NTNU – Trondheim
Norwegian University of
Science and Technology

The Distribution of Spatial Phases of Grid Cells

Daniel Wennberg

Master of Science in Physics and Mathematics
Submission date: July 2015
Supervisor: Jon Andreas Støvneng, IFY
Co-supervisor: Yasser Roudi, KAVLI

Norwegian University of Science and Technology
Department of Physics

Abstract

Grid cells, found in the medial entorhinal cortex of mammalian brains, are among the most significant discoveries from the research into the brain's system for spatial navigation during the last decades. A lot of work has been done to uncover the details of their regular spatial firing pattern and how they operate together in permanent cohort across different environments. In particular, we now know that grid cells are organized in multiple largely independent modules, consisting of cells that respond coherently to new environments and have approximately equal spatial firing patterns, except for a spatial offset known as the *phase*.

A lot of work has also been done on the theoretical front to develop models for how grid cells are part of a network that brings about their characteristic behavior. Some of these models make predictions about the distribution of the phase within a grid cell module. In particular, models based on continuous attractor networks predict that the distribution must be uniform. To evaluate the merit of these and other models, we must therefore develop rigorous methods of analysis to find out what experimental data can tell us about grid phase distribution.

Here, we develop novel ways of analyzing the firing pattern and computing the spatial phase of a grid cell. To analyze the distribution of phases within a module, we employ methods from point process statistics that have not previously been used for this purpose. Our results indicate that there exist grid cell modules with phase distributions that deviate significantly from uniformity.

Sammendrag

Oppdagelsen av gitterceller i hjernen hos pattedyr er et av de største fremskrittene innen forskning på hjernens stedsans de siste tiårene. Det har blitt gjort et betydelig arbeid for å avdekke detaljene om gittercellenes regelmessige romlige aktivitetsmønster og hvordan de fungerer sammen i permanente enheter på tvers av ulike omgivelser. Vi vet nå at gittercellene er organisert i flere mer eller mindre uavhengige moduler, som består av celler som reagerer enhetlig på nye omgivelser og har mer eller mindre identiske aktivitetsmønstre, bortsett fra en romlig forskyvning som kalles *fasen*.

Mye arbeid har også blitt gjort på det teoretiske planet for å utvikle modeller for hvordan gittercellene er del av et nettverk som fremkaller den karakteristiske atferden deres. Noen av disse modellene inneholder prediksjoner om fordelingen av faser i en gittercellemodul. Spesielt forutsier modeller basert på kontinuerlige attraktornettverk at fordelingen av faser må være uniform. For å vurdere verdien av slike og andre modeller må vi derfor utvikle nøyaktige fremgangsmåter for å finne ut hva eksperimentelle data kan fortelle oss om fasefordelingen.

I denne oppgaven utvikler vi nye måter for å anlaysere aktivitetsmønsteret og regne ut den romlige fasen til en gittercelle. For å undersøke fordelingen av faser i en modul tar vi i bruk vertøy fra punktprosesstatistkk, som ikke har vært brukt til dette formålet tidligere. Resultatene våre indikerer at det finnes gittercellemoduler med fasefordelinger som avviker betydelig fra en uniform fordeling.

Acknowledgements

First of all I would like to extend a huge thank you to my supervisor, professor Yasser Roudi, for taking me into his group and giving me this exciting project to work on, as well as encouraging my general development as a student and scientist.

Next, I would like to thank Benjamin Dunn for introducing me to the Kavli Institute and the world of grid cells, for getting me started on the project and feeding me with data, and for our numerous engaging and enlightening discussions and his general excitement about the project. I would also like to thank Ziwei Huang for excellent company in the office.

I would like to thank Jon Andreas Støvneng for taking on the duty as main supervisor, enabling me to do this project as my master's thesis.

A special thanks goes to Eivind Rørvik for five years of mutual joy and despair while navigating courses, exercises, dinners and exams, and in particular for his patience with my questionable laboratory skills for semesters on end. Furthermore, without the joint effort of him, John Alfred Brennsæter and Henrik Fiskerstrand Gjørtz, I don't know if I would have managed to finish the writing of this thesis. A big thanks goes to all three!

Finally, a very special thank you goes to Liv Julie for always being my greatest fan.

Contents

List of figures

List of tables

1

Introduction

Finding one's place is not easy. Interpreted figuratively this statement is certainly true, as witnessed by virtually all of human art, literature and history, from Moses to Hamlet to Simba in Disney's *The Lion King*.

However, as any scout or satellite navigation engineer can testify,[1] the literal interpretation is also dead on. Finding your location with cross bearings requires skill and attention to detail, as well as a good map and unambiguous reference points on the horizon. Satellite navigation systems such as GPS are made extra complicated by technical details such as relativistic time dilation and ionosphere refraction, but regardless of these, a receiver needs to relate to at least four remote units whose positions are already exactly known in order to calculate its own position. In general, our ability to navigate with exact knowledge of our position through mountains, along roads or across oceans depends on someone already having undertaken the laborious work of creating a map or otherwise giving us references that we can relate our position to.

In light of this, our innate ability to effortlessly navigate our immediate surroundings, trivial as it seems, is quite remarkable from a scientific perspective. Decades of research into how mechanisms in the brain conspire to generate these abilities have taught us a great deal, but there is still a lot left to learn.

A presumably important piece of the puzzle is the grid cell, a kind of cell found in the brains of mammals that is active only at specific locations as you (or a rat) walk(s) around a room. The active locations form a pattern with high degree of regularity, and may play a part in

[1] This author happens to have experience as both.

representing or computing quantities related to location and movement in space.

While the existence of grid cells is an experimental fact, the knowledge how they are connected to each other and other cells such as to generate the patterns of activity displayed by grid cells is limited. However, several models have been developed to represent in a potentially plausible way the dynamics of a grid cell network.

