



# **Neural synchrony as image familiarity in ant route navigation**

**Timothy Parker Russell**

Submitted for the degree of Master of Science  
University of Sussex  
September 2018

## ABSTRACT

Ants use visual information to quickly and accurately learn routes through their environment, despite a small brain and low-resolution visual system. This navigation may be driven by a search for familiarity between a current view and the views previously experienced along the target route. There is little consensus on how this familiarity measure is implemented, at least in a biologically plausible way. A recently proposed general familiarity measure, whereby an input history is encoded in a spiking neural network and the synchrony of spike timing is measured for a new input, could work well here. This project evaluates the use of this measure by extending it to a basic navigation task, using real images and various metrics. Its performance is found to be relatively weak, but experimental shortcomings and the plausibility of the method show that further investigation is warranted.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Background</b>	<b>3</b>
2.1	Visually guided ant navigation . . . . .	3
2.1.1	IDF and RIDF . . . . .	4
2.2	Cortical spike synchrony as a familiarity measure . . . . .	4
<b>3</b>	<b>Methods</b>	<b>8</b>
3.1	Replicating ‘Cortical Spike Synchrony as a Measure of Input Familiarity’ . . . . .	8
3.2	Extension for visually guided ant navigation . . . . .	9
3.2.1	Datasets and routes . . . . .	10
3.2.2	$R_{syn}$ vs. similarity and distance . . . . .	12
3.2.3	Calculating RIDF and rotational $R_{syn}$ . . . . .	12
3.2.4	Basic navigation simulation . . . . .	12
3.2.5	Parameter tuning . . . . .	12
<b>4</b>	<b>Results</b>	<b>15</b>
4.1	Replication . . . . .	15
4.2	Comparing $R_{syn}$ , similarity and distance . . . . .	18
4.3	Comparing RIDF and rotational $R_{syn}$ . . . . .	19
4.4	Navigation ‘simulation’ as a quiver plot . . . . .	22
<b>5</b>	<b>Discussion</b>	<b>23</b>
<b>6</b>	<b>Conclusion</b>	<b>25</b>
	<b>Bibliography</b>	<b>26</b>

# Chapter 1

## Introduction

Considering the small size of their brains, together with the low resolution of their visual systems, it is surprising to find that ants exhibit very robust navigational behaviour based on vision [1, 2, 3]. There is much research, but little agreement on the mechanisms underlying this fast route-learning and accurate traversal displayed by ants when moving between nests and foraging locations. One mechanism that appears to be a constant, however, is the comparison of views as part of a search for familiarity [2, 4]. This often takes the form of the ‘snapshot’ approach, where a view of the environment as seen from the goal location is memorised in some way, before being compared to the view from the ant’s current location [5]. Navigation thus becomes a search for the goal using a measure of familiarity. While having a single goal presents issues over long distances [6], leading to proposed techniques including the entirety of the route, the method is relatively sound in general and forms the basis of this project: finding a novel and biologically plausible method of determining image familiarity, which can be used for navigation in a fast and robust way.

There are of course many proposals for such a measure, as are outlined in 2.1. Here we try to take an approach different to that of feature detectors [2], classifiers [2, 3] and the like, or neural networks specifically designed to output a probability [3, 7, 8]. Is it possible to create a biologically ‘realistic’ spiking neural network and analyse what happens within it during a navigational task to derive a method of image comparison? Korndörfer et al. [9] have recently published their implementation of a phenomenon that could be of use here. They claim that neuronal spike synchrony—that is, the proportion of neurons firing simultaneously—in a spiking neural network can be used as an estimate of the similarity between an input to the network and the historical inputs to the network. This of course relies on the input history being encoded in the network, and it has been shown that ants should be able to encode a visual route history in the ‘mushroom body’ area of their brain, with image capacity theoretically in the hundreds [3].

Using the simulation found in [9] as a basis for experimentation, this project aims to evaluate its use for the aforementioned method of visual navigation centred around determining image similarity. This uses real images, rather than the randomised patterns found in [9], and is a technique that compares the view at a current location with the entire, encoded route (rather than the snapshot method).

# Chapter 2

## Background

### 2.1 Visually guided ant navigation

While an overview of visually guided ant navigation is given in Chapter 1, it would be useful to explain specific past methods for simulating ant navigation and determining familiarity. The snapshot model, as previously described, dominates in terms of the general navigational strategy [10], although there are issues with longer distances [6, 10] and with the need for the insect to align its current orientation with the snapshot [10].

It has been stated that there is no reason that ants could not encode images along their route-to-memorise, and perform evaluations on these multiple views for route guidance [3, 10, 11]. In [2], a (boosted) classifier is trained with on and off-route images, using downscaling and feature detection on the images for low-dimensional image representation. They are successful in proving that this approach can produce a working route navigation system. However, there is no claim that this is similar to the actual neural processing of the ant. The requirement of a suitably sized dataset of labelled images is also infeasible.

Baddeley et al. [4] use the ‘infomax’ learning algorithm [12] to eliminate this problematic focus on specific training images. The two-layer artificial neural network is trained on the route views, which are presented individually and then discarded. This encoding of the views experienced along the route has the advantage of lessening the reliance on a ‘perfect memory’ of the views—in fact, there is a direct comparison in this paper with the ‘perfect memory model’ of navigation, where the sum squared pixel difference is used for image discrimination. A probability of a novel view being part of the encoded route views is output, driving route navigation with much success. Again, however, the infomax algorithm cannot be said to represent the ant’s neural processes, although the encoding process here is important, as it appears to have some biological plausibility. One approach in [7] is the use of a Restricted Boltzmann Machine

(RBM) as an ‘autoencoder’ network, which can ‘learn a compact representation of the distribution of views experienced along the route’ and outputs a probability of a novel view being part of this training set. This sounds promising, but is a computationally heavy algorithm and requires a ‘burn-in’ period that is at odds with the ant’s fast, even ‘one-trial’ learning of a route. Additionally, an early exploration of ‘deep autoencoders’—compressing/encoding images for navigation using deep learning methods—can be found in [8].

Ardin et al. [3] attempt a more directly biological, as well as holistic, approach. They consider how insects perform path integration (PI) [13], and how this can be combined with their visual systems and memories in various ways. In particular, vector information may be stored together with visual memories, or the PI information may be used to decide which views to store. With the idea that PI can therefore be used as a type of—or to boost—reinforcement, they successfully implement the infomax technique from [4] in their model of an ant mushroom body. This gives credibility to the theory that the navigational processes being studied happen, to some extent, within this structure, which should be an aid to the attempts of future research to learn more about the neural implementation of visual navigation in insects.

### 2.1.1 IDF and RIDF

Zeil et al. introduce what has come to be known as the image difference function (IDF) and rotational image difference function (RIDF) [14]. Both are based on a pixel comparison (this paper uses the root mean squared difference), with the former illustrating how image difference increases with distance—see Fig. 10, bottom row for its usage in this paper—and the latter showing how difference between a reference and rotated image changes with rotation, with a ‘V’ shape expected, particularly when using the same image—see Fig. 11. The IDF and RIDF are compared against this paper’s familiarity measure on the understanding that they give a representation of the information, or information change, realistically available for the measure to use.

## 2.2 Cortical spike synchrony as a familiarity measure

Korndörfer et al., in their 2017 paper [9], attempt to demonstrate that when an input history is encoded in a spiking network, the synchrony of neuron firing provides a good estimate of the ‘match’ between further inputs and that history.

The network is implemented as an Izhikevich spiking neuron model [15] in two dimensions. Fig. 1 provides a visual summary, with details in [15] and [9, p. 15]. In brief, a neuron  $i$  is

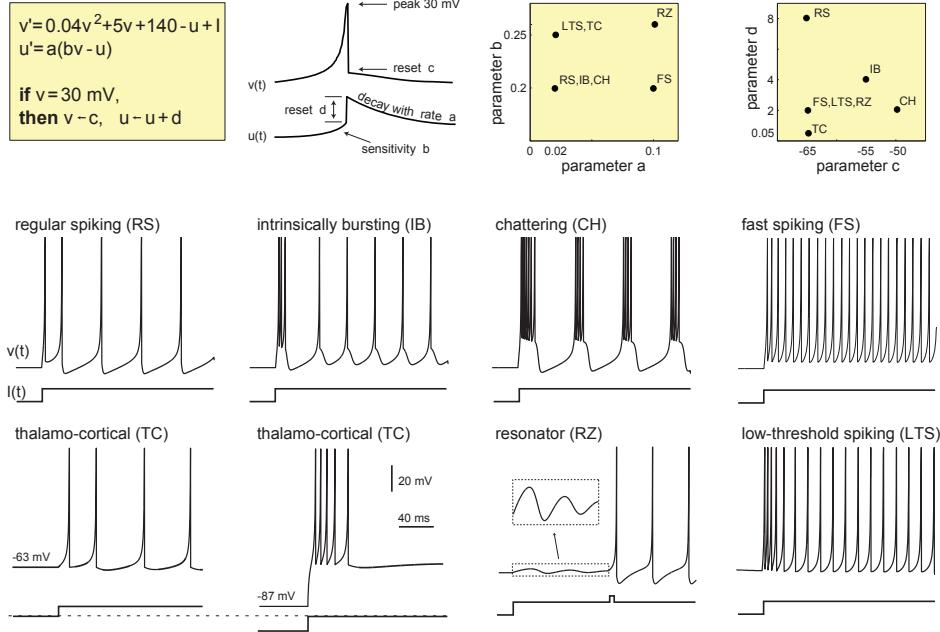


Figure 1: A summary of the Izhikevich model. Electronic version of the figure and reproduction permissions are freely available at [www.izhikevich.com](http://www.izhikevich.com)

