The Role of Grid Cells in Spatial Localisation and Computation

Daniel Manson
I, Daniel Manson, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm this has been indicated in the thesis.
Abstract

From recording single neurons in rodents and primates, and observing the effects of lesions in humans, it is well recognised that the mammalian entorhinal cortex, and the broader hippocampal formation, perform cognitive tasks relating to both space and memory. In this work I propose a number of specific roles for grid cells and experimentally test aspects of their function. I also give an overview of a piece of software developed for streamlining the process of researching grid cells.

The main theoretical chapter discusses how grid cells might be used to perform two-dimensional vector calculations. Simulations show that computing the vector to a goal can be performed accurately, and in a short time window, using a network topology that might conceivably be initialized during development. It is then suggested that such a network might also support vector addition. A short experimental chapter follows, in which some preliminary data is presented to test this hypothesis.

The main experimental chapter examines how grid cells respond to variations in the level of spatial information present in the environment. In the data presented, grid cells appear to be more stably anchored in a cue rich arena, as compared to an otherwise identical cue poor arena. Differences in grid scale are examined in relation to the inaccuracy of self-localisation, as inferred from spatial tuning.

The final chapter discusses the above findings further and makes suggestions as to their broader significance and the significance of other related recent findings.
Publications

I contributed to the following publications while working towards this thesis:


http://doi.org/10.1016/j.cub.2016.06.043


http://doi.org/10.1098/rstb.2012.0532
# Contents

Abstract ................................................................................................................................. 3
Publications ............................................................................................................................ 4
List of Figures ......................................................................................................................... 8
Preface .................................................................................................................................... 10

1. Introduction ....................................................................................................................... 13
   Overview ............................................................................................................................... 13
   Gross structural features of the hippocampal formation .................................................. 14
   Entorhinal cortex anatomy ............................................................................................... 19
   Representation of space in the hippocampal formation ................................................... 20
      Place cells ......................................................................................................................... 20
      Head direction cells ........................................................................................................ 21
      Border/boundary (vector) cells ...................................................................................... 22
      Grid cells ......................................................................................................................... 23
      Other functional classes ............................................................................................... 23
      Temporal patterns .......................................................................................................... 25
   Theories of memory and spatial cognition ........................................................................ 28
      Memory ......................................................................................................................... 28
      Space ............................................................................................................................... 32
      Memory and Space ......................................................................................................... 34

2. Methods ............................................................................................................................ 37
   Practical methods ............................................................................................................... 37
      Microdrives .................................................................................................................... 37
      Surgery ........................................................................................................................... 37
      Subjects and feeding .................................................................................................... 38
      Single unit recording ..................................................................................................... 38
      Screening/baseline arena ............................................................................................. 38
   Basic analyses .................................................................................................................... 40
      Ratemap, autocorrelogram ............................................................................................ 40
      Gridness ......................................................................................................................... 40
      Jitter/rotation ................................................................................................................. 41
      Cell inclusion criteria ................................................................................................. 41
      Behavioural metrics ...................................................................................................... 41
Adaptively smoothed ratemaps ................................................................. 41
Stability ........................................................................................................ 42
Grid fields ..................................................................................................... 42
Additional analysis methods ........................................................................ 43
  Estimate of composite grid scale ................................................................. 43
  Delaunay method for estimating scale ......................................................... 47
  Maximum likelihood estimate of drift ......................................................... 48
  Estimate of consistent within-trial drift ...................................................... 51
3. Waveform: a GUI for data exploration ................................................... 53
  Abstract ....................................................................................................... 53
  The data ....................................................................................................... 53
  User requirements ....................................................................................... 54
  Design .......................................................................................................... 54
  Implementation ............................................................................................. 58
  Further work ............................................................................................... 59
4. Grid Cells: theoretical considerations .................................................... 60
  Characterising grid cells ............................................................................ 60
    A logical introduction ................................................................................ 60
    Grid cell modules and the attractor ........................................................... 61
    Information density .................................................................................. 65
    The uniquely addressable range ............................................................... 67
  Generating and utilizing the grid cell pattern ........................................... 70
    Explaining phase precession .................................................................. 70
    Advantages of phase precession ............................................................... 72
  Calculating with grid cells .......................................................................... 74
    Overview .................................................................................................... 74
    1D navigation: the problem .................................................................... 76
    1D navigation: the solution ..................................................................... 77
    1D navigation: possible algorithms ......................................................... 78
    2D navigation ............................................................................................ 81
    Simulating the rate coded vector cell model .......................................... 82
    Computing the representation of distal locations .................................... 84
5. Grid cells in the presence of a robot ....................................................... 88
Introduction .................................................................................................................. 88
Methods ......................................................................................................................... 90
   The robot and its control and tracking ................................................................. 90
   Subjects and protocol ............................................................................................ 91
   Rat-robot distance analysis ............................................................................... 91
   Theta skipping analysis ...................................................................................... 91
Results ......................................................................................................................... 93
Discussion ................................................................................................................. 98
6. Grid cells: dealing with uncertainty ................................................................. 99
   Introduction ......................................................................................................... 99
   Methods .............................................................................................................. 103
      Subjects and rich/poor protocol ................................................................. 103
      Boosting familiarization of rich/poor ........................................................ 104
Results .................................................................................................................... 105
   Selecting grid cells for analysis ................................................................. 106
   Behaviour ......................................................................................................... 108
   Stability ............................................................................................................ 112
   Gridness ............................................................................................................ 120
   Scale ................................................................................................................ 122
Discussion ............................................................................................................. 127
7. Discussion .......................................................................................................... 130
   Computing with 2D vectors ........................................................................... 131
   The significance of spatial information ........................................................ 134
   Conclusions ...................................................................................................... 136
Bibliography .......................................................................................................... 138
Acknowledgements ............................................................................................... 150
List of Figures

Figure 1. The connections from the retina to the entorhinal cortex and hippocampus........ 15
Figure 2. Connections of the hippocampus ................................................................. 16
Figure 3. Cross species comparison of the HPC and EC .................................................. 17
Figure 4. Entorhinal cortex anatomy .............................................................................. 19
Figure 5. Spatial cells in the hippocampus ...................................................................... 20
Figure 6. A hypothesised “bionic autoencoder” ................................................................ 24
Figure 7. The theta-speed relationship ............................................................................. 26
Figure 8. Phase precession ............................................................................................... 27
Figure 9. Idealised characteristics of memory retention over different time scales ............ 31
Figure 10. Using the cosine/sine rule for 2D path integration ........................................ 33
Figure 11. Estimating composite grid scale ...................................................................... 45
Figure 12. The Delaunay method for estimating scale .................................................... 47
Figure 13. Maximum likelihood estimate ......................................................................... 48
Figure 14. The warp metric .............................................................................................. 51
Figure 15. A “tile” showing data for a single cluster ......................................................... 55
Figure 16. The mean time of spikes at each point in spike-amplitude space .................... 56
Figure 17. The “file panel” ............................................................................................... 57
Figure 18. Cartoon depicting the canonical grid pattern .................................................. 62
Figure 19. Cartoon of grid patterns from three different modules .................................... 62
Figure 20. Cartoon showing two valid tile shapes that tessellate in a 2D environment ...... 63
Figure 21. Cartoon depicting the conceptual stages of transformation ............................ 64
Figure 22. Estimating a hidden variable from a Gaussian function ................................... 66
Figure 23. Cartoon depicting how resolution was defined by Fiete et al(2008) .................. 68
Figure 24. From 1D to 2D ............................................................................................... 69
Figure 25. The problem of computing a vector, d = b-a ................................................ 77
Figure 26. The differences in the four navigation implementations ................................... 78
Figure 27. The rate coded vector cell model ..................................................................... 80
Figure 28. Composing two 1D axes to navigate in 2D .................................................... 82
Figure 29. Simulations of the rate-coded vector cell model ............................................ 83
Figure 30. Cartoon contrasting vector subtraction and vector addition .......................... 84
Figure 31. From 3D to 2D ............................................................................................... 86
Figure 32. Rat-ball distance for two example trials ......................................................... 94
Figure 33. Rat speed distribution for all trials ................................................................... 94
Figure 34. Example grid cells recorded in the presence of a robot ball ............................ 96
Figure 35. Examples of the Climer MLE method for estimating theta skipping ......................... 97
Figure 36. Variation in theta skipping .................................................................................... 97
Figure 37. The rich/poor experimental protocol .................................................................... 103
Figure 38. Checking for remapping between rich and poor ..................................................... 106
Figure 39. The brittleness of gridness and scale measurement .............................................. 107
Figure 40. Mean speed changes over time ............................................................................. 108
Figure 41. Behavioural differences in rich and poor ............................................................... 109
Figure 42. A selection of grid cells from various animals and days of the experiment ..... 111
Figure 43. Stability ............................................................................................................. 113
Figure 44. Stability of interior 50% of the arena ................................................................. 115
Figure 45. Maximum likelihood estimate of the distance of drift ...................................... 116
Figure 46. Average “warp” distance ..................................................................................... 117
Figure 47. MLE of the distance of drift, with resampling of runs ........................................ 118
Figure 48. MLE of the distance of drift as a function of distance from the arena wall .. 119
Figure 49. Gridness ............................................................................................................ 121
Figure 50. Scale ................................................................................................................ 123
Figure 51. Comparing scale measurements .......................................................................... 124
Figure 52. Scale as measured with Delaunay triangulation on ratemaps .............................. 125
Figure 53. Composite scale ............................................................................................... 126
Preface

From an evolutionary perspective, a “job description” for the position of Mammalian Executive Organ might read as follows:

Seeking an organ to extend life by ensuring a steady supply of food and water, whilst avoiding perilous situations. A proven ability to survive when encountering difficult scenarios is a must, with problems potentially manifesting suddenly and including novel hazards and dilemmas. Ultimately a mate must be secured and offspring successfully reared.

Work must be conducted within strict space and energy budgets and fail gracefully when infrastructure is compromised.

Results will be benchmarked regularly against peers, and must remain competitive.

Successful candidates will be supplied with a starter pack of useful tools, however much of the help provided will be in the form of on-the-job training, over the course of several years.

Candidates will be expected to accept a small number of input data streams, covering a range of sensor modalities, to include some high-bandwidth channels, and some channels with high-temporal resolution. Output will be a stream of muscle contraction commands. Preferred candidates will excel in tasks demanding low-latency responses, as well as in tasks requiring accurate prediction long in to the future. It is thus presumed that candidates have a mechanism for using information from the past to improve performance in the present.

Writing in the 1940s, Tolman imagined that the brain may store incoming knowledge using physical location in the real world as an “index” or “key”. He suggested that such a system would permit a broad range of computational queries to be executed on the data, and would ultimately provide a means to fulfil the demands outlined above (Tolman 1948).

In order to appreciate some of the benefits of this “cognitive map” hypothesis, in which information is annotated to location, consider the following:

An apple falling from a tree generates a blur of colour and a gentle rush of air as it travels, a thud as it hits the ground, and either a strong aroma as it lies decaying beneath the tree or a sweet taste if placed in the mouth.
In this scene, each sense is provided with some information relating to the event, and can (in principle) independently establish the position of the common data source (relative to the organism’s current location). Additionally, all processes occurring at the location in question can meaningfully be said to be a part of this event, because in the real world only one thing/event ever occupies a given location at a specific moment. Taking these two assertions together leads us to the conclusion that all incoming data about an event can be uniquely bound to an “index” representing that location, assuming a timestamp is also annotated in some way.

Tolman, and most subsequent investigators, did not discuss this general form of “information at a distance”, focusing instead on the special case of stimuli at the animal’s own location. In this restricted case, the system no longer needs to do vector calculations to project incoming data back onto a real world coordinate, but it may still do vector calculations when solving spatial problems such as:

- What is the quickest route home? Where can I get a decent meal around here? Am I at that point in the path where I have to be more careful with my footing or should I go quickly to avoid other dangers?

However, exactly how often “space” ought to be considered when performing common actions is not entirely clear. In some cases a full two dimensional spatial calculation can be simplified into a one dimensional spatial calculation, or even a one dimensional temporal calculation. Other times, recognition of a location rather than distances within that space may be important. Many actions may not require any extended knowledge of space at all:

- Eating some food, quietly in a corner. Grooming. Digging a hole. Recognising an acquaintance and predicting their likely behaviour. Deciding if it is likely to remain dry for the next few hours.

Although, one of the benefits of indexing information by location is that truths about the real world tend to have a diminishing significance the further you travel from the original observation: suppose a specific “truth” is the knowledge of the location of a commodity, then travelling a sufficiently large distance will bring another equivalent resource into closer range; equally, awareness of a particular adversary is helpful when close to the known sphere of activity, but generally not relevant outside this region. This is not to say that skills obtained in one location are worthless in other locations: the ability to recognise a particular danger or resource in one location should ideally be transferred to other locations,
however this is a general “skill” rather than knowledge about specific items existing in the world.

Finally, having an “address system” for points in the real world is an advantage in that it becomes possible to store items in space and return to them later. This means an animal can avoid carrying all its possessions with it. Addressed locations can also be employed within a collaborative endeavour in which individuals dedicate their personal resources to building some larger structure of shared utility.

Thus it seems that space is a fairly powerful choice of index. It may also be the easiest choice when it comes to building an implementation: two locations in the real world tend to be very similar in most respects if they are close together, and become less similar the greater the distance between them. This is true of distance in both space and time. So, perhaps the computation which must take sensory data and transform it into a spatial index could simply take this similarity metric, reinterpret it as a distance metric and arrive at a roughly Euclidean/Cartesian-like system for indexing space. Even if this process is not viable in general, it may be helpful during development, allowing a proper metric space to be setup for use during later life.

The last few decades of research in neuroscience and artificial intelligence have brought us closer to an understanding of how the brain is representing and utilizing space. But looking to the future, it is difficult to say whether we should be expecting a long list of incremental improvements to this knowledge, or whether we shall soon encounter a paradigm shift, forcing us to reassess the importance of the questions that appear so significant to the investigators of today.
1. Introduction

Overview

In this thesis I will explore the nature and purpose of the grid cell code, and its role within the larger spatial cognition and memory systems. I shall consider these questions first from a broad theoretical perspective, but then go on to present two experiments that seek to test specific hypotheses.

I begin this introductory chapter by providing an overview of the anatomy of the hippocampus, which is the site of spatial cognition in the mammalian brain. Following this I describe the functional properties of the major known classes of spatial cell within the region. Finally I consider the question of memory, which is the other main function associated to the hippocampus. The chapter concludes by recognising the interplay between memory and the representation of space, and asking what role the grid cells can play in this elaborate system. This question is taken up in earnest in Chapter 4, which focuses entirely on grid cells; it provides a literature review, together with some novel insights, and ends with a substantive section of theoretical work that is largely my own.
Gross structural features of the hippocampal formation

The mammalian hippocampal formation evolved from a simpler structure that existed 300 million years ago; birds and reptiles have a homologous region of similar origin (Bingman 1992). As shown in Figure 1, the entorhinal cortex (EC) and hippocampus (HPC) are two of the regions at the “pinnacle” of the visual processing pathways: signals from the retina must go through a complex web of ten or more intermediate processing steps before reaching these two structures (Felleman and Essen 1991). Somatosensory, and olfactory pathways have similar, though slightly simpler, hierarchies that also reach up to the HPC and EC.

These many processing steps serve to extract - and abstract - meaning from the simple detections of photons, sound waves, odorant molecules, and mechanical forces incident on the organism. For example, near the bottom of the visual hierarchy, V1 responds to all sharp edges in the visual image, whereas near the top, the parahippocampal cortex can distinguish between images of rooms/landscapes and pictures of isolated objects (Epstein and Kanwisher 1998; Ekstrom et al. 2003). Information takes several tens of milliseconds to propagate through the network (Suzuki, Miller, and Desimone 1997; Zugaro et al. 2003), so by the time it reaches the HPC and EC not only is it already highly processed, but there is little time left for these two regions to process it further, and potentially influence the system before the next saccade is initiated. Saccades occur every 50-1000ms in humans, and once the image on the retina changes, further visual processing is halted, even if the system had the potential to do more (Hegdé 2008). This reflects the fact that much of the cortex uses retinotopic coordinates to express information, including Brodmann area 7 and the superior colliculus which are part of the circuit that selects the location of the next saccade (Kandel, Schwartz, and Jessel 2000, Chp 39).

Figure 2 shows the connections internal to the EC and HPC, as well as some of the inputs and outputs from/to the basal forebrain and diencephalon. At a gross level, information flows from all the various sensory modalities to the EC, from the EC out into the hippocampal circuit, then returns back to the EC and out to the rest of the brain. One could picture a “wave” of information rolling in from the ocean, up onto the beach of the hippocampus, depositing little bits of knowledge in rock pools at different heights up the beach as it travels: firstly the subiculum, then CA1, CA3, and finally the dentate gyrus (DG). As the wave reaches the DG it begins to recede back down the beach again, possibly picking up old bits of knowledge as it goes, carrying those fragments out into the ocean.
Figure 1. The connections from the retina to the entorhinal cortex and hippocampus, in primates. 187 links are shown, most of which are reciprocal, representing approximately 30-40% of all the possible interconnections. Red linkages represent possible errors or inconsistencies. The hippocampus and entorhinal cortex are not alone at the top of the hierarchy: the frontal, cingulate, retrosplenial, insular regions could also be pictured here. From Felleman and Essen (1991).
Figure 2. Connections of the hippocampus. Created based on Chapter 3 of Anderson et al. (2007) and Figure 1 in Aggleton et al (2000).
This is, however, a simplified picture: in reality the perirhinal, retrosplenial, (and frontal, not shown in the diagram) cortices offer alternative routes in and out of the hippocampus via CA1 and the pre- and para-subiculum. Additionally, the basal forebrain and diencephalon are connected, via the fimbria/fornix, to many places within the hippocampus: for example, the medial septum (MSN) sends afferents to Sub, CA3, and DG; return afferents are sent to the lateral septum (LSN) from Sub, CA3, and CA1.

As with most structures in the brain, there are two hippocampi: one in each hemisphere. In rodents the two are highly interconnected via the dorsal and ventral commissure, however the connections are weaker in primates and humans. The location and relative size of the hippocampus also varies across the species: in rodents the formation corresponds to a much larger fraction of the whole brain and is orientated dorso-ventrally rather than the antero-posteriorly – see Figure 3, (Strange et al. 2014).

One of the important details omitted from Figure 2 is that there are variations in connectivity and other properties along the long axis of the HPC and EC (in our analogy this axis would correspond to the shoreline, i.e. perpendicular to the sloping axis rising up from the water). For example, the afferents to the LSN are arranged such that dorsal HPC projects primarily to dorsal LSN, and ventral HPC to ventral LSN (Strange et al. 2014). The connections to the amygdala (entirely omitted in Figure 2) also vary along this axis, with the implication being that different amygdala nuclei are preferentially innervated by different sections of the HPC’s

Figure 3. Cross species comparison of the HPC and EC. Taken from Strange et al. (2014).
long axis. And, even within a region there are variations along the axis at the level of ion channels: for example, in stellate cells of medial EC, the channels which restore the membrane voltage in response to hyperpolarisation (HCN1-4) exhibit a gradient in the prevalence of a certain subtype (HCN1), which appears to cause a corresponding gradient in the membrane resonance frequency of the cells (Giocomo et al. 2007; Giocomo and Hasselmo 2008; Giocomo and Hasselmo 2009).
Entorhinal cortex anatomy

Grid cells, the primary focus of this work, are located in the medial entorhinal cortex, and I thus focus on the architecture of this region in a little more detail.

As alluded to in Figure 2 and shown in more detail by the right pane of Figure 4, the EC nominally has six distinct layers, which is more than that of the three-layer hippocampal archicortex. Layers I and IV contain only dendrites and axons, however the other four layers are characterized by their selection of principal and inhibitory neuron types, and their specific sets of afferent and efferent connections.

As highlighted in Figure 3 and shown again in the left pane of Figure 4, in rodents, the EC is found only at the most ventral end of the hippocampal dorso-ventral axis, which corresponds to the posterior surface of the brain. And, as shown in the left pane of Figure 4, a major feature of the region is its division into medial and lateral fields (MEC, and LEC). A distinction not made in Figure 2 is that connection to/from the postrhinal cortex (POR) is more prominent in MEC, whereas the connection to/from perirhinal cortex (PER) is more prominent in LEC (Hargreaves et al. 2005). These two regions differ somewhat in their information content, with POR being largely vision-specific, compared to PER which encompasses a broader range of sensory modalities (Burwell and Amaral 1998; Aggleton et al. 2000). Both MEC and LEC project, via the performant path, into the hippocampus proper, and receive return projections, with the exact topography of these connections varying not only between the MEC and LEC, but also along multiple axes.
Representation of space in the hippocampal formation

The HPC and EC have been extensively studied using single unit recording in freely moving rodents (see methods chapter for details on this technique), with additional evidence collected from bats, monkeys, and humans. Using this method, several functional classes of cell have been identified, with each defined by the type of spatial information represented – see Figure 5.

![Figure 5. Spatial cells in the hippocampus. Top: plots showing the path (in black) of four different rats as they ran around a 1mx1m enclosure (with the exception of the data on the far left which is from a 62cmx62cm enclosure). The green dots show the location of the rat each time the relevant cell fires an action potential. Left-to-right: a place cell, head direction cell, grid cell, and border cell. Bottom: alternative representation of the data from the top row, here showing the mean (smoothed) firing rate in each spatial/directional bin. Taken from Hartley et al. (2014).](image)

**Place cells**

The first such class are “place cells”, which exist in CA1 and CA3 (O’Keefe and Dostrovsky 1971; O’Keefe 1976; O’Keefe and Conway 1978) as well as in DG, Sub and EC (Muller 1996). These cells have high firing rates in one or more zones of an experimental arena, and fire at a much lower rate outside these special “fields”. The fields are of the order of 20cm in diameter at the dorsal end of the HPC, but increase to around 10m wide at the ventral end (Kjelstrup et al. 2008). Each cell responds maximally in a slightly different location, so that when considered as a whole, the population labels each spot in the arena with a unique code and constitutes a “cognitive map” of the entire environment (O’Keefe and Nadel 1978). This code develops within a few minutes of exposure to a novel environment (Wilson and McNaughton 1993; Monaco et al. 2014; Alme et al. 2014), and only changes gradually over the course of repeated visits to the environment (Thompson and Best 1990; Lever et al. 2002;
Mankin et al. 2012; Ziv et al. 2013). Modifying various items in the environment demonstrates that place cells are generally anchored to a combination of one or more local and/or distant cues rather than individual features (O’Keefe 1976; O’Keefe and Conway 1978; Shapiro, Tanila, and Eichenbaum 1997; Manns and Eichenbaum 2009; Komorowski, Manns, and Eichenbaum 2009), however such environmental manipulations can lead to significant changes in mean/peak firing rate (Hayman et al. 2003; Fyhn et al. 2007; Leutgeb et al. 2005), especially if the manipulated item is important to the animal (Fyhn et al. 2002). When sufficiently major alterations are made, the representation will change completely, “remapping” to represent the newer version of the arena (Muller and Kubie 1987; Hayman et al. 2003; Wills et al. 2005; Leutgeb et al. 2005; Fyhn et al. 2007), possibly over the course of several days (Lever et al. 2002). In general, when examining the representations of two distinct environments, the pattern of activity in one arena provides little information about the activity in the other (O’Keefe and Conway 1978; Alme et al. 2014), with typically one third of cells active in a given experimental arena (Dombeck et al. 2010). In addition to the above rodent studies, place-like cells have also been recorded in bats (Ulanovsky and Moss 2007), monkeys (Georges-François, Rolls, and Robertson 1999), and humans (Ekstrom et al. 2003).

