

A neural network model for familiarity and context learning during honeybee foraging flights

Jurek Müller¹ · Martin Nawrot² · Randolph Menzel³ · Tim Landgraf¹ 

Received: 6 March 2017 / Accepted: 30 August 2017 / Published online: 15 September 2017
© Springer-Verlag GmbH Germany 2017

Abstract How complex is the memory structure that honeybees use to navigate? Recently, an insect-inspired parsimonious spiking neural network model was proposed that enabled simulated ground-moving agents to follow learned routes. We adapted this model to flying insects and evaluate the route following performance in three different worlds with gradually decreasing object density. In addition, we propose an extension to the model to enable the model to associate sensory input with a behavioral context, such as foraging or homing. The spiking neural network model makes use of a sparse stimulus representation in the mushroom body and reward-based synaptic plasticity at its output synapses. In our experiments, simulated bees were able to navigate correctly even when panoramic cues were missing. The context extension we propose enabled agents to successfully discrim-

inate partly overlapping routes. The structure of the visual environment, however, crucially determines the success rate. We find that the model fails more often in visually rich environments due to the overlap of features represented by the Kenyon cell layer. Reducing the landmark density improves the agents route following performance. In very sparse environments, we find that extended landmarks, such as roads or field edges, may help the agent stay on its route, but often act as strong distractors yielding poor route following performance. We conclude that the presented model is valid for simple route following tasks and may represent one component of insect navigation. Additional components might still be necessary for guidance and action selection while navigating along different memorized routes in complex natural environments.

This article belongs to a Special Issue on Neural Coding.

Electronic supplementary material The online version of this article (doi:[10.1007/s00422-017-0732-z](https://doi.org/10.1007/s00422-017-0732-z)) contains supplementary material, which is available to authorized users.

✉ Tim Landgraf
landgraf@inf.fu-berlin.de

Jurek Müller
jurek_mueller@yahoo.de

Martin Nawrot
mnawrot@uni-koeln.de

Randolf Menzel
menzel@neurobiologie.fu-berlin.de

¹ Institute for Computer Science, Free University Berlin, Berlin, Germany

² Computational Systems Neuroscience, Institute for Zoology, University of Cologne, Cologne, Germany

³ Institute for Neurobiology, Free University Berlin, Berlin, Germany

Keywords Insect navigation · Mushroom body · Spiking neural network model · Learning and plasticity · Artificial agent · Insect cognition

1 Introduction

Within the insect world, honeybees exhibit extraordinary navigational capabilities. Experimental evidence for different strategies, such as path integration and visual guidance using picture memories, has been presented (Collett and Collett 2002; Srinivasan 2014). However, it remains controversial how those components are combined and at which level of abstraction the different components are available to a navigating bee (Cruse and Wehner 2011; Cheung et al. 2014; Menzel and Greggers 2015).

Studies using harmonic radar suggest that bees can robustly find their nest, even when invalidating the path integrator through displacing the animal in a dark box, or when

disturbing the sun compass through pausing the internal clock using anesthesia (Menzel and Manz 2005; Cheeseman et al. 2014). A unique perspective on the internal representation of locations is provided by the waggle dance communication system with which foragers can direct nestmates to field locations (von Frisch 1967). After decoding a dance, honeybees have been shown to perform shortcut flights between known and dance-advertised sites over novel terrain, a behavior indicating that geometrical relationships between sites of interest are represented (Menzel et al. 2011).

Analytical approaches to investigate the neural correlates of navigation face a technological dilemma: To this date, there is no laboratory-based protocol available to study all aspects of long-range navigation in flying honeybees, e.g., using virtual environments in the laboratory as used in walking bees or flying fruit flies (for a review see Jacobs and Menzel 2014). Due to its extraordinary learning capabilities, the honeybee is the most popular insect model for the investigation of insect cognition and complex forms of learning (Menzel and Giurfa 2001; Menzel 2012; Avarguès-Weber and Giurfa 2013). Therefore, neural correlates of learning and memory formation have been extensively studied in the restrained honeybee and this anatomical and physiological knowledge has been used in a number of recent model approaches to learning and memory formation (see Sect. 4). The mushroom body (MB), a higher-order brain center in insects, has long been shown to be involved in associative learning in insects (for review see Heisenberg 2003; Menzel 2012, 2014). The computational resources in terms of neuron number and network complexity are highly developed in the MB of bees and ants. The MB circuit might thus also play a vital role in storing and retrieving higher-order information as required in navigation (Menzel 2012; Seelig and Jayaraman 2015; Devaud et al. 2015).

A neural network model for one component of navigation has recently been proposed by Ardin et al. (2016). This model relies on plasticity in the mushroom body, which is used as a visual matching unit. Walking agents (e.g., desert ants) were shown to robustly follow routes in a virtual world. The model maps the current view of the navigating agent to a single dimension representing how familiar the agent is with its current visual input. By maximizing this *familiarity* with respect to the heading direction, a target location can be reached without explicit knowledge of the field location and without other higher-level representations of the world such as a cognitive map (Baddeley et al. 2012).

Ants typically live in environments full of panoramic cues. Flying insects, however, might navigate over flat terrain, reducing the significance of the panorama and often limiting visual input to ground cues. Here, we investigated whether the familiarity model is applicable to the visual world of flying insects, particularly of the honeybee *Apis mellifera*, and how an agent performs navigational tasks in flat environ-

ments with sparse or dense object population. To this end, we developed virtual 3D models that provide realistic training and testing environments for flying agents. We implemented the spiking neural network model proposed in Ardin et al. (2016) and adapted our system to approximate the honeybee's visual input. Once trained on a sequence of images, the network represents the route memory for one unidirectional path. We extended the model to incorporate and learn an additional route context. This allowed us to train and test outbound and inbound foraging flights, enabling the agent to use one model for approaching food sources and the other for finding the nest on its way back.

2 Methods

2.1 Artificial worlds

We used a quadcopter (DJI Inspire One) to obtain aerial recordings of an experimental field site (N 50.814207, E 8.872498) at an altitude of approx. 100 m. In total, an area of 800 m × 800 m was covered. The recordings were processed with the software Pix4Dmapper to create a 3D texture object which was imported to Blender, a 3D rendering software. The recorded field, mostly open spaces used for agriculture, exhibited only few trees and bushes, features whose depth information was only partially recoverable due to the flight altitude of the drone. No panoramic features were transferred to the map; thus, the 3D scene was named *flat world* (see Fig. 1).

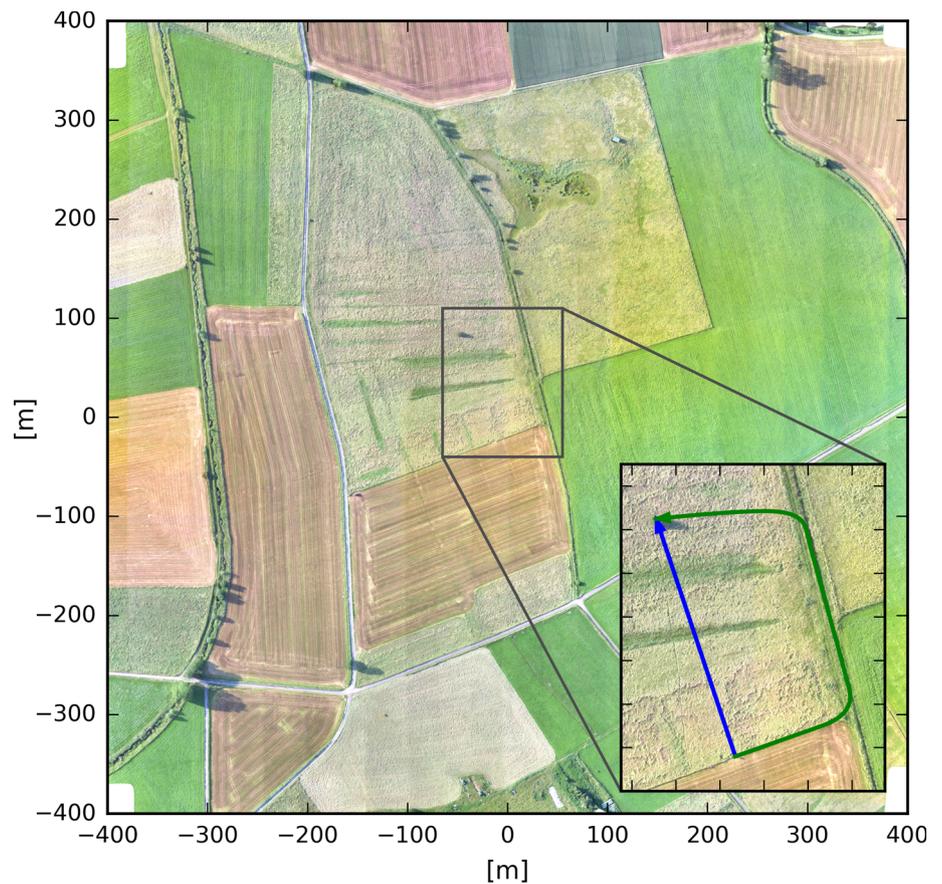
A second scene was composed by manually distributing models of trees, bushes and rocks. In this *tree world*, no ground texture was used. From this world, two conditions were derived (shown in Fig. 5), one densely occupied with objects, called high-density condition (HD), and the other with most of the objects removed. We call this variant of the tree world low-density condition (LD).