modelled with a membrane potential  $v_i$  and a recovery variable  $u_i$ :

$$\dot{v}_i = 0.04v_i^2 + 5v_i + 140 - u_i - I_i^{net} - I_i^{up} \quad (2.1)$$

$$\dot{u}_i = a(bv_i - u_i) \quad (2.2)$$

Korndörfer et al. set  $a = 0.01$  and  $b = -0.1$ . The input current  $I$  seen in the original Izhikevich equations (Fig. 2.1) is implemented as  $I_i^{net}$ —current from other cells in the network—and  $I_i^{up}$ , current from ‘upstream’ (that is, external) sources. A spike occurs when  $v_i$  crosses the spike detection threshold, set at  $30\text{mV}$ . This resets the model:  $v_i$  is set to  $-65\text{mV}$  and  $u_i$  is increased by 12.

There is also a complex synapse model, which will not be described in detail—but as summarised in [9, p. 15]: ‘Intuitively, these synapses cause quickly increasing currents in response to incoming spikes, diminishing somewhat more gradually back to zero if no additional spikes arrive’. Fig. 2 shows this visually.

Here we concentrate on Korndörfer et al.’s ‘familiarity’ experiment shown in [9, Fig. 4] and later reproduced for this paper: see 3.1. They use a network of size  $15 \times 15$ , with cells initially being connected in a random and local way: specifically, the euclidean distance between each pair of cells is used in determining the probability of connection  $P_{connect}(u, v) = \max(0, d(u, v)^{-1} - c)$ . For the ‘imprinting’ stage, 10 random stimulus patterns (cells clustered around a randomly chosen center point, with the probability of being activated decreasing with distance) are inten-

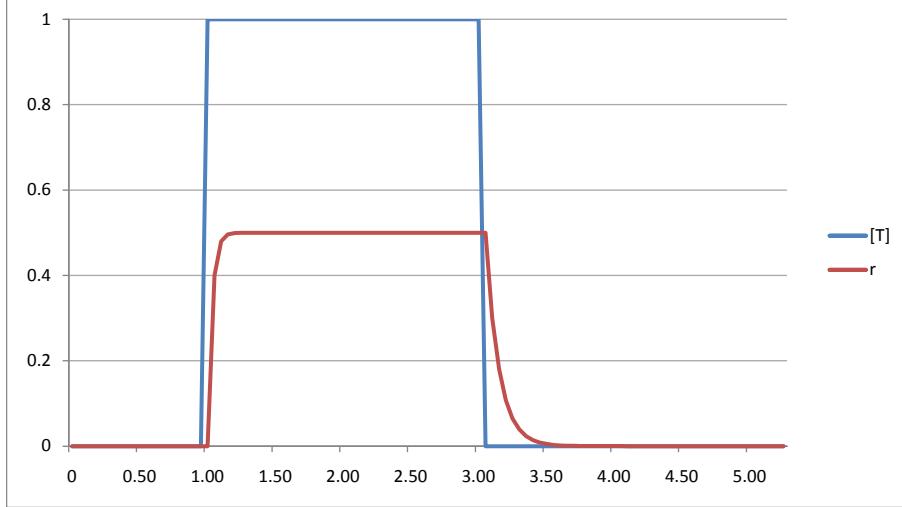


Figure 2: A basic graph of the synapse model responding to input over time.  $[T]$  can be seen as analogous to the input spike, here acting over a  $dt$  of 1 to 3, and  $r$  as the current generated.

ded to become the stimulus history of the network. This is achieved by modifying the connection parameters between cells that co-occur in the pattern and the network. In the given connection probability equation, the cutoff  $c$  is relaxed from  $c = 0.3$  to  $c = 0.15$ , allowing for connections of a greater distance to occur (and, therefore, for cells to be more directly connected overall).  $g^{net}$ , the maximum lateral synapse conductance, is also increased, from  $g^{net} = 1$  to  $g^{net} = 15$ , meaning that these connections are stronger. The network structure is kept fixed from this point onwards.

Further stimulus patterns, generated in the same way, are then used as new stimuli to drive the network. For the purposes of this experiment, the ‘familiarity’ between a ‘new’ pattern and the patterns used to define the network is measured as the ‘fraction of its cells shared’ with those patterns. Via rejection sampling, patterns are generated and sorted by familiarity into 9 bins (in range  $[0, 1]$ ), with one pattern in each bin. The entire procedure is then repeated until there are 100 patterns in each bin.

The synchrony measure is then applied as each new pattern is presented to the network, in the form of spike trains, causing the network to respond. Specifically, the paper utilises the method described in [16] by measuring ‘the average degree of zero-lag synchrony of a particular subpopulation in the network’. First the causal exponential kernel  $k(t) = e^{2t}$  is used to convolve spiketrains from each neuron, which gives an activation trace  $A_i$  for each neuron. The synchrony of a population of neurons,  $S$ , during an interval,  $T$ , can then be calculated ‘by the variance of

the mean field of  $S$ , normalized by the average variance of the members of  $S'$  [9, p. 16]:

$$R_{syn}(S, T) = \frac{\widehat{\text{Var}}[\langle A_i(t) \rangle_{i \in S}]_{t \in T}}{\langle \widehat{\text{Var}}[A_i(t)]_{t \in T} \rangle_{i \in S}}$$

Generally, they find that neural synchrony, and therefore  $R_{syn}$ , increases as the similarity between the presented and imprinted patterns increases: see [9, Fig. 4] for the box plot presenting their results. The same result is found in this project's replication of the experiment in section 4.1.

# Chapter 3

## Methods

### 3.1 Replicating ‘Cortical Spike Synchrony as a Measure of Input Familiarity’

Much of the code used by Korndörfer et al. [9] has been made available online by Korndörfer himself [17], under the permissive MIT license. This code was used in this project to replicate—and later used as the basis for extensions to—the ‘familiarity’ experiment displayed in [9, Fig. 4].

After running initial tests on a personal computer via a virtual machine running the ‘Lubuntu’ Linux distribution, it was found that the experiments planned for this research were taking an inconvenient amount of time to complete. It was therefore decided to move codebase to the ‘Cloud9’ integrated development environment<sup>1</sup> for Amazon Web Services (AWS), in order to execute what was very computationally complex and long-running code on a powerful Intel® Xeon® E5-2666 2.90GHz CPU. If time had permitted, converting the simulation to run on a GPU, for example using the ‘GPU enhanced Neuronal Networks’ (GeNN) project<sup>2</sup>, would have further shortened the runtime. Apart from the hardware considerations, the replication was of course quite simple given that the code was available.

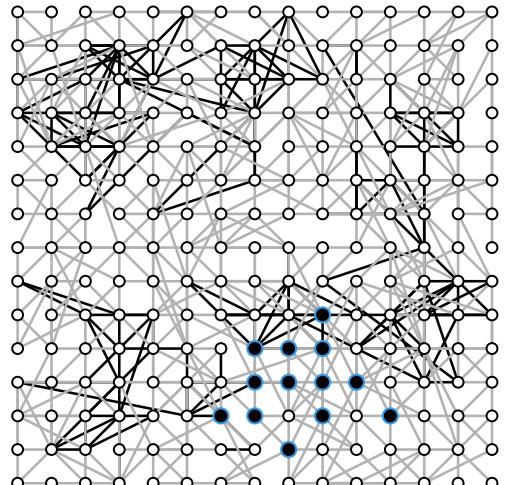


Figure 3: Example network, with a presented pattern as black nodes with a blue outline. Pattern similarity = 0.46.

<sup>1</sup><https://aws.amazon.com/cloud9>

<sup>2</sup><http://genn-team.github.io/genn>

The code itself consists of three main parts. The first, written in C, is a numerical simulation of a noise-driven Izhikevich spiking network. When called, it runs a single simulation of the network and writes the resulting voltage traces into a buffer. Apart from the modification of certain parameters, as will be discussed in 3.2.5, this remained unchanged throughout the project.

The second part is a Python ‘wrapper’ for the network simulation. It provides many functions for preparing experiments, retrieving data from the simulation; and interpreting and visualising the results. The existing functions were very useful in creating new experiments and helping to visualise the results of the extension. Functionality was also added here for the retrieval, manipulation and conversion to patterns of images used for the extension, as well as the generation of the figures displayed in this document. Details of these will be found later in this chapter.

The third part consists of the Python scripts used to set up the details of and run each experiment, as well as handling the creation and modification of the network. It was decided to replicate Korndörfer et al.’s ‘familiarity’ experiment [9, Fig. 4], as this gives an excellent overview of the method, its performance, and potential suitability for a navigation task, whereas other figures in the paper focus on smaller details of the simulation. The script for this replication was then used as the basis for running the simulation based on image data, as will be outlined in the following section.