**Head direction cells**

The second class of spatial cells are “head direction cells”, which exist in many places within the HPC and EC, as well as outside these two regions¹ (J. Taube, Muller, and Ranck, JB 1990; J. S. Taube 1995; Sargolini et al. 2006; Jeffrey S. Taube 2007; Giocomo et al. 2014). Unlike place cells, these cells fire irrespective of the location of the animal, instead being selective for the direction in which the animal is facing: typically a cell will fire at 5-100Hz in a band of 60-150 degrees, and produce almost no spikes outside this band (Jeffrey S. Taube 2007), with the pattern robust to partial (less than 90 degree) rolls or pitches of the head. The directional tuning of these cells persists in the absence of all visual cues (i.e. in the dark); however, when available, the cells strongly anchor to features in the environment: plunging a rat into darkness, rotating the cues, and then restoring light triggers the head direction cells to re-align to the new reference frame within 80-140ms of the light returning (Zugaro et al. 2003).

Unlike place cells, the *relative* preferences of head direction cells are fixed across all environments and all experimental scenarios, including those periods when the absolute orientations drift slowly or undergo abrupt shifts.

---

¹ Specifically the head direction signal has been identified in the pre- and para-subiculum, anterior and lateral dorsal thalamic nuclei, retrosplenial cortex, dorsal striatum, medial precentral cortex, medial prestriate cortex, dorsal tegmental nucleus, and even CA1 (Review by Jeffrey S. Taube 2007).
Soon after the discovery of head direction cells, it was suggested that they constitute an attractor network (B. L. McNaughton, Chen, and Markus 1991; William E. Skaggs et al. 1995): cells representing similar orientation are likely connected together via strong excitatory synapses, and cells representing dissimilar orientations are connected via weaker excitatory, or inhibitory, synapses. The pattern of activity in such a network would tend to favour representing only one “valid” orientation at a time, and would be capable of sustaining that representation for some seconds after the removal of external inputs. In addition to guaranteeing the uniqueness and stability of this “bump”, the network would need to provide a mechanism for shifting it at a rate specified by vestibular (and/or other idiothetic and/or optic flow) data. This shifting could be done using a full lookup table that maps an input orientation and angular velocity to an output orientation (B. L. McNaughton, Chen, and Markus 1991), or by having subpopulations of head direction cells that encode angular velocity in addition to orientation – such cells would need to project their activity asymmetrically in the direction indicated by their preferred velocity (William E. Skaggs et al. 1995; K. Zhang 1996). The final component of this hypothesised network is a mechanism for anchoring orientation to observable features in the environment: Hebbian learning could presumably associate cues with specific arcs of orientation in a manner that reflects both the angular extent and reliability of the cue (William E. Skaggs et al. 1995; Cressant, Muller, and Poucet 1997; K. J. Jeffery and O’Keefe 1999).

Border/boundary (vector) cells
When the dimensions of a rectangular box are stretched or squashed, the fields of place cells appear to also stretch or squash, in some cases decomposing into multiple fields (O’Keefe and Burgess 1996; Gothard, Skaggs, and McNaughton 1996). This behaviour can be neatly modelled by treating the place fields as sums of Gaussian bumps that run parallel to two or more of the walls. Such a model predicts the existence of cells with receptive field “lobes” that are tuned to arena boundaries at a specific distance and compass direction (O’Keefe and Burgess 1996; Tom Hartley et al. 2000; N. Burgess et al. 2000; Barry et al. 2006). Cells of this kind have been termed “boundary vector cells” or “border cells” and have indeed been identified experimentally, both in the subicula regions and in EC (Boccara et al. 2010; Lever et al. 2009; Sharp 1997; Solstad et al. 2008; Savelli, Yoganarasimha, and Knierim 2008). A single such cell will respond to walls of any colour and texture, as well as to “boundaries” defined by vertical drops, even if the animal is able to cross the drop to reach a second experimental platform (Barry et al. 2006; Lever et al. 2009; Solstad et al. 2008). And, as with head direction cells and place cells, the boundary pattern is maintained in the dark. Note that although the majority of cells in this class respond close to a boundary, there are recordings of
true boundary vector cells with non-zero “vector length”, i.e. cells which respond several centimetres away from a boundary (Lever et al. 2009; Solstad et al. 2008).

**Grid cells**

The last of the known functional classes is the “grid cell”. These cells, which exist in medial EC as well as pre- and para-subiculum (Hafting et al. 2005; Boccara et al. 2010; Fyhn et al. 2008), are similar to place cells in so far as they respond maximally at certain locations in an arena, however in this case there are multiple peak firing locations that are arranged in a regular triangular/hexagonal/rhomboid lattice (Bruce L McNaughton et al. 2006). Grid cells are the primary focus of this thesis and will be discussed in more detail in later sections. Here I just note a few important points: the grids are grouped into modules in which all cells have the same lattice size and orientation (H. Stensola et al. 2012; Yoon et al. 2013; Fyhn et al. 2007; Barry et al. 2007), right from the moment the pattern develops (Wills, Barry, and Cacucci 2012); some grid cells are also directionally selective, although less so than pure head direction cells (Sargolini et al. 2006); on exposure to a novel environment, patterns form that are stable on the order of minutes and hours, but over longer periods the regularity, scale, and orientation tend to shift (Barry et al. 2007; Carpenter et al. 2015; T. Stensola et al. 2015). In situations where place cells fully remap, the grid pattern tends to do undergo an equivalent major change, which in this case equates to just a shift and/or rotation (Fyhn et al. 2007).

**Other functional classes**

For most areas in the hippocampal formation (excluding perhaps CA1 and CA3) we are currently able to identify meaningful spatial information in only a minority of cells, perhaps 10-50% (see references above; note in particular that it still not even known whether grid cells are stellate, pyramidal, or both and exactly which parts of the MEC they are found in (Domnisoru, Kinkhabwala, and Tank 2013; Couey et al. 2013; Ray et al. 2014; Tang et al. 2014; Burgalossi et al. 2011)). A few other types of signal have also been observed, such as speed modulation (Bruce L. McNaughton, Barnes, and O’Keefe 1983; Giocomo et al. 2014; Kropff et al. 2015) and sensitivity to elapsed time (Eichenbaum 2014; Kraus et al. 2015).

A number of further studies have examined whether HPC and EC contain spatial information about objects/odours, behavioural states, and abstract concepts such as “goals” (O’Keefe 1976; Cressant, Muller, and Poucet 1997; Fenton and Muller 1998; Wood, Dudchenko, and Eichenbaum 1999; Frank, Brown, and Wilson 2000; Komorowski, Manns, and Eichenbaum 2009; Deshmukh and Knierim 2011; Deshmukh and Knierim 2013; Tsao, Moser, and Moser 2013; Kim et al. 2015). It is clear that these stimuli can exert some influence, especially in lateral EC, however as soon as we depart from foraging in the simple two-dimensional empty
field, the number of possible experiments undergoes a combinatorial explosion and it becomes very difficult to test the precise tuning of individual cells. In the protocols that have been tried, a broad range of cellular responses has been observed. Many of these experiments also encounter semantic (or perhaps logical) problems when attempting to construct a dichotomy between binding an item to a location, and defining a location by the history of its objects/events.2

Figure 6. A hypothesised “bionic autoencoder”, in which a biological neural network is quasi-reflected into an artificial network by adjusting the weights in the artificial network so as to predict real world stimuli and behaviour from observed neural data (e.g. as recorded from the HPC or EC). This entire system acts like an autoencoder in so far as there are only a small number of neurons at the central layer, but they encode sufficient information so as to reconstruct the

2If the item in question is a red ball, and we analogise the naïve hippocampus to a blank puzzle jumbled up in its box, then we are asking: should we position a new puzzle piece on the table and then paint our red ball on it, or should we paint the red ball on a new piece and only then add it to the table? In both cases the puzzle piece (representing the hippocampal cell) encodes ball and location, but in the first case we annotate a location with a ball, whereas in the second case we annotate a ball with a location. The distinction is weak: painting and placing may occur equally well in either order, or indeed simultaneously.
real-world scenario with (some degree of) accuracy. The information encoded by individual neurons in this “central layer” may not correspond neatly to specific categories such as “location” or “object”, however if the artificial network can be successfully trained then we will have demonstrated that the population is encoding certain features of the real world. Indeed if this is possible, we shall end up with something that constitutes a very rough copy of the entire biological network: a computational device of interest to both neuroscientists and AI researchers.

It is possible, perhaps likely, that most cells in the hippocampus do not encode strict categories of information, rather they simply have a propensity for certain classes of data such as “location”, “time”, and “object”. Attempts to segregate and label the cells through endless experimental protocols will thus provide disappointingly meagre results. Difficult as it may be to comprehend, it may be that the network functions perfectly well with these amorphous blobs of data, and that the best we can do is analyse them with something like a “bionic autoencoder” as suggested in Figure 6.

Temporal patterns
Thus far, for specific single cells, I have been describing correlations between firing rate and externally observable variables, such as heading direction and location. While these correlations have strong predictive power for the cells in question, we can do better, predicting major variations in firing rate on the scale of 10 to 100ms.

At a fundamental level, the neuron membrane has two design requirements: it must support integration of voltage change over time and it must support all-or-nothing spiking. Both processes must be achieved with minimal energy, and meet certain temporal constraints. In this context, it is perhaps not surprising that a neuron’s membrane voltage will oscillate in response to certain levels of depolarisation (Giocomo et al. 2007), often having a resonance frequency between 5-12Hz. Further, when thousands of these oscillators are loosely coupled together into networks it is reasonable to expect a degree of synchronisation (Acebrón et al. 2005), creating well-defined relationships between single cell activity and the network mean. Large-scale fluctuations in network activity have been known about for over a century (Haas 2003), and have long been recognised as correlating with certain behavioural states: in particular, the 5-10Hz hippocampal “theta” rhythm increases in amplitude when an animal performs voluntary movements involving the whole body (Vanderwolf 1969) and increases in frequency as the animal runs faster (Sławińska and Kasicki 1998; Jeewajee et al. 2008; Wells et al. 2013), see Figure 7. Lesion studies have examined the extent to which different regions, specifically the MS, exert a driving influence on this oscillation, as well as asking which ion channels/neurotransmitters are most involved (Green and Arduini 1954; Koenig et al. 2011;
Brandon et al. 2011; Brandon et al. 2014; Newman, Climer, and Hasselmo 2014). And recently we have begun to better understand how current flows in/out of the cell at dendrites/soma are responsible for the observed fluctuations (Buzsáki 2002).

Figure 7. The theta-speed relationship. Left: example power spectrum of local field potential in mEC with theta peak highlighted in red, in grey is the raw spectrum, with the black line showing the spectrum after smoothing. Right: example of the relationship between running speed and frequency of theta, the blue line is a least squares fit to the red medians shown for each speed bin. (Both plots are R2112, 21-Feb-2014 trial 3.)

In fact, not only are the firing rates of hippocampal cells modulated by the theta oscillation, but this modulation is dynamic: originally observed in the case of place cells (O’Keefe and Recce 1993; W E Skaggs et al. 1996) and then subsequently for grid cells (Hafting et al. 2008; Jeewaje et al. 2014; Climer, Newman, and Hasselmo 2013), as an animal runs through the field of a given cell, the spikes produced by the cell tend to occur at different phases of the theta cycle; initially the spikes occur at the end of the cycle, but on each of the subsequent cycles the spikes occur at slightly earlier points. This phenomenon is known as phase precession – see Figure 8. When presenting the original discovery (O’Keefe and Recce 1993), the authors listed several potential mechanisms that might be responsible for the effect, as well as a number of possible functions that it might serve. I shall discuss these in more detail later, in the context of grid cells.
Figure 8. Phase precession. Top: example of phase precession for two place cells as the animal runs through their respective fields: the x-axis corresponds to time and spans approximately two seconds, with the vertical lines delineating the cycles of the theta oscillation (displayed along the bottom of the plot). Observe how the spikes for both cells initially come at the end of the theta cycle but then precess towards the start of the cycle. Bottom left: example of phase precession for one place cell over many runs along a linear track. Observe that, as in the top plot, when the rat enters the place field the spikes initially occur at the end of the theta cycle and then over the course of each run they occur at earlier points in the cycle. The diagonal line highlights this trend. Right: phase precession in 2D of a grid cell for “normalised” runs through the cell’s fields: colours correspond to smoothed phase for each spatial bin, and the run-normalisation is such that the mean direction for each run is from left to right. Top and left are taken from O’Keefe and Recce (1993), right is taken from Jeewajee et al. (2014).
Theories of memory and spatial cognition

I shall dedicate a full chapter to discussing the role of grid cells specifically, however here I give a brief overview of theories of spatial cognition and memory more generally.

Memory

In the 1950s, neurosurgeons began experimentally removing large swaths of neural tissue in a small number of patients who presented with serious and seemingly intractable neurological disorders. It was soon realised that bilateral lesioning of the hippocampus had a striking effect on a patient’s ability to form new memories, but without significantly altering other aspects of the patient’s cognition (Scoville and Milner 1957; review by Milner, Squire, and Kandel 1998). For example, the most famous of the subjects, Henry Molaison, (original known as simply HM), was able to remember a three digit number for a few minutes if allowed to concentrate solely on that one task, but would immediately forget the number when distracted. Raw “semantic” information of this kind, as well autobiographical information, are together referred to as “declarative” memories as they can be declared (i.e. described in words) by the subject. Formation of all such memories is thought to be almost entirely hippocampus-dependant. Other types of learning, such as the development of manual skills, can be accomplished without the hippocampus (Milner, Squire, and Kandel 1998).

A decade or so after the original work of Scoville and Milner, Marr hypothesised on the nature of memory, using a reasonably precise, though somewhat restrictive mathematical and computational formulation. His first work concerned the nature of the neocortex (Marr 1970) but shortly after he turned to archicortex (Marr 1971). In particular, he proposed that the hippocampal system temporarily stores a “hash” of the neocortical activity in such a manner that the pattern can later be recovered using only a subset of the original activity. The idea was that during waking hours the hippocampal memory system would gradually become more saturated with stored patterns and eventually begin to falter in its ability to recall and distinguish between them. Then, during sleep, the most important memories would be transferred from the hippocampus to neocortex for longer term storage, and presumably hippocampal connections would be in some sense “reset” to provide a blank slate for subsequent days.

An aside: this idea parallels the memory model used in some programing language runtimes. In particular, the V8 JavaScript engine in Google Chrome uses two “generations” of memory, one for short term and one for long term. Every so often, when the short term memory is too full, the main code execution goes offline
Around this time, O’Keefe and Dostrovsky (1971) made their seminal discovery that principal cells in the rodent hippocampus represent some notion of location. While this result spawned a field dedicated to understanding the cognition of space, it was also of great interest to those concerned with memory as it was now possible to record memory “in action” in the rodent model. In time, it emerged that Marr may well have been correct: during sleep, and especially during specific bursts of hippocampal activity known as “sharp wave ripples”, it seems that spatial events from earlier in the day are “replayed” within the hippocampus (Wilson and McNaughton 1993; Ólafsdóttir et al. 2015).

Further to this finding, and inspired by developments in machine learning, McNaughton and colleagues (McClelland, McNaughton, and O’Reilly 1995) extended Marr’s model, suggesting that the role of the hippocampus is to act as a repository of “training data” (in the machine learning sense) for use by the neocortex. Their argument began with the idea that neocortical connections were unlikely to be updated in direct response to stimuli because the statistics of the very recent past are rarely an accurate reflection of long term statistics; the implication being that a network trying to always learn the statistics of the recent past will be constantly overwriting and corrupting its previous knowledge rather than reaching a long-term stable (and optimal) state. Instead, by having a library of “compressed” patterns stored in the hippocampus, it was imagined that the neocortex would have the opportunity to learn more gradually, with an even sampling of real-world statistics. Further, the existence of a bi-directional connection from hippocampus to neocortex was highlighted as evidence that the system may support some form of back propagation, as implied by the analogy to machine learning.

Indeed, taking additional inspiration from machine learning, we might also observe that it is usually helpful to augment any given training data set using generative processes that add noise, geometric transformations, or other more complex distortions to copies of the original data – doing so requires the original training items to be available for an extended period of time (as is being suggested here), but means that robust generalisations are then learnt from a small number of examples.

And in fact, the concept of generalising was something that Eichenbaum et al. explored in a review a few years earlier (1992), though from a slightly different angle: they proposed that (analogous to sleep) and the “garbage collector” goes through the short term memory checking to see what can be discarded and what should be moved into long term storage. This paradigm works effectively because, like the brain, a large fraction of the newly created memories can be thrown away soon after use.
outside the hippocampus, memories could be formed gradually over the course of many repetitions (that is repetitions of quasi-identical\(^4\) stimuli and/or of patterns of motor activity); however not only would this memory formation be a slow process, but the results would also be atomic in nature, in so far as their constituent parts would not be available for examination or manipulation. In other words, and in agreement with earlier authors, such memories would be non-declarative (due to being unable to examine the constituent parts and associate language to them); in contrast, memories formed in, or via, the hippocampus would be “relational”, that is they would be composed of divisible components which could be examined and manipulated as needed. Such a system would allow for post-hoc modifications of memories, including generalisations of a yet-to-be determined nature. And, if used over a lifetime, this system would allow an animal to develop a complex understanding of the relations between things in the real world, a concept originally introduced\(^5\) by Tolman (1948), and broadly referred to as a “cognitive map” (although this term has since come to refer specifically to spatial cognition).

Eichenbaum et al. (1992) provide a cartoon, reproduced here in Figure 9, giving the temporal range and strength of memory formation in and outside of the hippocampus. Then in support of this picture and their general “relational” hypothesis they cite a number of lesion studies in both primates and rodents. Specifically, they note that even without a hippocampus, animals are eventually able to learn to recognise an object as being familiar, however after each individual presentation of the object they quickly forget the fact that they have recently viewed it (or are unable to demonstrate such a memory within the limits of the match/non-match to sample task). Also, in a more direct test of the relational hypothesis, when the animals are trained on a task involving arrangements of multiple objects, they fail to recognise the objects after the various sets have been shuffled together. And, in a similar vein, they refer to the Morris water maze task, in which an animal is placed in a pool of opaque water and must learn to swim to a platform hidden at a specific location (Morris et al. 1982). The finding is that lesioned animals are able to learn the skill of swimming to a platform, and they are able to learn a specific start-end route, but they fail to build a representation of the arena that allows them to reach the platform from any generalised starting location.

\(^{4}\) One can never step in the same river twice!

\(^{5}\) Not only did he introduce it scientifically, but he proposed it as the basis of a kind of socio-political philosophy that he believed might have prevented the atrocities that took place in the first half of the twentieth century.
Figure 9. Idealised characteristics of memory retention over different time scales, specifically retention in monkeys following a single presentation of an object. Taken from Eichenbaum et al. (1992).
Space

Which brings us to the question of representing space, and locomoting within that space. Humans are unique in the animal kingdom in having constructed an infallible global positioning system, but this technical and cognitive milestone is only one of many in the long history of life on Earth. Without making any strong claims as to the exact timeline, I shall now imagine - based largely on a priori assertions – a possible evolutionary story. This will serve to familiarise the reader with the important concepts in the field. And it will demonstrate how spatial cognition has been maturing slowly, with plenty of scope for complex genetic hard-coding to accumulate (contrast this with the development of language, say, which likely occurred on the order of 1 million years, or less; also note that this notion of complex genetic hard-coding will be especially relevant in the context of theories of grid cell mechanisms and function).

We might label the first step in spatial cognition as simply being able to recognise a pair of locations at two ends of a straight line “commute”: perhaps a dark zone for home and a nutrient-rich zone for feeding. Simply maintaining a fixed body orientation would be sufficient for executing and then reversing this commute as needed, without any specialised orientation or distance measurements occurring.

The next step might have been the development of an accumulator to track distance covered along that one-dimensional axis. This extra information could perhaps have freed the organism to travel at faster speeds without the risk of overshooting the two ends of the commute. And it would have permitted some form of primitive cost/benefit analysis in terms of venturing a long way in any one direction. Also, adding some level of memory for distance into the system would permit a comparison between multiple potential goals, further improving the cost-benefit analysis. In the insect kingdom as well as in mammals, increments in distance are thought to be estimated based on some combination of (a) planned movements, the “motor efference copy”; (b) the output of force receptors between body joints, “proprioception”; (c) optic flow; and (d) integrated acceleration from the vestibular system (Etienne and Jeffery 2004).

The addition of a "compass” based on sunlight or on another readily available global cue, would free the organism to venture out over larger distances without fear of straying from an intended bearing. However, at this stage paths would still be constrained to straight lines away from the home zone.
Once paths become more circuitous it becomes a little more involved for the organism to maintain a record of the vector from its current location to its home zone: the 1d-accumulator and compass bearing could be gradually augmented via an approximation of the cosine and sine rules (as indicated in Figure 10), or perhaps the problem could be decomposed into two or more 1d axes, with independently tracked 1d accumulators ("band cells", have been putatively identified in rodents as performing just such a role (Krupic, Burgess, and O’Keefe 2012; Bush et al. 2015)). Tracking a vector in this way is referred to as “path integration” and is clearly demonstrated by ants, which can take a circuitous outward path when leaving the nest, but still return along a straight line. Indeed if an experimenter moves the ant before it returns to the nest, the subsequent trajectory taken indicates that the ant has re-orientated relative to the sun, but has not updated its homing vector (Etienne and Jeffery 2004).

Figure 10. Using the cosine/sine rule for 2D path integration. Suppose an animal is tracking the vector from its home to its current location, and I use the notation as shown in the diagram here. In words: initially the animal is at location $p_1$, on a bearing $\alpha$ from home, and at distance $a$. It then moves to location $p_2$, covering distance $d$ and at an angle of $\delta$ relative to $\alpha$. In the process it must update its homing vector to have magnitude $b$, and bearing $\beta$. The cosine rule gives us that $b^2 = a^2 + d^2 - 2ad \cos \delta$, which may at first sight look somewhat complicated to implement in a neural network, but in practice each step in the calculation can likely be meaningfully approximated with a reasonably simple transfer function or summation/inhibition step.

Similarly, using the sine rule we find that $\beta = \alpha + \sin^{-1} \left( \frac{d}{b} \sin \delta \right)$.

Next, suppose an animal is able to recognise multiple places (based on visual/odour cues etc.), and it is able to perform path integration when traveling between any pair of locations. With these two facilities at its disposal, we can now imagine the construction of a “network” of learnt locations, with the edges (in the graph theory sense) of the network corresponding to an internally consistent 2D Euclidean geometry (due to the precise nature of the mapping from the real 2D world).

At this point we encounter something of a false dichotomy imposed in the literature: authors typically draw a distinction between “egocentric” and “allocentric”/”geocentric” reference
frames, where in the former case spatial information is expressed with respect to self-location, but in the latter case it is expressed with respect to some fixed external anchor (Cheng and Newcombe 2005; Neil Burgess 2006). However, as can be seen from the exposition here, when encoded in a network, a set of free vectors (i.e. vectors defined by their magnitude and orientation rather than by their position relative to some specific origin) will satisfactorily encode all the spatial information obtained by the animal, i.e. without recourse to any kind of reference frame. Another way of saying this is that when looking at the data encoded at a single node in the network, it appears that the information is egocentric because distances are defined relative to that one location, but since this is true of all nodes, the network as a whole could be thought of as allocentric. Again, note that the valid allocentric interpretation arises only because the external world is consistent and the path integration apparatus can measure distance and angles with reasonable accuracy.

Once all this spatial and non-spatial (odour, visual etc.) information is encoded in the network, one can then imagine performing “virtual” path integration across the nodes in the network in order to answer questions such as: what is the straight line vector joining A to C, where I already know the vectors joining A to B and B to C? This ability to take shortcuts was discussed at length by Tolman (1948) and further developed by O’Keefe and Nadel (1978), who capture the idea of a “cognitive map…as a set of ordered connected places”.

Memory and Space
The structure of this chapter and the discussion thus far, should hopefully have highlighted to the reader the existence of a certain blurring between memory and spatial cognition, with spatial cognition seeming to arise naturally out of the memories formed during locomotion. One exception to this may be the grid cell system, which, with its fixed internal topography, appears to be tied specifically to computing in Euclidean space rather than to storing/recalling arbitrary memories. Perhaps it is thus no coincidence that the entorhinal cortex, the site of the grid cells, is a more recently evolved structure than the hippocampus and its associated subcortical nuclei, in which the more primitive place and head direction signals are located. Note, however, that one function often ascribed to the grid system is path integration itself (Bruce L McNaughton et al. 2006), presumably not in the ultra-simplistic sense discussed

---

6 The oscillatory properties of grid cells (or their neighbours in MEC) could make them good candidates for a more general type of memory involving time as well as space (Hasselmo, Giocomo, and Zilli 2007). Also note that in a certain sense all parts of cortex may play at least some role in memory (Milner, Squire, and Kandel 1998).
above (since we acknowledged that ants are capable of doing this without anything resembling a neocortex), but perhaps in a more general/complex (though usually unspecified) sense. The chapter on grid cells will explore this question in more detail.