The Blender Python API served to define camera position and orientation and to obtain the corresponding view of a given agent.

2.2 Obtaining views of the environment

A relatively simple model of the bees visual apparatus was used, following Ardin et al. (2016). We consider the imaging resolution (angle between neighboring ommatidia) as constant at 2.5°, which corresponds to the resolution measured in the frontal part of the honeybee compound eyes (Laughlin and Horridge 1971). The horizontal field of view was set to 295°. Following Ardin et al. (2016) we limited the vertical field of view (FoV) to 75°. Although this reduced the amount of ground cues for agents in the flat world, we kept this parameter constant for the sake of comparison with

Fig. 1 Reconstructed field environment in top view. The inset in the lower right depicts two predefined routes used in the experiments



results obtained in the tree world. Here, a larger vertical FoV would have resulted in large ground areas without any visual input.

To accelerate computations, we prerendered bee views in Blender as grayscale images from an altitude of 1.5 m and with a field of view of 360° horizontally and 100° vertically (0° being parallel to the ground). With an angular resolution of 2.5°, this resulted in a 144 × 40 pixel matrix. Corresponding to the agents direction of movement, the images were then cropped to the final field of view (295° horizontal and 75° vertical FoV), resulting in an 118 × 30 pixel image. Depending on the world the experiment took place in, the view was cropped differently: We removed the lower 10 pixel rows in the tree world, but removed the upper 10 pixel rows in the flat world. We therewith shifted the viewing axis of the 75° FoV 12.5° upward in the tree world and by the same amount downward in the flat world to increase the amount of visible cues in the respective visual field.

In the final steps of preprocessing, the image was inverted and flattened into a 3540 × 1 array. Figure 2 shows preprocessed example views from all three environments.

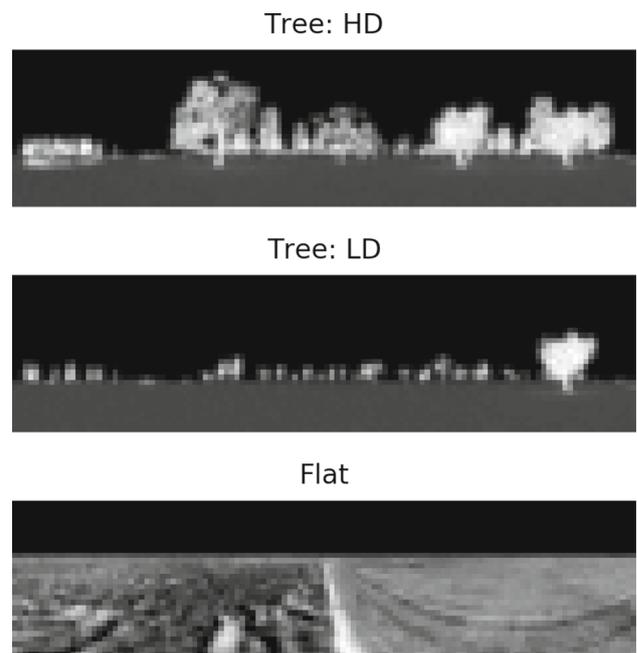


Fig. 2 Visual Input generated from the different environments described in Sect. 2.1: The tree-HD, tree-LD and flat world

Table 1 Neuron parameter

Parameter	PN	KC	EN
Number of neurons	360	20,000	1
C	100	4	100
a	0.3	0.01	0.3
b	-0.2	-0.3	-0.2
c	-65	-65	-65
d	8	8	8
k	2	0.035	2
v_r	-60	-85	-60
v_t	-40	-25	-40
ξ	$N(0,0.05)$	$N(0,0.05)$	$N(0,0.05)$

2.3 Spiking neural network model and simulation

We implemented the neural network model presented in [Ardin et al. \(2016\)](#) and adjusted the number of input neurons to match the dimensionality of our bee views as well as to allow for additional context input.

We use the values reported in [Ardin et al. \(2016\)](#) for the majority of model parameters (see [Table 1](#) for details). The model architecture as depicted in [Fig. 3](#) consists of three layers of Izhikevich neurons ([Izhikevich 2007](#)). This is a flexible neuron model able to reproduce a variety of spike patterns depending on the model parameters. The two dynamical variables in the model are the membrane voltage v and an abstract recovery variable u :

$$C\dot{v} = k(v - v_r)(v - v_t) - u + I + [\xi \sim N(0, \sigma)], \quad (1)$$

$$\dot{u} = a(b(v - v_r) - u), \quad (2)$$

where C is the membrane capacitance, v_r and v_t are resting and threshold potentials, respectively, and a , b and k are model parameters that determine the specific spiking properties. Only the dynamics of the upstroke of the spike are modeled and the variables are reset when v exceeds a peak value:

$$\text{if } v > v_t \begin{cases} v = c \\ u = u + d \end{cases} \quad (3)$$

After preprocessing, the input values are taken as input current I to the respective neurons.

The network topology resembles the fan-in–fan-out architecture of the MB ([Heisenberg 2003](#); [Jortner et al. 2007](#); [Huerta and Nowotny 2009](#); [Caron et al. 2013](#)). The first layer represents the population of projection neurons (5057 PNs) that relay the input information to the mushroom body. There are two PN subpopulations, one for receiving visual input (3540 vPNs) and one population that encodes the agents nav-

igational context (1517 cPNs). Their projection targets, the MB intrinsic Kenyon cells (20,000 KCs), form the second layer. Each KC receives input from 10 randomly ([Caron et al. 2013](#)) selected PNs ([Szyszka et al. 2005](#); [Turner et al. 2008](#)) with connection weights set to 0.25. All neurons in this feature layer project to a single mushroom body output neuron or extrinsic neuron (EN) in the output layer. The respective synapse parameter are given in [Table 2](#).

The input current to the second and third layer is given by:

$$I = gS(v_{\text{rev}} - v), \quad (4)$$

where g is the synaptic weight resembling a conductance and S represents the amount of active neurotransmitter which increases with presynaptic spikes and decays with

$$\dot{S} = \frac{-S}{\tau_{\text{syn}}} + \phi \delta(t - t_{\text{pre}}). \quad (5)$$

The synaptic weights between first and second layer are fixed at $g = 0.25$, and the plastic weights between second and third layer are initialized with $g = 2$ and if subjected to learning are quickly reduced to 0. There is no fixed quantile in weight change. The amount of change is governed by a three factor rule. A synapse becomes tagged, i.e., eligible to plastic change, according to the spike-timing-dependent plasticity (STDP) rule (for review see [Morrison et al. 2008](#)). In this case the STDP rule tags all synapses where during the simulation time both the presynaptic and the postsynaptic neuron has fired. The tag change is inversely proportional to the temporal difference between those spikes. The third factor is the coincidence of the reinforcement signal. The amount and duration of the reinforcer r are determined by the injected signal $R(t)$ and its time constant τ_r . The reinforcer in turn determines the amount of weight change Δg . Parameters are chosen such that the tagged synapses are silenced over a 10 ms window after the presentation of the reinforcement signal.