## 3.2 Extension for visually guided ant navigation

The aforementioned scripts were heavily modified—including extensive refactoring for code clarity and the creation of modular, reusable functions—in order to change the experiment from a comparison of  $R_{syn}$  and similarity based on randomised patterns, to a comparison based on real images simulating ant vision. It was decided that images of a route through an environment could be used in place of the ‘imprinting’ of random stimulus patterns, with the features of the image determining the pattern. Further images, converted into patterns in the same way, were then used to drive the network and obtain synchrony readings. See 3.2.1 for more detail on the images/image datasets used. The network itself was converted to a size of (90, 7): the same aspect ratio as the ant vision-like images, and as large as gave a reasonable time and space complexity for the simulation. The images themselves were downsampled to have a pixel dimension of (90, 7), before being converted into binary images using the Otsu thresholding method [18]—that is, each originally greyscale pixel was considered ‘on’ or ‘off’ depending on how light it was. Light regions of the image became ‘on’, shown as black, and were used to drive

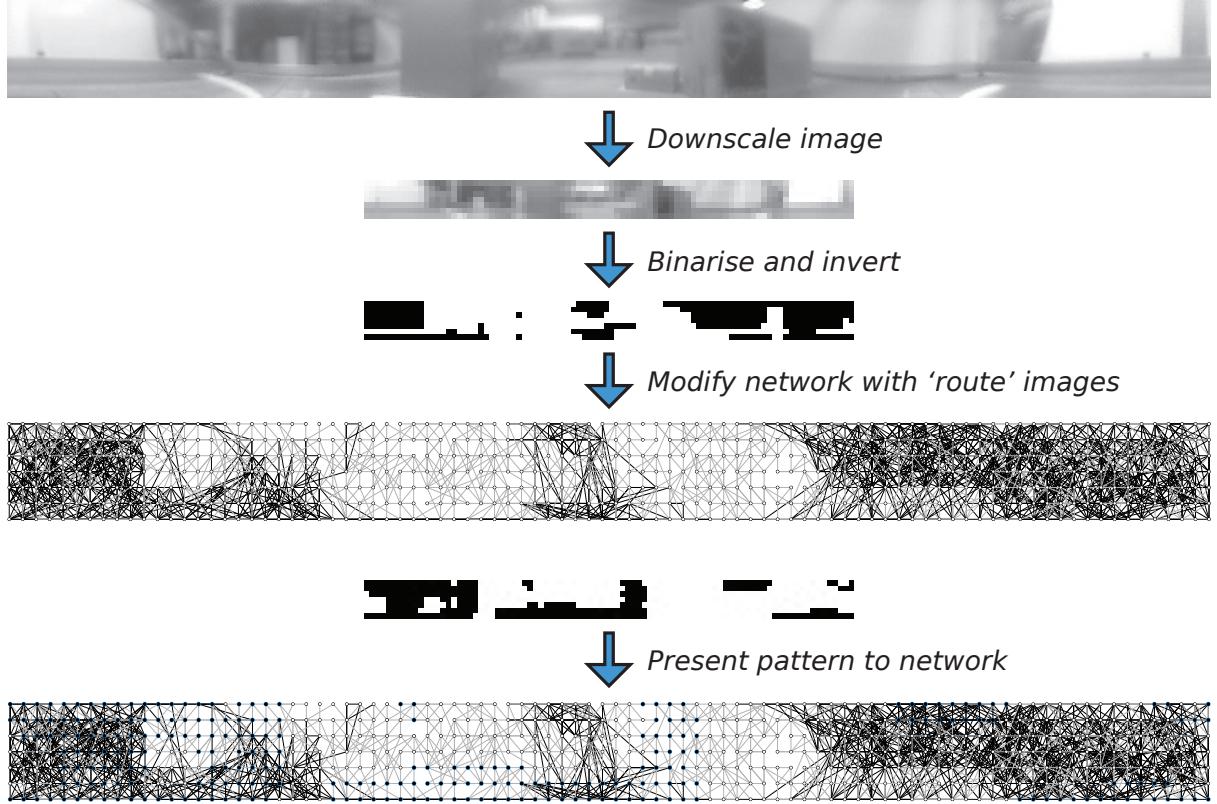


Figure 4: The ‘image process’—downscaling and binarising (720, 58) pixel images to (90, 5) then using them to construct and later drive the Izhikevich spiking network.

the network. Having the conversion this way around made for a slightly sparser pattern when using the ‘boxes’ dataset, which was considered advantageous, although the opposite was true for the ‘plants’ dataset. As the images were 360° panoramic images, the (90, 7) dimensions give a horizontal resolution of 4°/pixel. According to Baddeley et al. [2], this resolution “represents our best guess for the ants’ visual acuity”. This means that the vertical dimension of the image would ideally be 29 pixels; however, the resulting (large) network size would be too computationally demanding for this simulation.

Fig. 4 outlines the above process; and a 40-second animation of network activity for one trial, showing network structure, neuron activation and oscillation cycles can be found online<sup>3</sup>.

### 3.2.1 Datasets and routes

The ‘boxes’ and ‘plants’ image datasets used for the navigation extension consist of 274 and 196 360° images respectively, taken from a larger dataset of images taken using a gantry robot in a 2800mm × 1600mm × 1200mm environment. Further technical details of the gantry setup are given in [2, pp. 4-5]. In order to more realistically recreate an ant’s experience, the images used

<sup>3</sup><https://youtu.be/cJIfgjZyjng>

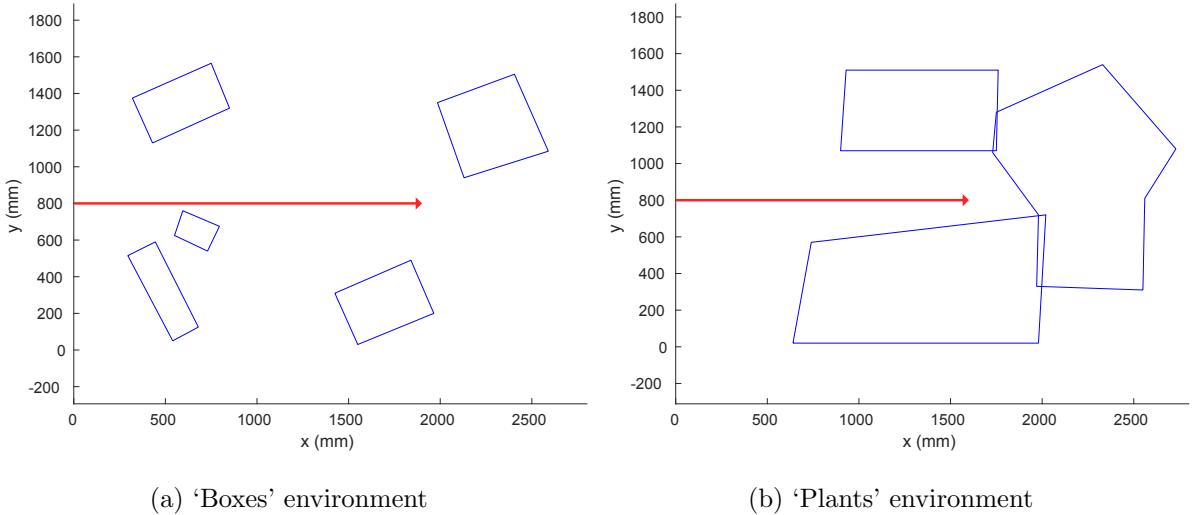


Figure 5: Top-down view of the environment used in the ‘boxes’ (a) and ‘plants’ (b) datasets. Blue-outlined shapes represent the location and  $(x, y)$  dimensions of the boxes in (a) or planters containing the plants in (b), while the red line shows the route location and direction when used in the experiments. Route length in (a) = 20 images across 1900mm. Route length in (b) = 17 images across 1600mm.

should be close to the ground, and so the subset of images taken from the larger, ‘3D’ dataset, for use in this project, were a single horizontal plane as close to the floor as possible. For the ‘plants’ environment, this was 50mm, as low as the camera could go. For the ‘boxes’ environment, this was 150mm, as it was deemed necessary to clear a low obstacle in order to increase the number of possible images, although some ecological validity was sacrificed here. This obstacle was a cardboard box—objects were placed in both environments, to recreate the obstacles found in and used for real-world navigation. For the ‘boxes’ dataset these were cardboard boxes, acting as large, well-defined and simply-shaped features in the images. Although these were simple, the boundaries of the environment were not covered, and it was possible to see into the room beyond, providing more (and less well-defined) features to the images. The environment used for the ‘plants’ dataset included a number of white plastic planters, from which sprout various plants. While the plants present relatively complex shapes, they are very distinct against the background of the environment, which was this time shrouded by a white material. The aforementioned image process lead to the binarised images in this dataset being much more densely populated by ‘on’ pixels, and generally more varied, than the ‘boxes’ dataset. Fig. 5 gives a top-down view of both environments, showing these obstacles, and was provided with the datasets. It also shows the route that was determined in each dataset, used in the imprinting stage to construct the network and act as a route ‘memory’.