The value of such a model is of course its ability to predict things that are later confirmed by experiment. For grid cells, a possible prediction is the distribution of the grid *phases*, that is, the distribution of offsets between the patterns of activity for two cells with otherwise very similar patterns. The purpose of this project is to develop rigorous ways of analyzing this distribution for experimental data, and to apply these methods on dataset and evaluate the results. The findings may thus have significant implications for the plausibility of grid cell models that predict something about the phase distribution.

2

Grid cells

The medial entorhinal cortex (MEC) and hippocampus are areas in the mammalian brain containing activity that is central to the brain's representation of external space [1–3]. The first evidence of this was the discovery of neurons in the hippocampus of rats that only fire when the animal visits a single specific location in its environment, the so-called *place cells* [4, 5]. More than 30 years later, a different spatially modulated activity pattern was discovered in neurons in the MEC. These neurons, known as *grid cells*, fire at multiple regularly spaced locations spread across the environment [6, 7].

A grid cell's *firing fields*, the areas where it is most active, form a periodic pattern variously described as a triangular, hexagonal or rhombic lattice [1, 7]. The characteristic feature of this pattern is that each firing field is surrounded by six other fields at approximately equal distance from the center field and each other, resembling the vertices of a tiling of the plane by equilateral triangles. The three principal axes that can be drawn through a field and its neighbors are thus spread out uniformly, approximately 60 degrees apart. Hence, the pattern is a subset of a triangular Bravais lattice, as shown in Figure A.1. See Appendix A for an introduction to Bravais lattices. The periodicity and triangularity of a grid cell firing pattern is readily observed, at least qualitatively, in Figure 2.1.

The firing pattern of a grid cell is anchored to visual cues in the environment, but is stable in the sense that it persists when light and other sensory stimuli are removed as well as across multiple exposures to the same environment, and it is independent of where in the environment the animal started its run [1, 7].

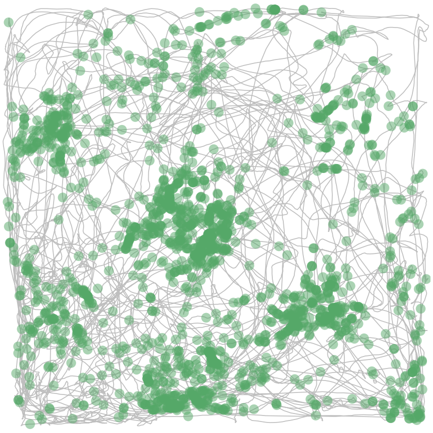


Figure 2.1: Spatial firing of a grid cell. The gray line shows the trajectory followed by a rat during an experimental session in a square environment, and the green dots show the locations at which an action potential was fired from a single cell in the MEC during this session. The presence of discrete regions of high activity organized in a regular pattern is evident.

To the degree that real-life grid cell firing patterns deviate from the ideal triangular lattice described above, they very closely approximate a lattice obtained by applying a *shear mapping* to a perfect triangular lattice, at least when the environment explored by the animal is a square [8]. A shear mapping translates all points along some axis (the *shear axis*), and each point is translated a distance proportional to the distance between the point and the shear axis. Shear mappings deform shapes, (for example, circles are mapped to ellipses), but since they only consist of parallel translations of points, they preserve area (the deformed ellipse will have the same area as the original circle). Grid cell firing patterns in square environments tend to be consistent with a lattice obtained by placing a perfect triangular lattice in the environment with one axis parallel to one of the walls in the environment, and applying a shear mapping along the perpendicular wall.[1]

[1] This is true for environments of the size considered in this work, but for larger environments the situation is more complex, see [8].

2.1 Modules

The MEC contains grid cells with firing patterns of different lattice spacings and orientations. The cells are organized in *modules* with similar spacing, orientation and deformation from perfect triangularity [9]. Cells in the same module are located topographically close to each other: the *dorsal* end of the MEC (towards the back of the rat) is dominated by modules with small spacings, while modules with larger spacings are located closer to the *ventral* end (towards the belly of the rat). Up to four distinct modules have been observed in a single rat. The modules operate as coherent and independent units, in that cells in the same module are always found to have similar spacing and orientation to each other, while the relative orientation between cells in different modules may change from one environment to the next.

Although the spacing and orientation is similar between cells in the same module, they can have completely different spatial phases. The spatial phase of a grid cell is the spatial offset of its firing pattern with respect to some reference. No evidence has been found against the assumption that the phases of cells in a module are uniformly distributed over the set of possible phases, and no correlation has been observed between topographical proximity and proximity in phase [7, 10]. However, the relative phases between cell pairs are always similar from one environment to the next [11].

Using methods that have not previously been applied in this context, we will here attempt to investigate further whether the distribution of phases within a module is uniform. This question is of particular importance when evaluating the plausibility of different theoretical models for the how grid cells function and interact with each other and the rest of the brain.

2.2 Grid cell models

The coherent response of whole modules of grid cells when moving to a different environment, despite substantial changes in the firing patterns of the cells, suggest that the intrinsic properties of the local circuit of neurons in the MEC play an important role in the forming of grid cell firing patterns [3]. Different theoretical models for this circuit have been suggested. Here we will briefly review two classes of models:

continuous attractor networks and *adaptation in feedforward networks*.

Continuous attractor networks

The main idea behind continuous attractor networks is that certain connectivity configurations in a network of neurons can lead to the emergence of a low-dimensional continuous manifold of stable activity patterns, known as a continuous attractor. In particular, if neurons considered “close” according to some metric are preferentially coupled and tend to fire in conjunction with each other, while more “distant” neurons are weakly coupled and tend to fire at different times, the continuous attractor will contain states where activity is confined to a group of “close” neurons, with the rest of the network inactive [12]. If the network is translationally invariant with respect to the chosen metric, there can be no intrinsic preference for the location of the bump of activity – it is free to move around in response to external input.