Formal definition of learning rule and parameters:

$$\dot{g} = cr, \quad (6)$$

$$\dot{r} = \frac{-r}{\tau_r} + R(t), \quad (7)$$

$$\dot{c} = \frac{-c}{\tau_c} + \text{STDP}(t_{\text{pre}} - t_{\text{post}})\delta[(t - t_{\text{pre}})(t - t_{\text{post}})], \quad (8)$$

$$\text{STDP}(t_{\text{pre}} - t_{\text{post}}) = \begin{cases} A_+ e^{-\frac{t_{\text{pre}} - t_{\text{post}}}{\tau_+}} & \text{if } t_{\text{pre}} - t_{\text{post}} < 0 \\ 0 & \text{if } t_{\text{pre}} - t_{\text{post}} = 0 \\ A_- e^{-\frac{t_{\text{pre}} - t_{\text{post}}}{\tau_-}} & \text{if } t_{\text{pre}} - t_{\text{post}} > 0 \end{cases} \quad (9)$$

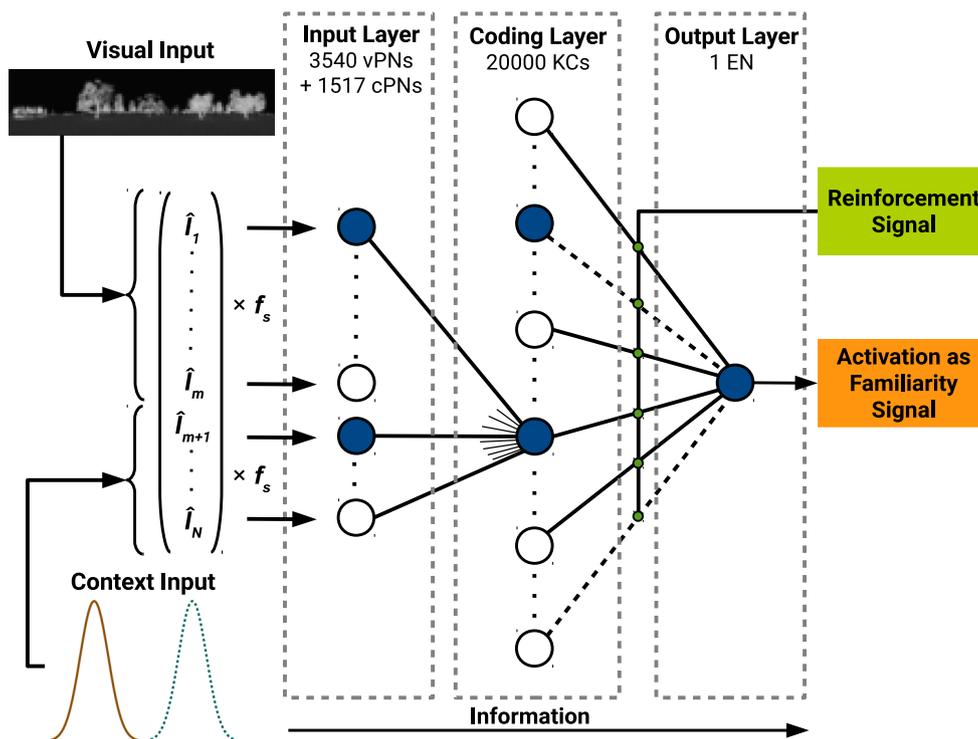


Fig. 3 Schematics of information processing in the neural network model. Visual input and context input are combined in a normalized 1D vector \hat{I} scaled by f_s and fed into the network. Here N is the number of PN and m the number of vPNs. The population of PNs corresponds in size to the input vector and constitutes the input layer. Their projections diverge on to the coding layer in which the input is sparsely represented by a set of active KCs. The projections of the coding layer converge on

a single EN, in the output layer. These projections are during training subject to learning, which is induced by a reinforcement signal BA. The learning results in the silencing of the connections that are involved in exciting the EN. During evaluation the spike count of the EN is taken as familiarity measure for the input. Connections already silenced are indicated with *dashed lines*. Active neurons are depicted in *blue*

Table 2 Synapse parameters: time constants for the decay of S (τ_{syn}), c (τ_c) and d (τ_d)

Parameter	PN to KC	KC to PN
τ_{syn}	3.0	8.0
ϕ	0.93	8.0
g	0.25	[0,2.0]
τ_c	N/A	40 ms
τ_d	N/A	20 ms
A_+/A_-	N/A	-1
τ_+/τ_-	N/A	15 ms

Quantile of S releases at each presynaptic spike ϕ and weights of synapses g . STDP amplitudes A_+ for $\Delta t = t_{pre} - t_{post} < 0$ and A_- for $\Delta t > 0$ as well as the respective time constants τ_+/τ_-

The KC population in the MB encodes input features in a sparse manner (Kloppenburg and Nawrot 2014). Only a small fraction of $\sim 5\text{--}10\%$ of all neurons are active for any given sensory input (Szyszka et al. 2005; Perez-Orive et al. 2002; Ito et al. 2008; Honegger et al. 2011). This *population sparseness* is mainly due to the divergent connectivity

scheme (fan-out) from the input to the coding layer, and it supports associative learning as synaptic plasticity can act on distinct sparse patterns in the large space of KCs (Huerta and Nowotny 2009; Nowotny and Huerta 2012). Moreover, individual KCs react only briefly with few spikes to a given change in the sensory input. This *temporal sparseness* has been attributed to feedforward inhibition (Perez-Orive et al. 2002, but see Gupta and Stopfer 2012), feedback inhibition (Kee et al. 2015), or cellular adaptation (Nawrot 2012; Farkhooi et al. 2013).

The network was implemented and simulated with ANNarchy, a neural network simulator with a Python interface and a C++ code generator to efficiently simulate custom spiking neural models and networks (Vitay et al. 2015). The simulation time step parameter of the Izhikevich model was set to $dt = 0.25$ ms. The differential equations were solved using the explicit Euler method.

2.4 Normalization of visual input

The overall visual input varies greatly across different input scenes which requires some form of input normalization.

Normalization mechanisms have not been studied in the visual pathway of the honeybee. However, in the olfactory pathway, experimental evidence and theoretical models indicate normalization, or gain control, through different mechanisms. For example, the interneuron network in the antennal lobe is hypothesized to provide gain control through lateral inhibition as one contribution to normalization (Wilson and Laurent 2005; Olsen and Wilson 2008; Asahina et al. 2009; Schmuker et al. 2011; Serrano et al. 2013). The mechanism of cellular adaptation additionally contributes to response normalization (Farkhooi et al. 2013).

In the present model, we did not attempt to model any biological mechanism explicitly. Rather, we normalized the input vector to unit length. Still, depending on the structuredness of the environment and on the agents view, the input might either activate too many or too few feature neurons in the KC layer; both scenarios lead to a drop in performance. We circumvented this by scaling the input vector following a simple heuristic. In preliminary experiments, we sampled random views for each environment from a fixed spatial grid and computed the number of active KC neurons given that input. The scaling parameter f_s was then manually adjusted such that the number of active KC neurons was on average 200. This value corresponds to a sparse representation by 10% of the total KC population. We set $f_s = 18,300$ for the flat world and $f_s = 18,800$ and $f_s = 17,200$ for the HD and LD condition of the tree world, respectively. These values differ from the ones chosen in Ardin et al. (2016).

2.5 Context-dependent input

The agents behavioral context is represented as an array of activations I_{m+1}, \dots, I_N that is fed into respective input neurons (cPNs) which in turn provide input to the KCs in parallel to the input from the vPNs. We limit the model to two contexts, such as inbound or outbound flight, represented by a population of neurons ($N_c = 1517$) whose activations correspond to a given context (see Fig. 4). The activations I are scaled with the same factor f_s as chosen for the activations of vPNs.

The cPN population size was experimentally determined to constitute 30% of the total PN population [$0.3 = N_c / (N_v + N_c)$]. The same proportion was taken for the overall input strength. At this size, an input vector learned in one context shows no significant familiarity when evaluated in the other.