### 3.2.2 $R_{syn}$ vs. similarity and distance

‘Similarity’ here refers to the root mean square (RMS) difference between patterns or images, or in most cases the average RMS difference between an image and the particular dataset’s defined route of images. All references to distance use the perpendicular distance between an image and the route, unless otherwise specified.

### 3.2.3 Calculating RIDF and rotational $R_{syn}$

2.1.1 gives an explanation of RIDF. This is compared with ‘rotational  $R_{syn}$ ’ in 4.3 for various images, where similar characteristics were analysed: having  $R_{syn}$  be similar to the corresponding IDF value gives an indication of whether the measure is performing as hoped.

The implementation worked as follows. The downscaling and binarising process described in 3.2 was completed, but the resulting ‘pattern’ of the binarised image was then rotated to give further patterns around the  $360^\circ$  of that image, here in steps of 2 pixels, giving  $8^\circ$  steps. For rotational  $R_{syn}$ , the  $R_{syn}$  was then measured for all patterns, for a number of replications, to give the result, plotted as in section 4.3. A similar process occurred for RIDF, except for IDF being the measurement taken: the RMS difference between the image and a reference image, usually the route image at the same  $x$ -value, was calculated.

### 3.2.4 Basic navigation simulation

It was also decided to visualise an initial impression of the performance of  $R_{syn}$  for navigating along the route in an environment. Time constraints meant that no real-time simulation could take place; however, by selecting various points around the route and calculating which image orientation at that point gave the highest  $R_{syn}$ , a simulation can be approximated. The process was simply a repeat of the rotational  $R_{syn}$  experiment, afterwards finding the index of the maximum value and converting that to a heading. This was then displayed using a quiver plot, as is shown and described in 4.4.

### 3.2.5 Parameter tuning

Here, a ‘parameter’ can refer to multiple things: a variable in the network construction, imprinting or simulation equations, the parameters of the images used and how they were converted to be used with the network, the number of images used, et cetera. Unfortunately, time did not permit a thorough investigation, and so two important areas were prioritised: the ‘route imprinting’ of the network, and the resolution of the images.

The equation giving the probability of connection between two cells, as described in 2.2, gives us the distance ‘cutoff’ parameter  $c$ . This can modify the network structure quite drastically and as such it is useful to study its effects in determining the ideal network, both in terms of biological plausibility and for generating an  $R_{syn}$  useful for visual navigation. Fig. 6 shows performance as rotational  $R_{syn}$ , with part of the network structure at  $0^\circ$  below. Briefly,  $c = 0.15$  was chosen as the default in all experiments due lower values causing a very dense structure of the network, despite the relatively sparse nature of the specific patterns used for this comparison, and a high variance in  $R_{syn}$  to be seen across repetitions of the experiment. See  $c = 0.10$  in 6 for an example of this, and also see  $c = 0.20$  as an example of why higher values were not chosen. It seems that there is no difference caused past a certain value/connectivity structure. This perhaps suggests that there is a threshold of connectivity involved in the firing, or synchronous firing, of neurons in this network.

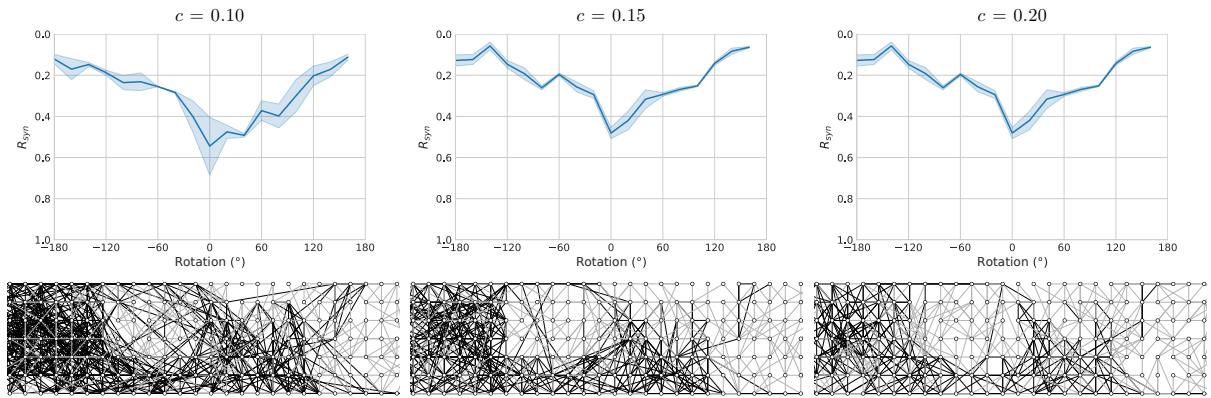


Figure 6: The effects of adjusting  $c$  in the network’s connection probability equation (see 2.2). 0.15 is the default and was used for the main experiments. Rotational synchrony graphs (3.2.3) are shown above, with representations of the network generated at rotation 0 below.

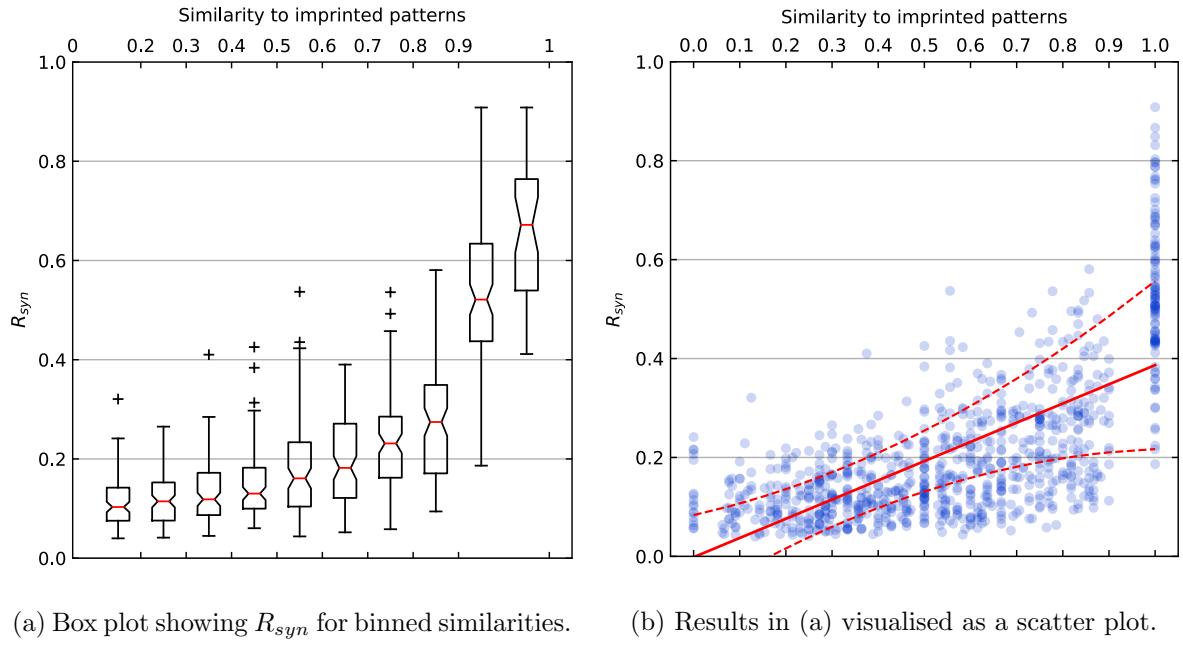
Wystrach et al. [19] explore the effect of resolution and field of view on the information available for visual navigation, finding that low resolution vision, when combined with a wide field of view, can aid navigation rather than hindering it. They similarly use insect navigation combined with RIDF, and indeed claim that ‘a lower visual resolution is better suited to recovery of a route direction’ in these circumstances. This information, combined with the realism of a  $4^\circ/\text{pixel}$  resolution, lead to this realistic horizontal resolution being chosen, combined with a relatively small vertical resolution in order to have a faster running simulation. However, this was initially (90, 5), before it was discovered that (90, 7) appeared to give a marginally better (task) performance while keeping a relatively reasonable runtime. Ideally, this vertical

resolution would have been explored further, especially considering the research around the amount of information contained in the horizon (see [11] for example). There was not an obvious horizon in the ‘boxes’ dataset; however, it could be argued that the ‘plants’ dataset did, only for it to be ‘cut off’ in the original images taken. It would be useful to know how much of a flaw this was—that is, whether it decreased the task performance of the familiarity measure artificially.

# Chapter 4

## Results

### 4.1 Replication

(a) Box plot showing  $R_{syn}$  for binned similarities.