It has been shown that stable triangular firing patterns of similar orientation and spacing can emerge from continuous attractor networks when the metric is the relative phase between cells [1, 13, 14]. Such networks are therefore natural candidates for grid cell models. It is typically hypothesized that an external mechanism, based on cells that code for head direction and speed, drives the bump of activity around in the attractor, bringing about the spatial triangular firing pattern of individual cells. This mode of operation is known as *path integration*. The network must also receive input based on sensory stimuli, correcting for errors that accumulate through path integration and keeping the firing patterns anchored to external landmarks.

An important requirement for the plausibility of continuous attractor network models for grid cells is that the cells in a module must cover the set of available phases evenly [14]. This is necessary for the attractor to approximate a continuous and translationally invariant manifold through which the activity bump can translate freely. Given uniform phase coverage, a continuous attractor network predicts many of the characteristic properties of grid cell modules observed in experiments, such as similarity and spacing and orientation.

Adaptation in feedforward networks

In grid cell models based on adaptation in feedforward networks, the geometry of the firing pattern is developed in each grid cell individually in response to external input, presumably derived primarily from sensory stimulus [15]. In the model studied by Kropff and Treves in [15], the input is reminiscent of the firing of a collection of place cells. Because the connections are one-way, from the input units to the grid cells, this configuration is called a *feedforward* network, as opposed to a *recurrent* network such as in the continuous attractor model.

The grid cells in this model experience *adaptation* to the input signals: when a grid cell fires in response to an input unit, the connections from this particular unit to this particular grid cell is strengthened. After a period of firing, neuronal fatigue sets in and temporarily prevents the grid cell from firing in response to new inputs and thus strengthening other connections. Both theoretical arguments and simulations confirm that this adaptation process may result in grid cells acquiring a triangular firing pattern. The grid spacing will be similar for all cells receiving the same input, but the orientation is completely random. The model therefore requires additional dynamics to account for the orientation alignment within grid cell modules.

The properties of models based on adaptation are quite different from those of continuous attractor models. For example, path integration is not required – the grid pattern emerges using only inputs that can be derived from sensory stimulus. However, grid cells in this model may also be part of a system that performs and/or represents some form of path integration.

Most importantly for the purpose of work, adaptation models do *not* require uniform coverage of the spatial phase space of a module. In the general framework of adaptation models, phase is just at random as orientation, and the extra dynamics necessary for alignment of orientation may even contribute to the clustering of phases. For example, the mechanism for orientation alignment suggested in [15] consists of a connection between grid cells such that a cell B will tend to fire in response to a cell A, but only if the head of the animal is pointing in a particular direction. This will contribute to aligning the firing pattern orientations of the two cells at a small, but nonzero, phase offset. Depending on how such connections are arranged throughout the network, they may perhaps contribute to global clustering of phases.

As models of grid cells, both continuous attractor networks and models based on adaptation have strengths as well as weaknesses when compared to experimental data. However, they are not mutually exclusive, and more advanced models containing features from both are possible [3].

3

Point process statistics

Point process statistics is an area of study in the field of spatial statistics, which aims to study the geometrical properties of a set of points in space that was presumably generated by some random process. This chapter introduces some important definitions, results and tools from the theory of point processes, focusing on the elements that are used in other chapters of this text.

Unless otherwise noted, the reference for everything presented in this chapter is [16].

3.1 Notation

Here we list some definitions and notational conventions and that are useful for expressing the upcoming ideas.

- We will only consider point processes in two dimensions, also known as planar point processes, so all sets mentioned in the text are subsets of $\mathbb{R}^2$, and all points mentioned are elements of $\mathbb{R}^2$.
- *Minkowski addition:* For sets A and B, we define their Minkowski sum as
$$A \oplus B = \{x + y : x \in A, y \in B\} \ . \tag{3.1}$$
This operation involves both translation and enlargement of the sets.
- *Translation:* The addition operator is used to denote the translation of a set by a point. If A is a set and x is a point, we define
$$A + x = \{x + y : y \in A\} \ . \tag{3.2}$$

- *Reflection:* The reflection of a set A through the origin is denoted $\tilde{A}$ and defined as
$$\tilde{A} = \{-x : x \in A\} \ . \tag{3.3}$$

- *Probability:* We denote the probability for a proposition Q to be true as $P(Q)$.

- *Expectation:* We denote the expectation value (mean) of a stochastic variable X as $\boldsymbol{E}(X)$.

- *Indicator function:* The symbol **1** denotes the indicator function, which evaluates to 1 if the argument is true and 0 otherwise:
$$\mathbf{1}(Q) = \begin{cases} 1 & \text{if } Q, \\ 0 & \text{otherwise.} \end{cases} \tag{3.4}$$

- *Measure:* We denote the area of a set B as $\nu(B)$.

- *Disc:* We denote the ball of radius r around a point x as $b(x,r)$.

- *Origin:* The origin $(0,0)$ is sometimes referred to as o.

3.2 Fundamentals

A *point process* is a stochastic model for sets of points. A sample from a point process is called a *point pattern*, and is simply a set of points

$$N = \{x_1, x_2, ...\} \ , \tag{3.5}$$

embedded in some space.

The symbol N is used in two different ways:

- to denote a stochastic point pattern, as in Equation (3.5),

- as a *counting measure* on sets, that is, a function that takes a set as an argument and returns the number of points from the pattern that are elements of the set:
$$N(B) = \sum_{x \in N} \mathbf{1}\,(x \in B) \ . \tag{3.6}$$

The term *point process* may sometimes lead to confusion concerning whether some temporal dynamics or development over time is inherent in the stochastic model. This is not the case – a point process is only concerned with the geometric properties of the point patterns it generates. In particular, the numbering of points in Equation (3.5) is arbitrary and bears no significance. Occasionally, point processes are referred to as *point fields* to reduce the potential for misinterpretation.