Before concluding this section I return to an important point that I introduced earlier when citing Marr (1971): the question of distinguishing between similar but non-identical patterns. A healthy memory system must be able to achieve this “pattern separation”, or else risk similar memories blending together into a blurry mix that emphasises commonalities and fails to encode any specific details. Indeed, the need for an explicit separation is particularly important in the hippocampus as – already mentioned - information is thought to be stored in a highly compressed form (Marr 1971; Rolls and Kesner 2006). Specifically, it is hypothesised that the recurrent network architecture of CA3 allows for the small “hashes” of individual memories to be maintained independently in such a manner that partial recall of a specific memory will be sufficient to drive full recall, likely using CA1 and EC to decompress the hash, adding information back into the memory as it leaves the hippocampus. This was the picture hinted at earlier, when I described a wave of information rushing up onto the rock pools of a beach, depositing kernels of truth as it travels in towards land, and then picking up older memories with its fall back out into the ocean.

The DG, with its privileged location “highest up the beach”, is believed to be the main site of pattern separation. This notion is supported by the fact that, unlike almost anywhere else in the mammalian brain, neurons are constantly being created here throughout an animal’s life (Kee et al. 2007), endowing each day with a marginally different network dynamic from the previous day, and thus providing a means to distinguish between otherwise similar memories. Indeed it is likely that the survival probability for newly born cells in the DG is a function of the rate at which new memories are to be formed, with exposure to novel activities and environments preventing newly born cells from dying prematurely (Kee et al. 2007). The second feature of note for the DG, which again appears to support the region’s role as pattern separator, is the unusually large nature of its axons’ synapses onto CA3 principle cells: the action potential from a single DG cell has the capacity to single-handedly cause downstream CA3 cells to fire (Lawrence, Grinspan, and McBain 2004), and thus a small change in DG network activity can have significant effects downstream.

7 This gradual turnover of cells may also serve to emphasise the recall of recent memories in preference to older ones. Such a feature is particularly relevant in spatial cognition, where knowledge of a location may become invalidated over extended periods of time, especially during any prolonged absence from the area.
In spatial cognition, it is particularly helpful to be able to separate similar patterns, while still being able to reconstruct specific memories from partially recalled information (“pattern completion”). In this context, pattern separation is witnessed as spatial “remapping”, a phenomenon introduced earlier. In fact, if two environments are nearly perceptually identical it can take some time before the pattern separation occurs, and may be a process that involves the grid cells rather than (or just) the DG (Hasselmo, Bodelón, and Wyble 2002; Lever et al. 2002; Hayman et al. 2003; Carpenter et al. 2015).
2. Methods

Practical methods

Microdrives
Each microdrive was formed from eight tetrodes, made by twisting together four strands of 17 micron platinum-iridium wire (90% - 10%) (California Fine Wire, USA). The eight tetrodes were wound onto a pair of 16-pin Omnetics connectors (Omnetics Connector Corp, MN 55432, United States) and attached to a single windable screw mechanism that permitted the experimenter to adjust the depth of the electrodes with a precision of 30 microns (Axona Ltd, St. Albans, UK). Electrode tips were cut, examined under magnification to check for the presence of a clean cross-section, and then electroplated using platinum solution to reduce their impedance, measured to be roughly 150kOhm for a 1kHz sine wave in saline. Note that in a small number of cases, the electroplating step was skipped so as to provide an opportunity to benchmark its effectiveness.

Surgery
Surgical protocol was similar but not identical to that of Barry et al. (2007). Anaesthesia was induced and maintained with an isoflurane-oxygen mix (1.5-3 l/min). Post-operative analgesia was provided by 5mg/kg Carprieve (carprofen – subcutaneous injection) and for a further three days with 1mg/kg/day Metacam (meloxicam – edible jelly mixture). Additionally, animals received a microbial prophylaxis in the form of Batyril for seven days after surgery (4.0 ml enrofloxacin/100 ml water). During surgery Vaseline or Viscotears was used to cover the rats’ eyes to prevent corneal damage and the animals’ rested on a heating pad maintained at 37 degrees in order to maintain body temperature. The animal’s head was fixed in a stereotaxic frame with lambda and bregma in the horizontal plane. The skull was revealed and one or two 1mm trephine holes made to locate the transverse sinus. Electrodes were implanted above the dorsolateral medial entorhinal cortex (MEC), 4.5 mm lateral to the midline, 0.2 – 0.5 mm anterior to the sinus, angled forwards in the sagittal plane at 8 - 10°, and to a depth of 1.0 – 2.0 mm. Microdrives were secured to the skull using five-seven stainless steel bone screws and dental acrylic mixed 3:1 with Aureomycin (chlortetracycline hydrochloride) powder. One of the anterior screws was used as a ground electrode.
Subjects and feeding
The subjects consisted of male Lister Hooded rats, weighing 290-350g at the time of surgery. A seven-day free-feeding recovery period was provided post-surgery, after which the animals’ food was restricted so as to reach and maintain 90% of the free-feeding body weight. During the experiments the animals were required to forage for sweetened cooked rice. All work was conducted according to institutional (University College London) and national ethical guidelines in accordance with the UK Animals (Scientific Procedures) Act of 1986.

Single unit recording
Using hardware and software from Axona Ltd. (St. Albans, UK) extra-cellular voltage traces were amplified, referenced, digitized at 48kHz, high-pass filtered, and finally thesholded online. Each thresholding event triggered the recording of 1 millisecond of data from all four channels of the given tetrode. Throughout the trial, the voltage from one or more electrodes was low pass filtered, and stored at the reduced rate of 250Hz, and in some cases additionally at 4.8kHz. Single units were identified in a two-step process, the first stage of which was to use a mixture of Gaussians model to automatically cluster the first three principle components of the waveforms; the second stage was to manually review the output of this algorithm, which entailed merging similar clusters, occasionally splitting erroneously joined clusters, and reordering clusters to maintain cell identity across trials. The automatic clustering was performed by KlustaKwik (Kadir, Goodman, and Harris 2013), and the manual adjustments were made using an application custom-written by me (see section here on “Waveform”) and on occasion using the TINT software provided by Axona Ltd.

Animals were tracked using the standard small/large head-mounted infrared LEDs in conjunction with a ceiling camera. (In some cases only one LED was used.) The LEDs’ location and pixel extent were obtained online using a simple thresholding algorithm operating at 50Hz, and then filtered and smoothed offline. As is standard practice, filtering consisted of two steps: (a) sections of path with spuriously high speed (> 4ms/s) are discarded and a linear interpolation is use to fill the gap; (b) the labels “small” and “large” for the two groups of pixels are switched whenever the z-scores for pixel counts and the Euclidean distance between consecutive tracking samples both exceed certain thresholds. (Note that this has little bearing on position tracking only on directional tracking.) The smoothing step consisted of a 400ms moving average (“boxcar”) operation.

Screening/baseline arena
A square arena (side 1m, 50cm high black walls, black floor), was used as the screening environment, meaning that rats performed a free foraging task in this arena for 20 minutes or
more each day until the experimenter was confident that the electrodes had reached a region with grid cells. The furniture of the room (shelves, computer, recording equipment, table etc.) were visible from the arena, although only dimly illuminated by the light of a single desk lamp that faced away from the arena towards a wall of the room.
Basic analyses

The main analysis was performed using Python 2.7, NumPy 1.10.4, SciPy 0.16.0, Matplotlib 1.10.4, and Numba 0.23.1 (Python Software Foundation 2010; Jones et al. 2001).

Ratemap, autocorrelogram

Spatial ratemaps were obtained in the standard manner, by binning spike count and dwell time into a grid with spacing of 2cm in each axis, smoothing with a 5x5 box-car kernel, and then dividing counts by dwell to obtain rate in each spatial bin. Spatial autocorrelograms were obtained by computing the Pearson correlations of the ratemap with each of the many shifted versions of itself. As stated by Sargolini et al. (2006), the value of the autocorrelogram at offset \((\tau_x, \tau_y)\) can be expressed as follows:

\[
\begin{align*}
    r(\tau_x, \tau_y) &= \frac{n \sum \lambda(x, y)\lambda(x - \tau_x, y - \tau_y) - \sum \lambda(x, y) \sum \lambda(x - \tau_x, y - \tau_y)}{\sqrt{n \sum \lambda(x, y)^2 - (\sum \lambda(x, y))^2} \sqrt{n \sum \lambda(x - \tau_x, y - \tau_y)^2 - (\sum \lambda(x - \tau_x, y - \tau_y))^2}} \\
\end{align*}
\]

where \(\lambda(x, y)\) is the average rate of a cell at location \((x, y)\) and the summations are over all \(n\) pixels where it was possible to estimate a value for both \(\lambda(x, y)\) and \(\lambda(x - \tau_x, y - \tau_y)\).

Here the computation was done using multiplication in frequency space rather than convolution, which is the slower, but perhaps more commonly used algorithm in this particular analysis.

Gridness

The gridness score was computed according to the previously existing definition (Sargolini et al. 2006). Briefly: identify the six maxima closest to the central maxima of the autocorrelogram; find the extent of the peaks around these maxima by thresholding at half peak height; draw a circle that encompasses all six peaks; ignore values outside this circle and values within the single central peak; for rotations of the autocorrelogram at 30, 60, 90, 120, and 150 degrees, compute the correlation with the unrotated version; finally take the maximum value of the correlations for 30, 90, 150 and subtract it from the minimum value for the 60 and 120 correlations. This gives an estimate of the amount of local 60-degree rotational symmetry in the ratemap. One small modification to the standard algorithm was to use a watershedding\(^8\) step as part of the peak-extent calculation, rather than a simple threshold and

\[^8\] Watershedding is an algorithm that simulates, for each point in a height map, where a drop of water would eventually flow to, thus producing a finite set of labelled “drainage basins” (mathematically this corresponds
identify the contiguous region around each maximum. I do not characterise the effect of this change in this work, except to say that it will only affect the gridness scores for a small subset of the more poorly-defined grid cells, which are well known to produce unstable gridness scores using the current working definition of gridness ("unstable" with respect to binning and smoothing).

**Jitter/rotation**

The gridness and scale measurements involve one or more peak-detection steps, which can be brittle when the peaks are not entirely clear. In an attempt to reduce the error due to this brittleness, the raw position data was rotated around the centre by 30 and 60 degrees and shifted slightly before binning, so as to produce three separate estimates of gridness and scale rather than one. The median of the three values was taken as the "correct" value.

**Cell inclusion criteria**

All putative grid cells were analysed on all (i.e. either 1 or 2) available baseline trials each day. Only cells with gridness score > 0.2 were used in the main analysis. For all rats, this value corresponded to approximately the 97th percentile of a baseline distribution, obtained by shifting the spike times relative to the position data and re-running the gridness algorithm.

**Behavioural metrics**

"area covered" is the total area of a grid of 2.5x2.5cm bins, where the animal spent at least 0.25s in each bin. Note that the total area of the arena was 1.13m². "dir variance" is a novel metric, calculated as the mean over all bins, taking the circular variance of the trajectory of directions in each bin (thus high values mean high variance and low stereotyping). Other metrics are explained in the results text.

**Adaptively smoothed ratemaps**

In (W E Skaggs et al. 1996), the concept of adaptive smoothing was introduced to the literature. As with a standard ratemap, the end result is a grid of points with rate values; here, however, the rates are not computed by binning the data into a square grid and smoothing, instead for each point, the smallest radius, \( r \), is found such that the following equality holds:

\[
N_{\text{spikes}} > \frac{\alpha}{r^2 T^2}
\]

where \( N_{\text{spikes}} \) is the number of spikes within a circle of radius \( r \), and \( T^2 \) is the dwell time within that circle. The value of \( \alpha \) used by Skaggs et al. (1996) was 2,500 (when expressed in the to a form of iterative gradient descent). Note that in this case we actually invert the height map before performing the watershedding as we wish to segment the maxima not the minima.
form shown here) and I use that same value. Where the inequality can be satisfied, the rate is recorded as \( N_{\text{spikes}}/T \), otherwise (i.e. if the inequality can not be satisfied for large \( r = 15\text{cm} \)) an “invalid” value is recorded.

**Stability**

For intra-trial stability, data for each trial was split into two halves: the first 10mins and second 10mins. The two halves had separate ratemaps created using adaptive smoothing, with a 1.4cm-spaced 2D array of points (i.e. bins). A Pearson correlation was then computed for the common valid bins.

When testing the stability of the interior half of the arena, the adaptively smoothed ratemaps were masked to a radius of \( 60/\sqrt{2}=42.4\text{cm} \).

The test of remapping was conducted on pairs of standard ratemaps (i.e. not adaptively smoothed maps). Again, here a Pearson correlation was computed on the common valid bins.

**Grid fields**

Some of the more complex analyses require identification of discrete grid fields. Designing algorithms that perform satisfactorily for a specific use cases is something of an art and here we have used more than one such algorithm.

Method (a) is based on that of Hardcastle et al. (2015) and requires the creation of an adaptively smoothed ratemap using a 2.5cm-spaced 2D array of points (i.e. bins). The result is thresholded at the 75\(^{th}\) percentile of valid bins, and topological holes are filled\(^9\). Finally, the fields are filtered, to remove fields with an area less than 40cm\(^2\) or less than 60% of the area of the second largest field.

Method (b) uses the same topological filling step and final filtering procedure, but constructs the fields using a normal ratemap, binned using 2.5cm bins and smoothed with a Gaussian filter, with sigma 2 bins. Here the threshold is set at the 50\(^{th}\) (rather than 75\(^{th}\)) percentile of the valid bins.

\(^9\) “Topological holes” in a binary map are contiguous regions of zeros that are completely surrounded by ones; filling these regions simply means switching the zeros to be ones.
Additional analysis methods

Estimate of composite grid scale

Evidence suggests that grids exist in attractor-like modules (H. Stensola et al. 2012; Yoon et al. 2013), in which all cells share the same orientation and scale. Thus it is possible to pool the data from multiple recorded cells so as to improve the accuracy of parameter estimates, with scale (rather than orientation) being of particular interest here.

The standard method for determining scale from a single cell is to compute the spatial autocorrelogram from the ratemap and then measure the distance from the central peak to the inner six peaks, ultimately taking the mean or median of the obtained (Barry et al. 2012). As stated in an earlier section, the quantization and anisotropy effects of the binning/peak-detection process can be partly mitigated against by performing the operation multiple times, using jittered and/or rotated versions of the original position data. A large variance in the set of scale values produced indicates that the process is brittle and/or detecting noise.

When estimating scale across multiple cells within a trial, one option is to just use the mean (Barry et al. 2012). Although simple, this method produces values that fluctuate significantly between trials as the proportion of cells from different modules changes. In principle one could assign cells to specific modules according to their scale value, however performing this pre-processing step before computing the mean would result in a rather circular/biased estimate of the mean due to the large variance on the estimate of scale for individual cells (as compared to the scale difference between modules (H. Stensola et al. 2012)). A possible alternative is to entirely avoid quoting a single absolute value for scale, instead computing trial-to-trial differences for individual cells and then aggregating on that data, expressed either in absolute centimetres or as a percentage (Barry et al. 2012).

However, a better alternative is to consider the observations of scale as measurements of an exact value plus the observation of some Gaussian error term, of known standard deviation, $\zeta$. Then the 95% confidence interval for the scale is approximately given by the mean $\pm 2\zeta/\sqrt{n}$. We can then ask which is more accurate: (a) estimating scale separately for multiple cells, followed by taking the mean; or (b) combining the raw data from the cells to be used in a single, but more accurate, estimation of scale. If $n=4$, which is a typical example of the number of co-recorded grid cells in an experiment, then we require that the composite method (which uses $n=1$) has measurement errors that are less than half the single-cell measurement error. For example, if the scale measurement for a single cell has an error of 6cm, then we are...
better using the compositied method if it can offer an error below 3cm. If the composite method also addresses the question of module membership then it is more powerful still.

The approach I take here is to compute the standard spatial autocorrelogram for all cells as normal, and then compute all pairwise correlations between autocorrelograms for the cells on a given trial. However, at the edges of the autocorrelograms the values correspond to correlations with small n, and are thus rather noisy and would be best ignored in the pairwise correlation. So here, rather than use the full autocorrelogram or even set an arbitrary cut-off point, I opt to weight the bins according to their n value. An example covariance matrix is shown in Figure 11 with some weighted autocorrelograms.

---

10 This approach could be applied to gridness as well.
Figure 11. Estimating composite grid scale. Above: example covariance matrix for six cells on one trial. Along the top and left edges are the six ratemaps, and down the far left is shown the six individual spatial autocorrelograms, with transparency indicating the relative weight, n, of each bin. At the top is shown the hierarchy resulting from agglomerative clustering, with the green section marking the cells in the largest group having all pairwise correlations above threshold. The composite autocorrelogram for the group is also shown above. Below: example results from this analysis when applied to data from a single animal (the animal visited a pair of environments multiple times over several days).
Once we have obtained the covariance matrix we can perform some form of clustering to identify one or more modules of grid cells. The approach followed here is to use hierarchical agglomerative clustering with “complete” linkage, a process that identifies sets of cells in which all pairwise correlations are above a particular threshold. In particular, if we choose a specific threshold (e.g. 0.4) we can identify the largest group of cells that form such a set and then define those cells as belonging to the same module. In principle there may be sets corresponding to multiple modules, but in practice one module always dominates in terms of cell numbers and we would gain little by attempting to analyse the other minority cells. Once the module has been identified, we can add together the individual autocorrelograms (in a similar manner to Fyhn et al. (2007)) from all the cells and then use this composite autocorrelogram to measure the scale, employing the standard peak-detection method.

If we require consistent module membership across trials, we can extend this method as follows: compute the covariance matrix for each trial, and then combine the two (or more) matrices using the minimum correlation for each element in the matrix. This ensures that when we perform the clustering and set selection we can still claim that all pairwise correlations within our newly defined “module” are above the required threshold (within each of the relevant trials). This combined covariance matrix is not illustrated in Figure 11, however some sample data is shown that utilizes this method in order to estimate scale in a pair of environments (note that in this analysis, module membership is only consistent for single pairs of trials, not necessarily for whole days or across days).
Delaunay method for estimating scale

As mentioned in the previous section, measuring scale from the spatial autocorrelogram (SAC) can be brittle and noisy. I have already proposed one solution - the composite method – but here I describe another. In this method, I measure peak-to-peak distance on the ratemap directly, rather than building an autocorrelogram and measuring centre-to-peak on that. The SAC method works well for highly regular grids, because all pairs of adjacent ratemap peaks contribute to the same six inner SAC peaks, however, for less regular grids the SAC will not have such clear peaks and thus it will be more difficult to identify scale from it.

Figure 12 outlines the stages of computing scale using this method. Firstly we identify the grid fields using Method (a) described in the previous section. For each field, we then find the location of the peak rate (using the ratemap computed during the field identification process) and feed these positions into a Delaunay triangulation function which establishes the adjacency of peaks.

![Figure 12. The Delaunay method for estimating scale. From left: spikes on path; adaptively smoothed ratemap; fields identified from the thresholding and filtering process; peaks and triangulation superimposed on a standard (i.e. non-adaptively smoothed ratemap). In the final plot, peaks are shown as black dots, with the basic Delaunay triangulation as solid black lines; the triangles which passed through the filtering process are highlighted in white.](image)

Delaunay triangulation has previously been used in this context (Barry et al. 2012; Climer, Newman, and Hasselmo 2013) though the steps employed here vary somewhat: we compute the angles in each triangle, and discard any triangles with angles outside the range 45-75 degrees; next we discard triangles with area less than 60% of the second largest triangle. Finally, for each triangle we take the mean of its side lengths, and then the median length over all triangles. By filtering (twice) relative to the size of the second largest field/triangle we remove a potentially long “tail” of triangles which would spuriously reduce the final median. And by using the second largest field/triangle, and by taking the median rather than the mean, we avoid spuriously increasing the final value due to omitted or fused fields.
Maximum likelihood estimate of drift

If spike data from a grid cell is aggregated into chunks of several minutes, then over the course of a 20 minute trial, the pattern typically does not appear to change much. However, if one considers the data for individual short runs through a field, it is possible that the pattern is jittering around its long-term stably anchored position. Here I describe a method for quantifying the “drift” for single runs: for each run we output a distance (i.e. in centimetres) as well as a likelihood of the drift.

We assume a Poisson spiking model for the grid cell, with the rate parameter given by the mean rate (over the whole trial) for the relevant spatial bin, i.e. we read it from the ratemap. However, because we need to be able to evaluate the rate in bins that the animal never visited, we must first extend the ratemap to any unpopulated bins using nearest-neighbour interpolation – see Figure 13 for an example extended ratemap, note this applies to bins outside the environment as well as to unvisited areas within the environment.

![Figure 13. Maximum likelihood estimate. Main panel: ratemap, with nearest-neighbour interpolation, and an example run through a field shown in white, spikes also superimposed in white. The red path and spikes are the same data but shifted down and to the right by the amount indicated by the magenta arrows. Small panel top left: likelihood of observing the spike data for the specific run depicted, evaluated at a grid of x and y offsets from the correct location. The white square at the centre corresponds to zero shift, and the red square in the lower right quadrant corresponds to the peak identified.](image)

The likelihood of observing the spiking data for the whole of a specific run, $L_R$, is equal to the likelihood of observing the data for all the $n_R$ short time windows within the run:
\[ L_R = \prod_{w=1}^{n_R} L_{R,w} \]

For simplicity, we allow the position sampling frequency to dictate the width of the short time windows: thus at 50Hz the width is 20ms. For each of these short windows, \( w \), we look up the Poisson rate parameter, \( \lambda_{R,w} \), from the relevant bin in the ratemap. We also count the number of spikes within the window, \( k_{R,w} \) and then we can evaluate the Poisson probability:

\[
Poisson_{\lambda}(k) = \frac{\lambda^k e^{-\lambda}}{k!}
\]

In practice, we prefer to evaluate the log likelihood: this allows us to replace the product of near-zero values with a sum of larger-magnitude values (or in fact as three separate summation terms, one for each of the factors in the Poisson equation).

As indicated in Figure 13, we evaluate the likelihood of each run on a 17x17 array of \((x, y)\) offsets centred on an offset of \((0, 0)\) and using the same bin size as the ratemap. The final step in decoding the grid signal is to identify some kind of maximum from this 2D array. However, rather than simply use the global maximum, we adopt a slightly more nuanced approach, taking the local maximum which is closest to the centre, and which is also an “inward-looking” maximum, meaning no other point closer to the centre has a higher likelihood. This choice of double criteria ensures that small maxima close to the zero-offset point can be overridden by larger maxima further out, so long as the slope leading up to the larger maxima begins close to the centre. Once the maximum has been identified we can measure the Euclidean distance to it from the centre of the grid, and provide this as the “drift” distance for the run. The likelihood at the maximum is also of interest, but it must first be normalized: here we require that the total likelihood on the offset grid sum to one, so it is this sum which provides the normalizing factor. Note that the process described here is different from that normally used with place cell decoding because we are confronted here with a periodic pattern which cannot be decoded to a single global location (this would not be the case if we had data from many simultaneously recorded grid modules).