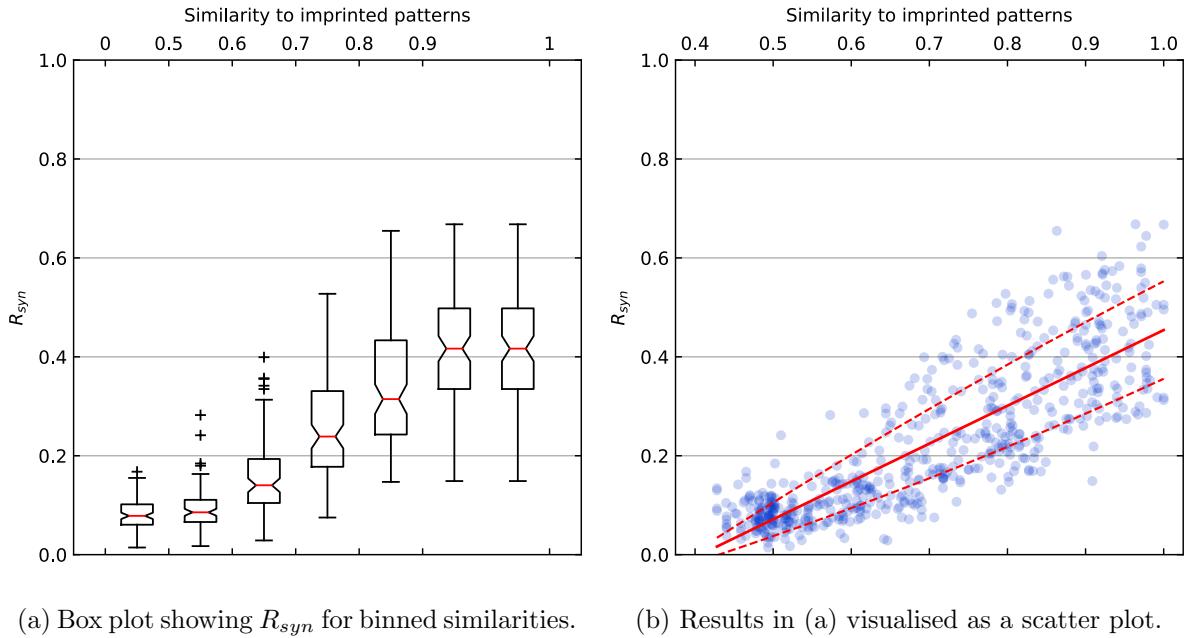
(b) Results in (a) visualised as a scatter plot.

Figure 7: ‘Synchrony as a familiarity measure’: Replication of [9, Fig. 4].  $R_{syn}$  is the measure of zero time lag spike synchrony as described in 2.2. There are 100 samples for each of the 10 similarity bins in (a), each using a random stimulus pattern to drive one of 100 different networks: see 2.2 for more details. The final bar in (a) ‘shows the subset of the penultimate bin where two or more connections per activated cell have occurred’ [9]. (b) visualises the results as a scatter plot in order to see the distribution in more detail. A regression line is shown as a solid red line, with dashed lines showing error.  $R^2 = 0.46$ .

Fig. 7 shows what was found to be a successful replication of Korndörfer et al.’s ‘Synchrony as a familiarity measure’ experiment [9, Fig. 4]. The same general trend can be observed, and

very similar values were found for each similarity bin. In fact, as well as there being fewer outliers overall, where there are small, non-significant differences between the original experiment and the replication—for example, the 0.7 - 0.8 bin having a lower average  $R_{syn}$ —they appear to create a ‘smoother’ increase in the mean.

A scatter plot is also displayed (Fig. 7b) in order to present the data more precisely. Note the vertical bands and the ‘attraction’ of the similarity measure to 0.0, 0.5 and 1.0. This appears to be an artifact of the pattern generation method combined with the small network size, and as such can be considered a limitation of that method. It would be useful to have a similar visualisation of the results from the original paper, to see if the same thing occurred. There is one occurrence of this when using the ‘plants’ dataset, as will be discussed, but the issue is otherwise absent from the image-based extension.

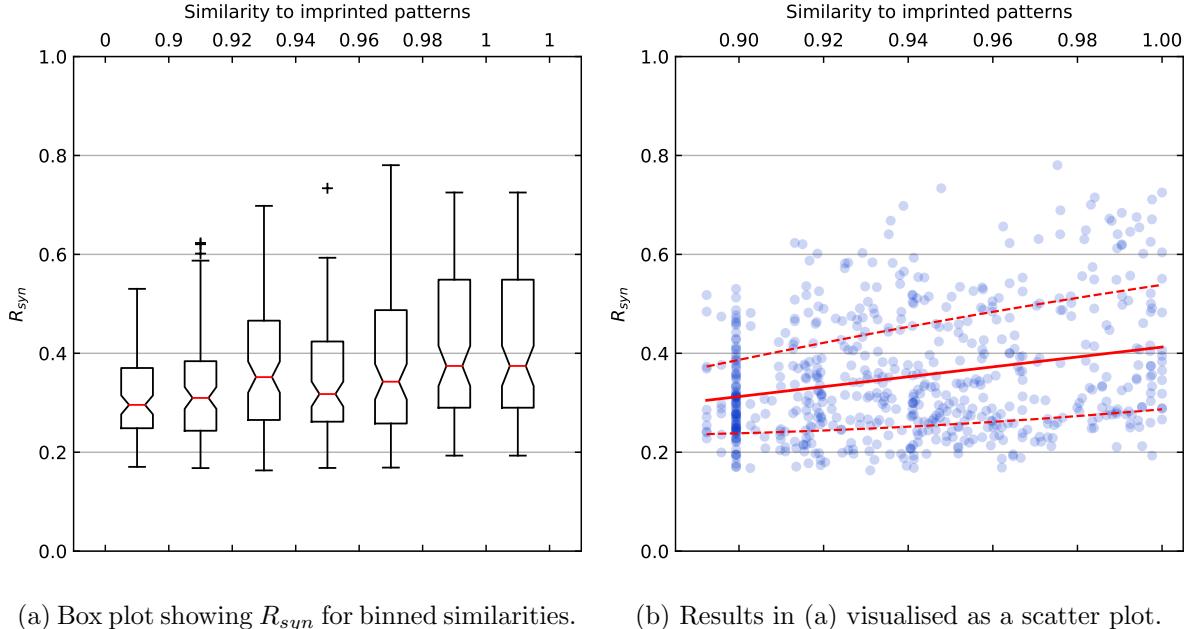


(a) Box plot showing  $R_{syn}$  for binned similarities.      (b) Results in (a) visualised as a scatter plot.

Figure 8: As Fig. 7, showing the results of modifying the experiment with the network and ‘boxes’ image dataset as described in 3.2. In (b),  $R^2 = 0.69$ .

Fig. 8 displays the results of that extension, for the ‘boxes’ dataset, in the same manner. Due to limited variance in the patterns, caused by limited variance in the image datasets, the lowest similarities calculated were around 0.48, hence the limited number of bins. Nevertheless, Fig. 8a shows a strong positive correlation, as does the corresponding scatter plot in Fig. 8b. In fact, when comparing this to the results of the original experiment—both visually and using the coefficient of determination (0.69 to 0.46)—the measure appears more reliable. This is an interesting result: when trying to introduce more natural conditions, with real-world images and

a larger network constructed with images along a route, we find  $R_{syn}$  to be a better predictor of similarity. It is likely that this is caused by the informational properties of the images used (compared to randomised patterns), as will be discussed in Chapter 5.



(a) Box plot showing  $R_{syn}$  for binned similarities. (b) Results in (a) visualised as a scatter plot.

Figure 9: As Fig. 8, with the ‘plants’ image dataset. In (b),  $R^2 = 0.06$ .

It should also be noted that such a clear result was not obtained for the ‘plants’ dataset. This, again, was down to the dataset. The nature of the environment—relatively sparse, but with less clearly defined objects—made for binarised images that were more densely packed with ‘on’ pixels, less well defined and more varied. Considering that individual images were also being compared to the 10-image route, differentiation would intuitively be much more difficult. This had two main effects: making the environment more ‘challenging’ considering the task, and causing similarity values to fall in a smaller range. Thus, Fig. 9 must be approached with care: while the correlation is weak, it is over a small similarity range—but might this not be considered a more natural environment than the ‘boxes’? Also, these high similarities are intuitively the more important when considering the ant’s behaviour, as an accurate similarity measure may be more important when the ant is near the route it is course-correcting towards: the ant would drift far from the route less often and precision becomes less important when needing to make a major correction.