A point process is formally defined by the probability distribution

$$P\left(N \in \mathscr{A}\right) \ , \tag{3.7}$$

where $\mathscr{A}$ is any set of point patterns. This definition is very general, but all the probability distributions one could possibly consider for point processes are contained in it as marginal distributions. Examples of interesting cases are the joint probability distributions

$$P\left(x_1 \in A_1, x_2 \in A_2, \ldots\right) \ , \tag{3.8}$$

for sets A_i, and the number distributions

$$P\left(N(A_1) = n_1, N(A_2) = n_2, \ldots\right) \ . \tag{3.9}$$

3.3 The Poisson process

The most fundamental model in point process theory is the *homogeneous Poisson process*, characterized by two properties:

1. The number $N(B)$ of points in a bounded set B is Poisson distributed with mean $\lambda v(B)$:

$$P\left(N(B) = k\right) = \frac{\left(\lambda v(B)\right)^k}{k!} \exp\left(-\lambda v(B)\right) \ . \tag{3.10}$$

The density λ is the called the *intensity* of the process.

2. The number of points in disjoint sets form *independent* random variables. If $B_1, \ldots, B_m$ are disjoint bounded sets, we then have that

$$\begin{aligned} &P\left(N(B_1) = k_1, \ldots, N(B_m) = k_m\right) \\ &= \frac{\left(\lambda v(B_1)\right)^{k_1} \cdots \left(\lambda v(B_m)\right)^{k_m}}{k_1! \cdots k_m!} \exp\left(-\lambda \sum_i v(B_i)\right) \ . \end{aligned} \tag{3.11}$$

The process is called *homogeneous* because the intensity λ is a single-valued parameter. A general (non-homogeneous) Poisson process is obtained by letting λ be a function of position, $\lambda = \lambda(x)$, and replacing the mean $\lambda v(B)$ in the equations above with $\int_B \lambda(x)dx$. Non-homogeneous processes will not be considered here, and the term *Poisson process* is henceforth understood to refer to the homogeneous Poisson process.

The Poisson process is the most fundamental model in point process theory because it is a model of the property known as *complete spatial randomness* (CSR), which means that points are scattered completely independently of each other. This is satisfied by the Poisson process because there is no underlying distribution favoring some locations over others, and the presence of a point at a location does not affect the probability of finding other points in its vicinity. This makes the Poisson process an appropriate null model for point processes.

In particular, CSR implies *stationarity* and *isotropy*,[1] which are defined as follows:

- *Stationarity:* A stationary point process is a point process that is invariant under translations,

$$P(N \in \mathscr{A}) = P(N + x \in \mathscr{A}) \ \forall x \in \mathbb{R}^2 \ . \tag{3.12}$$

- *Isotropy:* An isotropic point process is a point process which is invariant under rotations,

$$P(N \in \mathscr{A}) = P(R_\theta N \in \mathscr{A}) \ \forall \theta \in [0, 2\pi) \ . \tag{3.13}$$

  Here, $R_\theta N$ is shorthand for $\{R_\theta x : x \in N\}$, and R_θ denotes rotation by an angle θ around the origin.

If a process is both stationary and isotropic, we say that it is *motion-invariant.*

Non-CSR processes may exhibit interactions between points, meaning that the events $x_1 \in A$ and $x_2 \in A$ for two distinct points x_1, x_2 in N are not statistically independent. Point interactions may be classified by the effect they have on the pattern:

- *Clustering*: A process is said to be clustered if the presence of a point at a location x increases the probability of finding another

[1] But these properties alone are not sufficient for CSR.

point in the neighborhood of x. We can say that the points attract, and patterns from such processes tend to form clusters.

- *Regularity*: A process is said to be regular if the presence of a point at a location x decreases the probability of finding another point in the neighborhood of x. We can say that the points repel, and patterns from such processes tend to form be more spread out than what one would expect if they were independent.

The two effects may be combined – for example, a process can have an absolute minimum distance between points, imposing regularity at small distance scales, while a long-range attraction makes the patterns cluster on a global scale.

3.4 The binomial process

A binomial process can be understood as the intersection of a Poisson process and a bounded set W, called the *window* of the process, under the condition that the number of points within W is fixed, $N(W) = n$. The binomial process is thus denoted $N_{W^{(n)}}$. It is equivalent to taking n independent samples from a uniform distribution over W. The process is characterized by the joint probabilities

$$P\left(x_1 \in A_1, \dots, x_n \in A_n\right) = \frac{\nu(A_1) \cdots \nu(A_n)}{\nu(W)} \ . \tag{3.14}$$

The name of the process owes to the fact that the counting measure on a subset of W is binomially distributed: defining $p(A) = P(x \in A) = \nu(A)/\nu(W)$ and the intensity $\lambda = n/\nu(W)$, we have

$$\begin{aligned} P\left(N_{W^{(n)}}(A) = k\right) &= \binom{n}{k} p(A)^k \left(1 - p(A)\right)^{1-k} \ , \\ \boldsymbol{E}\left(N_{W^{(n)}}(A)\right) &= np(A) = \lambda \nu(A) \ . \end{aligned} \tag{3.15}$$

The binomial process is an example of a *finite* point process, since the number of points is finite. Although the patterns generated from it can be considered sets of independent samples from a uniform distribution, the binomial process cannot be an appropriate model of CSR, since the number of points in disjoint subsets are not independent. For example, if $N_{W^{(n)}}(A) = m$, then $N_{W^{(n)}}(W \setminus A) = n - m$. But this applies to all finite processes with fixed n, and the binomial process is usually the appropriate null model for finite processes, analogous to the Poisson process for general point processes.

3.5 The periodic binomial process

If the set W tiles the plane by repeated translation, the binomial process can be extended to fill the plane by tiling the plane with copies of W and $N_{W(n)}$. Clearly, the resulting process is stationary, since an arbitrary translated window $W + x$ will still contain exactly one copy of each of the points in $N_{W(n)}$, which are distributed with uniform probability. However, it is not an appropriate model of CSR for the same reason that the binomial process is not.

In [16], only parallelograms are considered as possible windows for periodic binomial processes. However, hexagons can also tile the plane by repeated translation, provided they are central-symmetric such that opposite sides are parallel and equal in length [17]. In general, any valid unit cell of a Bravais lattice (see Appendix A) can be used as window for a periodic process.