In this analysis, grid fields are identified using Method B (see previous section) and runs are defined as contiguous segments of path through a field that have a duration greater than 0.25s. The ratemap defining \( \lambda \) is constructed with 2cm bins and Gaussian smoothing, using a sigma of 3 bins.
Where noted, this analysis is not performed on all runs, instead some runs are removed so as to better match the speed distributions of a pair of trials. Specially, the mean speed for each run is computed, and the runs are then grouped into bins of 4cm/s. For each such bin, the trial with the greater number of runs has a random sample of its runs removed, so as to force the counts to match across the two trials. Note that mean speed is only a one dimensional quantity which does not capture the full range of behavioural differences between runs and trials. Also note that differences in speed may contribute to variation in the properties of the ratemaps, and that I make no attempt to account for this here.
Estimate of consistent within-trial drift

The previous method was concerned with identifying weaknesses in the anchoring of the grid pattern on the scale of 1-2 seconds; I assumed that the overall pattern was stable throughout the trial. In this section I wish to examine whether regions of the pattern undergo a consistent drift during the course of a 20 minute recording. The approach I take is to split the trial into two parts, as done for within-trial stability analysis. As depicted in the cartoon of Figure 14, at each location in the arena, we correlate a patch of ratemap from the second 10 minutes of the trial, with patches of ratemap from the first 10 minutes of the trial, evaluating this correlation for a grid of 13x13 different patches that are offset slightly from the location in question. We use a Gaussian weight for the bins within each “patch”, where bins are 4x4 cm and the Gaussian kernel has sigma = 8 bins.

Figure 14. The warp metric. From left: (1) standard ratemap. (2) red and green channels show separate ratemaps from the first 10 mins and second 10 mins of the trial, respectively. (3) “warp” map with key, showing which direction to warp each bin in the second 10 minutes of the trial so as to bring it into closer agreement with the first half of the trial. (4) cartoon depicting how the data surrounding one location is weighted and correlated with data at a variety of offsets, with the data from the central location from the second 10 minutes of the trial, and the offset data from the first 10 minutes of the trial. Data in (1)-(3) from R2174 t11c2, 2014-10-23 trial 1.

Although this requires over 100k weighted correlations for a 1m x 1m ratemap, it is possible to evaluate this efficiently using just six Gaussian convolutions per offset, i.e. about 1000 convolutions total. We can see this simply by noting that all the terms in the correlation consist of weighted sums, and that the Gaussian convolution performs exactly the required weighted sum operation: i.e. for each bin it provides a weighted sum of the values in the surrounding bins. The six convolutions correspond to the six terms $S_{ab}, S_{a}, S_{aa}, S_{bb}, S_{ab}$ and $n$, and are used as follows:

$$
\rho = \frac{S_{ab}n - S_{a}S_{b}}{\sqrt{S_{aa} - S_{a}^2} \sqrt{S_{bb} - S_{b}^2}}
$$
Once we have evaluated this for each bin at each of the points on the 13x13 grid of offsets, we can identify (again, for each bin), the maximum correlation and the offset to which it corresponds. Thus, we arrive at a “warp” map, which gives the optimal offset (specific to each point in the arena), by which to shift the data so as to align the pattern from the later part of the trial with the pattern from the earlier part of the trial. See Figure 14 for an example with real data.
3. Waveform: a GUI for data exploration

Abstract
I have created a graphical user interface (GUI) for exploring and working with the data generated during single unit recording spatial experiments. The application is primarily implemented in HTML5/JavaScript, but with some rendering using the GL Shader Language. The application is open source, with both the source and live application hosted on github, at github.com/d1manson/waveform and d1manson.github.io/waveform respectively. An eight minute video demo is available on YouTube and is linked to from both these urls (the reader is encouraged to view this video in conjunction with reading this chapter). The design and features have evolved over a number of months, with the current version now serving the needs of a handful of researchers at UCL. The application builds on older software used for similar purposes, but provides additional features, and aims to serve as a central tool in a more efficient experimental pipeline for working with ever larger datasets.

The data
When conducting a single-unit recording experiment, the voltage signals from each electrode are typically passed through an online processing pipeline that firstly bandpass filters the signal and then performs some type of putative action potential detection. A one millisecond section of the filtered voltage signal is saved to disk for each detection event, together with a timestamp identifying when the event occurred. In the case of tetrode recording, the system is usually configured to save the voltage trace from all four electrodes when any one of them passes the detection threshold.

Thus the data from a tetrode recording experiment primarily consists of a long list of waveforms. For example, the files produced by Axona systems are of the following form:

<table>
<thead>
<tr>
<th>timestamp for event 1</th>
<th>50 x voltage readings for event 1 on channel 1</th>
<th>... channels 2 and 3 ...</th>
<th>timestamp for event 1</th>
<th>50 x voltage readings for event 1 on channel 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>timestamp for event 2</td>
<td>50 x voltage readings for event 2 on channel 1</td>
<td>... channels 2 and 3 ...</td>
<td>timestamp for event 2</td>
<td>50 x voltage readings for event 2 on channel 4</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>timestamp for event n</td>
<td>50 x voltage readings for event n on channel 1</td>
<td>... channels 2 and 3 ...</td>
<td>timestamp for event n</td>
<td>50 x voltage readings for event n on channel 4</td>
</tr>
</tbody>
</table>
where each of the timestamps is a 32-bit integer, each of the individual voltage readings is an 8-bit integer, and the table is serialized in row-major order.

In addition to this voltage data, the experiment will usually record a location trace (and/or other variables of interest). For example, positional data can be sampled at 50Hz using an overhead camera, with the location and size of one or more LED spots recorded in a fixed-width data-structure.

A typical 20-minute recording session may involve 16 tetrodes each of which records 10,000-80,000 putative action potentials, i.e. a session generates 16 files each of size 2-10MB. As discussed below, the experimenter must somehow explore all this data, clustering the putative action potentials into groups corresponding to individual neurons, and identifying cells of interest. The “interest” in this case referring to the presence of some form of structure in the firing pattern – significant patterns are usually (putatively) identified by manually examining plots of the data. Such plots are created using a combination of the spike time data and the location/other trace data.

**User requirements**

Generally, when working with the data the user will have one or more explicit objectives, however they may also have a range of less explicit aims, or be acting within a set of rather loosely defined constraints.

The most significant of the vaguely defined constraints are time and experimenter effort. If the time spent exploring the data is kept to an absolute minimum, the experimenter can focus on the physical data collection, analysis, and other more intellectual aspects of the research process. It also provides the experimenter with a means and motivation (or disinhibition) to collect larger volumes of data. Similarly, if the mental effort involved in data exploration can be minimized then the experimenter can allocate their efforts elsewhere or perform a more thorough exploration of the data and/or explore a larger volume of data.

**Design**

The waveform application was designed and built to serve not only the explicit needs of single-unit hippocampologists, but also to satisfy their more subjective preferences.

Its design is focused around the concept of a “tile”, in which the user is presented with all the relevant plots and information for a given cluster of spikes – see Figure 15.
By showing all these plots side by side, the user can, at a glance, distinguish between a cluster of noise and a true neuron. They can also assign a vague “class” label to the cell using whatever heuristic intuition they may have developed from experience. For example:

*Is this a directional grid cell, or a theta-modulated interneuron, or a non-spatial principle cell?*

In fact, not only does it make it easier to draw on previous experience, but it hopefully improves the rate at which individuals can develop, and/or be taught, the necessary recognition skills.

Showing one tile at a time is useful, but the application does better, showing the user a “wall” of tiles, one for each cluster on the tetrode. On modern monitors there is usually plenty of screen real estate available for displaying plots (and the application can be dynamically resized to fill it), however if the tiles do not all fit on screen simultaneously, the user can simply scroll through the list. This everything-visible-simultaneously design allows us to extend the above heuristic/intuition arguments to the level of a whole tetrode: the user can now, at a glance, ask questions about the region in which the given tetrode is currently located. For example:

*Is the tetrode close to the entorhinal cortex, or some way above it? Is it in the deep layers or the shallow layers? Are all the grid cells of similar orientation and scale? Are the spatial patterns equally neat across cells?*

Obviously, they can also find the cells of interest more quickly and with minimal effort. One additional benefit of showing all this information to the user in one (scrollable) page is that the user has an opportunity to spot interesting patterns that they may not have been explicitly searching for.
In some cases the user may want slightly more information about a specific cell than can be provided within the context of the “tile”, in such situations, a simple mouse-over action brings up additional data in a side-panel of the application window.

Similarly, some support for comparing pairs of groups is built in: dragging one tile over a second tile, displays a looping animation that oscillates between the two sets of plots, allowing the user to quickly identify differences or similarities.

In addition to the “wall of tiles”, the spike groupings are shown in a more traditional set of cluster plots, which take the amplitude (maximum-minimum) for each wave on each channel and renders a two-dimensional plot for each pair of channels. Traditionally such plots have labelled each point with a colour identifying which group it has been assigned to; this convention has been maintained here, but we add an additional rendering mode in which each pixel in the plot shows the mean time of spikes for that location in amplitude space – see Figure 16. This mean calculation includes a smoothing step that hides the exact variation in timing from one amplitude bin to the next, but makes it clear to the user where a cluster has drifted in amplitude space during the experiment, or perhaps a cluster that appeared / disappeared part way through the trial.

Figure 16. The mean time of spikes at each point in spike-amplitude space, referred to as “drift plots”. As is commonly done, the six subplots show all six possible pairings of the four sets of amplitude readings. Red pixels correspond to early mean time and green pixels to late mean time, thus a gradient from red to green indicates a drifting cluster.

By providing these plots to the experimenter, before they have even clustered the data, it reduces the need to observe the oscilloscope throughout the experiment (so as to avoid spuriously splitting a cluster into two groups, or computing invalid plots for cells only available for specific intervals within the trial). It can also alert the experimenter to issues with the tetrode/microdrive stability that they may not have otherwise been aware of. This is of
particular relevance to novices, who most benefit from additional feedback on the quality of their handiwork.

Although the application does provide a few methods for modifying cluster assignments through merging, splitting, “painting”, and swapping operations, the interface has evolved to be used in combination with an automated cluster assignment step (such as provided by KlustaKwik (Kadir, Goodman, and Harris 2013)). The combination of automation and manual review/editing has proven to be popular with a number of experimenters in the lab.

The final design consideration to mention is the support for switching between files: if multiple cut files are available for a specific tetrode, the user can select between them with a single click, they can also switch from one tetrode to any other with a single click, and even from one trial to another using a single click. All these files are automatically arranged neatly in an interactive tabular form when the user drags them into the application – see Figure 17.

![Figure 17. The “file panel”, here showing 3+ trials, with tetrode seven currently selected for the second trial.](image)

The file loading process is heavily optimised so that a tetrode file with 60,000 events can be read off disk and have all its plots rendered in under 400ms on a typical experimental PC. This disinhibits the experimenter from switching files, allowing them to ask questions that may previously have had low priority:

*Did that cell have the same pattern on the first trial of the day? Did I lose a cell on this tetrode between these two experiments? Have I been lucky and found some cells on the tetrode I thought wasn’t going to produce anything? Did I really see a grid cell in the hippocampus yesterday?*
The user is also provided with a simple action with which to “grab” a copy of a tile (click with
the mouse while the space key is down). This grabbed copy remains visible when switching
between trials/tetrodes/cuts so the user can quickly perform any necessary visual comparisons
without recoursing to additional screenshot tools, opening multiple instances of the
application, or taxing their own memory.

Implementation
Since the application’s main aim is to be an interactive graphical interface, it was decided that
the web browser would be a powerful platform upon which to build. This decision was
initially only intended to apply at the prototype stage, however the application has proven
sufficiently performant that it has not been migrated to another environment. The browser
offers a number of benefits, including a high performance scripting language, a highly
optimised and flexible rendering engine and a powerful debugger and profiler. No single
other platform currently offers all these tools with such a wide range of features and high
quality. For an application such as this, the main drawback of JavaScript is its lack of support
for the tools normally employed in numerical computing: SIMD (single instruction multiple
data parallelism), shared memory across multiple threads (including atomic operations and a
memory ordering model), and a way to guarantee optimal machine code for the most carefully
designed algorithms. Such limitations are currently being addressed by standards committees,
so it is likely that the situation will soon change and it may become more common to build
scientific applications in the browser, but for now we have to work around these issues.

A detailed discussion of the implementation is beyond the scope of this work, however I shall
briefly highlight a couple of the tools employed as they go some way to circumventing the
apparent limitations mentioned above:

- The heaviest disk access and plot generation tasks are carried out off the main thread
  using WebWorkers, which communicate with the main thread via a combination of
  message passing and transfer of memory ownership (but not true shared memory).
- The most computationally intensive task, rendering the waveform data, is carried out
  by the GPU using a custom-written kernel that shows the “density” of voltage traces in
each pixel of the plot. This kernel is designed such that updates to the cluster
assignments can be reflected in the waveform plots with very low latency.

More information can be found within the readme file provided with the source.
Further work

The application was initially developed in a series of small increments, spread over the course of many months. During that period much was learnt about the possibilities for improving user experience – some of which has already been discussed above. As with any software project, there remains a long list of features awaiting implementation – this can be viewed within the github web interface. Also provided on the github page is an extensive design document with diagram and links to resources for anyone looking to further develop the application.
4. Grid Cells: theoretical considerations

Although the first two parts of this chapter consist largely of literature review - as indicated by the concertation of references - they also contain some additional insights of my own. The third part continues on a similar theme, but presents new (now published) contributions made by my colleagues and I.

Characterising grid cells

A logical introduction
Natural selection would lead us to believe that any phenomenon that can be robustly observed in healthy subjects across a number of species is (almost certainly) performing a useful function for the organisms in question. Grid cells (however one defines them) are clearly such a phenomenon (Fyhn et al. 2004; Fyhn et al. 2008; Doeller, Barry, and Burgess 2010; Yartsev, Witter, and Ulanovsky 2011; Jacobs et al. 2013), so presumably they serve a function of some sort. Or, perhaps they serve a family of (conceptually distinct) functions. Or perhaps they act only as an optional aid to other regions of the brain, offering one or more of the following:

- increased precision
- reduced errors
- acceleration of slow algorithms or…
- parallelisation of slow algorithms or…
- completely distinct algorithms, optimized for speed/energy/accuracy
- better reutilization of previously acquired/computed knowledge
- an opportunity to offload specific compute-tasks, freeing up more general-purpose compute resources to work on other tasks

For readers with some experience in computer science, these concepts will no doubt appear familiar: their silicon/software counterparts are referred to by such names as “coprocessors”, “GPGPU computing”, “FPU”, “cache hierarchy”, “memorization”, “precomputed lookup table”, “big-O complexity”, “checksum”, and maybe even “branch predictor” and “profile guided optimization”. While I do not intend to draw a direct analogy between any specific concepts, I note that the brain is fundamentally a computational device and thus it must
ultimately jump through the same logical hoops that a silicon computer would, when confronted by an equivalent situation. This sort of parallel was recognised early on by the pioneers in computing (Von Neumann 1958), and continues to inspire neuroscientists to this day (for example Erdem and Hasselmo 2014).

In reality, however, there is only a weak distinction between being an “optional aid” and serving a distinct purpose: if an operation can be made ten times faster or much more accurate or can be achieved with far less energy, then the new version of the operation will likely find more uses than the original. For example:

*The automobile encouraged regular long-distance travel; the computer facilitated the development of instruction manuals (i.e. programs) taking billions of steps to execute; the shipping container allowed workers in one country to make cheap goods for consumers thousands of miles away.*

In this light, it is not immediately clear what questions should be asked of grid cells: simply “turning them off” and performing behavioural tests in different scenarios will provide some empirical data points (Miller and Best 1980; Remondes and Schuman 2004; Parron and Save 2004; Parron, Poucet, and Save 2004; Steffenach et al. 2005; Parron, Poucet, and Save 2006; Winter et al. 2013), but is unlikely to lead us to a concise and conclusive description of their functional role. Furthermore, any artificial manipulation of a highly-interconnected network is liable to cause unusual reverberations on short timescales and may cause complex changes over longer timescales. Thus, even if there were a “pure” functional purpose waiting to be discovered, it would not be possible to observe the “pure” impact on behaviour of removing the grid cells. And in a similar vein it is difficult to imagine isolating the specific mechanistic roles played by the various components in the network.

Despite such caveats and warnings, I shall nevertheless make an attempt to guide the reader through a variety of hypotheses, and inspect some of the empirical evidence that has been collected to date in favour and against them.

**Grid cell modules and the attractor**

As shown in Figure 18, the canonical grid cell displays a pattern of firing peaks that is parameterized by four values:

- scale - the distance between adjacent peaks
- orientation – the angle between a chosen real-world axis and an axis of the pattern.
- x and y offsets – the two-dimensional vector from some chosen real-world location to the nearest peak in the pattern (possibly expressed in fractions of the scale).

Figure 18. Cartoon depicting the canonical grid pattern. Note how the pattern is parameterized by a peak-to-peak distance (red lines, centre panel) and by an angle of orientation (blue construction markings, centre panel). Individual cells are offset relative to one another, and relative to a chosen point in the real-world (magenta, right panel).

Cells close together in the entorhinal cortex tend to have very similar scale and orientation, i.e. differing only in their offsets (Hafting et al. 2005; Barry et al. 2007; H. Stensola et al. 2012). At the dorsal end of the entorhinal cortex, cells typically have a scale of 30cm; moving ventrally, one finds cells with increasingly large scales: 40cm, 60cm, …2m, and beyond (Brun, Solstad, et al. 2008), see Figure 19.

Figure 19. Cartoon of grid patterns from three different modules.

It thus becomes natural to consider a set of grids cells as forming a “unit tile”, which tessellates the whole environment. One can use any single shape that tessellates in this way, i.e. without gaps, overlaps, or rotations (Kubie and Fenton 2012). Note that triangles are not valid as they require a rotation (or the use of “two” triangles, which are then better described as a single quadrilateral).

The rectangle shown in Figure 20 is one simple option, but note that the ratio of its sides is fixed at 1:1/sqrt(3). This is required in order to maintain equal distance between any point on the tile and the six adjacent repeated copies of that point. Although the choice of a rectangle is sometimes easier to reason with it is perhaps more natural to use a rhombus (Bruce L McNaughton et al. 2006). Using the rhombus form we can derive an affine transformation to
take a point in Cartesian space and map it to a point on the unit grid tile (i.e. \([0,1] \times [0,1]\)). The inverse transform will map the point back into Cartesian space, but with ambiguity – we will no longer know which instance of the tile the point came from.

Figure 20. Cartoon showing two valid tile shapes that tessellate in a 2D environment in a manner compatible with the grid cell pattern. The two annotated and enlarged tiles in the centre show, for both cases, the location of three cells within the unit: the magenta cell highlights the different forms of periodicity in the two arrangements.

The transform conceptually consists of multiple steps (shown in Figure 21), although the steps can in practice be combined using matrix multiplication to give a single transform matrix, paired with a modulo-one operation.

\[
A = \begin{bmatrix}
\cos(\theta) & -\sin(\theta) \\
\sin(\theta) & \cos(\theta)
\end{bmatrix} \cdot \begin{bmatrix}
1 & 0 \\
-1/\sqrt{3} & 2/\sqrt{3}
\end{bmatrix} \cdot \frac{1}{\text{scale}}
\]

\[
f(x, y) = f(x) = (xA) \mod 1
\]

This matrix transformation does not seem to appear in the literature (or at least not often), perhaps because its utility is not entirely clear. I suggest that one potential use would be as part of a fast algorithm for fitting idealised grid patterns to real data\textsuperscript{11} either at the level of ratemaps (Carpenter et al. 2015) or autocorrelogram peaks (Yoon et al. 2013).

\textsuperscript{11} I shall not give a full description of potential algorithms here, but simply suggest the following partial recipe for interested readers (I do not have a fully working optimized implementation in python):

- The original (x, y) coordinates are transformed into unit grid space using the transformation described above.
- A ratemap is formed in this new coordinate space, rather than in standard coordinate space, it is then transformed into Fourier space.
Figure 21. Cartoon depicting the conceptual stages of transformation which convert real-world Cartesian coordinates into points in a canonical grid tile. See main text for details.

Although one could use a variety of two-dimensional shapes to represent the flat canonical tile, none of these shapes conveys the periodicity (i.e. the true distance between all pairs of points on the tile). Instead, we must use the two-dimensional surface of a three-dimensional torus. One slight confusion with this is that mapping the Cartesian axes onto the torus should not produce a pair of perpendicular lines: rather one of the lines must be at a slight angle that allows it to circle the torus twice before joining up with itself (confused readers are recommended to examine the picture showing the rectangular version of the unit tile, and imagine walking along one axis and then along the other). This topology is referred to in the literature as a “twisted torus” (Guanella, Kiper, and Verschure 2007; Bush et al. 2015).

Given that the cells within each grid module collectively appear to represent a single location on a torus, it has been suggested that perhaps the network of synaptic weights within the module actually reflects this topology: i.e. cells representing nearby points on the torus have strong excitatory connections, and those further away have weaker, or inhibitory connections (Fuhs and Touretzky 2006). Such a network is termed an “attractor” because population activity is always attracted towards the representation of a valid “bump” on the torus: other patterns of activity are unstable.

The exact neural implementation of this attractor may be complex, involving various forms of inhibition (Burak and Fiete 2009; Couey et al. 2013) as well as excitation. However, if such an attractor exists it leads us to expect that any set of cells from within a single module will always have the same relative spatial phases: e.g. if cell C fires midway between cells A and B

- A canonical unit grid ratemap is generated from cosines and also transformed into Fourier space (this need only be done once).
- The real and ideal data are multiplied together and then transformed back to real space, using the inverse Fourier transform. According to the convolution theorem, this is equivalent to having performed a periodic convolution in real space.
in one location it will always fire midway between those two cells. This should be true even if the firing locations of A and B appear to drift (translate/rotate/scale) relative to real-world coordinates.

Of course, the synaptic weights corresponding to the toroidal attractor (Fuhs and Touretzky 2006; Guanella, Kiper, and Verschure 2007; Couey et al. 2013) may not (always) dominate the grid module: external inputs to the network may have a stronger influence on activity, and there may be neuromodulators which can alter the gain on the attractor relative to those inputs (for an equivalent statement regarding place cells, see e.g. Wallenstein and Hasselmo (1997)), or even to other competing internal topologies. In principle there could even be two or more distinct attractors encoded in the same population: the gains on the separate attractors might be controlled by different neuromodulators; or use the same neuromodulator but in inhibitory and disinhibitory roles; or not require neuromodulation at all, instead forming a single “metatopology” in which the network is attracted to a point on any one of multiple tori, and preferentially shifts smoothly on the surface of that torus rather than jumping to a separate one (or perhaps neuromodulation or other external switches could be used to control when the network jumps between tori). In fact, this discussion may be more relevant at the level of the grid system as a whole: it has been suggested that the multiple grid module attractors might act as components of a single larger attractor with more complex topology (Sreenivasan and Fiete 2011), the existence of this larger attractor would ensure that no single grid module can be arbitrarily perturbed by noise.

It has also been suggested that the attractor might be a finite sheet (Fuhs and Touretzky 2006; Burak and Fiete 2009), where if one travels far enough from the centre an edge is encountered and the system must somehow arbitrarily reset itself to some other point on the surface (presumably another edge). A final proposed topology is a stack of sheets, each of which supports a single bump at any one time. These two alternatives have received less attention in the literature, perhaps because one might expect both to degenerate into the torus case if any hebbian learning is permitted to occur after the proposed topology in question has otherwise stabilised (Widloski and Fiete 2014).

**Information density**

In the previous section I introduced the concept of a grid cell module, a group of perhaps 1000 cells (in the rat at least, assuming 50,000 cells per entorhinal cortex, approximately ten modules, and around 20% of cells in the region actually being grid cells (Brun, Solstad, et al. 2008; H. Stensola et al. 2012)) which we might like to treat as a system for robustly representing two numbers: an x and a y phase within the unit tile for the given scale.
The rat entorhinal cortex is thought to contain perhaps as many as ten separate modules, of increasing scales. But, before I consider the reasons for having these multiple modules, the reader may like to refer back to the introduction section of this chapter, and ask why it is that the brain sees fit to use 1000 cells to represent just two numbers. One way to justify this quantity of cells would be if the precision of each of those two numbers was very high (i.e. each number was accurate to many “decimal places”). It is difficult to put a robust estimate on the actual resolution: in principle, even if cells are firing action potentials stochastically, with a sufficient number of samples (aggregated over time and/or across a pool of “identical” cells) one could make an arbitrarily precise estimate of the “hidden variable” of rate for each cell/cell pool, and then examine a vector of rates across the population to resolve position with arbitrary precision (Sreenivasan and Fiete 2011) – see Figure 22, although note that noise correlations can complicate matters (Mathis, Herz, and Stemmler 2013).

