance with a low-cost setup, even compared to GPU-powered alternatives. Our approach reduced the memory footprint by 99.2% compared to storing raw images. By narrowing the scan range from ±100° to ±45° and increasing the scan resolution from 20° to 5°, we enhanced system stability without sacrificing accuracy. This architectural adjustment addressed oscillations previously attributed to slower refresh rates, highlighting the importance of optimized visual sampling over simply increasing temporal refresh rates. As a result, we emphasized here the trade-off between scan resolution and scan amplitude to limit the computational load. These findings highlight the value of parsimonious, low-cost solutions in autonomous navigation. The perception-focused architecture developed here can be applied to resource-constrained systems such as drones or planetary rovers and is compatible with both spiking and non-spiking insect-inspired AI. Future work will explore extending this model to longer routes in more complex environments and integrating opponent processes for familiarity assessment to further reduce computational demands.

REFERENCES

[1] G. C. H. E. De Croon, J. J. G. Dupeyroux, S. B. Fuller, and J. A. R. Marshall, "Insect-inspired AI for autonomous robots," *Science Robotics*, vol. 7, no. 67, p. eabl6334, Jun. 2022.

[2] P. E. Glick, J. B. Balaram, M. R. Davidson, E. Lyons, and M. T. Tolley, "The role of low-cost robots in the future of spaceflight," *Science Robotics*, vol. 9, no. 91, p. eadl1995, Jun. 2024.

[3] M. Simon, G. Broughton, T. Rouček, Z. Rozsypálek, and T. Krajník, "Performance Comparison of Visual Teach and Repeat Systems for Mobile Robots," in *Modelling and Simulation for Autonomous Systems*, J. Mazal, A. Fagiolini, P. Vašík, A. Bruzzone, S. Pickl, V. Neumann, P. Stodola, and S. Lo Storto, Eds. Cham: Springer International Publishing, 2023, vol. 13866, pp. 3–24.

[4] M. Mangan, D. Floreano, K. Yasui, B. A. Trimmer, N. Gravish, S. Hauert, B. Webb, P. Manoonpong, and N. Szczecinski, "A virtuous cycle between invertebrate and robotics research: Perspective on a decade of Living Machines research," *Bioinspiration & Biomimetics*, vol. 18, no. 3, p. 035005, May 2023.

[5] M. Kohler and R. Wehner, "Idiosyncratic route-based memories in desert ants, Melophorus bagoti: How do they interact with path-integration vectors?" *Neurobiology of Learning and Memory*, vol. 83, no. 1, pp. 1–12, Jan. 2005.

[6] S. Heinze, "Visual Navigation: Ants Lose Track without Mushroom Bodies," *Current Biology*, vol. 30, no. 17, pp. R984–R986, Sep. 2020.

[7] A. Wystrach, A. Dewar, A. Philippides, and P. Graham, "How do field of view and resolution affect the information content of panoramic scenes for visual navigation? A computational investigation," *Journal of Comparative Physiology A*, vol. 202, no. 2, pp. 87–95, Feb. 2016.

[8] B. Webb and A. Wystrach, "Neural mechanisms of insect navigation," *Current Opinion in Insect Science*, vol. 15, pp. 27–39, Jun. 2016.

[9] T. S. Collett and M. Collett, "Memory use in insect visual navigation," *Nature Reviews Neuroscience*, vol. 3, no. 7, pp. 542–552, Jul. 2002.

[10] P. Ardin, F. Peng, M. Mangan, K. Lagogiannis, and B. Webb, "Using an Insect Mushroom Body Circuit to Encode Route Memory in Complex Natural Environments," *PLOS Computational Biology*, vol. 12, no. 2, p. e1004683, Feb. 2016.

[11] J. Zeil, M. I. Hofmann, and J. S. Chahl, "Catchment areas of panoramic snapshots in outdoor scenes," *Journal of the Optical Society of America A*, vol. 20, no. 3, p. 450, Mar. 2003.

[12] B. Baddeley, P. Graham, P. Husbands, and A. Philippides, "A Model of Ant Route Navigation Driven by Scene Familiarity," *PLoS Computational Biology*, vol. 8, no. 1, p. e1002336, Jan. 2012.

[13] D. D. Gaffin and B. P. Brayfield, "Autonomous Visual Navigation of an Indoor Environment Using a Parsimonious, Insect Inspired Familiarity Algorithm," *PLOS ONE*, vol. 11, no. 4, p. e0153706, Apr. 2016.

[14] J. C. Knight, D. Sakhapov, N. Domcsek, A. D. Dewar, P. Graham, T. Nowotny, and A. Philippides, "Insect-inspired visual navigation on-board an autonomous robot: Real-world routes encoded in a single layer network," in *Artificial Life Conference Proceedings*. MIT Press One Rogers Street, Cambridge, MA 02142-1209, USA journals-info . . . , 2019, pp. 60–67.

[15] P. Furgale and T. D. Barfoot, "Visual teach and repeat for long-range rover autonomy," *Journal of field robotics*, vol. 27, no. 5, pp. 534–560, 2010.

[16] G. Gattaux, R. Vimbert, A. Wystrach, J. R. Serres, and F. Ruffier, "Antcar: Simple Route Following Task with Ants-Inspired Vision and Neural Model," *HAL (preprint)*, 2023.

[17] L. Zhu, M. Mangan, and B. Webb, "Neuromorphic sequence learning with an event camera on routes through vegetation," *Science Robotics*, vol. 8, no. 82, p. eadg3679, Sep. 2023.

[18] M. Aitsam, S. Davies, and A. Di Nuovo, "Neuromorphic Computing for Interactive Robotics: A Systematic Review," *IEEE Access*, vol. 10, pp. 122 261–122 279, 2022.

[19] C. Denk, F. Llobet-Blandino, F. Galluppi, L. A. Plana, S. Furber, and J. Conradt, "Real-time interface board for closed-loop robotic tasks on the spinnaker neural computing system," in *Artificial Neural Networks and Machine Learning–ICANN 2013: 23rd International Conference on Artificial Neural Networks Sofia, Bulgaria, September 10-13, 2013. Proceedings 23*. Springer, 2013, pp. 467–474.

[20] V. Aksoy and Y. Camlitepe, "Behavioural analysis of chromatic and achromatic vision in the ant formica cunicularia (hymenoptera: Formicidae)," *Vision research*, vol. 67, pp. 28–36, 2012.

[21] O. O. Jesusanmi, A. A. Amin, N. Domcsek, J. C. Knight, A. Philippides, T. Nowotny, and P. Graham, "Investigating visual navigation using spiking neural network models of the insect mushroom bodies," *Frontiers in Physiology*, vol. 15, p. 1379977, May 2024.