3.6 Characteristics of point processes

Several summary characteristics defined in the point process literature to capture some of the statistical properties of a process. Here we will present the ones that are used in later chapters.

Intensity

The intensity measure Λ is the analog of the mean of a random variable for point processes. It is defined for a set B as

$$\Lambda(B) = \boldsymbol{E}\left(N(B)\right) \ . \tag{3.16}$$

For most processes of interest, the intensity measure may be expressed in terms of an intensity function $\lambda(x)$:

$$\Lambda(B) = \int_B \lambda(x) dx \ . \tag{3.17}$$

If λ is a constant (which must be true for stationary processes), it is simply called the *intensity*.

Ripley's K-function

Ripley's K-function is the most frequently used second-order characteristic for point processes. It is defined as the mean number of points within a distance r from a typical point in the pattern, normalized such that it is independent of the intensity λ. For stationary processes, one may write down an intuitive definition of the K-function as

$$K(r) = \frac{1}{\lambda}\boldsymbol{E}\left(N(b(o,r) \setminus \{o\}) \mid o \in N\right) \ , \tag{3.18}$$

that is, $K(r)$ is equal to the number of points lying within distance r of the origin but not at the origin itself, given that the process has placed a point at the origin. The latter condition embodies what is meant by the term *typical point* above: the typical point can be any location in the plane, given that our process has placed a point there. For a stationary process we may use translational invariance and take the typical point to be at the origin without loss of generality.

Equation (3.18) cannot actually be used as the definition of K because the condition $o \in N$ has exactly zero probability for any point process. However, it is sufficient for an intuitive understanding of the K-function and how it can be estimated from an observed pattern. A formal definition relies on the theory of Palm distributions, which we will not go into here.

For a Poisson process, an explicit expression for $K(r)$ is easy to derive: the expected number of points in $b(o,r) \setminus \{o\}$ is $\lambda\pi r^2$, that is, the intensity multiplied by the area of the disc, regardless of whether there is a point at o or not. Hence, for a Poisson process,

$$K(r) = \pi r^2 \ . \tag{3.19}$$

A stationary process will typically have $K(r) > \pi r^2$ if it is clustered at small length scales, and $K(r) < \pi r^2$ if it is regular at small length scales.

For finite processes, the definition above is not applicable, because the finite window W in which they are defined breaks stationarity. Even for a process giving independently and uniformly distributed points, such as the binomial process, the expected number of points within distance r from a point close to the boundary of W can be much smaller than the expected number of points within the same distance from a point close to the center of W, since in the former case a disk of radius

r can extend outside W where there by definition are no points. For a finite process with n points, the K-function is therefore defined as

$$K(r) = \boldsymbol{E}\left(\sum_{i=1}^{n}\sum_{\substack{j=1\\ j\neq i}}^{n} \frac{\mathbf{1}(\|x_i - x_j\| \leq r)}{\nu(W + x_i \cap W + x_j)}\right) \bigg/ \frac{n(n-1)}{\nu(W)^2} \ . \tag{3.20}$$

The denominator in the double sum gives greater weight to distant pairs of points than to close pairs, compensating for the relative sparseness of such pairs due to the finiteness of the window. The double sum will then be roughly proportional to $\lambda^2 = n^2/\nu(W)^2$, but the correct normalization is to divide by $n(n-1)/\nu(W)^2$, with the factor $n-1$ appearing due to the fixed total number of points in W.[2]

Using this definition, the value of $K(r)$ for the binomial process is

$$K(r) = \nu\left((W \oplus \tilde{W}) \cap b(o, r)\right) \ . \tag{3.21}$$

Recalling the definitions in Section 3.1, we see that $W \oplus \tilde{W} = \{x - y : x, y \in W\}$ is the set of vectors between points in W. For example, if W is a rectangle with sides a and b, then $W \oplus \tilde{W}$ is a rectangle with sides $2a$ and $2b$ centered at the origin. Defining r_1 and r_2 as the shortest and longest distance, respectively, from the origin to the boundary of $W \oplus \tilde{W}$, we may write

$$K(r) = \begin{cases} \pi r^2, & r \leq r_1 \ , \\ \nu(W \oplus \tilde{W}), & r > r_2 \ . \end{cases} \tag{3.22}$$

The shape of K between r_1 and r_2 depends on the shape of W. If we restrict our consideration to $r < r_1$, we find the same relationship between the K-function and clustering/regularity as for stationary processes.

We conclude this discussion by considering an appropriate definition of the K-function for processes with periodic boundary conditions, such as the periodic binomial process. This topic is not given a systematic treatment in [16], but the reasoning that follows is based on the same ideas as everything above. Although periodic processes can technically

[2] To understand this, consider a binomial process with n points. If a point is known to be at a location x, there are only $n-1$ points left to consider, and so the expected number of points in $b(x, r) \setminus \{x\}$ is $(n-1)\pi r^2/\nu(W) = \lambda \pi r^2 \cdot (n-1)/n$.

be interpreted as infinite processes defined in the whole plane, we will consider them as finite processes, containing only n unique points in a finite area, but with redefined distances give by a metric that wraps around according to the periodic tiling defined by the window W. We denote this distance $\|x_i - x_j\|_p$ for two points x_i and x_j in a window W that tiles the plane.