If we choose a value of 1cm for the smallest module, we can treat that module as having approximately $32=2^5$ possible values or 5 bits per dimension, and thus a total of 10 bits shared across the 1000 cells, or equivalently 0.01 bits per cell.\(^{12}\) Is this surprising? Perhaps this is just an example of standard redundancy; or maybe this pair of numbers is so important to the animal that it can afford to dedicate an exceptional amount of resources to storing it? Both seem unlikely: firstly, all the cells in a module are located close together (ignoring the contralateral hemisphere) so there can be only limited scenarios in which they provide useful redundancy (i.e. local violations of homeostasis will affect all cells); and secondly it is unlikely that investigators have happened upon one of the few locations in the brain that encode exceptionally significant information (i.e. presumably the information density across cortex is roughly uniform, so long as you know what to look for –a counter argument would be that researchers have only succeeded in identifying grid cells precisely because they have an exceptionally low information density).

Figure 22. Estimating a hidden variable from a Gaussian function. Sufficiently precise measurements of a Gaussian function can be used to estimate its hidden variable with a

\(^{12}\) Fiet et al. (2008) arrive at an even lower estimate, 0.001bits/cell.
comparable high level of accuracy (i.e. estimation of x in this case). Simultaneous measurements of multiple such Gaussians serves to resolve ambiguities arising from symmetry, offer additional precision, and enable the estimation to be made in a relative manner rather than relying on absolute magnitudes.

An alternative explanation, which appeals to me, is that by representing the pair of values using a “large” module of cells, the system then becomes capable of efficiently executing one or more special algorithms, which otherwise would have been prohibitively difficult to run. I shall describe one or two such possibilities later in this chapter.

**The uniquely addressable range**

To return to the question of multiple modules, let’s first consider a one-dimensional version of the system. Fiete et al. (2008) made several statements about the capacity of the system to uniquely address different ranges with varying resolutions. Firstly they imagine that each module represents integer values, and the scales of the modules are coprime. They conclude, using the Chinese Remainder Theorem, that the capacity of this system is given by the lowest common multiple of all the module scales. Next, they consider that modules are quantized to resolve a fixed number of states, r (e.g. 32). This allows them to put an upper bound on the total number of states the system can be in: $r^N$, where N is the number of modules (e.g. ten).

The upper bound on the width of each state corresponds to the resolution of the smallest scale, i.e. $s_0/r$ (which would be approximately 1cm if $s_0=30$cm). Thus, a rough upper bound for the total width of uniquely addressable space is simply $s_0 r^{N-1}$ (which is approximately $10^{15}$cm for the numbers given). They observe that this formula is exponential in the number of modules but only polynomial in the resolution of each module. In other words, adding extra cells to a module to increase its resolution is less efficient than adding an additional module of suitable scale (if one is looking to increase range). They then go on to numerically evaluate a slightly more realistic scenario in which the modules’ phase values are not quantized, but still have a specific resolution $1/r$, i.e. they look for the smallest distance (larger than $1/r$), for which no module has a phase difference, d, larger than $1/r$ – Figure 23 depicts this notion of resolution.

They see that the earlier formula gives a good approximation, so long as an extra dampening factor is included in the exponent, making the total addressable range now proportional to $r^{0.6(N-1)}$. Even this numerical method is an approximation of sorts as the true calculation should be done probabilistically, taking into account the joint distribution across modules as well as within modules (e.g. as done by Towse et al (2014)).
Fiete et al. (2008) compared a selection of numbering systems and noted that if the ratio of scales across successive modules is small, i.e. in the experimentally observed range of 1.2-1.5 times (or preferably smaller still), then a step/increment in the globally represented location, corresponds to a similar increment in the phase of all modules. Contrast this with the more familiar decimal numbering system, in which small increments usually only affect the first digit, with the other digits having to “liaise” with earlier digits so as to increment suddenly at the correct moment. Additionally, in the decimal system the importance of errors varies widely depending on which digit is in error (e.g. 602 is rather different from 502, but 503 is not), whereas the importance of errors within a grid module varies only marginally across modules.

![Figure 23. Cartoon depicting how resolution was defined by Fiete et al (2008). The 1/r length is the same in all four panels: d1 and d2 are greater than 1/r and thus s in those cases the locations a and b can be resolved; whereas d3 and d4 are less than 1/r and so their a’s and b’s cannot be resolved.](image)

This brings us again to the comments in the introductory section: Fiete et al. (2008) note that if the capacity of the grid system is much greater than the animal actually needs in its lifetime then perhaps it can use some of the redundant capacity as a form of error correction (Sreenivasan and Fiete 2011).

So far I have only looked at the one-dimensional case. To extend the arguments into two dimensions the easiest option would be to assign separate sets of modules for each Cartesian axis, or for at least some pair of non-collinear axes (as any pair of non-collinear axes form a basis with which to uniquely address points in two dimensions). However, as we have already seen, the grid system does not do this, using a two-dimensional tile rather than a pair of one-dimensional lines. As depicted in Figure 24, the benefit of this may be that there is no preferred/optimal direction in which increments can be accumulated most accurately: were a pair of one-dimensional axes used, then off-axes movements would need to be projected back
onto the axes, resulting in smaller absolute changes. This is true no matter what the angle between the axes.

Figure 24. From 1D to 2D. Cartoon highlighting how the magnitude of a 2D vector can be projected onto two 1D axes (left). Note how the magnitude on both those axes is less than (or at most equal to) the magnitude of the 2D vector. If vectors/space is truly represented in 2D then the precision is isotropic (right).
Generating and utilizing the grid cell pattern

This section reviews several important debates that have featured prominently in the literature; the subsequent section then moves into less explored territories that may offer fertile ground for future debates.

Explaining phase precession

As stated in the introductory chapter, when phase precession was first discovered in place cells (O’Keefe and Recce 1993), several explanatory mechanisms were immediately proposed. These hypotheses mostly framed phase precession as an inevitable side effect of the general functioning of the spatial-modulation apparatus, however suggestions were also made as to what benefits could be derived from the phenomenon itself.

One possibility for the mechanism is that the pattern is being inherited from the inputs to the hippocampus: if the network could be sufficiently influenced by external signals, and with an appropriately short time constant, then any spatially-correlated phase precession in upstream regions would be imprinted onto the activity of HPC cells (W E Skaggs et al. 1996). With the discovery of phase precession in grid cells (Hafting et al. 2008), which are likely one of the inputs to place cells, this seemed like a possibility; indeed Schlesinger et al. (2015) recently showed that the EC is required for phase precession in place cells. Of course, even if this were the case - and it does indeed appear that theta is more critical to the functioning of grid cells (Brandon et al. 2014; Newman, Climer, and Hasselmo 2014) - it only shifts the problem elsewhere rather than fully addressing it.

A second possibility is that the pattern at the start of each theta cycle is created by external inputs, and then during the remainder of the cycle it evolves according to a learnt pattern of recurrence, with connection strengths strongest for cells having nearby fields. This model was originally proposed for place cells (Tsodyks et al. 1996; Wallenstein and Hasselmo 1997), but is perhaps better suited to grid cells as their fixed phase relationships remove the need for environment-specific learning (e.g. as suggested by Erdem and Hasselmo (2012)). The existence of directionally tuned grid cells may also hint at how the evolving activity is restricted to a certain direction when the environment is truly 2D (Navratilova et al. 2012; Kubie and Fenton 2012). One outstanding issue is how the pattern would be reset at the start of each cycle in the absence of external stimuli: perhaps the intrinsic oscillatory nature of the neuron membrane could provide a solution, showing a secondary depolarisation 100ms after each burst of spikes (Navratilova et al. 2012).
A third, and not entirely distinct, possibility is that as the animal passes through a cell’s place/grid field, there is a general increase in excitation due to external and recurrent inputs, i.e. the cell is depolarised, which indeed seems to be the case (Harvey et al. 2009; Schmidt-Hieber and Häusser 2013). For a fixed spiking threshold, superimposing an oscillation on this slow depolarisation “ramp” would create spikes at progressively earlier phases of the oscillation (Mehta, Lee, and Wilson 2002; Hopfield 1995). However the excitation ramp must drop off as the animal leaves the cell’s field, so we should actually expect a symmetric late-early-late phase profile, rather than the linear late-to-early result that is observed (Tsodyks et al. 1996). Perhaps this mechanism is responsible for the initial part of the precession, but a separate process (e.g. synaptic short term depression) causes the phase pattern at field-exit, or maybe rate is more strongly anti-correlated with phase than initially claimed (Mehta, Lee, and Wilson 2002).

A final possibility is that the precessing spikes correspond to peaks in an interference pattern: the pattern would be composed of two voltage oscillations added together, one at a fixed frequency, and one whose phase is advanced faster/slower in proportion to the speed of the animal (O’Keefe and Recce 1993; Neil Burgess, Barry, and O’Keefe 2007; Hasselmo, Giocomo, and Zilli 2007; Neil Burgess 2008; Erdem and Hasselmo 2012). There are many possibilities for exactly how/where these oscillators might be implemented and how the “beating” wave pattern could be made to work in two dimensions; if necessary one could even relax the requirement on the first oscillation being of a fixed frequency. However, despite the vague nature of this model, there is a variety of evidence in its favour: (a) there is a fairly robust correlation between running speed and theta frequency (Sławińska and Kasicki 1998; Jeewajee et al. 2008); (b) if this correlation flattens, as seems to occur in unfamiliar environments, the change is reflected by a roughly concurrent increase in grid scale (Barry et al. 2012; Jeewajee et al. 2008; Wells et al. 2013); (c) the resonant frequency of stellate cells decreases along the dorso-ventral MEC axis, approximately mirroring the increase in grid cell spatial scale, and the scale increases when the resonant frequency is artificially reduced (Giocomo et al. 2007; Giocomo and Hasselmo 2008; Giocomo and Hasselmo 2009; Giocomo et al. 2011); (d) there is some evidence of cells that spike at a frequency modulated by speed in a particular heading direction (Welday et al. 2011); (e) when the theta signal is disrupted the grid pattern disappears (Koenig et al. 2011; Brandon et al. 2014; Newman, Climer, and Hasselmo 2014). All these findings are however rather inconclusive, for example the resonance/scale correlation in (c) was not as strong as might be expected and disappeared under the artificial manipulation; also the speed-theta relationship in (a) and (b) is very noisy, possibly non-linear, and its precise significance in the model is not clear; and the
direction/frequency modulation measured in (d) is only of the order of 0.1Hz, right at the limit of what is detectable. Also, contrary to prediction, when measuring the sub-threshold voltage of patch-clamped putative grid cells, Schmidt-Hieber and Häusser (2013) found that increases in firing rate were significantly correlated with a non-oscillatory voltage “ramp”, but only weakly with increases in the amplitude of the theta (although, when working with place cells, Harvey et al. (2009) did identify some evidence of phase precession in the sub-threshold theta, in addition to the non-oscillatory voltage ramp). And even for (e), when the grid pattern disappears in the absence of theta, the system is still able to generate and maintain place cell firing (Brandon et al. 2014), indicating that oscillations are not critical to the spatial code. Indeed it may be that external sensory input is far more critical than originally thought (Chen et al. 2016), reducing the opportunity for a integrative oscillatory mechanism to play a role.

Most of the above proposals incorporate some form of continuous attractor, a concept introduced in an earlier section, in which a stable bump of activity is moved around the network using a combination of symmetric and asymmetric connection weights (Fuhs and Touretzky 2006; Couey et al. 2013); even the oscillatory interference system has been extended to include multiple attractor properties (Navratilova et al. 2012; Bush and Burgess 2014). The exact nature of the attractor and why it generates phase precession is still open for debate, but the consensus now seem to be that it does at least exist in some form, likely an inhibition-only form (S.-J. Zhang et al. 2013).

Advantages of phase precession

When first encountered, the phenomenon of phase precession may appear to pose a problem, since it is no longer ideal to simply readout the firing rate of a cell in order to “decode” its value. However, encoding information in terms of phase has many potential benefits, particularly as the strength of the signal is potentially no longer relevant, only its timing counts (Hopfield 1995). And if the cell is subject to many rate-modulating influences, such as seems to occur with velocity, then by encoding information in phase the data is no longer ambiguous (O’Keefe and Recce 1993; Huxter, Burgess, and O’Keefe 2003).

It is also interesting to note that learning rate (LTP) seems to vary as a function of theta phase (Pavlides et al. 1988), meaning that if the system is encoding different information at each phase of theta it will be resulting in very specific forms of learning, perhaps even allowing for the head-direction signal to be combined with the place code in such a way as to produce a gradient field. Such a learning mechanism might permit a limited form of gradient descent (Neil Burgess, O’Keefe, and Recce 1993), although it might also be that such skewed learning “corrupts” an otherwise more homogeneous spatial code (Mehta, Lee, and Wilson 2002).
Another possibility is that it is not the phase in particular which is important, rather the fact that the spatial sequence is compressed into such a short timespan and then repeated over many cycles (W E Skaggs et al. 1996; Neil Burgess and O’Keefe 1996) – that is, perhaps this would aid in efficient learning (Jensen and Lisman 1996; Wallenstein and Hasselmo 1997; Monaco et al. 2014), specifically addressing the requirements for long term potentiation (LTP), and/or spike time dependant potentiation (STDP) (B. L. McNaughton and Morris 1987; Kubie and Fenton 2012; Widloski and Fiete 2014).

Finally, it may be that different phases of theta correspond to different stages in a top-down vs bottom-up or encoding vs retrieval process (Hasselmo, Bodelón, and Wyble 2002; Newman et al. 2013). By separating the different components of the computation temporally in this way, the system potentially improves its ability to integrate information and to learn with minimal “copying” of data across redundant cells/synapses. This notion that different phases of theta could have different computational roles is something that I shall return to in the next section.
Calculating with grid cells

With the exception of the discussion on “vector addition”, much of the material in this section is adapted from the work of Bush et al. (2015), though with a focus on the “rate coded vector cell” model that I conceived. Note that the text, diagrams, and mathematical notation used here are largely my own (either specifically created for this thesis or through my original contribution to the previously published work). However the simulation presented was not performed by me. As an accompaniment to the text here, readers may like to view the interactive diagram I created (after the publication of the above article); it is available online at d1manson.github.io/toy-grids.

Overview

As already noted, the hippocampus encodes a variety of spatial information, and without it the performance of many spatial tasks is impaired. Such discoveries have encouraged investigators to form hypotheses on the exact classes of spatial problems that are being solved by the region, and what algorithms are being used to solve them.

One important class of problem is navigating in straight lines from a current location to a previously visited location, referred to in this context as a “goal”. Restricting the problem to straight lines is an appealing simplification, however its real-world applicability could be questioned: any region in which a straight line path between (most) pairs of points constitutes a valid (i.e. unobstructed) route, is a region in which navigation could be performed simply using distal cues, i.e. without complex internal machinery for dealing with space. However, over long enough distances, it may be the case that obstructions exist but can be easily circumnavigated; in this scenario having a mechanism to compute vectors between pairs of points becomes useful. At the other extreme is a scenario where most areas are impassable, with only a limited set of valid corridors connecting distant locations: this situation is familiar to human city dwellers but may also be of relevance to rats, which often live in human-constructed environments and seem to develop networks composed of narrow “ratruns” (Davis, Emlen, and Stokes 1948; Feng and Himsworth 2013).

In recent years, considerable effort has been invested in designing algorithms to solve these problems – most current solutions implement Dijkstra’s algorithm supplemented by some form of heuristic “field” (Sanders and Schultes 2007). In brief, Dijkstra’s algorithm involves starting at one location and iteratively computing the distance/cost of traveling to all adjacent locations until the goal is reached. To ensure the optimal route is identified, the nodes must be processed in order of distance/cost (of reaching the given node from the start); a heuristic field
can be used to prioritise nodes that are more likely to lead to the destination (but in such a way as to maintain the optimality of the solution). A basic form of the algorithm has inspired a treatment of CA3 as a “cognitive graph”, where connection weights between place cells are interpreted (through some function) as representing distances (Muller, Stead, and Pach 1996), however it is not clear how this information could be appropriately utilized by the network unless a much more elaborate network topology is learnt (Erdem and Hasselmo 2012). As such, the question of computing the “shortest path” over a network will not be considered any further here.

A simpler, and more limited mechanism, still using only place cells, would be to compare the representation of the goal location with that of the current location and move so as to increase the similarity between the two (Neil Burgess and O’Keefe 1996). Navigating in this way is limited in range to the diameter of the largest place fields, unless combined with experience-dependent learning over multiple trials, which will tend to bias behaviour toward previously learned routes. Beyond this range, the similarity of the current and goal place cell representations will be zero, providing no gradient in similarity leading to the goal location. Although large place fields have been recorded (~10m; Kjelstrup et al. 2008), these properties clearly limit the utility of place cell representations for large-scale vector navigation.

As already discussed, and in contrast to place cells, grid cells have the potential to provide a context-independent spatial metric that spans large spaces. It is widely believed that this property is utilized during path integration - the process of updating the representation of self-location by a vector describing the animal’s recent motion (Hafting et al. 2005; Bruce L McNaughton et al. 2006). However, this same property may also allow for a more general computation that calculates the vector between any pair of points, even when the two locations are much farther apart than the largest grid scale (Bush et al. 2015; Neil Burgess, O’Keefe, and Recce 1993). A third, and more controversial proposal, is that the grid system could be used for activating the representation of distal locations using current location plus a known vector, such as those found in Marr’s two-and-a-half-dimensional sketch (Marr, David 1982) (when used in conjunction with the allocentric head direction signal).

The first of these proposed functions requires that the grid be regular only on the shortest scales, unless a homeward-pointing vector is included in the definition of “path integration”, in which case grid regularity is required over larger distances. The third function requires regularity within the “perceptual horizon”, failing with some degree of gracefulness in the presence of irregularities. However, the second function requires grid regularity over large spaces (Carpenter and Barry 2016). Whether or not sufficient regularity exists is unclear: in
the open field the grid pattern can take many hours to become regular (Barry et al. 2012), and may actually become less regular over time (Krupic et al. 2015; T. Stensola et al. 2015). When obstacles (in the form of walls) are present, the pattern can take longer still to span the whole space or perhaps never succeeds in doing so (Derdikman et al. 2009; Carpenter et al. 2015). Indeed, if the famed regular metric of grid cells is not available until after many hours of experience within an environment, it is not obvious that the grids are providing any functional service during this initial exposure, which is when a universal metric is most needed: a prolonged period of grid regularisation provides plenty of time for place cells to develop a simple metric/vector representation of their own (as mentioned above in reference to place-cell-only models of navigation) and thus make grid cells redundant. This leads back to the question posed at the start of this chapter: is it more accurate to treat grid cells as an optional, but possibly highly useful, aid to place cells rather than a computational device in their own right?

Here, I consider the problem of large-scale vector navigation with grid cells at Marr’s three levels of analysis. First, I outline the computational problem to be solved: how to compute a translation vector between co-ordinates encoded in an idealized grid cell system. Second, I describe an algorithmic solution to this problem, based on the grid cell network and related to the Fourier shift theorem. This solution is focused on resolving ambiguity between the multiple, periodic locations represented by activity within each grid module, rather than optimizing the efficiency and accuracy of the grid cell code for location within the scale of the largest grid. Finally, I describe several plausible neural network implementations that use grid cells to calculate the translation vector between start and goal locations in 2D space over distances that can exceed the largest grid scale. I focus on proposed mechanisms that can perform vector navigation relatively rapidly (i.e., without an exhaustive search of the numerous possible solutions) and that provide experimentally testable predictions.

1D navigation: the problem

We parameterize the grid cell spatial representation as follows: there are M grid cell modules with spatial scales $s = s_1, \ldots, s_M$ (with $s_1$ being the smallest and $s_M$ the largest). In 1D space, we can visualize each module of grid cells as a ring that supports a population activity bump centred at phase $p_i$ on the interval $[0, 1)$. The 1D vector navigation problem can be stated thus: given the grid cell representations of two locations a and b, calculate the displacement between those locations, i.e. compute $d = b - a$. More specifically, if we denote the phase of the activity bump in module $i$ at location $x$ as $p_i(x)$, then the grid cell representations of
locations a and b are given by $p(a) = p_1(a),...,p_M(a)$ and $p(b) = p_1(b),...,p_M(b)$, and we must use this information to compute d.

For example, let's take $s = 20cm, 30cm, 50cm$, and for the sake of simplicity (but without loss of generality), let's say $p(a) = 0,0,0$. Now if the correct answer were d=75cm, then the question would be posed with $p(b) = \frac{75 \mod 20}{20}, \frac{75 \mod 30}{30}, \frac{75 \mod 50}{50} = 0.75, 0.5, 0.5$ and the task would be to recover d. See Figure 25.

**Figure 25** The problem of computing a vector, $d = b - a$. Cells that are active at the start location are marked in red and those active at the goal location are marked in yellow. See main text for a full description. Adapted from (Bush et al. 2015).

**1D navigation: the solution**

The cyclical nature of the grid representation within each module i is such that an activity bump at phase $p_i$ implicitly represents an infinite set of “unwrapped” phases $p_i + n_i$, where $n_i$ can take any integer value, corresponding to an infinite set of distances along the 1D axis: $s_i(p_i + 0), s_i(p_i + 1), s_i(p_i + 2), ...$. If we initially treat a as being zero, then $p(a) = 0,...,0$.

The solution corresponds to the vector of integers $n = n_1,...,n_M$ for which:

$$d = s_i(p_i(b) + n_i) \text{ for } i = 1,...,M$$

Graphically, as seen in Figure 25, if modules are depicted as being periodic and shown at appropriate relative scales, then the solution corresponds to the straight line which has the correct phase in each module. For arbitrary $p(a)$, we simply replace $p(b)$ in the above equation with $\Delta p = p(b) - p(a)$:

$$d_{ab} = d_{ba} - d_{oa} = s_i(p_i(b) + n_{bi}) - s_i(p_i(a) + n_{ai}) = s_i(\Delta p_i + n_i) \text{ for } i = 1,...,M$$

Note that when the above equation is divided by $s_i$ and phase is expressed in radians rather than on the interval $[0, 1)$, we obtain a statement related to the Fourier shift theorem (Orchard, Yang, and Ji 2013).
It is important to note that because the entire grid pattern (i.e. across all modules) repeats after a certain distance, there are in principle an infinite number of possible solutions to the question of what value of $d$ satisfies $d=b-a$. However, I have already discussed the size of the uniquely addressable range in an earlier section so will not dwell any further on it here.

**1D navigation: possible algorithms**

As summarised in Figure 26, Bush et al. (2015) proposed four algorithmic implementations for solving the navigation problem.

![Figure 26. The differences in the four navigation implementations provided in Bush et al. (2015), this summary figure is not part of that paper.](image)

I conceived implementation 2, the “rate coded vector cells”, and this will be discussed in some detail in this section. Briefly, the other implementations were:

1. The “distance cell” model - a string of cells represent individual points on the number line, and these number line cells are then connected to a single read-out cell that expresses the number line’s value as a rate. Two sets of such cells provide the rate at location $a$ and location $b$, and the rate difference provides the distance from $a$ to $b$. 


3. The “phase coded vector cells” model – similar in some respects to the “rate coded vector cells”, here it is assumed that when at position a, if the grid pattern for position b is represented by the network, then the phase (with respect to theta) of each cell’s spikes reveals $\Delta \phi_i$, i.e. the phase difference between a and b for that particular module. This is similar to phase precession as observed during open field experiments. Assuming this phase code is present, then a population of vector cells could read out the set of values and identify the single vector that matches across all modules. The general notion of computing with phase/temporal deltas was a concept championed by Hopfield (1995).