## 4.2 Comparing $R_{syn}$ , similarity and distance

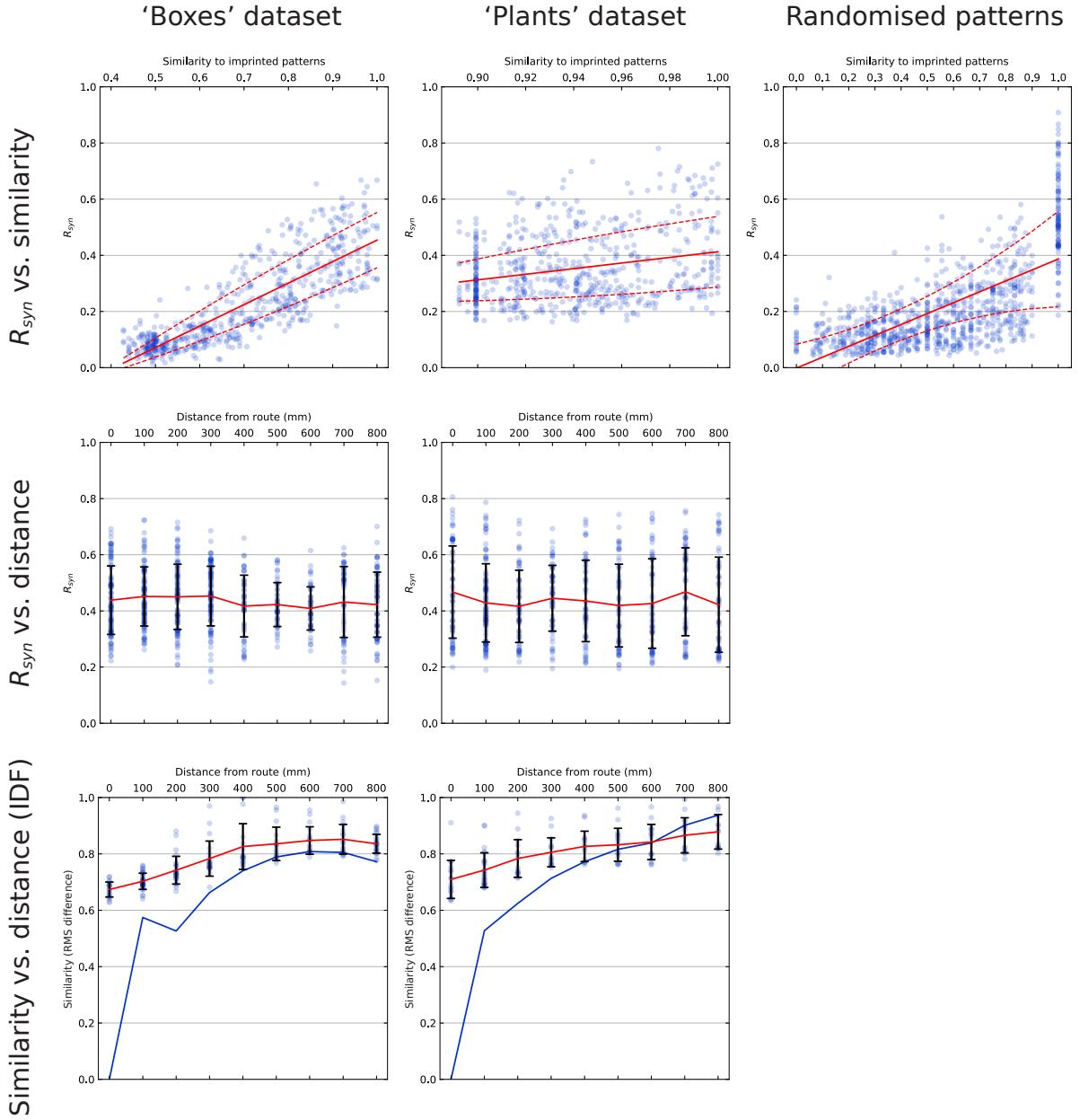


Figure 10: A comparison of results over different datasets, using three different measures. The top row shows  $R_{syn}$  vs. a pattern's similarity to the route, plotted as described in Figs. 7 and 8. The second row shows  $R_{syn}$  vs. an image's distance to the route (with rotation being in the same direction as the route). Bottom row shows similarity vs. distance—or the IDF as introduced in 2.1.1. The red average line, scatter plot and error bars represent the IDF calculated as an average between individual images at varying distance from the route, and all route images, whereas the blue line represents the IDF between the comparison images and their ‘corresponding’ route image (i.e. the image located at the same  $x$ -position).

Fig. 10 allows a direct visual comparison of three methods used to test the familiarity measure and the dataset. The first,  $R_{syn}$  vs. pattern similarity, has already been discussed, but is presented here for completeness and ease of comparison.

The second,  $R_{syn}$  vs. distance, was calculated as described in 3.2.2, and gives a measure of how  $R_{syn}$  changes as the viewpoint for the image being compared shifts in a perpendicular manner away from the route, though maintaining the same orientation. Due to the unchanging orientation, it could be argued that only a small change in the image will occur, whereas an ant at the same distance is likely to be oriented differently from the route and therefore see a more different image. A measure of this can be seen in the bottom row of Fig. 10, showing image similarity vs. distance for the same images used in  $R_{syn}$  vs. distance. The difference, measured as a normalised IDF is indeed relatively small, falling in the range of around 0.1. Although not a direct comparison, it should be noted that this is the similarity range of the plants dataset, and the same pro-and-con discussion of the small range applies.

Nevertheless, the results are not encouraging for  $R_{syn}$  as a familiarity measure. Note the high average  $R_{syn}$  at 300 and 700mm distances for the plants dataset: the measure does not seem to be robust to these image ‘coincidences’ found far from the route, although it works well near the route. This would be a problem in particularly sparse environments. It works better in terms of the general trend on the boxes dataset, however, near the route there is little variation, with the average  $R_{syn}$  even rising slightly from 0-300mm. This will be discussed further in Chapter 5.

### 4.3 Comparing RIDF and rotational $R_{syn}$

So far, the performance of the familiarity measure, calculated with a route, has been mixed. It is interesting, therefore, to consider the results produced without the inclusion of a route. Fig. 11 shows the similarity between the RIDF for an image compared with itself, and the  $R_{syn}$  computed with rotations of that image and a network generated using 10 copies of the same image (see 3.2.3 for details). If we take the RIDF to be an accurate, though relatively simplistic and biologically implausible, measure of image similarity, then if  $R_{syn}$  generated in the way described here ‘follows’ it well, it has performed well. This appears to be the case: although there isn’t a dramatic increase in  $R_{syn}$  at rotation 0, it is still the maximum, falling away with rotation before rising as the images happen to become more similar according to the RIDF, then falling once more. From here on,  $R_{syn}$  will be again be generated using more realistic routes of images moving through the environment (see 3.2.1), making for a less direct comparison but a more realistic test of what is, after all, intended to be a measure used in real-world navigation.

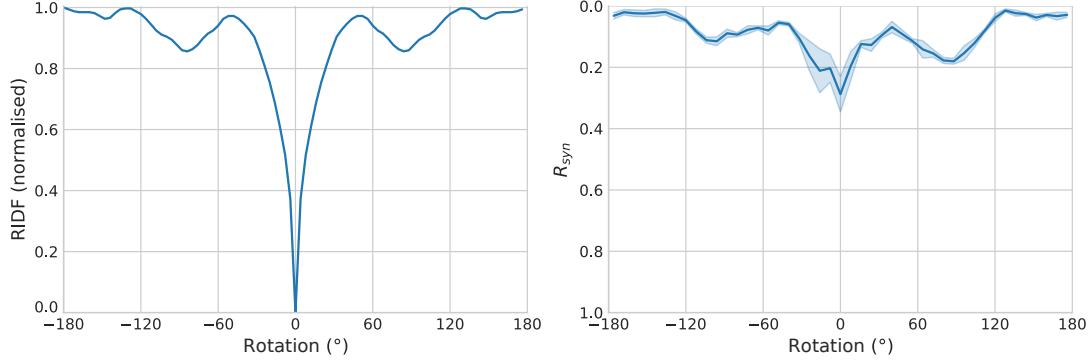


Figure 11: Comparing RIDF and  $R_{syn}$  when using a ‘route’ comprised of 10 of the same image (‘boxes’ dataset, position (1100, 800)) to generate the network. That image was used to calculate RIDF (against itself, left) and rotational  $R_{syn}$  (see 3.2.3, right). The results for the latter were generated with 5 replications at each rotation, moving in steps of  $8^{\circ}$ , or 2 pixels in the 90 pixel-wide image. Blue line shows the mean, with error bands  $\pm$  one standard deviation. This is applicable to all figures in section 4.3