2.6 Route learning

A number of routes through the artificial worlds were predefined, and visual inputs were sampled along these routes at a frequency of 0.5 m^{-1} . Each of these inputs was combined with a context vector corresponding to the direction

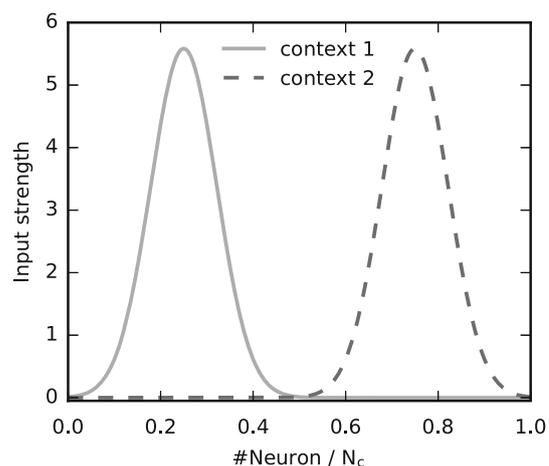


Fig. 4 Distribution of input activations over the population of cPNs, for two distinct motivational contexts, e.g., inbound and outbound travel. The index n denotes the n th neuron out of a population of size N_c

of the agent and then presented to the network for 40 ms. The reinforcement signal is then injected, and the simulation continues to run for another 10 ms. All connections of the feature layer to the output neuron are subject to learning. The learning rule is a form of spike-timing-dependent plasticity (STDP) paired with a reinforcement signal, as proposed in Izhikevich (2007). The STDP tags the downstream synapses activating the EN. Only if a reinforcement signal is present, each synaptic tag is translated into the (negative) change of the synaptic weights. STDP actualized by a reinforcement signal has been found in the locust MB (Cassenaer and Laurent 2012). The parameters of the learning rule are chosen such that it implements one-shot learning. The weight of an active connection is quickly reduced to zero. The same input will therefore not elicit any more activity in the EN during subsequent inputs.

2.7 Network evaluation

Learning is disabled during testing (Ardin et al. 2016). Due to the learning procedure, low EN spike rates during the evaluation phase represent high familiarity with the input, and high EN spike rates are produced by features that were not present in learning and therefore signal low familiarity. In our experiment, a given input vector is presented for 40 ms to the trained network. To save computation time, the period between presented inputs, in which the honeybee moves and the network returns to its equilibrium state, is not simulated. Instead all neuron and synapse variables, except for the synaptic weights, are reset to their initial values after each presented image, for both training and evaluation. To determine the most familiar direction at a given location, we evaluate the network output for a number of views obtained by scanning the environment horizontally. Each view to be

tested is produced by cropping the 360° view provided via the Blender API. Given an initial heading, only views from -90° to $+90^\circ$ enter the network for familiarity evaluation. This implies a bias to move forward. The minimum in EN spike count of this sequence of measurements represents the most familiar direction. If multiple minima are found, the closest to the initial heading direction is chosen.

3 Experiments

3.1 Setup

To evaluate whether an agent can learn routes in the tree world, we defined three routes [tortuous (T), curved (C) and S-shaped (S)] in a low-density (LD) and high-density (HD) tree world (see Fig. 5). These are representative for the diverse routes seen in field experiments, e.g., see (Riley et al. 2005; Capaldi et al. 2000; Riley et al. 2005). An agent was first trained to the corresponding views along the focal route (2 m step size). In the experiments, the agent is reset to the

routes beginning, the most familiar direction is determined as described in Sect. 2.7 and a step of 2 m is performed into this direction. This scheme is performed repeatedly until the agent either reaches the end of the route (distance <3 m), counted as a success, or leaves the boundaries of a route corridor (8 m wide). This hyperparameter was experimentally determined in preliminary tests. Once the agent goes astray, it is reset to the nearest point on the route. This way we obtain a second measure of the networks performance, the number of resets. After a reset and at the beginning of a simulation, the agent performs a scan over the full 360° range. If the agent reaches a maximum number of resets, it is considered lost.

To quantify the route following performance, 20 runs per route were simulated. The output of the network and therefore the route an agent takes is dependent on the specific connectivity of the layers, which is generated randomly within certain constraints. Each run was performed with a different instance of the MB network and therefore a different connectivity. Distributions and means of the number of resets were taken as a measure for the route following performance.

In a second experiment, we investigated whether an agent can learn following routes with only ground cues present. Thus, we defined two routes in the flat world, with a shared start and end point. Both locations, an actual hive location that was situated there at the time of the recordings and a bush on a meadow north of the hive, were only visible as ground texture and therefore might only offer decisive information in close range. One route directly connects start and goal location, and the other route arrives at the goal in a detour, following clear linear ground structures (see Fig. 1). The routes consist of 58 (straight) and 107 (detour) training views. To take a probable increase in salience of a goal cue in this environment into account, the success radius in this experiment was set to 5 m. The rest of the test procedure follows the one described for experiment 1.

In the third experiment, we investigated whether the context input scheme enables an agent to learn and discriminate routes that share a portion of the path. The experiment was conducted in the tree world. Two routes were designed with an overlap in their first half. The routes bifurcate then into an east and a west route. Both routes are 90 m long resulting in 45 training views along each route. Preliminary tests showed that an agent can learn and then follow both routes independently. In the experiment, we trained an agent by presenting both routes in random order. We then tested the agent by following the scheme described above with the context input set to either one of the routes. As a sanity check, we trained agents by presenting both routes but in only one context. These agents should not be able to follow the correct route. We performed 16 experimental trials per training type and counted in how many trials the agent was able to reach the goal. Resetting was disabled in this experiment.

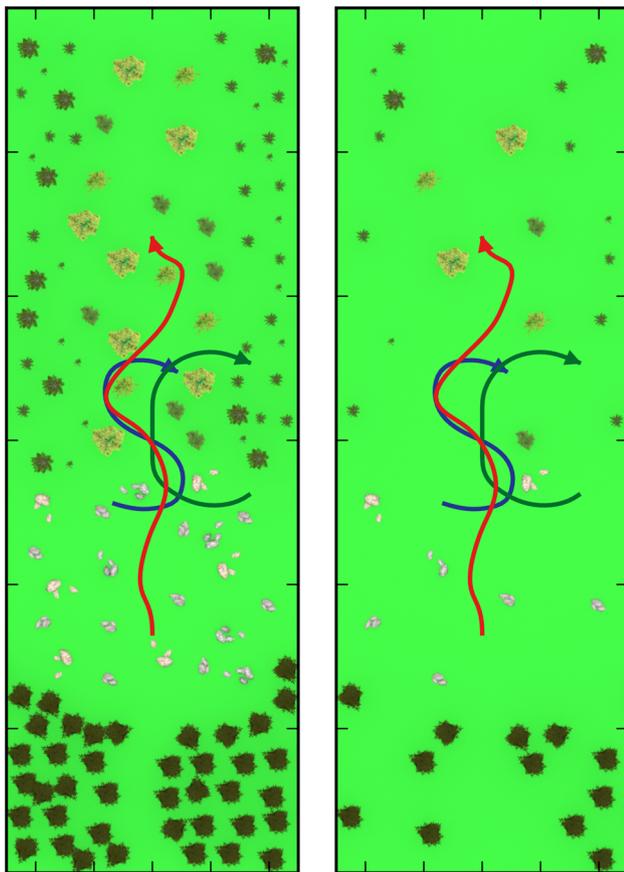


Fig. 5 Top views of the tree world in HD (left) and LD (right) condition. The three routes constructed for the route following experiment are depicted in color: T in red, S in blue and C in green

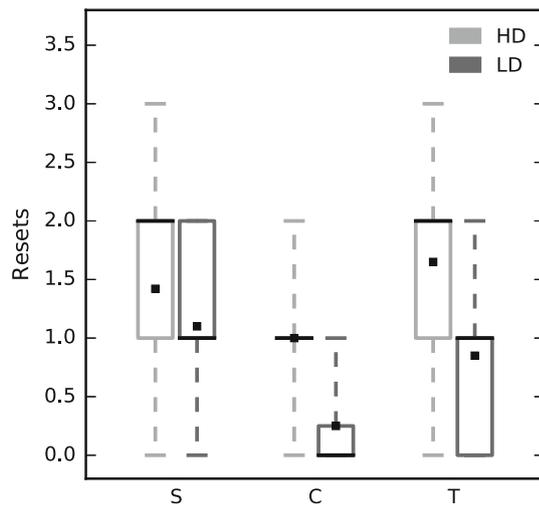


Fig. 6 Boxplots of reset counts for the routes: S, C and T both in the HD and LD condition. The *whiskers* indicate the range of results, whereas the boxes encompass the upper and lower quartiles, with the median shown as a *black line*. The *black squares* denote the means of the distributions. In total 20 runs were conducted per route and condition

All simulations were performed on a machine with a AMD Phenom II X6 1090T Processor and 24G RAM. The run time for one simulation step was about 50 s resulting in about 440 hours of runtime for all experiments.