4. The “linear look ahead” model – rather than calculate the distance in a fixed number of computation steps, here the network uses an approach closer to Dijkstra’s algorithm, but for the trivial 1D case: a bump of activity is rolled out across the space until the desired location is reached, meanwhile a counter of some kind tracks the length of the vector needed. Models of this kind exist elsewhere in the literature (Kubie and Fenton 2012), and include the field-based optimization for Dijkstra’s algorithm mentioned earlier (Erdem and Hasselmo 2012), as well as “hierarchical” architectures that operate over multi-resolutions (Erdem and Hasselmo 2014).

The rate coded vector cell model, shown in Figure 27, posits that there should be a set of vector cells, with each cell connected to specific pairs of grid cells. These paired connections must be set up such that the two grid cells within the pair are only able to depolarise the vector cell when they fire together (or perhaps at specific phases with respect to theta, or with a specific delay between cell 1 firing and cell 2 firing). Assuming this is indeed possible, then during development, the vector cell representing distance d, should receive inputs from all pairs of grid cells that represent that span d. For example, as shown on the left of Figure 27, if module 1 has a scale of 20cm, then each pair of cells separated by phase 0.25 can represent the span d=5cm, 25cm, 45cm, 65cm, … and should thus be connected to the vector cells with corresponding lengths. If each vector cell receives paired inputs from all grid cell modules, then a winner take all dynamic in the vector cell population will select the single vector that is valid across all modules.

13 We do not make claims about the details of this developmental process, however one might imagine a pair of waves of activity rolling across each grid module, with the phase difference between the waves varied so as to cover all possible offsets. Patterns of activity, perhaps similar to this, are thought to occur at various stages during development (Review by Bruce L. McNaughton et al. 2006).
The distances represented by vector cells may be more coarsely defined at larger scales, so that the system can represent a wide range of distances without sacrificing precision at the low end: as the animal approaches its goal, the vector would be recalculated one or more times so as to home in on the target’s exact location (a similar mechanism is suggested by Erdem and Hasselmo (2014) when performing linear lookahead over place cell maps with multiple levels of resolution).

Figure 27 The rate coded vector cell model. Cells that are active at the start location are marked in red and those active at the goal location are marked in yellow. Left: in module 1, various pairs of grid cells that are separated by a phase of 0.25 connect to a set of vector cells representing the spans of \(0.25s_1, 1.25s_1, 2.25s_1\) .... Right: three modules of grid cells connect to the single population of vector cells. The multi-peaked trace adjacent to each module shows the excitation afferent on the vector cells due to that grid module. Due to winner-take all dynamics in the vector cell population, a single bump of activity develops representing the correct distance between start and goal. Adapted from (Bush et al. 2015).

Although not discussed by Bush et al. (2015), the system may not support navigation on the scale of the entire addressable range. If, as seems to be the case, the animal uses “pockets” of the addressable range in different environments, without joining these pockets into a single universal map, then vector calculations cannot, and should not, be attempted at this most extreme scale. One advantage of relaxing the requirements in this manner is it simplifies the connections required: the vector cell population no longer needs to disambiguate quite so many vectors, and thus each grid module can dampen/truncate its connections to the vectors that lie far beyond the scale of that particular module – this is depicted on the right of Figure 27.
**2D navigation**

The problem of navigation in 1D can be extended to 2D by treating a, b, p and d as two-element coordinates/vectors (rather than scalar values), with the two elements corresponding to non-collinear axes. The equation to be satisfied remains the same, but now represents two independent scalar constraints rather than one.

The obvious approach to solving this pair of equations is to deal with each separately, i.e. as though it were from the 1D case. However, there are some potential problems: if the alignment of grid modules is not fixed, then the vector cell model breaks down: every choice of 1D axis produces a different set of distances between a given pair of cells, and thus if no particular axis is chosen during development (because the modules freely rotate relative to one another, and thus relative to the head direction system) then there can be no learning of the grid-pair-to-vector relationship. This could be overcome if the model is extended into true 2D rather than a pair of 1D axes, i.e. the grid-pair-to-vector relationship must now capture all two-dimensional vectors between the pair of grid cells. However, even this is insufficient: you would now need a separate vector population for each grid module, with some complex form of coupling across vector populations that permits a choice of alignment. The distance cell and linear look ahead models would deal better with this scenario as both use the grid code in a more direct/simplistic way.

A second possibility is that the grid modules have fixed relative orientations but are not exactly aligned. In this case, the full 2D vector cell model would work perfectly, without any complex adjustable coupling (as suggested in the previous paragraph). The doubled 1D approach may work to some extent, however in this scenario the choice of axes cannot be aligned to all grid modules: as shown in Figure 28, this is problematic in that it greatly increases the number of potential vectors in the two axes (making it harder to identify the winning choice), and leads to an unfortunate cross-dependence on the choice of vectors in the two axes.

---

14 For example every vector cell in population 1 representing length d would be connected to every vector cell in population 2 representing length d, and shunting inhibition would be used to ensure that only one of the connections is active in any particular environment.
Figure 28. Composing two 1D axes to navigate in 2D. The cell active at the start location is marked in red and the cell active at the goal location is marked in yellow; no specific “true goal” location is marked here. The two axes may be aligned to the grid (left) or at some other orientation (right). When aligned to the grid, the set of possible 2D vectors is given by the full Cartesian product of the u and v sets. However, when the axes are misaligned, the u and v sets are larger and the set of possible 2D vectors is now a subset of the Cartesian product, i.e. many pairings of u and v are not valid in 2D.

Simulating the rate coded vector cell model

Bush et al. (2015) simulated the rate coded vector cell to test its viability. I reproduce here the details of those simulation and the results figures (with some minor changes in notation). Note that the original work was performed by DB not by me.

Locations are encoded by \( M = 10 \) grid cell modules whose spatial scales are arranged in a geometric progression: \( s = 25 \text{cm}, 1.4 \times 25 \text{cm}, 1.4^2 \times 25 \text{cm}, \ldots, 1.4^9 \times 25 \text{cm} \) (i.e. the maximum scale was approximately 5m). Each module consists of \( N_{\text{GC}} = 400 \) grid cells distributed evenly among \( N_p = 20 \) equally distributed spatial phases. We note however that the exact distribution of grid scales has no effect on the simulations presented here, aside from determining the capacity of the grid cell system to encode unique locations (Fiete, Burak, and Brookings 2008; Mathis, Herz, and Stemmler 2012; Towse et al. 2014) and the ability of the grid cell system to deal with neural noise (Sreenivasan and Fiete 2011; Mathis, Herz, and Stemmler 2013).

\( N_{\text{sim}} = 1000 \) simulations were run using randomly assigned current and goal locations in 2D. These locations were picked uniformly from a rhomboid region with sides of 500m and \( 60^\circ/120^\circ \) internal angles. We note that the size of this area is significantly less than the capacity of this grid cell network to encode locations with a unique set of grid cell phases across modules. Although I noted above that the double 1D solution is not necessarily
sufficient for 2D navigation, that is what is implemented here, using axes aligned to the grid. In fact, when modelling grid cell firing rates here, I ignore the implications of firing rate being modulated along two axes, and instead consider each axis in isolation. (In a sense we thus emulate a larger population of grids that have a much smaller active subset.) Grid cells in module i at location a, are modelled as a Poisson process with the following rate:

\[ r_i(a) = r_{\text{max}} \frac{1}{2} \left[ 1 + \cos \left( \frac{2\pi}{s_i} [a - s_i p_i] \right) \right] \]

where \( r_{\text{max}} = 30 \text{Hz} \) and \( p_i \) gives the phase of the bump in module i. The spike counts are simulated for a 100ms window.

As shown in Figure 29, the vector cells’ magnitudes are pseudo-exponentially distributed from zero to 500m. This range is covered by \( N_{\text{VC}} = 1250 \) cells in the positive and negative direction in both axes, giving 5000 vector cells in total. The grid cell pairs for each axis are connected to the relevant vector cells as previously discussed (including the use of a logical AND operation, however note that there is no attenuation in weights at larger scales). Winner-take-all dynamic in the vector cell population is achieved with an E%-max algorithm, such that cells only fire if their input is within \( k=1\% \) of the maximum feed-forward excitation, a result that can likely be achieved with suitably arranged recurrent inhibition (de Almeida, Idiart, and Lisman 2009).

Figure 29. Simulations of the rate-coded vector cell model. Left: pseudo-exponential distribution of vector cell magnitudes. Centre: the distribution of navigation errors (i.e. distance) in the final iterative step of 1000 separate simulations. Right: navigation error (distance) in the first iterative step of the 1000 separate simulations, plotted against the true distance to the goal. Correlation shown is \( r=0.61, \ p < 0.001 \). From (Bush et al. 2015).

Each iteration of the simulations was performed independently along each axis, and consisted of simulating grid cell counts, followed by computing vector cell rates, and finally decoding the output vector from the weighted average of all the vector cells, i.e. using the firing rates as
the weighting. The current location was then updated by 80% of the decoded vector, and the next iteration begun. Once the goal was within 1m of the current location the iterations ceased.

As shown in Figure 29, the simulated system was able to navigate to within 4cm of the target. The mean number of iterations required was 3.83, with a range of 2-4. As expected, the size of the error in the initial step correlated with the true distance required (due to the decrease in vector cell precision at greater magnitudes), again see Figure 29 and its caption.

**Computing the representation of distal locations**

In the previous sections I looked at how the grid system may solve (for \( d \)) the equation \( d=b-a \). If we rearrange this to give \( b=a+d \), and ask to solve for \( b \) given \( a \) and \( d \), we now have a different question – see Figure 30. This new question corresponds to knowing the hippocampal representation of one location, and wishing to find the representation of a distal location, given some specific vector - this question is rarely addressed in the literature, but see Gallistel and Cramer (1996).

![Figure 30. Cartoon contrasting vector subtraction and vector addition.](image)

In both cases the animal is at the location \( a \), illustrated in red in the grid code. The second input to the calculation, illustrated in yellow, and the output, illustrated in green, differ between the two paradigms: in the case of vector subtraction, the second input is the distal location, \( b \); and the output is the vector \( d \); whereas in the case of vector addition, the second input is the vector \( d \), and the output is the distal location, \( b \).
As suggested in an earlier section, if a is the current location, then the vector could represent the distance and direction to an object pictured on the retina: binocular rivalry as well as various other techniques allow the visual system to compute such vectors (Marr, David 1982) and the head direction system could anchor the vector in an allocentrically-oriented reference frame. Similar feats are possible for auditory stimuli, direction in particular, and indeed the visual and auditory localisation systems meet at the optic tectum (Knudsen 2002). Before going further, however, we must consider the importance of the third dimension: due to the constraints of gravity, most land-based mammals experience a planar world populated by objects that have a clear footprint in the 2D world: trees, rocks and other animals are all well approximated as (elliptical) cylinders with a central vertical axis perpendicular to the plane. A large amount of useful spatial information can thus be recorded using only two dimensions, and indeed it appears that the hippocampal system in rodents is largely restricted to representing planar surfaces (Kate J. Jeffery et al. 2015). However, when sensing in the 3D world, the animal requires a mechanism for projecting distances down to that 2D plane: if the eyes are not aligned to the (pseudo-)horizontal, or the object of interest is not centred on the retina, then the distance to the object (as measured by the visual system) will not represent the distance along the plane.

As shown in Figure 31, this projection corresponds to multiplying the distance by the cosine of the viewing angle. I shall not discuss here the details of how this calculation could be performed, however I do note that the head direction system may well include cells that are tuned to the vertical angle (especially in bats, see review by Jeffery et al (2015)), and that within a range of ±30°, cosine is actually within 15% of unity.

Unlike in the previous section, I shall also not give a full description or simulation for the solving of the b=a+d problem, instead I simply note that the rate coded vector cell model could be extended by introducing a reciprocity in the vector-cells/grid-cell connections, and that this extra apparatus would likely be all that was needed for the calculation.

Were this system to exist it would have the capacity to compute the hippocampal representation of distal locations many times a second, i.e. at a similar rate to visual saccading.

---

15 Distance estimation is really only feasible when the acoustic environment produces reverberations, i.e. when in a room or close to a large barrier (Bronkhorst and Houtgast 1999).

16 A select group of rats have actually had the chance to “escape” gravity, during several days in orbit; with some rats even successfully providing place cell data during the flight (Knierim, McNaughton, and Poe 2000). That is despite the unfamiliar 3D arrangement of the recording arena.
Salient features of the environment could thus be annotated to their own location (as represented by the hippocampus) rather than to the current location of the animal. The system could also be used for path integration, i.e. updating current location based on the instantaneous velocity vector.

Figure 31. From 3D to 2D. An animal, depicted here as a pink rectangle, looks up towards the tops of two blue cylinders that stand vertically on the horizontal plane. The head direction system provides the angle of inclination of the head/eyes and the visual system computes the distance to the top of the cylinders. To project the distance onto the plane, you must multiply by the cosine of theta.

The calculation, as envisaged here, would be composed of an “input” phase and an “output” phase, which would be evident in the EC as a phase-precession-like phenomenon (or even as a shift in the mean of a bimodal distribution (W E Skaggs et al. 1996; Hafting et al. 2008)), with one set of grid cells active at the start of theta and a different set at the end of theta. If used primarily for path integration, the observed pattern would indeed match that seen in empirical data, namely with the current location coming early in the cycle (due to being the input to the calculation) and future locations come later in the cycle (due to being the output of the calculation). If the EC signal is propagated from/to HPC then the phase precession observed in HPC would be more pronounced than in EC (due to the input phase coming even earlier and the output phase coming even later), which is indeed the case (Jeewajee et al. 2014). Additionally, if directional grid cells correspond to part of the computation machinery it seems appropriate that they should lack any sign of phase-precession (as reported), since their firing would occur during the middle phase of the calculation, i.e. between the input and output phases.
In the framework of this model, the phenomena of theta flickering (Jezek et al. 2011) and theta skipping (Brandon et al. 2013) would correspond to periods in which the grid system is being used for something more complex than path integration: whether a particular grid cell was active on a given theta cycle would depend on the computation taking place during that cycle, with adjacent cycles potentially working with quite distant locations.
5. Grid cells in the presence of a robot

I performed all data collection and analysis presented in this chapter, as well as designing and programming the control of the robot.

Introduction

In this chapter I address the hypothesis posed at the end of the previous chapter, that is I ask whether there is any evidence that cortex, and specifically the grid cells of the entorhinal cortex, are capable of performing the b=a+d vector calculation. Or in words, can the animal add a vector to the representation of one location so as to obtain the representation of another location. And in particular, I hypothesise that such a computation could be done on a timeframe similar to that of saccading and/or hippocampal theta.

In monkeys there is some evidence that the hippocampus can represent the location that the animal is currently viewing independent of the animal’s current location (Georges-François, Rolls, and Robertson 1999; Killian, Jutras, and Buffalo 2012), although little data has been collected to date. Similar, though slightly less direct, evidence exists for rats (Ólafsdóttir et al. 2015). If this were true, it implies there is some mechanism for doing the b=a+d calculation even if it is not implemented in the way suggested. In rats, examples of true boundary vector cells (Lever et al. 2009; Solstad et al. 2008) demonstrate that rodents are able to find d (i.e. a vector oriented in allocentric space), meaning that they at least have the inputs for the calculation even if they do not go so far as to compute the result. Additionally, given that grid cells may turn out to be poor at performing path integration (Derdikman et al. 2009; Carpenter et al. 2015; Chen et al. 2016) due to inevitable error-accumulation over multiple iterative steps (Erdem and Hasselmo 2014), one might conclude that the system instead favours single-iteration calculations, such as b=a+d. There is also some evidence from lesion studies (Parron, Poucet, and Save 2004) that the entorhinal cortex plays a critical role in storing/processing information about (specifically) distal landmarks.

Here I present preliminary data from experiments in which rats shared an arena with a robot ball. Grid cells were recorded from the rat as it foraged in the arena, with the robot ball programmed to approximately pursue the rat. I predicted that the typical, neat, grid pattern would be somewhat disrupted during this paradigm due to the grid machinery taking part in a wider variety of computations than usual. Specifically, I hoped to see that some fraction of
the time, perhaps at specific phases of theta, the grid cells would encode the location of the robot rather than the location of the rat.
Methods

See main methods chapter for details of electrodes, surgery, recording, animal handling, and basic data processing.

The robot and its control and tracking

A spherical robot toy, Sphero 2.0 (Sphero, Colorado USA; gosphero.com) was programmed to follow rats within a square arena. The robot consisted of a hard white plastic spherical shell, diameter 3 inches (7.6 cm), encasing a battery, gyroscopic sensors, LEDs and electronic control circuits. It was controlled via Bluetooth from a custom written Python (Python Software Foundation 2010) script running on the main recording computer. A USB webcam, with a view of the whole arena, was connected to the same computer and was used to track the location of the robot. The arena used was the same as that in baseline/screening trials in the main methods chapter, i.e. its floor and walls were black. During recording sessions, only the small blue LED “tail light” on the robot was turned on, so that when viewed from the webcam the robot appeared as a clear bluey-white disk against a dark black background. (The rat was tracked using the standard arrangement of overhead infrared camera and head-mounted LEDs, thus there was minimal spectral overlap in the two tracking operations.) The webcam achieved a framerate of approximately 30 Hz, with data being processed online using a custom written Python script employing routines from OpenCV (Bradski 2000). The spatial and temporal alignment between the two tracking systems was not explicitly quantified for this experiment; rather the setup was judged purely on its capacity to steer the robot in pursuit of the rat. In each 2 s command period, the robot was instructed to travel in a nominally straight line at a fixed speed of approximately 30 cm/s. During these straight line movements, its direction was estimated from the webcam data and a (damped) angular error correction was applied to future commands. If the robot’s sensors detected a collision or the computer observed that the robot was not covering large enough distances, the computer instructed the robot to rotate 90 degrees and attempt moving in the new direction. During “pursuit” periods, the Python script read the position data file being generated by the Axona software and used the latest rat coordinate to compute a target direction for the following 2 s of robot movement. In non-pursuit periods, the script randomly picked a location in the central part of the arena and used that to compute a target direction for the following 2 s of robot movement. Some trials consisted entirely of pursuit periods, in other trials each pursuit period was followed by two non-pursuit periods. One charge of the battery (using the inductive charging base station) lasted for well over 20 minutes of movement, and could be topped up in the intervals between trials. Although the control of the robot was fully automated, very occasionally (at most once
or twice per trial) the experimenter physically intervened, moving it away from the corners/walls.

**Subjects and protocol**

Two rats were used in this experiment: R2112 performed two days of three trials. The first trial used constant pursuit mode (“hot pursuit”), the second trial used pursuit mode for 2s every 6s (“cold pursuit”) and the third trial acted as a baseline, with the robot absent or inactive in a corner. R2175 performed a similar protocol, but for five days, and with days 2-5 having an additional baseline trial at the start.

The robot contained a blue LED light, which was used for tracking, but also made it easier for the rat to identify its location. Further, the robot made distinct sounds as it rolled across the floor of the arena, as it made purposeful turns, and when it knocked into the walls. Most trials were conducted with a low level of ambient lighting provided by a desk lamp outside the arena, but in some trials the desk lamp was not used and instead the only illumination available was from the robot itself. Although in most trials grains of rice were provided, as done in standard open-field foraging tasks, the amount of rice was sometimes reduced significantly so as to encourage the rat to attend more closely to the robot.

**Rat-robot distance analysis**

The heterogeneously sampled robot path data was resampled to the same frequency as that of the rat (i.e. as output by the Axona tracking system; 50Hz). Then, for every sample, the Euclidean distance between the rat and ball was computed. In Figure 32 this distance data is expressed in cumulative form and as a fraction of the trial duration, i.e. it shows the fraction of time during the trial for which the rat and robot were within a certain distance of one another. The shuffled distribution in that figure was generated by temporally shifting and wrapping the robot and rat paths relative to each other. A minimum shift of ±60s was enforced and 100 shuffles were performed. At each point along the x axis the 1, 10, 50, 90, and 99\textsuperscript{th} percentiles of the cumulative distributions is shown. Note that for most of the distribution the 1-10 and 90-99\textsuperscript{th} percentiles span very narrow ranges.

**Theta skipping analysis**

As introduced by Climer et al. (2014), the shape of the temporal autocorrelogram was captured in a model that included terms for exponential decay and oscillation at two frequencies. The easiest way to comprehend the meaning of all the parameters in the model is to view the interactive widget put online for the publication of the original paper (short url: goo.gl/Zaf6sU). The parameter of most interest here is s, which captures the strength of theta
skipping: it varies from 0 to 1, and governs the extent to which power is shifted from theta frequency (at 0) to half theta frequency (at 1). See the original paper for further details. Note that the implementation used here was custom written in Python, but translated and adapted from the Matlab code published with the paper rather than created de-novo from a mathematical description. Here I used a frequency search range of 6-13Hz.
Results

As described in the methods section, the rats had strong multi-modal sensory cues with which to locate the robot and (due to the possibility of being run into) had an incentive to be aware of its location. Unlike most tasks investigating goal-navigation, here there was no finite set of points of interest, and the rate at which spatial calculations must be done was, perhaps, much higher: in goal-navigation tasks, approximately one calculation must be done every trial, and this calculation may be very brief and occur at an unknown moment, thus making it difficult to study; whereas here the rat would benefit from constantly understanding exactly where the robot was and its trajectory, and using this knowledge to pick locations in the environment that are safe to navigate to, i.e. there are many truly two dimensional spatial calculations that could be occurring.

When exposed to the robot, rats initially appeared fearful, rearing up many times around the edges of the arena, and holding their tail off the floor so as to avoid being knocked by the ball. That rats are fearful of a locomoting robot is not surprising; indeed Kim et al. (2015) claim this is as an amygdala-dependant phenomenon. However, after one or two trials the animals became more confident, with a definite reduction in rearing (although never quantified) and occasionally “mounting” the ball or trying to bite into it (which proved not to be possible).

The rats appeared capable of looking at the ball from a distance, indicating both that they were interested in it and that they were able to recognise and locate it from afar (although “locating” here may simply indicate aligning the visual field with the stimulus rather than placing it within a true allo- or even ego-centric reference frame). However, once the robot had been identified as largely harmless, the animals seemed to ignore it on the whole, making little attempt to actively avoid it much of the time. While this was desirable from the perspective of not causing distress to the animal, it raised the possibility that the hoped-for spatial calculations were not being performed.

Note that after exposure to the robot, it is likely that the animals behaved differently in the baseline trials, due to (a) the knowledge that the experimental arena would sometimes contain a “threat”; and (b) because a fight-or-flight response may still have been in effect from the most recent robot trial. No attempt was made to quantify either potential effect.

Figure 32 demonstrates that the robot ball was able to pursue the rat successfully, spending a much greater proportion of the trial in close proximity to the rat than expected by chance. The data also shows that the rat and robot were rarely closer than about 5-10cm, presumably because the two “creatures” could not physically or at least “willingly” occupy the same
location. We can thus conclude that the two tracking systems were reasonably aligned (at least within the rough requirements of this preliminary experiment). Figure 33 shows the rat’s speed distribution for all trials, and does not appear to indicate any significant patterns, except perhaps that the robot caused an increase in the general variability of the speed profile.

**Figure 32.** Rat-ball distance for two example trials. Raw data is shown in red, expressed as cumulative fraction of trial, and grey and black area shows the distribution of shuffled data. The trial on the left used hot pursuit mode, whereas the trial on the right used cold pursuit mode. Data is from R2112 28-03-2014.

**Figure 33.** Rat speed distribution for all trials. 25 trials are shown here, with “baseline” including both pre- and post- exposure.

Although there may be some heterogeneity in the properties and functions of individual grid cells, if the system truly represents a metric for space one would expect all aspects of space to be defined using that same metric: since the grid cells are the coordinates, whenever a particular coordinate is being referred to, the corresponding grid cells should presumably become active. This is in contrast to place cells, which have no well-defined metric identity:
individual cells can be co-opted as and when necessary to represent objects in space. Thus, if the system is indeed being used for “exotic” spatial calculation, i.e. not simply path integration, cells that are normally neat grids, should show a significant amount of out-of-field activity, possibly at specific phases of theta.