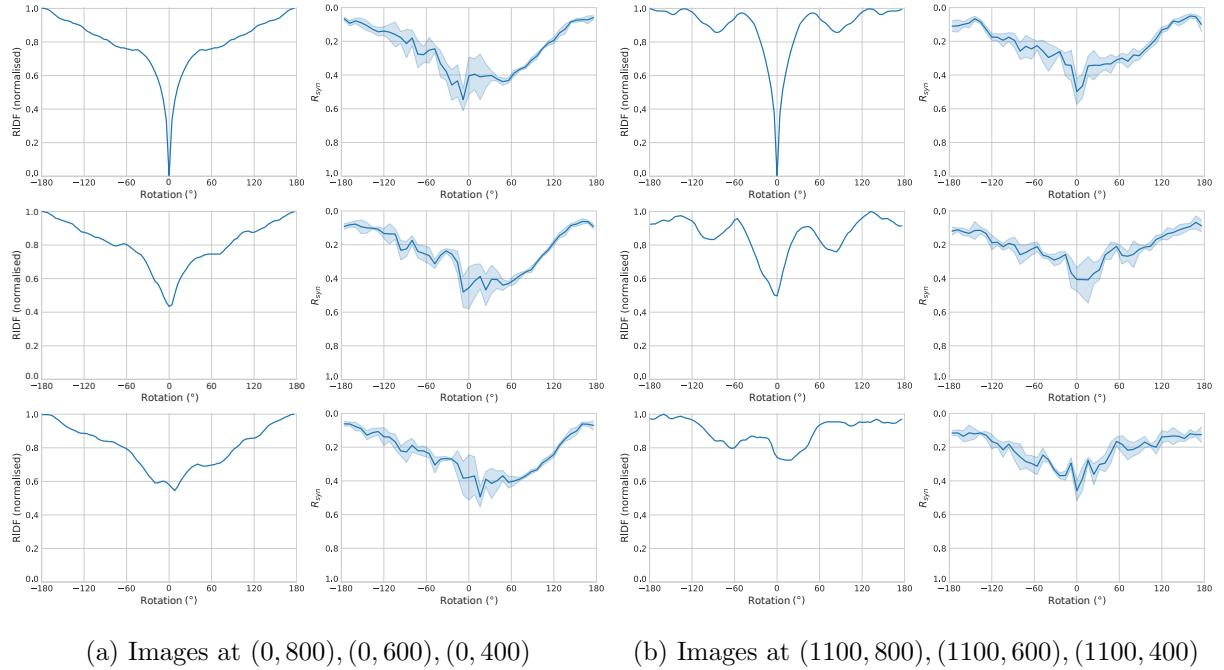


Figure 12: Comparing RIDF and rotational  $R_{syn}$ : ‘Boxes’ dataset. Image locations given as  $(x, y)$ , in mm.

Fig. 12 compares the RIDF and results of running the rotational synchrony experiment (described in 3.2.3) for various images from the ‘boxes’ dataset. In (a), the first image from the route used in the dataset is the comparison image for the RIDF calculation. It is therefore considered as having 0 distance from the route. The image is then used for the rotational  $R_{syn}$

calculation with the route-generated network. While this image is used for the top graphs, the middle and bottom graphs use images increasing in perpendicular distance in steps of 200 mm. This is also true of (b), except for the comparison image being partway along the route. Fig. 13 shows the same for the plants dataset.

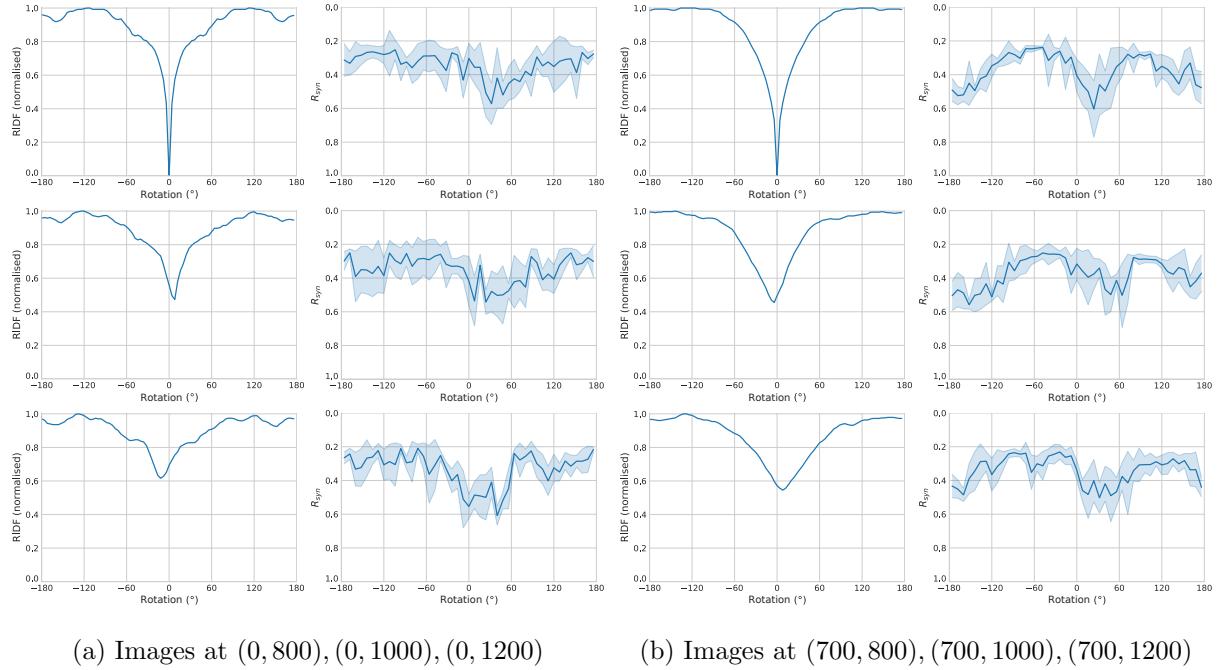


Figure 13: Comparing RIDF and rotational  $R_{syn}$ : ‘Plants’ dataset. Image locations given as  $(x, y)$ , in mm.

Using this rotational method gives more promising results than using perpendicular distance, at least for the ‘boxes’ dataset. The comparison between RIDF and rotational  $R_{syn}$  is somewhat obfuscated by the former being a 1:1 image comparison, with the latter being ‘compared against’ the route. However, we get an idea of where the familiarity ‘peak’ should occur (at a rotation of  $0^\circ$ , shown as a dip due to the inversion of the rotational  $R_{syn}$  graphs), and where we may find unexpected variances in similarity caused by the dataset. For an example of this, see Fig. 12b: after an increase in RIDF with rotation, as expected, there is then a significant fall, reaching a ‘local minimum’ at around  $\pm 85^\circ$ . This is matched by the corresponding rotational  $R_{syn}$ , with a local rise around this rotation.

However, the example given is a rare occurrence, although it may partially explain the aforementioned image ‘coincidences’. Looking for accuracy on a small scale here is likely a poor method of analysis—and indeed most of Figs. 12 and (especially) 13 show poor accuracy. A more thorough and fine-grained simulation would be better, with substantially more replications

than this limited trial. So we move to looking at a general trend. Introducing perpendicular distance as a method of comparison could here be useful. It is expected—and backed up by the RIDF—that the range of similarities over the rotation decreases with distance, caused mainly by the maximum similarity (found at a similar orientation to the route) decreasing. This does seem to occur for rotational  $R_{syn}$  in the ‘boxes’ dataset, albeit by a small and likely insignificant amount.

In Fig. 13, it could be argued that the same occurs in (b), but the overall inaccuracy of the method with this dataset makes analysis difficult, beyond pointing out the ‘spike’ in (b) becoming less pronounced at a similar rate to the corresponding RIDF. Notice the increase in  $R_{syn}$  around  $\pm 180^\circ$  here, though. This is likely due to the route itself having a large variance between its images. They are much different in the latter half of the route, and so could be coincidentally more similar to facing in the reverse direction. If true, this demonstrates the sensitivity of the method to a varied environment.

#### 4.4 Navigation ‘simulation’ as a quiver plot

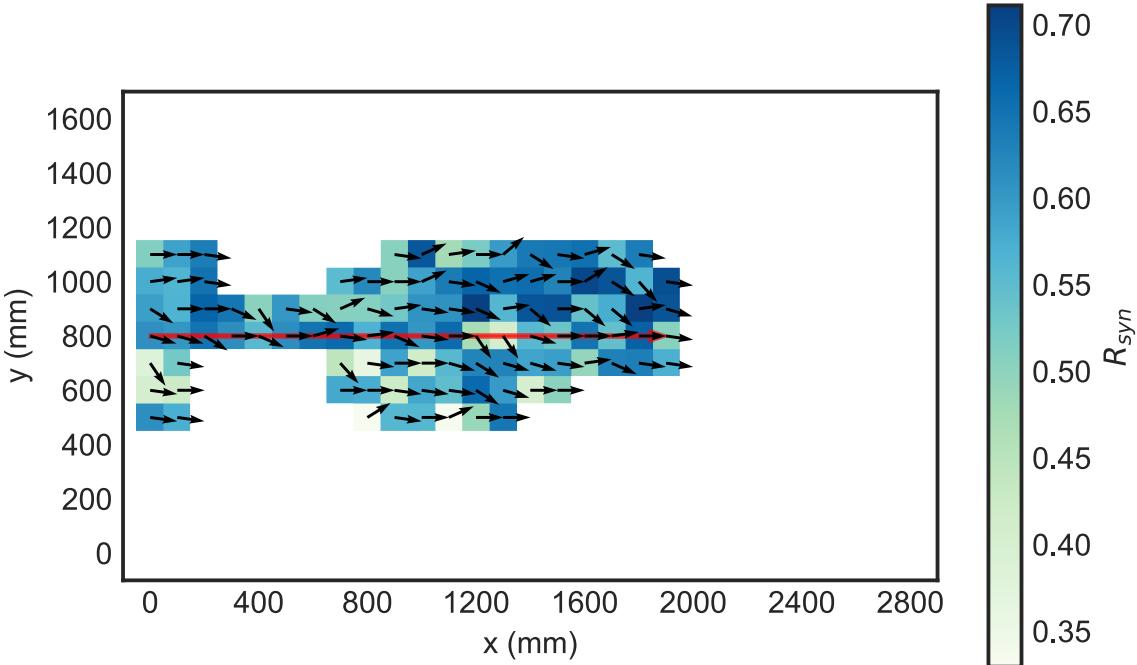


Figure 14: A rudimentary simulation of navigation along a route, as outlined in 4.4. A selection of locations around the route, shown in red, were tested against a network constructed using the route. For each location, images  $\pm 64^\circ$  from the route orientation, moving in steps of  $8^\circ$ , were used. Arrows show the direction where the strongest  $R_{syn}$  was found, with the background colour representing the strength—the numerical value given by the measure.