We can think of the distance defined by $\|x_i - x_j\|_p$ as the Euclidean distance from x_i to the copy of x_j that is closest to it after having tiled the plane with copies of the pattern. Since any window of a periodic process must be a unit cell of a Bravais lattice, we can formalize this notion: Let $\mathcal{R}$ be the set of all lattice vectors of the Bravais lattice corresponding to W, that is, $\mathcal{R} = \{n_1 \boldsymbol{a}_1 + n_2 \boldsymbol{a}_2 : n_1, n_2 \text{ integer}\}$, where $\boldsymbol{a}_1, \boldsymbol{a}_2$ are primitive vectors of the lattice. Then, we can write

$$\|x_i - x_j\|_p = \min_{\boldsymbol{R} \in \mathcal{R}} \|x_i - (x_j + \boldsymbol{R})\| \, . \tag{3.23}$$

The periodicity eliminates edge effects, so we do not need to consider weightings like in Equation (3.20). Thus, we arrive at the following definition of the K-function for periodic processes:

$$K(r) = \boldsymbol{E}\left(\sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} \frac{\mathbf{1}(\|x_i - x_j\|_p \leq r)}{\nu(W)} \right) \Bigg/ \frac{n(n-1)}{\nu(W)^2} \, . \tag{3.24}$$

By analogy with the K-function for finite processes, we realize that the value of $K(r)$ for a periodic binomial process using this definition can be written

$$K(r) = \nu\left(V \cap b(o, r)\right) \, , \tag{3.25}$$

where V is the Voronoi cell of the Bravais lattice for which W is a unit cell. To see the latter fact, realize by analogy with Equation (3.20) that V must be set of all vectors between a point x in W and the closest periodic copy of a point y in W. Formally, for any point y in W, let y_x be the element of $\{y + \boldsymbol{R} : \boldsymbol{R} \in \mathcal{R}\}$ such that $\|x - y_x\|$ is minimal (and thus defines $\|x - y\|_p$ via Equation (3.23)). Then, we can write $V = \{x - y_x : x, y \in W\}$. This coincides with the definition of the Voronoi cell of the Bravais lattice with lattice vectors $\mathcal{R}$.

Still in anaolgy with Equation (3.22), we define r_1 and r_2 as the shortest and longest distance, respectively, from the origin to the boundary of the Voronoi cell V, and we may thus write

$$K(r) = \begin{cases} \pi r^2, & r \leq r_1 \, , \\ \nu(W), & r > r_2 \, . \end{cases} \tag{3.26}$$

Here we used the fact that $\nu(W) = \nu(V)$, since both are unit cells of the same Bravais lattice. Once again, clustering is indicated by $K(r) > \pi r^2$, while regularity is indicated by $K(r) < \pi r^2$, as long as we only consider $r < r_1$. In everything that follows, the restriction $r < r_1$ for the appropriate definition of r_1 will be implicit whenever the K-function or derived quantities are discussed in the context of finite or periodic processes.

The L-function

The K-function is easy to understand intuitively, but can be improved upon for more rigorous analysis. In particular, the variance of K-function estimators $\widehat{K}(r)$ generally increases with r, making hypothesis tests based on intuitively appealing statistics such as $\max_{r \in I} |\widehat{K}(r) - K(r)|$ heavily biased towards structure at large length scales. This motivates the introduction of the L-function, which is defined as

$$L(r) = \sqrt{\frac{K(r)}{\pi}} \, . \tag{3.27}$$

The variance of common L-function estimators is empirically found to be close to constant. The L-function has the added benefit that for Poisson and binomial processes it is a straight line, $L(r) = r$, simplifying visual judgement of the fit of an estimator to the null model.

The pair correlation function

The K- and L-functions are cumulative in nature: their value at a given r is related to the probability of finding pairs of points at all distances smaller than or equal to r. This ensures monotonicity and is good for testing, but obscures details about the scales at which a process is clustered or regular. For example, if a process tends to cluster at a small scale r_0, this increases the values of K and L for all $r > r_0$. A function that makes details about the structure more accessible is the pair correlation function, which can be defined via the derivative of the K-function,

$$g(r) = \frac{dK/dr}{2\pi r} \ . \tag{3.28}$$

For Poisson and binomial processes we thus get $g(r) = 1$. A process is clustered at length scales around local maxima of g with values greater than 1, and regular at length scales around local minima of g with values smaller than 1.

3.7 Estimators

So far the discussion of point process statistics has been predominantly theoretical. To apply this theory in the analysis of empirical point patterns, we must find ways of estimating the summary characteristics.

A point pattern recorded in the real world consists of a number n of points distributed across some finite window W. As such, the distinction between finite and infinite processes vanishes on the empirical level (in particular, it is impossible to distinguish between a binomial process and a Poisson process using a single pattern), and based on our knowledge of where and how a pattern originated we can choose to analyze it as a realization of either kind of process.

Intensity

Assuming the pattern originated from a stationary process, a straightforward and unbiased estimator of the intensity is

$$\widehat{\lambda} = \frac{n}{\nu(W)} \ , \tag{3.29}$$

which is simply the number of points in the pattern divided by the area of the window. For most purposes, this estimator is the most useful and appropriate. Other estimators have been developed for specific purposes, but they will not be considered here.

When computing estimators of other characteristics, we often need to plug in estimates of λ^2. For an unbiased estimator $\widehat{\lambda}$ of λ, the corresponding unbiased estimator of λ^2 is

$$\widehat{\lambda^2} = \frac{n-1}{n}\widehat{\lambda} \ . \tag{3.30}$$

The estimator of λ^2 corresponding to Equation (3.29) is thus

$$\widehat{\lambda^2} = \frac{n(n-1)}{v(W)^2} \; . \tag{3.31}$$

The factor $(n-1)/n$ is introduced because the mean and variance of a Poisson distributed variable is equal, which means that $\langle n^2\rangle - \langle n\rangle^2 = \langle n\rangle$ such that $\langle n\rangle^2 = \langle n(n-1)\rangle$.

For a finite process, if the intensity is a constant it is *defined* by the expression above, and no estimators are needed.

In the case of non-stationary processes or finite processes with non-constant intensity, the intensity function $\lambda(x)$ can be estimated by fitting a parametric model or using kernel density estimates.