3.2 Statistics

To verify the statistical significance of observed differences in route following performance, the sets of reset counts were compared between conditions with the Chi-square test of independence using the function “*chisq.test()*” of the statistics software R. This test takes as the zero hypothesis that the two compared samples are drawn from the same distribution. Considering the small sample population, the test was performed following Hope (1968).

3.3 Results

Route following in the tree world We trained an agent on each of the three route types and consequently tested its route performance over 20 trials per route type and world condition (120 trials in total). For each trial, different instances of the network, i.e., with different random connectivity, were used. Due to the reset procedure, the agents reached the goal in almost all runs. Only one flight on the S route failed because the agent got trapped in an infinite reset loop. In the analysis of the reset counts, this flight is excluded. It is to note that for all routes there were also successful flights where no reset was necessary. We call these tight successes (TS).

In all experiments, the mean reset count is smaller for routes in the LD environment (see Fig. 6). However, the statis-

Table 3 Mean reset counts and proportion of tight successes for the various routes in the HD and LD condition

Route	S	C	T
Mean (HD)	1.42	1.0	1.65
Mean (LD)	1.10	0.25	0.85
<i>P</i> value	0.485	8.9e−05	0.033
TS (HD)	0.15	0.10	0.10
TS (LD)	0.20	0.75	0.30

The given *P* values are the results of the Chi-square test comparing reset distributions between both conditions

tical significance of this result varies between the route types. The results are significant for the C and T route ($P < 0.05$), but not for the S route ($P \gg 0.05$). The decrease in reset counts is in turn manifested in an increase in tight successes for the LD condition. These results are summarized in Table 3.

Navigating on open fields We ran 20 trials for each route, with a different network each. From these, five runs, three along the straight and two along the detour route, had to be excluded due to the agent getting stuck in an infinite reset loop. In contrast to the experiments in the tree world, the agent never reached the goal without resets. The average reset count for the straight route is 3.12, slightly higher than for the detour route ($\mu = 2.94$). The standard deviations are 1.18 and 0.85, respectively.

We find no significant difference in reset count distributions ($p = 0.186$). However, the detour route is 98 m longer than the straight route, and thus, it is an unfair comparison. By normalizing by the route length, we obtain a reset rate of 0.027/m vs 0.014/m for the straight and detour route, respectively.

The detour route can be divided into three straight route segments with differing visual input and connected by two turns between these segments (see Fig. 5). The first segment (short edge) is a sharp edge between a field in the south and a meadow in the north. This segment was traveled without error in all runs. It is followed by a turn onto a long straight segment (long edge) oriented along a rut. In some of the runs, the agent attempted to turn in the other direction following the bend of the field edge, which accounts for 9.4% of overall resets. The long edge segment was followed with the second lowest reset count (7.5% of overall resets) although it was the longest route segment. The resets occurred in an area where the rut was briefly disconnected. The next transitional turn, connecting the long edge segment and the meadow segment, failed in all runs accounting for 35.8% of overall resets. In all runs the agent continued along the rut instead of turning onto the meadow. The meadow segment, which leads over the open meadow till the goal, was the most error prone with

Table 4 Success counts for 16 runs with both routes trained in the same context, either inbound or outbound

Context	West	East	Failures
Context 1	5	1	2 (East)
Context 2	4	0	4 (West)

Columns indicate which goal was reached, or which direction was chosen if the run was a failure

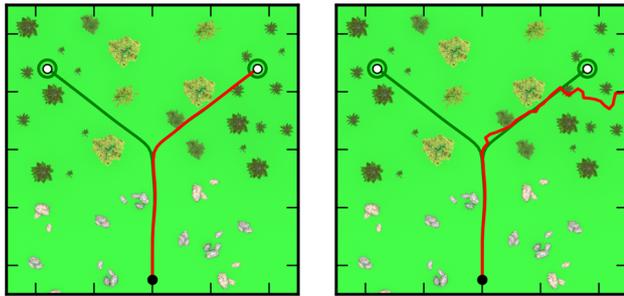


Fig. 7 Exemplary route following during the context discrimination experiments. Trained and flown routes are indicated in green and red, respectively. Start positions are shown as black and goal positions as white circles together with the success radius. *Left* successful run with the two routes trained in different contexts. *Right* failed run with both routes trained in the same context

43.8% of overall resets and only a single run without resets on this segment.

Context discrimination First, we trained both overlapping routes without discriminating between them in the context input. When tested, the agents predominantly followed the west arm of the Y-route and in 6 runs the agent did not reach either goal, which was counted as a failure (see Table 4; Fig. 7). We then trained agents, with a different context input for each route. When tested, all agents arrived successfully at their respective goal.

Baseline To compare the results obtained with the network model to a more tangible approach, we implemented two baselines. In the first baseline model, we kept all training views in memory and defined the familiarity of a given test view as the inverse Euclidean distance to its nearest neighbor. Not very surprisingly, all agents could follow the trained views. We have not investigated further how robust this baseline is against changes in starting position and how the route following performance depends on object density. This has been studied thoroughly elsewhere (Stürzl and Zeil 2007). To allow a fairer comparison, we modeled the set of training views with a multivariate Gaussian. The familiarity hence was just the probability of the given view to belong to the trained distribution. We reduced data dimensionality with PCA to $D = 50$ in order to avoid the covariance matrix to become singular. This baseline was tested along the three

route types in the LD and HD world and along both route types in the flat world. Since the method is deterministic, we ran 9 trials for each route, for a 3×3 grid of starting positions centered on the route’s actual start (2 m spacing). Similar to the neural network model, the baseline yields fewer resets in the LD world. For the S and C routes, we find that with the Gaussian model the agent can follow the route in all trials except for the HD world’s C route (7 successful arrivals). The model, however, fails on the N route (one arrival in the HD world, no success in LD) and the flat world (no arrivals). Particularly in the flat world, where visual features were sparse, the network model was better able to extract the relevant features and to represent the subspace of known views.

4 Discussion

4.1 The effect of environmental richness on model performance

The neural network model proposed by Ardin et al. (2016) was successfully adopted to control a simulated agent to follow previously trained routes in panorama-rich environments.

In Ardin et al. (2016) the average number of resets for routes trained with 80 images was 2.6. Intriguingly, our experiments in the tree world yield even lower mean values (HD: 1.6 resets, LD: 0.8 resets). This result can be attributed to the feature density of the worlds. Our virtual worlds exhibit less features than the one modeled in Ardin et al. (2016). However, in a world with even less structure, such as our flat world, the performance of the network deteriorates and reset counts increase to 3.25 resets on average. Two processes may have caused this result. First, an agent faces a larger parallax in densely populated worlds. Thus, the agent’s view changes more quickly as it leaves the route. The catchment area’s dependency on the object density in the world has been described earlier in Stürzl and Zeil (2007). A second factor that may have caused the differences in route following performance relates to the learning process. Each neuron in the central feature layer of the network (KC) represents a specific combination of active pixels. Each view presented in training activates a certain number of feature neurons in the KC layer. The more visual features are present in the view, the more KC neurons will be firing. By applying the learning rule, these features do not contribute to the activity of the output neuron anymore. Hence, by using many views along a route, the union U of all feature cells that were activated represent the route memory. If an agent deviates from the learned route, it is hence not necessarily caused by a view that closely resembles one of the trained views. The view that maximizes familiarity is just the one with the largest overlap with U (see Supplementary Information S1 for an illustra-

tive example). Feature-rich worlds exhibit more active KCs, which in learning lead to the silencing of more synapses that contribute to the output activity. To make different worlds comparable, we scaled the input activations such that on average 200 (1%) of all features cells are active. Thus, the HD condition does on average not activate more KC neurons. The scaling, however, affects which input features are represented by the KC neurons. In the LD world the input is sparse, and most of the KC features represent local image structure. In the HD world, and even worse in the flat world, active pixels are more uniformly distributed and, on average, the KC layer exhibits more active neurons that receive input from almost anywhere in the image. This is reflected by how random pairs of views overlap in feature space. To quantify the similarity of the internal representations of two views, we randomly sampled views from each environment and calculated the average overlap of the internal representations. In the flat world 39% of the feature neurons are shared on average. For the HD and LD tree worlds, this overlap is 15 and 8%, respectively. Similarly, if we apply the L2 norm to compare the corresponding input vectors directly, we find that on average the distance of two inputs is 0.370 for the flat world, 0.783 for the HD and 0.788 for the LD condition in the tree world.