# Chapter 5

## Discussion

Overall, the results indicate that although the  $R_{syn}$  method can be shown to be generally accurate in more abstract settings, such as when using randomised patterns, or even comparing individual natural images, some serious flaws are highlighted by the introduction of more ‘real world’ conditions such as the route and a real spatial distribution of images. The lack of accuracy in differentiating similar images is an issue, particularly as images in the wild will likely be quite similar on and off-route. This has been exposed by using a navigation task and real-world images, which have a very important difference to the randomised images used in 4.1: ‘features’. These specific structures are present across images and although they can be useful when recognised and used for tasks such as image comparison [11, pp. 449-450], if they are not utilised they simply make images taken from nearby views appear more similar. Perhaps this method does indeed not utilise this information as well as others.

There is also a lack of robustness when it comes to ‘coincidences’ of similarity, images that falsely look in some way like the ‘goal’ images despite being far away. Although this would be a problem for most measures and could perhaps be made less of an issue when considering that an ant could possibly combine this information with other sources of positional information, we could again see the lack of explicit feature detection as a hindrance.

Although the results are fairly negative for the performance of the method, it must be remembered that this is a very limited investigation, especially where it comes to the ‘tuning’ of the various parameters (3.2.5). Further investigations could also use just the image at the end of a route to imprint the network, as in ‘snapshot matching’ [5]. Fig. 11 demonstrated good performance where the comparison was based on a particular image, rather than a route, and although the flaws of snapshot matching have been discussed, it may at least turn out as a strong proof-of-concept for the method.

It is interesting to note the large difference in performance, by almost all metrics used,

between the two datasets. The level of variation along the route in the ‘plants’ dataset was certainly an issue, as well as the smaller total range of differences between the images. Fig. 9 also shows a ‘band’ of that have the same similarity to the route. This could highlight the lack of variance, or possibly an issue with the experiment’s implementation. Either way, if it is unnatural, then it could be considered an ‘unfair’ issue for the method to face—an intentionally simple method could not necessarily be expected to find a difference between images with the same RMS pixel difference. ‘Unnatural’ is of course the key word here, and using an established, realistic dataset would solve many of these problems and let us know more accurately where the method’s problems lie. For example, the ‘tussock’ simulation in [3], based on field data, would be an ideal fit.

# Chapter 6

## Conclusion

Spike synchrony, and certainly the timing of spikes, is an important area not just for the explanation of fast behavioural responses (as advocated in [20]), but also for the encoding and retrieval of information as complex as images, and image familiarity. Although this particular implementation was not found to have great success in simulating ant navigation, its methods were limited, and something as simple as tuning parameters may result in much better performance. Either way, a certain amount of feasibility has been demonstrated, and it is interesting to see a phenomenon such as spike timing used in such a way. After all, most research into neuronal spike synchrony utilises an input which is in some way temporal, whether it is a timed pattern or simply the time of the presentation that is being focussed upon. We can see here, however, that neural spike synchrony, which is temporal in nature, has a possible use as a measure of something which is not.

Other, more accurate proposed and simulated methods of visual ant navigation exist. It is hoped that this neural synchrony based method deserves further attention, however, for ranking high in biological plausibility. Also, although this simulation had long runtimes, it should theoretically be fast—making it useful for certain computer vision scenarios and having an association with literature discussing the use of spike timing for fast responses [20, 21]. A useful next step for this research would be to implement this method within the mushroom body model in [3], as they did with the Infomax algorithm. The mushroom body is a plausible area for this type of processing to take place, so it would be interesting to see if it is possible to find and use  $R_{syn}$  in a biologically plausible way, within a biologically plausible structure.

# Bibliography

- [1] T. Collett and J. Zeil, "The selection and use of landmarks by insects," in *Orientation and communication in arthropods*, pp. 41–65, Springer, 1997. 1
- [2] B. Baddeley, P. Graham, A. Philippides, and P. Husbands, "Holistic visual encoding of ant-like routes: Navigation without waypoints," *Adaptive Behavior*, vol. 19, no. 1, pp. 3–15, 2011. 1, 3, 10
- [3] 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, pp. 1–22, 02 2016. 1, 3, 4, 24, 25
- [4] B. Baddeley, P. Graham, P. Husbands, and A. Philippides, "A model of ant route navigation driven by scene familiarity," *PLOS Computational Biology*, vol. 8, pp. 1–16, 01 2012. 1, 3, 4
- [5] B. Cartwright and T. S. Collett, "Landmark learning in bees," *Journal of Comparative Physiology*, vol. 151, no. 4, pp. 521–543, 1983. 1, 23
- [6] A. Vardy, "Long-range visual homing," in *2006 IEEE International Conference on Robotics and Biomimetics*, pp. 220–226, Dec 2006. 1, 3
- [7] A. Philippides, P. Graham, B. Baddeley, and P. Husbands, "Using neural networks to understand the information that guides behavior: A case study in visual navigation," *Methods in molecular biology (Clifton, N.J.)*, vol. 1260, p. 227244, 2015. 1, 3
- [8] C. Walker, P. Graham, and A. Philippides, "Using deep autoencoders to investigate image matching in visual navigation," in *Biomimetic and Biohybrid Systems* (M. Mangan, M. Cutkosky, A. Mura, P. F. Verschure, T. Prescott, and N. Lepora, eds.), (Cham), pp. 465–474, Springer International Publishing, 2017. 1, 4
- [9] C. Korndörfer, E. Ullner, J. García-Ojalvo, and G. Pipa, "Cortical spike synchrony as a measure of input familiarity," *Neural Computation*, vol. 29, no. 9, pp. 2491–2510, 2017. 1, 2, 4, 5, 7, 8, 9, 15
- [10] A. Philippides, B. Baddeley, P. Husbands, and P. Graham, "How can embodiment simplify the problem of view-based navigation?," in *Biomimetic and Biohybrid Systems* (T. J. Prescott, N. F. Lepora, A. Mura, and P. F. M. J. Verschure, eds.), (Berlin, Heidelberg), pp. 216–227, Springer Berlin Heidelberg, 2012. 3
- [11] A. Philippides, B. Baddeley, K. Cheng, and P. Graham, "How might ants use panoramic views for route navigation?," *Journal of Experimental Biology*, vol. 214, no. 3, pp. 445–451, 2011. 3, 14, 23

- [12] A. Lulham, R. Bogacz, S. Vogt, and M. W. Brown, "An infomax algorithm can perform both familiarity discrimination and feature extraction in a single network," *Neural computation*, vol. 23, no. 4, pp. 909–926, 2011. 3
- [13] R. Wehner, "The architecture of the desert ant's navigational toolkit (hymenoptera: Formicidae)," vol. 12, pp. 85–96, 09 2009. 4
- [14] J. Zeil, M. I. Hofmann, and J. S. Chahl, "Catchment areas of panoramic snapshots in outdoor scenes," *J. Opt. Soc. Am. A*, vol. 20, pp. 450–469, Mar 2003. 4
- [15] E. M. Izhikevich, "Simple model of spiking neurons," *IEEE Transactions on Neural Networks*, vol. 14, no. 6, pp. 1569–1572, 2003. 4
- [16] J. Garcia-Ojalvo, M. B. Elowitz, and S. H. Strogatz, "Modeling a synthetic multicellular clock: Repressilators coupled by quorum sensing," *Proceedings of the National Academy of Sciences*, vol. 101, no. 30, pp. 10955–10960, 2004. 6
- [17] C. Korndörfer, "Source code for 'Cortical spike synchrony as a measure of input familiarity'. <https://github.com/cknd/synchrony>," 2018. 8
- [18] H. J. Vala and A. Baxi, "A review on otsu image segmentation algorithm," *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, vol. 2, no. 2, pp. pp–387, 2013. 9
- [19] 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, 2016. 13
- [20] R. VanRullen, R. Guyonneau, and S. J. Thorpe, "Spike times make sense," *Trends in Neurosciences*, vol. 28, no. 1, pp. 1 – 4, 2005. 25
- [21] D. A. Butts, C. Weng, J. Jin, C.-I. Yeh, N. A. Lesica, J.-M. Alonso, and G. B. Stanley, "Temporal precision in the neural code and the timescales of natural vision," *Nature*, vol. 449, no. 7158, p. 92, 2007. 25
- [22] W. Singer, "Neuronal synchrony: a versatile code for the definition of relations?," *Neuron*, vol. 24, no. 1, pp. 49–65, 1999.
- [23] R. Bogacz, M. W. Brown, and C. Giraud-Carrier, "Model of familiarity discrimination in the perirhinal cortex," *Journal of Computational Neuroscience*, vol. 10, pp. 5–23, Jan 2001.