In total, 12 putative grid cells were recorded from two animals (R2175: 10 cells for 1-4 days; R2112: 2 cells for 1-2 days). The top row of Figure 34 shows two simultaneously recorded grid cells in trials with and without the robot. Contrary to prediction, initial inspection does not reveal any clear increase in out of field activity for trials with the robot. The bottom row of the figure shows the same data, but with the path (and spikes) from the robot’s location rather than the rat’s. If the grid cells were occasionally being used to represent the robot’s location we should observe a grid pattern superimposed on a high level of baseline activity. There is, however, no such sign. Note that in the first trial the robot spent much of its time in close proximity to the rat (by design) so any hints of grid pattern may in fact be due to a partial correlation in the location of rat and robot.

Initial recordings suggested that in the presence of the robot, grid cells showed an increase in theta-skipping. This was quantified by performing a maximum likelihood fit to the temporal autocorrelogram, using the parametrized equation and methodology of Climer et al. (2014) – see examples in Figure 35. As already suggested, an increase in theta skipping might be a reflection of a diversification of spatial computations, although it may also be a sign of more coarse neuromodulatory changes (Yu and Dayan 2005; Newman et al. 2013). The results in Figure 36 show that although there was a significant increase in theta skipping during pursuit trials, the increase remained after the robot was removed. Most likely this hysteresis indicates that the phenomenon is caused by neuromodulatory changes, however in principle one might also suppose that the animal has switched into a different behavioural state: rather than aimlessly foraging in the arena, it is now taking more care to plan its movements and check for disturbances in its vicinity. Kim et al. (2015), who recorded place cells on a linear track in the presence of a robot, did not examine theta skipping, but they did analyse LFP theta power, finding an increase that was likely due to fear.
Figure 34. Example grid cells recorded in the presence of a robot ball. Top row: the rat’s path is shown in grey with spikes from two grid cells superimposed. In the first trial the robot was in hot pursuit mode, for the second trial it was in cold pursuit mode, and the final trial was a baseline with the robot absent/inactive. Bottom row: the robot ball’s path is shown in grey, with spikes from the same two grid cells superimposed. Data is from R2112 28-03-2014 with t1c1 (blue) and t4c1 (green).
Figure 35. Examples of the Climer MLE method for estimating theta skipping (Climer et al. 2014). All four plots are from R2175 2015-01-28 tSc1. The first and last plots are the baseline trials, i.e. with no robot ball, and the middle two are the hot and cold pursuit trials, respectively. The three values above each plot correspond to (some of) the parameters estimated using the “with skipping” model: skipping (s), frequency (f) and rhythmicity (r).

Figure 36. Variation in theta skipping. There are seven total days of data presented here, two from one rat, and five from another. Due to the preliminary nature of the experiment, the number and order of trials was not identical across days: some days began with a baseline trial, some ended with a baseline trial, and some had both such baseline trials. The hot and cold pursuit trials were always present and performed in that order. Each data point corresponds to a grid cell on one day. No effort is made here to de-duplicate cells across days. The bar plot on the right shows the mean ±SEM for all the available data. The two plots on the left show roughly the same data as each other, but with slightly different subsets (due to the variation in daily protocol), and using a different choice of x-axis. As is suggested by the plot on the left, when combined, the hot and cold pursuit data show significantly more theta skipping than on the pre-exposure baseline trial (2-sample t-test: T=-3.58, p=0.0009).
Discussion

Despite providing rats with a strongly spatial dynamic stimulus, I could identify no major sign that the grid system was behaving any differently to normal, i.e. it was not performing additional calculations. Does this mean that the $b=a+d$ problem is never solved by the EC? Possibly not: perhaps there are different groups of grid cells, that independently constitute attractors, but which are only loosely coupled together; certain calculations would be done using one group and others using a different group. Indeed, as mentioned in the introductory chapter, many cells in the MEC are spatially stable, but without clear patterns, implying they contain some unknown information about the environment; and many cells in the LEC encode complex histories of objects in the environment. Given their proximity in the brain, all these cells are likely strongly connected (functionally, at least) and perform computations on similar data: perhaps some of these cells are grid cells, but never represent the animal’s self-location and thus do not present a neat pattern to the experimenter: maybe what is needed is a simple form of “bionic autoencoder”, as described in the introduction chapter.

It may also be the case that the posited calculations are only performed by grid cells during the period of familiarization: this is when grid patterns are least clear (from the experimenter’s perspective, i.e. the cells could in principle be doing something different) and when the HPC has not yet had an opportunity to learn complex relationships about the space$^{17}$. Perhaps, once the grid pattern is formed the EC is no longer required for the posited calculations, or is not used for tracking moving objects as tested here.

$^{17}$ This does not provide an interpretation for the gradual changes in scale/skew/phase that occur in novelty, when a grid pattern has become visible to the experimenter but has not fully stabilized (Barry et al. 2012; Krupic et al. 2015; Carpenter and Barry 2016; T. Stensola et al. 2015).
6. Grid cells: dealing with uncertainty

I performed all data collection and analysis presented in this chapter. However in some cases the drive construction and/or surgical implantation was performed by someone else.

Introduction

As discussed in earlier chapters, it is not known what purpose grid cells serve, it is also not known exactly how the grid pattern is generated and stably maintained from minute to minute and trial to trial. The current consensus is that each grid module corresponds to an attractor network that stably represents a bump of activity on the surface of a (twisted) torus. As with a single idealised grid cell, there are four parameters that define the mapping from the real world environment onto an idealised grid module: scale, orientation, and offset in 2D. Soon after entering a new environment, each of these parameters must be given values, and those choices adhered to, otherwise the pattern will be irregular and/or unstable.

A priori, scale would seem to be the simplest parameter to set robustly because it could be tightly coupled to the animal’s idiothetic sense of movement rather than being specifically selected for individual environments. This, however, does not appear to be the case: not only is the scale of the module not fixed universally, but it even varies over time within a given environment, starting with a long length and then gradually shrinking down to the specific length observed in a familiar environment (Barry et al. 2012). It can also be artificially manipulated by stretching/squashing the dimensions of an otherwise familiar arena (Barry et al. 2007).

Given that both the mechanism and purpose of grid cells remain largely unknown, we can only guess at the process underlying this change in scale and what its purpose might be. Indeed, it is not immediately clear that there is any advantage in allowing the grid scale to vary: regions upstream of the EC that convey scale information from the real world will presumably require a more complex set of connections, as will regions downstream of the EC that wish to utilize information about scale; further, if the grid pattern in a single environment changes over time then any learning that occurs early on in this environment will seemingly become invalidated.

One proposed advantage of changing the scale is that it allows the EC to match the precision of spatial representation to the current level of spatial uncertainty (Towse et al. 2014): perhaps
on initial exposure to an arena the animal is “aware” (Yu and Dayan 2005) that it can be less accurate in locating itself in reference to the available cues, and thus it reduces the precision of the grid system, reassigning the bump of activity in each module to a larger region in space. If this were not done, then the smallest grid module would be almost entirely noise rather than signal. When a module is unavailable to the grid system, it is thought that the total uniquely addressable range is reduced, as is the system’s capacity for performing error correction (Fiete, Burak, and Brookings 2008).

This notion of matching precision to certainty is what I explore here, however one can imagine other hypotheses:

- The changing grid pattern in novel environments may drive the newly forming place cell representation into a new region of the hippocampal state space, i.e. it ensures the hippocampus fully remaps (Barry et al. 2012). However, while it is clearly desirable for the hippocampus to be able to separate the representations it is unlikely that a fully-functioning grid system would be the best candidate for this task, especially as the grid system itself would have to be informed of the need to remap in the novel environment.

- The slow contraction of the pattern may allow the final representation in the given environment to be more regular and/or stable, similar to how the annealing process in a cooling metal creates a highly regular ionic lattice. Evidence suggests that the grid pattern can indeed become more regular over time (Carpenter et al. 2015; Barry et al. 2012), but it can also become *irregular* and “customized” to the quirks of a particular arena (Krupic et al. 2015; T. Stensola et al. 2015). One problem with this argument is that in novel environments, the pattern will sometimes be highly regular and stable, and other times irregular but still stable, thus it is not clear that regularity or experience are required to ensure stability as proposed.

- A related proposal is that the EC holds “templates” for various types of environment which it can draw on in an unfamiliar location. “Out of the box” a template would fulfil some of the functions of a proper grid system; but only after customizations are made for a given arena would the grid system truly maximize its functional potential. “Bootstrapping” the representation of novel locations in this manner would be advantageous if there is a high degree of commonality across locations, which there likely is (for rats, consider the repeating trees in a forest, or meanders of a small stream; for humans, the rectangular rooms of a building, the layout of pedestrian road crossings, etc.). Indeed this concept could be more formally described as being a
Bayesian approach, starting with a prior formulated from previous experience of the larger world, and refining it over time as more details emerge about the specific location. However, this description offers no explanation for the specific change in scale observed or how remapping occurs. That is unless one considers that the small rectangular home cage is the only available template available to most laboratory animals.

- The larger grid scale in unfamiliar environments may translate into behaviour occurring over larger scales.\(^{18}\) For example, when encountering a poorly explored arena, the animal must cover long distances if it is to successfully locate valuable resources, whereas on familiar territory, prior knowledge of the environment should be used to minimize the path length needed to obtain resources (Davis, Emlen, and Stokes 1948). Despite this proposed optimality, the approach does not appear to be adopted by rats in the laboratory – for example, see Figure 40. Instead, perhaps it is not the distances traversed by the animal that are scaled but the sphere of its “situational awareness”, i.e. in unfamiliar areas it must constantly attend to a larger zone if it is to detect dangers and/or opportunities. This concept was explored to some extent in the previous chapter, and little evidence was found in its support.

- It may be that the rat is capable of locating itself accurately early on, but the extra precision is of limited value until a larger number of salient events have occurred in the arena, i.e. distinguishing between many episodic memories may require a fine-grained spatial representation. Conversely, having a coarser representation initially may allow memories to be annotated to wider regions of space, thus “flooding” the environment with experiential data faster than would be possible at a finer scale. This hypothesis could be tested in an experiment where rats have to be very precise in novel environments: for example the rat could be trained to dig for food buried in sand, having previously been briefly shown the location, but with spatial errors in digging penalised with an electric shock. This would presumably give the experimenter direct control over the precision needed in the hippocampal map. And so, if a finer scale grid is induced in these conditions it suggests that required precision is the controlling factor rather than certainty. However, one could also argue that the

\(^{18}\) One example of a known spatial uncertainty being translated into behaviour occurs in some insects when performing path integration: they seem to bias their return trajectory to intersect the familiar territory of the outward journey rather than aiming directly toward home and potentially missing familiar territory entirely (Etienne and Jeffery 2004).
animal may be attending to the environment more acutely in this experiment to expedite the familiarization process.

As with scale, a priori, one might expect the choice of orientation to be fixed in an environment-independent manner, being defined relative to certain generic features of an environment, such as its most prominent axis of symmetry or dominating boundary. Indeed there does seem to be some relation between environment geometry and orientation (Krupic et al. 2015; T. Stensola et al. 2015). However, here I present data from two circular arenas, neither of which has a well-defined primary axes of symmetry or dominating straight boundary, and thus I will not discuss orientation any further.

The final parameters, 2D offset, could potentially also be defined by features of the environment.
Methods

See main methods chapter for details of electrodes, surgery, recording, animal handling, and basic data processing.

Subjects and rich/poor protocol
The main part of the experiment consisted of a foraging task in a circular arena (diameter 1.2m, 50cm high beige walls) within a circular curtained environment (blackout curtains, diameter 2.4m), with a single brightly illuminated cue card hanging immediately in front of the curtains (60cm square white card), no other illumination was provided. On alternate trials the arena was either furnished with “rich” local cues, or it was cue “poor”, i.e. devoid of obvious cues. The layout of the rich cues was nominally the same each time, and consisted of several different textured sections of flooring arranged in a pseudo-random pattern plus several features on the walls – see Figure 37 for a diagram and further details, note in particular that the cues did not act as a barrier to the animal’s movement. The floor (including textured cues) was wiped down with detergent before the start of each trial. Note that the square “baseline” environment was similar to the cue rich/poor environment in terms of total area, wall height, floor colour and texture, and dim ambient lighting – see main methods chapter for details. Trials were separated by intervals of approximately ten minutes, during which time the animal was placed in an open transport box. Each experimental day began and ended with a trial in the baseline arena; for the first day there were six trials in the circular arena, and on subsequent days there were four trials (unless cells were never found or disappeared early on, in which case the experiment was cut short after ensuring equal exposure to the two test environments). The order of rich and poor trials was fixed for each animal, but counterbalanced across animals (for the animals in the main analysis, four experienced rich first and three experienced poor first).

Figure 37. The rich/poor experimental protocol consisting of a 20 minute foraging task in three different arenas. See text for explanation. Note that the cue rich arena depicted here is a literal representation of the cues used: i.e. there was indeed a large blue rectangle section of flooring
(smooth plastic) with various other irregularly shaped textured segments on top (ribbed rubber matting, gaffer tape, black plastic, rough marble-effect linoleum) and there were indeed three wall cues (two plastic columns the same height as the walls, one of which protruded 3cm into the environment, the other of which was flat; and a rectangle of white card positioned half-way up the wall, with a pair of black markings

**Boosting familiarization of rich/poor**

In an effort to quickly familiarize it with the two environments, one animal, R2241, was exposed to both environments for six additional days in between the primary recording days. These six days occurred as two blocks of three: the first coming after day three of recording, and the second block after day five of recording. A custom-built overhead automatic food pellet dispenser was used during this “boosting” period to reduce the work-load on the experimenter, who was then able to observe from a distance. Every effort was made to counter balance the experience in the two environments during this process. Note that this data is included in the main analysis, using the recording days as the “day” label rather than an absolute measure of exposure time.
Results

Eight rats foraged for rice in a circular arena for several 20 minute trials each day, over multiple days. Each trial, the arena was either furnished with “rich” local cues or was cue “poor”, i.e. devoid of local cues. Individual rats experienced alternating rich/poor trials in a fixed order, with five rats visiting rich first each day and three rats visiting poor first.
Selecting grid cells for analysis

As shown in Figure 38, the grid cells from most animals distinguished between the rich and poor environments, i.e. they clearly remapped as determined from a Pearson correlation of ratemaps. However, all eight of the grid cells for rat R2175 failed to make the distinction (mean Pearson R for rich/poor ratemaps is between 0.35 and 0.79). It was concluded that this animal was not treating the two versions of the area as different locations (i.e. not remapping) and the animal was excluded from further analyses. Note that it is possible that this purported lack of grid remapping is actually a “false negative”, with significant changes having actually occurred in one or more of the unobserved grid modules.

Figure 38. Checking for remapping between rich and poor. For all grid cells, the Pearson correlation is computed between paired rich and poor trials: the mean R value is taken within day and then across days. The eight colours correspond to eight rats: the black rat is R2175 and is excluded from all further analysis due to showing a failure to remap.

Putative grid cells were identified on baseline trials, which occurred at the start and (usually) at the end of the day. Gridness scores were computed for these cells, and if the value crossed a threshold of 0.2 on at least one baseline trial, the cell was classed as a true grid cell for the given day. As discussed in the main methods chapter, the gridness (and scale) values were computed three times using spatially jittered/rotated versions of the position data. Figure 39 compares the values obtained from these multiple iterations: as many investigators know, both gridness and scale can be “brittle”, i.e. there can be significant variation in value due to only minor change. Where scale data had an SEM > 2cm, it was discarded, otherwise the median value was used; the median gridness value was used in all cases. A total of 74 grid cells were
successfully recorded on pairs of rich/poor trials (counts per rat: 3, 3, 8, 8, 10, 15, 27), some examples are shown in Figure 42.

![Figure 39](image)

**Figure 39.** The brittleness of gridness and scale measurement: each dot shows, for a single cell on a single trial, the variability under spatial jitters/rotations of the data. (a): the range (max-min) of gridness scores; dotted line marks a gridness range of 0.1, with 83.7% of dots lying below this line. (b): the range of scales; here the dotted line marks a scale range of 2.5cm, with 86.2% of dots lying below this line. For baseline trials all putative grid cells are included, but for rich and poor trials only cells passing the daily gridness criteria are used.
**Behaviour**

In the rich environment, although never quantified, it appeared that some rats had a tendency to examine one or more of the three wall cues at the start of a trial. Occasionally a rat would do so mid trial, including in some cases trying to clamber up the single wall cue that could almost function as a ladder (though such behaviour was rare). In addition, the features on the floor of the cue rich arena appeared to attract some attention, with animals digging their noses into the taped edges of the textured patches. There was no noticeable difference in the frequency of rearing between the two environments.

As shown in Figure 41, to quantify behavioural differences, I examined the total area covered; the extent of directional stereotyping; and the variation in the shape of the speed distribution. No significant difference was observed in the area covered; but directional variance was lower in rich, meaning stereotyping was slightly greater. Additionally, the speed profiles in rich and poor also differed somewhat, with animals spending slightly less time stationary in the rich arena, but rarely reaching the higher speeds encountered in the poor arena. As is often recognised, the mean speed of animals tended to be lower at the start of the experiment and then increase over time Figure 40. All these numerical analyses support the observation that animals attended to the cues in the rich arena, allowing their behaviour to be guided by the locations of cues on the walls and textures on the floor. While these behavioural differences make interpretation of single unit results more complex, they demonstrate that the two environments were being treated in quite different ways, as originally desired.

![Figure 40. Mean speed changes over time: daily mean in rich (solid lines) and poor (dashed lines).](image)
Figure 41. Behavioural differences in rich and poor. Each plot shows the data for all animals on all rich and poor trials. (a) area covered and (b) directional variance are explained in the main methods chapter; (c) “active fraction” is the proportion of the trial spent running at a speed greater than 5cm/s; (d), (e) and (f) show the lower, upper and mean speed. Inset bars show the mean in rich and poor, and significance refers to a paired t-test between consecutive rich/poor trials.
Figure 42. A selection of grid cells from various animals and days of the experiment. Each row corresponds to one cell on a specific day, as stated on the left of the row. Each trial shows the ratemap, spatial autocorrelogram, grid scale and gridness score. All ratemaps use the same scale, from 0 to 20Hz. The gridness scores and grid scale values quoted are the median values taken over all jitters/rotations of the data.
Stability

Figure 43 shows an initial analysis of stability differences between rich and poor arenas. The rich arena shows a clear improvement in stability, with significance overall (paired t-test: p=0.031) and for some individual days. While one animal shows a positive correlation between faster speeds in poor and higher stability, the remaining animals do not and there is no overall correlation. Thus it seems there is no particular contribution from speed to the difference in stability.

In the past, it has been suggested that place and grid cell representations are anchored to the boundaries of an environment, and errors accumulated while traversing the centre of an arena are discarded in favour of a “reset” signal that dominates when in proximity to a boundary (O’Keefe and Burgess 1996; Hardcastle, Ganguli, and Giocomo 2015; Barry et al. 2007). If the presence of cues on the floor and walls of an arena can improve stability of the pattern, as shown here, then it suggests that this resetting and anchoring may not be exclusively controlled by the boundaries. Indeed, if we repeat the stability analysis, but now using only the interior part of the arena the difference between rich and poor is even more stark (when judged by the significance of t-tests for each day), see Figure 44. Note that strictly speaking, this effect could be due to the presence of the wall cues, which are clearly visible from the arena interior, rather than due to the floor texturing. And, in principle, the extra cues could also be improving stability of the pattern without providing an instantaneous “reset” mechanism, for example they could simply contribute an increased precision or amplification of the reset signal used at the boundaries.

In an attempt to ascertain the exact nature of the instability I performed two further analyses, as described in the main methods chapter. The first analysis questioned whether the instability was due to (translational) drift of the pattern on the timescale of individual runs across the arena. For each run through the fields of a grid cell, I computed a maximum likelihood estimate of the distance the pattern had drifted from its long-term anchoring. The second analysis questioned whether the instability was due to a warping of the pattern from one half of the trial to the next. Here, I obtained a warp vector for each spatial bin in the environment, where the vector gives the optimal shift needed to match up the ratemaps from the first and second halves of the trial at that location. I then averaged over the warp map to compute a single, scalar, warp metric for the cell on that trial.

Figure 45 shows the extent of pattern drift on individual runs. As expected, there is a greater drift in poor, with significant differences on all but the first day, and significance overall.
Also, as shown in Figure 46, there was a tendency for the warp magnitude to be larger in the poor environment, but the difference was not significant overall.

However, it is not clear that we really gain any insight from these additional analyses. In particular, the warp metric will presumably be very sensitive to the signal-to-noise ratio: wherever noise has a magnitude close to that of the signal, the warp magnitude will tend towards the intrinsic “scale” of the noise rather than indicating true shifts in the signal. And, as demonstrated in the lower central plot of Figure 45, the drift metric is sensitive to speed, which we know differs significantly between rich and poor. To try and reduce the effect of speed differences, I repeated the drift analysis, this time matching the distribution of run mean speeds between pairs of trials – the results shown in Figure 47, still show some significance, though less than the original version, but they also continue to show a weak correlation with speed.

As with the earlier basic analysis on stability, we can also examine whether there is a larger drift on the interior of the arena as compared to the periphery, near the walls. In agreement with Hardcastle et al. (2015), Figure 48 shows that in the cue poor environment, runs across grid fields in the interior of the arena show slightly greater drift than runs across fields in the periphery, but this difference largely disappears in the cue rich arena.

Figure 43. Stability. [Next page] All plots show the same underlying stability data, obtained by computing Pearson correlations between the ratemaps from the first and second half of each trial. The legend in the bottom right of the figure applies to all plots: specifically the colours indicate animal identity and solid/dashed lines indicate rich/poor respectively. (a): stability is greater in the rich arena; for each cell the daily mean stability has been computed for rich and poor, and the absolute difference taken (poor-rich), the plot then shows the mean and SEM over all cells for each day. Significance markings refer to t-tests within day, with no correction made for multiple testing. The first inset shows, for each day, the number of cells from each rat. The second inset shows the mean stability in rich and poor, with the significance referring to the result of a pair t-test; see (d) for underlying data. (b): similar to (a), but here showing the values for rich and poor rather than their difference. Values are normalised to the mean on day seven. (c): similar to (b), but with data broken down by animal. (d): each dot shows the mean value for one cell, t-test shown in inset of (a). (e): plot of difference in trial mean speed for paired rich/poor trials, and difference in stability for the same trials. Each dot corresponds to a cell on a single trial pair; lines indicate where significant correlations exist within animal (in this case there is a positive correlation for R2124 but not for any other animal, and there is no overall correlation). (f): a combination of (a) and (c), showing difference values broken down by animal.
Figure caption on previous page.
Figure 44. Stability of interior 50% of the arena. All plots have the same meaning as in Figure 43, except Pearson correlations are conducted on only the interior of the environment.
Figure 45. Maximum likelihood estimate of the distance of drift, averaged over runs. Plots have a similar meaning to that in Figure 43. The strong bright blue line in (e) shows a significant regression on all the data (Pearson R=0.18, p=0.0014; in previous plots no such significant line existed, the other two lines indicate separate correlations for R2149 and R2112).
Figure 46. Average “warp” distance to transform ratemap from second half of trial back onto the first half of the trial. Plots have a similar meaning to that in Figure 43.
Figure 47. MLE of the distance of drift, with resampling of runs. Panes have the same meaning as in Figure 45 (see also Figure 43), except that the distribution of mean speeds for runs is matched across paired rich and poor trials. Pearson correlation in (e): R=0.14, p=0.02.
Figure 48. MLE of the distance of drift as a function of distance from the arena wall. The two plots aggregate over all runs from all animals: data is binned into 4cm wide bins, with the central black line giving the median, the inner grey area giving the inter quartile range, and the outer grey area extending out to the 5th and 95th percentiles. There are 105,789 runs in poor and 94,764 runs in rich.
Gridness
As found by previous experimenters (Barry et al. 2012), the hexagonal regularity of the spatial pattern (“gridness”) increased somewhat over time – see Figure 49b. Initially it was believed that the gridness was higher in the rich environment, but one animal in particular (R2241) provided a strong counter example – see Figure 49b and first cell in Figure 42. It may be that although in the rich environment the pattern was more stable, it was also more susceptible to deformations (Krupic et al. 2015; T. Stensola et al. 2015) caused by perceived discontinuities or symmetries in the arena; indeed it has been shown in the past that border cells are sometimes active along non-obstacle “boundaries” (Lever et al. 2009), so perhaps deformations can occur along such non-obstacle boundaries. Correcting for ellipticity in the grid pattern improved gridness scores in both rich and poor, but did not separate the two conditions (data not shown).
Figure 49. Gridness. Plots have the same meaning as in Figure 43, but now for gridness. Note that in (a) here there are no days with significant rich/poor differences.
**Scale**

The original motivation for undertaking this experiment was to establish whether animals would use a smaller scale grid when they had greater certainty of self-location. The increase in stability observed in the rich environment, relative to the poor, suggests that the extra cues did aid the hippocampal system in some fashion, although I cannot claim that “certainty”, per se, was improved.