The effect of the KC overlap on the learning process is illustrated by comparing the average familiarity of randomly sampled views before and after training a specific route. In the flat world, we found the average EN spike count to be 79.8. After learning it drops to an average of 11.5 (straight route in Fig. 1). When training a similar route with the same amount of training views in the tree world, we observe a drop from 89 to 52.3 and from 89 to 75 in the average EN spike count for the HD and LD condition respectively. Hence, in the flat world, everything looks familiar at the end of training. Learning a route in this world leads to a situation where signal and noise are not well separated. We call this “signal flattening.”

The different degree of overlap in the KC representation and the subsequent signal flattening effects might be linked to the input sparseness and therefore to the structuredness of the respective environment. We calculated the Gini factor (Hurley and Rickard 2009) for all three environments (HD: 0.663; LD: 0.715; flat: 0.569), which shows a decreasing input sparseness from LD to HD and the flat world. Since all visual inputs are normalized to unit length, we observe stronger activations of active input cells in sparse worlds. Given the random PN-KC connectivity, sparse environments yield more KCs that are activated by only a few highly active PNs. In environments with more distributed visual input, KCs are activated by multiple inputs across the field of view and therefore code global rather than local features. A detailed examination of how the input structure relates to the representation of local and global features should therefore be in focus of future research.

4.2 Extended landmarks

Although in the examined case, traveling the detour route gives no significant advantage in reaching the goal, our results suggest that traveling along extended landmarks might improve performance on long journeys considerably. Extended landmarks are adding additional local features to an input that else is broadly distributed over the FoV. Training a route along such structures could therefore reduce the overall signal flattening as well as provide a prominent local feature to follow. This is illustrated by the mean EN spike count, which dropped to 11.5 after training the 58 views of the straight route compared to 21.2 for training 58 views of the detour route.

On the other hand, these types of structures pose a risk of the navigator to become bound to the landmark, not being able to break away. This was observed at the transition between the long edge and the meadow. This is likely due to the similarity, and hence familiarity, of the views along the landmark in an otherwise rather featureless environment. At the T intersection the agent turned into the wrong direction in 9.4% of the trials. Since the model does not integrate compass information, the directional ambiguity thus poses another risk when orienting along extended landmarks. Lastly, under certain conditions similar landmarks that are not part of the route could attract and mislead the navigator.

We performed an additional set of experiments in which we tested the context extension. Although we regarded this input component as representative for a categorical internal state (such as the motivation for outbound or inbound flights), we would like to stress the fact that this input could be virtually anything, such as low-level sensory input (such as UV or chemical cues), multimodal sensory integrations or high-level information, such as motivation, plans and decisions.

When both routes are trained without reflecting the route identity in the context subspace, i.e., when training both routes in the same context, we observe two effects: The agent either follows any of the two routes or it deviates from the route, both of which are more likely than in the context-aware case. Agents deviate from the route anywhere, not just at the bifurcation. Where it deviates from the routes, or which of the routes it follows, depends on the specific route the network was trained and tested on, and the noise intrinsic to the process.

Training the two routes with separate context inputs, the agents were able to successfully discriminate the paths according to the context. Generally, the route following performance was increased compared to the same-context case, likely due to a lower KC overlap between the two routes and therewith better separability and lower overall signal flattening. It remains future work to investigate how to integrate additional sensory components, such as compass informa-

tion, and how reliable the navigator can follow routes using such a continuous context.

4.3 Plasticity model

Experimental evidence for associative plasticity exists for multiple loci in the insect system. We have here assumed synaptic plasticity at a single site in the network, between KCs and the MB output neurons, in our case a single EN. This assumption follows earlier modeling approaches on unsupervised (Nowotny et al. 2005; Montero et al. 2015) and supervised associative olfactory learning (Huerta et al. 2004; Huerta and Nowotny 2009; Hausler et al. 2011; Wessnitzer et al. 2012; Smith et al. 2012; Helgadottir et al. 2013; Bazhenov et al. 2013; Schmuker et al. 2014; Haenicke 2015) in insects. There is accumulating experimental evidence for plasticity at this synaptic location (Menzel and Manz 2005; Menzel 2012; Cassenaer and Laurent 2012), and ENs have clearly been shown to encode valence in the honeybee allowing rewarded olfactory stimuli to be unequivocally distinguished from non-rewarded stimuli (Strube-Bloss et al. 2011, 2016; Menzel 2014). Valence encoding has also been shown for mushroom body output neurons of the fruit fly *Drosophila melanogaster* (Aso et al. 2014; Hige et al. 2015). This value code at the MB output might also reflect plasticity at an earlier stage of the network. In the honeybee, the MB constitutes a recurrent network where a population of GABAergic inhibitory ENs receive input from KCs and backproject to the input site of the MB. They terminate on synaptic densities in the microglomerular structures of the calyx where they may inhibit the PN boutons that connect with postsynaptic KCs. It has been shown experimentally that these inhibitory ENs change their activity in response to classical conditioning (Haehnel and Menzel 2010; Filla and Menzel 2015) and thus may modulate the olfactory input due to learning (Szyszka et al. 2005; Haenicke 2015). This fits the observation of learning-induced changes of odor responses in the PN boutons of the honeybee (Haenicke 2015). Also KCs of the honeybee (Szyszka et al. 2008) and of the fruit fly (Dylla et al. 2017) show plasticity in classical conditioning experiments indicating plasticity at the MB input site. There is additional evidence for associative plasticity in the early olfactory pathway, both for the honeybee (e.g., Fernandez et al. (2009)) and for the fruit fly (Schwaerzel et al. 2003). To our knowledge, plasticity in the visual pathway of the honeybee has not yet been demonstrated. With the accumulating evidence for plasticity at multiple locations in the MB network, we may hypothesize that the MB circuit establishes and makes use of multi-site plasticity and future models of navigation may exploit this feature.

5 Outlook

Although successful route following could be reproduced using the MB model, the performance highly depends on the structure of the environment and breaks down in flat worlds. We suggest that one limiting factor to the capacity of the network is the KC overlap, which induces global accumulation of familiarity that we called signal flattening. Reducing this overlap therefore could greatly increase performance. This would require further investigation into the representation of input features in the feature layer.

Ways to reduce overlap resulting from global environmental features could be to introduce a visual preprocessing or a PN-KC connectivity which samples for local image features, like edges or color gradients. If the average KC overlap measured here does stem from global as well as from local features, there might be a possible trade-off as some similarity in the representation of neighboring views has to be retained in order for the familiarity navigation to work. To resolve this conflict, future investigations should have the goal to reduce global while increasing local KC overlap.

An alternative approach to achieve this goal is to introduce heterogeneity in the KC population. In computational models, typically all model neurons share identical parameters. However, this assumption is not biologically realistic and parameter variation across neurons and synapses in biological networks have been suggested to support neural processing and network function while avoiding unwanted effects such as network synchronization (e.g., Lengler et al. (2013)). Montero et al. (2015) studied the effect of neuronal variability in unsupervised learning and classification in a MB network with binary neurons and synapses. Introducing a variable activation threshold across the population of KCs improved learning of classification in their MB model. The authors could further show that a combination of generalist-type KCs (low thresholds) responding to a larger set of inputs and specialist-type KCs (high threshold) responding to one or few very specific inputs allowed for an optimal performance. In our model, we used a fix average number of 200 activated KCs for all input scenarios. In the dense-world scenario, we obtained good performance. However, in the flat world we observed signal flattening and a large KC response overlap. This broad responses might correspond to the case of generalist-type KCs. Introducing a variable threshold in the KC population would allow for a variation in the KC excitability which might be better suited to adopt to different worlds scenarios. Combining specialist-type KCs to support highly specific associations and non-overlapping responses even in a flat world with generalist-type KCs that are suitable for high-density input should be tested in future model extensions. One step further, we might hypothesize that neuron excitability might be dynamically changed through

neuromodulatory input to adjust to different input scenarios (Nadim and Bucher 2014).