The K-function

For a stationary process, the finiteness of the window W in which the recorded pattern is sampled significantly complicates the estimation of second-order characteristics such as the K- and L-functions. Several edge correction methods have been developed to get around this problem. To derive a general approach, we can first figure out what we would do if we actually had access to an infinite pattern, and the window W with its n points only defined which points we could use as typical points in the estimation. This naturally leads to the expression

$$\begin{aligned} &\frac{1}{n}\sum_{i=1}^{n}\sum_{\substack{j=1\\ j\neq i}}^{\infty} \mathbf{1}\left(\|x_i - x_j\| \le r\right) \\ &= \frac{v(W)}{n}\sum_{i=1}^{n}\sum_{\substack{j=1\\ j\neq i}}^{\infty} \frac{\mathbf{1}\left(\|x_i - x_j\| \le r\right)}{v(W)} \end{aligned} \tag{3.32}$$

as an estimator for the factor $\boldsymbol{E}(N(b(o,r) \setminus \{o\}) \mid o \in N)$ in Equation (3.18). While the first form of the expression is the most intuitive, the second factorization shows that this is in fact a ratio-unbiased estimator of $\lambda K(r)$[3] – we recognize the first factor as $\widehat{\lambda}^{-1}$, and the double sum turns is in fact an unbiased estimator for $\lambda^2 K(r)$. This motivates a general factorization of estimators for $K(r)$ as

$$\widehat{K}(r) = \widehat{\kappa}(r)/\widehat{\lambda^2} \; , \tag{3.33}$$

[3] A ratio-unbiased estimator is the quotient of two unbiased estimators.

where $\widehat{\kappa}(r)$ is an estimator of $\lambda^2 K(r)$. This enables the construction of ratio-unbiased estimators of $K(r)$. From Equation (3.32) we have

$$\widehat{\kappa}(r) = \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{\infty} \frac{\mathbf{1}\left(\|x_i - x_j\| \leq r\right)}{\nu(W)} \tag{3.34}$$

Using this together with Equation (3.31) gives the most straightforward estimator of $K(r)$, with the slight inconvenience that it requires knowledge of the complete, infinitely large pattern. However, if extra points can be provided in some region outside W, extending as far as the largest r we care about, this is in fact a viable estimator, and this general strategy for dealing with a finite window is known as plus sampling. The extra points can be provided in different ways – for example, the pattern may have been sampled in a larger window than the one used for analysis, or a qualified assumption about the kind of process we are dealing with may be used to simulate the process outside W given the points inside W.

If plus sampling is not an opportunity, we need to modify $\widehat{\kappa}$ by weighting distant pairs of points more than close pairs. The general form of $\widehat{\kappa}$ then becomes

$$\widehat{\kappa}(r) = \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} \frac{\mathbf{1}\left(\|x_i - x_j\| \leq r\right)}{w(x_i, x_j)} \; . \tag{3.35}$$

Here, the window area has been replaced by an inverse weight which is a function of the pair of points. This is exactly what was done in the *definition* of the K-function for finite processes in Equation (3.20), with weight function

$$w(x_i, x_j) = \nu(W + x_i \cap W + x_j) \; . \tag{3.36}$$

In fact, by combining the resulting $\widehat{\kappa}$ with $\widehat{\lambda^2}$ from Equation (3.31),[4] we arrive at a K-function estimator that seems to be taken straight out of Equation (3.20) by simply removing the expectation value operation,

[4] Note that $\widehat{\lambda^2}$ is *not* an estimator of λ^2 here, since we are considering finite processes for which λ is known and does not need estimation. This only shows that while the factor of $(n-1)/n$ were introduced for different reasons in the finite and stationary cases, it serves a common purpose when we analyze empirical patterns.

and this is indeed the natural K-function estimator to use for finite processes.

Unsurprisingly, the same estimator works for stationary processes as long as we restrict the definition to $r \leq r_1$, where r_1 is the same as in Equation (3.22) (this is the r-value at which the theoretical K-functions for the finite and stationary processes begin to deviate). However, experience shows that this estimate can be improved even more by using a different estimator for λ^2 which is designed specifically to match this $\widehat{\kappa}$ for stationary processes. Since such estimators will not be used in the later analysis, we will not elaborate further.

Finally, the natural estimator for the K-function for periodic processes can be taken right from the definition in Equation (3.24) in the same way as for the estimator for the K-function for finite processes. Since this is the estimator that will actually by applied in the upcoming analysis, we state it here explicitly:

$$\widehat{K}(r) = \left(\sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} \frac{\mathbf{1}(\|x_i - x_j\|_p \leq r)}{\nu(W)} \right) \Bigg/ \frac{n(n-1)}{\nu(W)^2} \ . \tag{3.37}$$

The L-function

No direct estimators for the L-function are known, but once an appropriate estimator for the K-function is chosen, the L-function can simply be estimated by

$$\hat{L}(r) = \sqrt{\frac{\widehat{K}(r)}{\pi}} \ . \tag{3.38}$$

The pair correlation function

Estimators of g are factorized, in analogy with estimators of K, as $\hat{g}(r) = \widehat{\rho}(r)/\widehat{\lambda^2}$. For each definition of $\widehat{\kappa}$ used to define an estimator $\widehat{K}$, the corresponding definition of $\widehat{\rho}$ and thus $\hat{g}$ can be obtained by performing the substitution [18]

$$\mathbf{1}(\|x_i - x_j\| \leq r) \rightarrow \frac{k(\|x_i - x_j\| - r)}{2\pi r} \ , \tag{3.39}$$

where k is a kernel function, that is, a function satisfying $k(x) \geq 0$ and $\int k(x)dx = 1$. The box kernel,

$$k(x) = \frac{1}{2h}\mathbf{1}(|x| \leq h) \tag{3.40}$$

is usually the best choice of kernel, and it is the one that will be used here. The bandwidth h decides the balance between smoothing over noise and removing details, and trial and error is often necessary to find a good value.