Figure 50 compares grid scale in rich and poor over time. As previously identified (Barry et al. 2012), the scale shrunk with experience, particularly in the first couple of days. However, contrary to prediction, the scale in the poor environment was initially smaller than that in the rich; over time there was a significant trend for this pattern to reverse, but a smaller relative scale in the rich never reached significance.
Figure 50. Scale. Plots have the same meaning as in Figure 43, but now for scale, and with scale differences expressed as a percentage of the rich/poor mean rather than as a raw value. Note that in (a) here there is a significant correlation in the daily mean values (Pearson $R=0.79$, $p=0.04$).

Extrapolating the regression out beyond seven days, initially suggested that significance could be achieved if animals were exposed to the two arenas for a much longer period. To this end,
one animal (R2241) was put through a “boosting” protocol, to rapidly make it ultra-familiar with the arenas (see methods chapter). This, however, had no discernible effect, and I thus simply included the data in the main analyses already presented.

As previously noted, scale measurements can be brittle, especially where the grid pattern is irregular. I thus performed two additional analyses, with the hope of obtaining more accurate estimates of scale, and ultimately seeking to better distinguish between the patterns in rich and poor.

The first method, as described in the main methods chapter, used Delaunay triangulation to find the peak-to-peak distances directly on the ratemap, rather than using the distance to the inner six peaks on the spatial autocorrelogram. There were a large number of steps and parameters used in this algorithm, but as shown in Figure 51 the results show a reasonably close agreement with the spatial autocorrelogram (SAC) method. And, looking at a random selection of the 9.3% of the data having a difference greater than 5cm, it seems the values from the Delaunay method are on the whole still close to what a human might have manually recorded.

Figure 51. Comparing scale measurements. Left plot shows 569 scale values obtained using both the primary spatial autocorrelogram method (with jitters/rotations) and with the Delaunay method. The dashed black lines mark ±5cm around the diagonal, with 9.3% of points lying outside this region. Right: 6 examples of cells (from six different animals) that lie outside the central diagonal of the left plot. See Figure 12 for a full explanation of the annotated ratemaps.
Figure S2. Scale as measured with Delaunay triangulation on ratemaps. Plots have a similar meaning to that in Figure 50 (see also Figure 43), but note that slightly less data is available in this analysis due to the more stringent requirements of the algorithm.
Figure 52 is equivalent to Figure 50, showing the results for scale, but now using the Delaunay method rather than the SAC method. Note that slightly less data is available in this analysis due to the more stringent requirements of the algorithm. As might be expected from the fact that there is some agreement between the two methods, the general picture of the results looks similar: in the first few days scale is somewhat larger in the rich, though never significantly so; in the final two days, the scale is smaller in the rich, with significance obtained on the penultimate day, when four of the six animals show this pattern.

The second approach was to form a composite spatial autocorrelogram from a subset of the multiple grid cells recorded each trial – see the main methods chapter for details on the algorithm. Figure 53 shows the results for all animals, though note that even more data is being discarded than in the previous scale analysis. Of the five animals with composite scale available on the final day of the experiment, four show a smaller scale in rich and one shows a smaller scale in poor. As with the previous plots, there is an observable shrinking in scale over time, as well as a possible tendency for larger scale in rich at the start of the experiment (four of six have a larger scale in rich on the first available day of data).

![Figure 53. Composite scale. This plot is roughly equivalent to pane (c) in previous analyses figures, but here I have aggregated spatial autocorrelograms within animal for each trial before computing scale. Specifically, the plot shows the mean scale in each environment type each day, with error bars giving the maximum and minimum (from the 1, 2 or 3 available trials of that kind on the given day).](image-url)
Discussion

I sought to vary the spatial precision or “certainty” available to rats foraging in an arena, by varying the number of spatial cues, to test whether grid scale would adapt in line with this variation (Towse et al. 2014). I also controlled for the amount of exposure to the cue-rich and cue-poor environments, so that comparisons could be made independently of effects of environmental novelty on grid scale (Barry et al. 2012). The extra cues provided in the cue “rich” arena resulted in more stable spatial patterns on almost all days of the experiment, with particularly significant contrast observed in the central part of the arena. While it is possible that the measured instability was due in part to a more pronounced gradual warping of the pattern during cue-poor trials, the evidence suggests that the primary difference was due to the greater magnitude of the wobble around the long-term anchored position of the grid pattern.

Strictly speaking, this “wobble” is not necessarily “path integration error”, it could also be due to instability or imprecision of the anchoring of the pattern to the environment, what might be called a “reset error”. Hardcastle et al. (2015) addressed path integration error more directly, by calculating the “wobble” as a function of distance/time since leaving the boundary. It was concluded that since the two variables were indeed correlated, path integration error was accumulating while the animal was away from the boundary and resetting when in closer proximity. Our result can thus be seen to extend this finding, suggesting that non-boundary cues also play a role in “resetting”. Note however, that if the environment provides a constant level of weak spatial information (as in the cue rich situation), then there may never be a need for a binary “reset” operation to occur at the boundary; instead, the extrinsic and intrinsic signals could be continuously combined in a harmonious fashion, such as is done in a Kalman filter. Indeed, looking beyond grid cells, there is evidence in the literature that non-boundary cues do have a role in spatial cognition more generally (Cheng and Newcombe 2005; Benhamou and Poucet 1998; Scaplen et al. 2014).

Having concluded that the experimental protocol did indeed vary the spatial certainty/precision available to the animal, I next examined the scale. The results here were less conclusive: all three methods agreed that scales contracted with experience of the environments, as previously identified (Barry et al. 2012), however there was no clear sign that scale was smaller in the rich environment, as originally predicted. The null, or indeed negative result, was most prominent at the beginning of the experiment; by the latter days there was a suggestion that the prediction might have some merit. Thus it is possible that there was an interaction between novelty and cue-density, such that the cue-rich arena was initially
experienced as more novel than the cue-poor arena, potentially leading to a confounding additional expansion of grid scale early on in the cue-rich trials.

Should we regard this null/negative result as a disproof of the hypothesis? Perhaps, but final judgement might better be withheld until additional evidence is collected. While the protocol employed here did evoke measurable changes in behaviour and differences in grid cell metrics, the extent of the physical differences between the two types of arena was not as marked as it might have been: had the arena been larger, the central region would have been that much poorer in the “poor” setup, the measurement of scale would also have been more precise. Alternatively, the poor setup could have had its visual, olfactory and auditory cues removed/reduced by performing experiments in the dark with loud white noise and a strong aroma on the floor, even removing the boundaries entirely as done by Kate Jeffery’s lab, see Barry et al. (2006). Another possibility, though one not yet technically proven, would be to perform the experiment in a virtual environment in which space and cues could be artificially manipulated to create known levels of imprecision.

A less drastic change to the experiment would have been to familiarize the animals with the type of cues found in the cue rich arena before exposing them to the particular layout used. This would then have removed any initial effects due to the extra novelty of the cue rich arena as compared to the cue poor arena, and it would have provided animals with the chance to learn that the cues are stable and can be used as localization aids.

One potential problem with the setup was that if the cue rich arrangement was not perfectly reconstructed each trial, there was a possibility that the animals would recognise this and then subsequently treat the cues with contempt rather than see them as helpful, or experience increased novelty which itself might lead to grid expansion or remapping. There was no specific behavioural evidence for a sudden change of this kind, however (as already noted) animals often examined the cues closely on re-entering the rich arena, as though they were assessing the exact changes that had occurred (Eichenbaum, Otto, and Cohen 1992). Certainly, there is evidence from past cue-manipulation experiments that animals learn to ignore unreliable cues (Shapiro, Tanila, and Eichenbaum 1997; Manns and Eichenbaum 2009; Scaplen et al. 2014). Perhaps R2199, the animal showing the most significantly negative result, was more observant than the other animals, or the associated experiments were performed less precisely, and thus the animal was more reluctant to use the cues in the rich environment. This problem could be solved either by having two separate arenas that are never dismantled, or using virtual reality. The disadvantage of using separate arenas is that there is a possibility that other uncontrolled factors would become relevant.
In conclusion, non-boundary cues are relevant to the dynamics of the grid pattern, but their precise influence on the mechanisms generating grid cell firing remains an open question.
7. Discussion

The overarching purpose of this thesis was to explore the nature and purpose of the grid cell code, and its role within the larger spatial cognition and memory systems. In particular I asked how grid cells might evaluate simple operations on 2D location and translation vectors, and whether the grid code needs to reflect the quantity of spatial information present in the environment. Building on the introductory Chapter 1, Chapter 4 provided a theoretical framework for understanding the grid code as well as discussing some aspects of how it might be generated. It then progressed to ask what role the grid system may be fulfilling, with special emphasis on a model for solving the 2D large-scale navigation problem. Chapters 5 and 6 presented new experimental data, with the former addressing a specific case of 2D vector computation, and the latter examining the role of spatial information (or “certainty”) on the properties of the grid code. A separate passage, Chapter 3, was focused entirely on the development of a piece of software, which was of great practical benefit to my data collection effort as well as to that of my colleagues.

In this final chapter, I review and draw together the conclusions of the aforementioned sections. Open questions are noted and examined, in the wider context of the field.
Computing with 2D vectors

The last section of Chapter 1 proposed that grid cells are a spatially-specialised apparatus built on top of a general purpose memory system (located in the hippocampus proper). Then, at the start of Chapter 4, I considered the potential generic benefits of having a specialised apparatus. These benefits, which can broadly be described as optimising the precision and energy/time/space requirements of an algorithm, were noted as being non-critical. For example, place cells might also be able to guide navigation (although perhaps navigation over very large scales eventually does require the compactness of the grid code to work in practice). And indeed previous researchers have found only limited behavioural deficits when the entorhinal cortex is lesioned (Winter et al. 2013; Parron, Poucet, and Save 2004; Parron and Save 2004; Miller and Best 1980; Parron, Poucet, and Save 2006; Steffenach et al. 2005; Remondes and Schuman 2004). Similarly, recording from place cells in animals with a lesioned/disrupted EC, appears to demonstrate that the place cell code does not rely on the existence of the grid code (Brun, Leutgeb, et al. 2008; Van Cauter, Poucet, and Save 2008; Miller and Best 1980; Brandon et al. 2014; Koenig et al. 2011); a fact that can also be observed during development (Wills, Barry, and Cacucci 2012).

This led me to try and identify the algorithms that the grid system is optimised to perform, and I focused on computing with vectors in 2D. Drawing on work that my colleagues and I have already published (Bush et al. 2015), I cited some prior proposals that addressed navigation (Muller, Stead, and Pach 1996; Neil Burgess and O’Keefe 1996; Erdem and Hasselmo 2012; Kubie and Fenton 2012; Erdem and Hasselmo 2014), and I described the four specific algorithms my colleagues and I considered. These algorithms vary in their strengths and weaknesses, with the “rate coded vector cell model”, proposed by me, offering fast (perhaps on the scale of a single theta cycle) vector addition, with minimal requirements for additional cell types, and no environment-specific learning. The two main drawbacks of the proposal are its requirement for multiplicative (i.e. logical AND) synapses and the need for a complex network architecture to be lain down during development. Whether either is realistic is still a question for debate: London and Häusser (2005) are optimistic about the possibility of complex computation in dendrites, but note that - if needed - their hypothesised functionally could also be achieved using a multi-level network composed of simplistic neurons;19 and in

---

19 Containing as much computation as possible within the dendrites is in keeping with the notion that grid cells are specially optimised for a specific class of computations. Indeed if it transpires that dendritic computation is widely used by the brain but grid cells – or their allies - do not perform any, then it would be an argument against the grid cells being specialised for a particular algorithm.
terms of complex patterns of connectivity, I could cite the grid attractor (Fyhn et al. 2007; Yoon et al. 2013; Chen et al. 2016) itself as a prominent example, deferring again to London and Häusser (2005) for additional evidence. In addition, the idea of relatively sophisticated developmental mechanisms for organising neural connectivity are not implausible, perhaps benefitting from travelling waves of activity (Bruce L McNaughton et al. 2006).

Following the description of the rate coded vector cell model, evidence was presented of numerical simulations showing how the system could be used to navigate accurately within a rhomboid region of side 500m, with the virtual animal taking 2-4 iterations to home in on the target each trial.

Next, I introduced the inverse of the navigation problem, which was framed as the adding of a translation vector to a start location to get a second (goal) location. It was proposed that this problem could be solved in a similar manner to the navigation problem, meaning it could be performed by the grid system within a timeframe of perhaps 100ms. If the system were optimised to perform such a calculation, then incoming sensory information could be truly “indexed” by the location from which the information emanated, rather than at the location it was observed. However, I recognise that this “information storage at a distance” cannot be done in parallel for everything on the retina, or everything currently audible; rather, the storage process – as hypothesised – could only handle one item at a time, and thus would have to behave in a serial manner like foveation, directed attention or language production. Exactly how it is that potential spatial computation “requests” would compete for use of the grid hardware is not something I discuss here, but presumably some machinery in or around the hippocampus would have to enforce a winner-take-all filter at the point of input, to prevent multiple requests from corrupting each other during the computation.

In general, the question of using massively-parallel hardware to control a serially-behaving organism is one that will require further investigation in future. In computer science the difference and benefits between serial and parallel computing have gradually become more widely recognised (Asanovic et al. 2009), but it remains difficult for programmers and neuroscientist alike to think within the²⁰ parallel paradigm, in which efficiency is not measured just in terms of total work done, but also in total time taken.

In pursuit of evidence that the grid system was performing the vector addition computation, I designed an experimental paradigm in which a rat was exposed to a salient, spatially dynamic, 

---

²⁰ Of course there are in fact many variations on the concept of “parallel”, with each offering different pros and cons to the hardware designer and software engineer.
stimulus, operating in the same arena as the rat itself. That is, I programmed a small robot to chase the rat around an environment, while the rat performed a standard rice-foraging task, and I recorded its grid cell activity. The reasoning behind the paradigm’s design was that it would force the rat to constantly make truly 2D spatial observations and decisions, that is, unlike with a finite set of salient locations, it would never be able to resort to re-using stereotyped behaviour and zero-dimensional representations of space. Further, I recorded from grid cells rather than place cells because it was hypothesised that if the grid map really were involved in the posited computations (i.e. tracking the location of the robot in grid space), then all such computations would have to be funnelled through a single grid-system, rather than having multiple independent systems operating in isolation for different tasks (whereas individual place cells could likely be co-opted for representing an unpredictable multitude of things).

Although only limited preliminary data was collected from two rats, it was clear that the paradigm failed to produce the anticipated grid cell dynamics: there was no particular sign of an increase in out-of-field firing, which would have been the most basic prediction of the hypothesis. An increase in theta-skipping was detected, but its significance was difficult to interpret with the small amount of data available, and it is not clear that it had anything specifically to do with spatial computation in the grid cells.

In conclusion, although it is tempting to claim that the grid system is optimised for performing specific algorithms, such as vector addition, it remains to be seen whether it does indeed perform any such computations: further experiments exploring grid cell behaviour during navigation and in complex dynamic environments will be needed before the question can be resolved.
The significance of spatial information

In this section I explore the functional purpose of grid cells from a different perspective, asking how their pattern is influenced by the quantity of spatial information in the environment, and probing some of the mechanisms that might be involved.

In the past it has been suggested that it is specifically environment boundaries that anchor spatial representation in the hippocampus (Neil Burgess and O’Keefe 1996; Wang and Spelke 2002; Lever et al. 2009; Hardcastle, Ganguli, and Giocomo 2015), and that away from the boundaries the primary purpose of the grid system is simply to help update the place cell code by performing path integration (Bruce L McNaughton et al. 2006).

Somewhat at odds with this simplistic view of grid cells, is the observation that the scale of the grid pattern seems to shrink during the period in which an animal familiarises itself with a new environment (Barry et al. 2012); and worse, it is often highly irregular at the start of the familiarisation period, when the place cell code is most likely to benefit from an external path integrative mechanism. Indeed, given that grid cells in mice may be unable to perform pure path integration (i.e. losing regularity in the absence of visual environmental cues) (Chen et al. 2016) we might begin to wonder whether the interior-anchored-to-boundary hypothesis holds up at all. One of the hardest pieces of evidence to fit into this framework is that, even after changes in scale have taken effect, the pattern can still shift and warp over time (Krupic et al. 2015; T. Stensola et al. 2015; Carpenter et al. 2015), suggesting that either the mechanism or function of the grid cells adapts to some property of, or some experience in, the environment rather than providing a generic ready-for-use spatial metric.

Here, in Chapter 6, I presented data from a paradigm in which rats had access to visual and tactile cues in the interior of the arena, as well as on the boundary. We wondered whether the grid system would use a finer-scale metric to represent increased amounts of information within such an environment. Doing so, would show not only that the grid system was benefiting from the non-boundary cues, but also hint that it was adapting to known levels of uncertainty (Yu and Dayan 2005), supposing that spatial uncertainty reflects the number of available cues, so as to suitably match the metric coding to the level of certainty (Towse et al. 2014). Other hypotheses relating to changes in scale were also proposed, including that grid scale varies with novelty, and that grid expansion in a new environment might serve as a trigger for the formation of new mnemonic representations in the hippocampus by causing place cell remapping (Barry et al. 2012).
However, the data in this experiment were not conclusive with regard to grid scale – there not being a clear overall difference in scale between the cue-rich and cue-poor environments. Perhaps the time course of the grid scales in the two environments hints at an interaction between the factors of novelty and cue-related uncertainty so that enhanced novelty-related increase in the cue-rich arena confounded an overall decrease in the cue-poor arena. Examining these unexpected additional possibilities will require further experimentation.

One clear result that did arise from Chapter 6 was that the grid pattern was more stable in the cue-rich environment, indicating that non-boundary cues do play a role of some kind in spatial cognition. This increased stability should be of interest to future experimenters, who wish to record – and understand – the most ideal-looking grid patterns. It also serves as a reminder, that although the laboratory is a good place to create artificially simplistic paradigms for careful investigation, it is entirely possible that the most interesting features of neural activity thus end up being missed (Savelli, Yoganarasimha, and Knierim 2008). Instead, perhaps we would be better starting with large, complex, dynamic arenas and using better analytical tools to tease out information from the recorded neural signals.
Conclusions

The grid pattern identified in the entorhinal cortex is a fascinating example of neural computation in action. While it is clearly both functionally and mechanistically related to the space and memory apparatus of the hippocampus proper, it remains to be seen exactly what purpose it serves and how it fulfils this role.

The Waveform graphical interface, introduced in Chapter 3, and used by me as well as by others (Chen et al. 2016; Carpenter et al. 2015), was designed with the aim of widening the conceptual view of the researcher as he/she “peers down” the electrodes into the brain. It was hoped that by shifting the focus from the individual cell onto the population, the researcher would gradually come to better conceptualize the system as a massively parallel processing machine, and ultimately formulate novel hypotheses appropriate to such a paradigm. It was also hoped that by reducing the friction in the experimental pipeline, researchers would be keen to collect more data and take more risks experimentally, again with the ultimate goal of making new discoveries as to the function and mechanisms underlying the grid and place systems.

While it is true that several significant discoveries have been made using tetrode technology over the last four decades, one might also conclude that the technology (and experimental paradigms employed) is limiting: with enormous effort and resources, a scientist can use these techniques to record a handful of cells in a given region of the rodent brain, and with a bit of luck he/she may recognise the odd correlate in the outside world. However, if we are to continue to push our understanding we now have to turn to higher bandwidth, and higher throughput, techniques. Optogenetics, as well as pure genetic and pure imaging technologies, promise just such benefits (for example Dombeck et al. 2010), especially when used in conjunction with advanced processing pipelines, such as deep learning. Indeed, given the recent renaissance in the field of artificial neural networks, it may be that the next great breakthrough in understanding spatial cognition is of a more theoretical nature: it is only when we truly have a working knowledge of how networks can perform useful computation that we are likely to come up with the most meaningful hypotheses for the very special network inside the skull. Indeed, it may turn out that our current set of hypotheses prove to be severely misguided in their general quality.

However, let us now return specifically to the question of grid cells one final time. In this work I presented a model that demonstrated how grid cells might conceivably, in conjunction with a set of vector cells, perform navigation in 2D as well as perform other 2D vector calculations. However, despite preliminary attempts, I failed to provide convincing evidence
in favour of the generalised 2D computation hypothesis. This leaves us with the more vague claim, based on existing evidence in the literature, that grid cells are likely to be tuned for performing a non-critical, but highly optimised, algorithm of some kind. Whatever this algorithm might be, the work in the rich/poor paradigm shows that the system is incorporating multiple forms of spatial information, not simply that derived from the boundary/geometry of the environment. And theoretical considerations in Chapter 5 indicate that perhaps the optimally compact nature of the grid code (Sreenivasan and Fiete 2011) mean that its contributions to cognition will become most apparent when tested in very large scale environments.
Bibliography


Acknowledgements

Let’s not pretend. There were many times I thought I’d never get here. Many times I wished it would just go away. And I’m sure I’m not the only one.

But here we are, with a finished thesis!

Those who know me well may sense a lack of gratitude for the final product, and they would not be entirely wrong in doing so. However, I am extremely grateful to the people who gave of themselves along the way: to those who provided help and guidance, and to those who offered respite and moral support: each of your contributions were important to me, and I thank you for them.

Firstly, to my parents – I couldn’t have asked for a kinder or more stable home in which to grow up; I am always in awe of the love and care you show me. Tolman, the guy who first wrote about cognitive maps, would have approved: he concludes his 1948 article by saying that only by creating for ourselves, and for our children, a safe environment, free of unnecessary frustrations, can we hope to calmly understand the world and act well within it – something which I certainly aspire to do.

To my sister, Abi – although I have not seen a lot of you over the past few years, I have greatly enjoyed hearing all your exciting stories and have been inspired by your recent displays of determination and by your hard-won successes.

To Guifen – thank you for the 55 minutes of suffering you went through to finish that curry – you certainly made your point. Thank you also for everything that came before and everything that came after: I really am grateful for having had such a true, and wonderful, friend in my life.

To Ali – it was some time now since we were actually colleagues, but you have often been in my thoughts as “the one who escaped”, and I’m so glad that you did. I also have you to thank for my over-developed upper body strength!

Francis, Robin, and Freyja – not only are the three of you are exceptionally talented and (almost always) hardworking, but your willingness to share wisdom - both technical and otherwise – demonstrates a striking level of kindness.

To John, Steve and everyone else in the lab – thank you for accepting me into the fold of experimentalists. It certainly is a difficult job you have; I can only say that I have a huge amount of respect for those of you who have succeeded – and excel – at it.

To Neil - I am grateful that you responded to my original email way back in 2011, and for guiding me onto and through the CoMPLEX PhD scheme. Although I don’t see a future for myself in primary neuroscience research, I remain fascinated by the discoveries being made, and I will be forever impressed by the efficient and calmly intelligent manner in which you manage to deal with the realities of the process. I am so fortunate to have had an opportunity to be a small part of it.

And last, but certainly not least, Caswell. What can I say? I owe you so much – I doubt I could even recall half the ways you have contributed to my eventual submitting of this thesis: the pep-talks, the technical tuition, the long-term troubleshooting assistance, the casual intellectual discussion, the innumerable suggestions for analysis, and the surgical procedures performed for my benefit, to name but a few. Your unfaltering optimism, enthusiasm, and determination, are an inspiration to all. I have no idea how you manage it - I just hope some of it has rubbed off on me!