To investigate navigational strategies and their respective neural correlates with the computational approach, future research will make use of recorded flight paths of bees, either by deriving models from the data or by validating models against the data. We will further improve the three-dimensional reconstruction of realistic environments in which computational agents can be tested. This will be of great use when analyzing honeybee flight trajectories that were recorded in the same environment. By projecting the virtual world using a camera model that mimics the honeybee eye (Stürzl et al. 2010), we can approximate the visual input along the flight paths and can investigate how environmental and visual input features correlate with the animal's behavior.

The proposed model involves basic features of neural computation such as sparse coding and reward-modulated synaptic plasticity that are essential for MB learning. We want to add that models as presented here do not exclude additional neural processing stages that lead to more holistic representations as assumed, e.g., in the concept of a “cognitive map.” Higher-order cognitive concepts are always rooted in more basic neural operations, and neither the elementary functions nor the higher-order computations oppose each other. Elementary properties like “familiarity,” as modeled here, capture important components of navigation, but cannot be used to reproduce the full spectrum of honeybee behavior. Future work should therefore combine basic computations and higher-order neural processing. Comparing the behavior of computational agents with that of real animals requires close to equivalent test procedures allowing for the richness of environmental and cognitive conditions. Test procedures need to ask under which conditions agents and animals perform novel movement trajectories that are more efficient and less risky. Cognitive behavioral biology considers novel shortcuts as indicator for a form of navigational memory that is best conceptualized as cognitive map, the mental representation of the environment in its geometric layout (Tolman 1948), if elementary processes like image matching and path integration can be excluded. It will be necessary to include such a paradigm in future modeling approaches.

Acknowledgements We thank Barbara Webb for fruitful discussions and Julian Petrasch and Sebastian Krieger for creating the drone map. This work was partially funded by the Bundesministerium für Bildung und Forschung (BMBF) through Grant No. 01GQ0941 to the Bernstein Focus Learning and Memory Berlin (Insect Inspired Robots: Towards an Understanding of Memory in Decision Making) and the Dr. Klaus Tschira Stiftung through Grant No. 00.300.2016 (Robotik in der Biologie: Ziele finden mit einem winzigen Gehirn. Die neuronalen Grundlagen der Navigation der Bienen).

References

- Ardin P, Peng F, Mangan M, Lagogiannis K, Webb B (2016) Using an insect mushroom body circuit to encode route memory in complex natural environments. *PLoS Comput Biol* 12(2):e1004683
- Asahina K, Louis M, Piccinotti S, Vosshall LB (2009) A circuit supporting concentration-invariant odor perception in drosophila. *J Biol* 8(1):9
- Aso Y, Sitaraman D, Ichinose T, Kaun KR, Vogt K, Belliart-Guérin G, Plaçais PY, Robie AA, Yamagata N, Schnaitmann C et al (2014) Mushroom body output neurons encode valence and guide memory-based action selection in drosophila. *Elife* 3(e04):580
- Avarguès-Weber A, Giurfa M (2013) Conceptual learning by miniature brains. *Proc R Soc Lond B Biol Sci* 280(1772):20131907
- Baddeley B, Graham P, Husbands P, Philippides A (2012) A model of ant route navigation driven by scene familiarity. *PLoS Comput Biol* 8(1):e1002336. doi:10.1371/journal.pcbi.1002336
- Bazhenov M, Huerta R, Smith BH (2013) A computational framework for understanding decision making through integration of basic learning rules. *J Neurosci* 33(13):5686–5697
- Capaldi EA, Smith AD, Osborne JL, Fährbach SE et al (2000) Ontogeny of orientation flight in the honeybee revealed by harmonic radar. *Nature* 403(6769):537
- Caron SJ, Ruta V, Abbott L, Axel R (2013) Random convergence of olfactory inputs in the drosophila mushroom body. *Nature* 497(7447):113–117
- Cassenaer S, Laurent G (2012) Corrigendum: conditional modulation of spike-timing-dependent plasticity for olfactory learning. *Nature* 487(7405):128–128. doi:10.1038/nature11261
- Cheeseman JF, Millar CD, Greggers U, Lehmann K, Pawley MDM, Gallistel CR, Warman GR, Menzel R (2014) Way-finding in displaced clock-shifted bees proves bees use a cognitive map. *Proc Natl Acad Sci* 111(24):8949–8954. doi:10.1073/pnas.1408039111
- Cheung A, Collett M, Collett TS, Dewar A, Dyer F, Graham P, Mangan M, Narendra A, Philippides A, Stürzl W, Webb B, Wystrach A, Zeil J (2014) Still no convincing evidence for cognitive map use by honeybees: Fig. 1. *Proc Natl Acad Sci* 111(42):E4396–E4397. doi:10.1073/pnas.1413581111
- Collett TS, Collett M (2002) Memory use in insect visual navigation. *Nat Rev Neurosci* 3(7):542–552. doi:10.1038/nrn872
- Cruse H, Wehner R (2011) No need for a cognitive map: decentralized memory for insect navigation. *PLoS Comput Biol* 7(3):e1002009. doi:10.1371/journal.pcbi.1002009
- Devaud JM, Papouin T, Carcaud J, Sandoz JC, Grünwald B, Giurfa M (2015) Neural substrate for higher-order learning in an insect: mushroom bodies are necessary for configural discriminations. *Proc Natl Acad Sci* 112(43):E5854–E5862
- Dylla KV, Raiser G, Galizia CG, Szyszka P (2017) Trace conditioning in drosophila induces associative plasticity in mushroom body kenyon cells and dopaminergic neurons. *Front Neural Circuits* 11:42
- Farkhooi F, Froese A, Muller E, Menzel R, Nawrot MP (2013) Cellular adaptation facilitates sparse and reliable coding in sensory pathways. *PLoS Comput Biol* 9(10):e1003251
- Fernandez PC, Locatelli FF, Person-Rennell N, Deleo G, Smith BH (2009) Associative conditioning tunes transient dynamics of early olfactory processing. *J Neurosci* 29(33):10191–10202
- Filla I, Menzel R (2015) Mushroom body extrinsic neurons in the honeybee (*Apis mellifera*) brain integrate context and cue values upon attentional stimulus selection. *J Neurophysiol* 114(3):2005–2014
- Gupta N, Stopfer M (2012) Functional analysis of a higher olfactory center, the lateral horn. *J Neurosci* 32(24):8138–8148
- Haehnel M, Menzel R (2010) Sensory representation and learning-related plasticity in mushroom body extrinsic feedback neurons of the protocerebral tract. *Front Syst Neurosci* 4:161

- Haenicke J (2015) Modeling insect inspired mechanisms of neural and behavioral plasticity. PhD thesis, Freie Universität Berlin
- Hausler C, Nawrot MP, Schmuker M (2011) A spiking neuron classifier network with a deep architecture inspired by the olfactory system of the honeybee. In: 5th International IEEE/EMBS conference on neural engineering (NER), pp 198–202
- Heisenberg M (2003) Mushroom body memoir: from maps to models. *Nat Rev Neurosci* 4(4):266–275. doi:[10.1038/nrn1074](https://doi.org/10.1038/nrn1074)
- Helgadottir LI, Haenicke J, Landgraf T, Rojas R, Nawrot MP (2013) Conditioned behavior in a robot controlled by a spiking neural network. In: 6th International IEEE/EMBS conference on neural engineering (NER), 2013, pp 891–894
- Hige T, Aso Y, Rubin GM, Turner GC (2015) Plasticity-driven individualization of olfactory coding in mushroom body output neurons. *Nature* 526(7572):258–262
- Honegger KS, Campbell RA, Turner GC (2011) Cellular-resolution population imaging reveals robust sparse coding in the drosophila mushroom body. *J Neurosci* 31(33):11,772–11,785
- Hope ACA (1968) A simplified Monte Carlo significance test procedure. *J Roy Stat Soc B* 30(3):582–98. doi:[10.2307/2984263](https://doi.org/10.2307/2984263)
- Huerta R, Nowotny T (2009) Fast and robust learning by reinforcement signals: explorations in the insect brain. *Neural Comput* 21(8):2123–2151
- Huerta R, Nowotny T, García-Sánchez M, Abarbanel HD, Rabinovich MI (2004) Learning classification in the olfactory system of insects. *Neural Comput* 16(8):1601–1640
- Hurley N, Rickard S (2009) Comparing measures of sparsity. *IEEE Trans Inf Theory* 55(10):4723–4741
- Ito I, Ong RCy, Raman B, Stopfer M, (2008) Sparse odor representation and olfactory learning. *Nat Neurosci* 11(10):1177–1184
- Izhikevich EM (2007) Solving the distal reward problem through linkage of STDP and dopamine signaling. *Cereb Cortex* 17(10):2443–2452. doi:[10.1093/cercor/bhl152](https://doi.org/10.1093/cercor/bhl152)
- Jacobs LF, Menzel R (2014) Navigation outside of the box: what the lab can learn from the field and what the field can learn from the lab. *Mov Ecol* 2(1):3
- Jortner RA, Farivar SS, Laurent G (2007) A simple connectivity scheme for sparse coding in an olfactory system. *J Neurosci* 27(7):1659–1669
- Kee T, Sanda P, Gupta N, Stopfer M, Bazhenov M (2015) Feed-forward versus feedback inhibition in a basic olfactory circuit. *PLoS Comput Biol* 11(10):e1004531
- Kloppenborg P, Nawrot MP (2014) Neural coding: sparse but on time. *Curr Biol* 24(19):R957–R959
- Laughlin SB, Horridge GA (1971) Angular sensitivity of the retinula cells of dark-adapted worker bee. *Zeitschrift für Vergleichende Physiologie* 74(3):329–335. doi:[10.1007/BF00297733](https://doi.org/10.1007/BF00297733)
- Lengler J, Jug F, Steger A (2013) Reliable neuronal systems: the importance of heterogeneity. *PLoS ONE* 8(12):e80694
- Menzel R (2012) The honeybee as a model for understanding the basis of cognition. *Nat Rev Neurosci* 13(11):758–768
- Menzel R (2014) The insect mushroom body, an experience-dependent recoding device. *J Physiol Paris* 108(2):84–95
- Menzel R, Giurfa M (2001) Cognitive architecture of a mini-brain: the honeybee. *Trends Cogn Sci* 5(2):62–71. doi:[10.1016/S1364-6613\(00\)01601-6](https://doi.org/10.1016/S1364-6613(00)01601-6)
- Menzel R, Greggers U (2015) The memory structure of navigation in honeybees. *J Comp Physiol A Neuroethol Sens Neural Behav Physiol* 201(6):547–61. doi:[10.1007/s00359-015-0987-6](https://doi.org/10.1007/s00359-015-0987-6)
- Menzel R, Manz G (2005) Neural plasticity of mushroom body-extrinsic neurons in the honeybee brain. *J Exp Biol* 208(Pt 22):4317–32. doi:[10.1242/jeb.01908](https://doi.org/10.1242/jeb.01908)
- Menzel R, Kirbach A, Haass WD, Fischer B, Fuchs J, Koblöfsky M, Lehmann K, Reiter L, Meyer H, Nguyen H, Jones S, Norton P, Greggers U (2011) A common frame of reference for learned and communicated vectors in honeybee navigation. *Curr Biol* 21(8):645–650. doi:[10.1016/j.cub.2011.02.039](https://doi.org/10.1016/j.cub.2011.02.039)
- Montero A, Huerta R, Rodríguez FB (2015) Regulation of specialists and generalists by neural variability improves pattern recognition performance. *Neurocomputing* 151:69–77
- Morrison A, Diesmann M, Gerstner W (2008) Phenomenological models of synaptic plasticity based on spike timing. *Biol Cybern* 98(6):459–478
- Nadim F, Bucher D (2014) Neuromodulation of neurons and synapses. *Curr Opin Neurobiol* 29:48–56
- Nawrot MP (2012) Dynamics of sensory processing in the dual olfactory pathway of the honeybee. *Apidologie* 43(3):269–291
- Nowotny T, Huerta R (2012) On the equivalence of Hebbian learning and the SVM formalism. In: 46th annual conference on information sciences and systems (CISS), 2012, pp 1–4
- Nowotny T, Huerta R, Abarbanel HD, Rabinovich MI (2005) Self-organization in the olfactory system: one shot odor recognition in insects. *Biol Cybern* 93(6):436–446
- Olsen SR, Wilson RI (2008) Lateral presynaptic inhibition mediates gain control in an olfactory circuit. *Nature* 452(7190):956–960
- Perez-Orive J, Mazor O, Turner GC, Cassenaer S, Wilson RI, Laurent G (2002) Oscillations and sparsening of odor representations in the mushroom body. *Science* 297(5580):359–365
- Riley JR, Greggers U, Smith AD, Reynolds DR, Menzel R (2005) The flight paths of honeybees recruited by the waggle dance. *Nature* 435(7039):205
- Schmuker M, Yamagata N, Nawrot M, M R, (2011) Parallel representation of stimulus identity and intensity in a dual pathway model inspired by the olfactory system of the honeybee. *Front Neuroeng* 4:17
- Schmuker M, Pfeil T, Nawrot MP (2014) A neuromorphic network for generic multivariate data classification. *Proc Natl Acad Sci* 111(6):2081–2086
- Schwaerzel M, Monastirioti M, Scholz H, Friggi-Grelin F, Birman S, Heisenberg M (2003) Dopamine and octopamine differentiate between aversive and appetitive olfactory memories in drosophila. *J Neurosci* 23(33):10,495–10,502
- Seelig JD, Jayaraman V (2015) Neural dynamics for landmark orientation and angular path integration. *Nature* 521(7551):186–191. doi:[10.1038/nature14446](https://doi.org/10.1038/nature14446)
- Serrano E, Nowotny T, Levi R, Smith BH, Huerta R (2013) Gain control network conditions in early sensory coding. *PLoS Comput Biol* 9(7):e1003133
- Smith BH, Huerta R, Bazhenov M, Sinakevitch I (2012) Distributed plasticity for olfactory learning and memory in the honey bee brain. In: *Honeybee neurobiology and behavior*, Springer, pp 393–408
- Srinivasan MV (2014) Going with the flow: a brief history of the study of the honeybee's navigational odometer'. *J Comp Physiol A* 200(6):563–573. doi:[10.1007/s00359-014-0902-6](https://doi.org/10.1007/s00359-014-0902-6)
- Strube-Bloss MF, Nawrot MP, Menzel R (2011) Mushroom body output neurons encode odor-reward associations. *J Neurosci* 31(8):3129–3140
- Strube-Bloss MF, Nawrot MP, Menzel R (2016) Neural correlates of side-specific odour memory in mushroom body output neurons. *Proc R Soc B* 283(1844):20161270
- Stürzl W, Zeil J (2007) Depth, contrast and view-based homing in outdoor scenes. *Biol Cybern* 96(5):519–531
- Stürzl W, Böldcker N, Dittmar L, Egelhaaf M (2010) Mimicking honeybee eyes with a 280 field of view catadioptric imaging system. *Bioinspir Biomim* 5(3):036,002
- Szyszkka P, Ditzen M, Galkin A, Galizia CG, Menzel R (2005) Sparsening and temporal sharpening of olfactory representations in the honeybee mushroom bodies. *J Neurophysiol* 94(5):3303–3313
- Szyszkka P, Galkin A, Menzel R (2008) Associative and non-associative plasticity in Kenyon cells of the honeybee mushroom body. *Front Syst Neurosci* 2:3. doi:[10.3389/neuro.06.003.2008](https://doi.org/10.3389/neuro.06.003.2008)

- Tolman EC (1948) Cognitive maps in rats and men. *Psychol Rev* 55(4):189
- Turner GC, Bazhenov M, Laurent G (2008) Olfactory representations by. *J Neurophysiol*, pp 734–746. doi:[10.1152/jn.01283.2007](https://doi.org/10.1152/jn.01283.2007)
- Vitay J, Dinkelbach HÜ, Hamker FH (2015) ANNarchy: a code generation approach to neural simulations on parallel hardware. *Front Neuroinform* 9:19. doi:[10.3389/fninf.2015.00019](https://doi.org/10.3389/fninf.2015.00019)
- von Frisch K (1967) The dance language and orientation of bees. Harvard University Press, Harvard
- Wessnitzer J, Young JM, Armstrong JD, Webb B (2012) A model of non-elemental olfactory learning in drosophila. *J Comput Neurosci* 32(2):197–212
- Wilson RI, Laurent G (2005) Role of gabaergic inhibition in shaping odor-evoked spatiotemporal patterns in the drosophila antennal lobe. *J Neurosci* 25(40):9069–9079