The pair correlation estimator corresponding to the periodic K-function estimator from Equation (3.37) can be written explicitly as

$$\hat{g}(r) = \left(\sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} \frac{\mathbf{1}(|\|x_i - x_j\|_p - r| \leq h)}{4\pi r h v(W)} \right) \Bigg/ \frac{n(n-1)}{v(W)^2} \ . \tag{3.41}$$

3.8 Testing complete spatial randomness

The first question to ask when analyzing a point pattern, and the one we are most interested in here, is whether the pattern could have been generated by a process generating uniformly and independently distributed points, or whether it contains evidence of a process with more structure. Here we will only present one test, specifically the one that is most recommended by [16], based on the L-function and aptly named the L-test. It is appropriate for both regular and clustered processes. The test statistic is defined as

$$\tau = \max_{r_{\min} \leq r \leq r_{\max}} |\hat{L}(r) - r| \ . \tag{3.42}$$

Critical values for the test statistic τ can be found by Monte Carlo simulation: a large number k of patterns is simulated with n uniformly and independently distributed points in W,[5] and the test statistic is computed for each of them, giving a random sample of test statistics $\tau_i, i = 1, \ldots, k$ from the probability distribution for τ under the null

[5] The simulation of the null model is identical whether we test against the Poisson process, the binomial process or the periodic binomial process – only the estimators are different.

model. The critical value $\tau_{1-\alpha}$ for significance level α can be approximated from this sample, and the p-value of the empirical pattern under the null model can be computed from its rank among the τ_i when sorted in ascending order.

If a set of simulations have been performed, it can also be informative to plot a simulation envelope of $L(r)$, defined by the critical values $[\hat{L}(r)]_{\alpha/2}$ and $[\hat{L}(r)]_{1-\alpha/2}$ based on the distribution of simulated estimators $\hat{L}_i(r)$ at each value of r. This illustrates the width of the distribution for $\hat{L}(r)$ at different r. Note that the simulation envelope is not suited as a formal statistical test – even if the empirical estimator $\hat{L}(r)$ pokes outside the envelope for a chosen α, this does not support rejecting CSR at significance level α.

The value of $r_{\max}$ in Equation (3.42) depends upon the choice of estimator. For a square window, using half the diagonal length is recommended by [16]. However, this is clearly inappropriate for the novel K/L-function for periodic processes defined here, since this will make $r_{\max}$ larger than r_1, the value at which $L(r)$ for the periodic binomial process begins to deviate from r. Hence, we will use $r_{\max} = r_1$ when applying the L-test against the periodic binomial process.

The lower limit $r_{\min}$ in Equation (3.42) is not used by [16], but is introduced here to avoid having the test dominated by the observed increase in variance of $L(r)$ close to $r = 0$, especially for .patterns with few points. No general rules for choosing $r_{\min}$ have been derived, but a suitable value can be found by looking at a simulation envelope.

4

Data and methods

4.1 Data

The data analyzed here was recorded in the MEC area of a Long Evans male rat, using 12 tetrodes at independent recording locations covering large parts of the MEC. Neural activity was recorded during 30 sessions, lasting between 15 and 30 minutes each, in which the rat was foraging in a 150 cm × 150 cm square box. This square box will henceforth be referred to as the *experimental environment* or simply *environment*. The recording sessions were carried out over a period of 7 months.

This dataset is identical to the data from rat 14147 in an article by Stensola et al., and more information regarding recording procedures, spike sorting and cell classification can be found in the supplementary information to this reference [9]. Note in particular that an effort was made to eliminate duplicate recordings of cells from the dataset. Hence, we have assumed without reservation that each cell in the dataset is unique.

Only cells that were identified as grid cells by Stensola et al. have been considered, and we did not verify this identification in any way. A total of 176 grid cells, belonging to 4 modules, were identified in the dataset.

For each recording session, the dataset consists of a sequence of regularly spaced time points $t_i \approx i * 0.04$ s, and corresponding position samples x_i, y_i giving the position of the rat within the experimental environment at time t_i. The positions are given in arbitrary, dimensionless coordinates with axes approximately aligned with the walls of the environment. For each identified cell, a sequence of spiking times τ_j is provided, giving the times at which the cell fired an action potential.

4.2 Parametrizing the grid pattern

Many of the procedures used to extract parameters characterizing the firing pattern of a grid cell are fairly standardized, see e.g. the supplementary information to articles by Hafting et al., Stensola et al. [7, 9]. However, variations are possible at almost every step, and we have also developed some novel elements in this work. The steps followed are therefore laid out in full detail in the following sections.

Firing rate maps

The initial step in the analysis is to compute a map of the average firing rate of each cell across the environment. First, we transform the positions to fit into the set

$$\begin{aligned} B &= I \times I \ , \\ I &= [-L/2, L/2] \ , \end{aligned} \tag{4.1}$$

where $L = 150$ cm is the length of the sides of the environment. This transformation allows us to interpret the positions as centimeter measurements along axes aligned with and centered in the experimental environment. The transformation consists of three steps, each assuming that the animal visited several positions along each wall.

- First, the positions are rotated to maximize the alignment of the coordinate axes with the walls of the box. This is done by defining

  $$\begin{pmatrix} x_i' \\ y_i' \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x_i \\ y_i \end{pmatrix} \ , \tag{4.2}$$

  and finding the angle $\theta \in [-\pi/4, \pi/4]$ that minimizes the area of the rectangular bounding box around the data,

  $$A = \left(\max_i \ x_i' - \min_i \ x_i'\right) \times \left(\max_i \ y_i' - \min_i \ y_i'\right) \ . \tag{4.3}$$

- Next, the positions are centered over the origin by the transformation

  $$\begin{aligned} x_i'' &= x_i' - \frac{1}{2}\left(\max_i \ x_i' + \min_i \ x_i'\right) \ , \\ y_i'' &= y_i' - \frac{1}{2}\left(\max_i \ y_i' + \min_i \ y_i'\right) \ . \end{aligned} \tag{4.4}$$

- Finally, the positions are scaled to fit inside B, inserting a small (and somewhat arbitrary) margin between the most extreme position and the nearest wall:

$$\begin{aligned} x_i''' &= a x_i'' \ , \\ y_i''' &= a y_i'' \ , \\ a &= \frac{L/2 - 0.5 \text{ cm}}{\max\left\{\max_i |x_i''|, \max_i |y_i''|\right\}} \ . \end{aligned} \tag{4.5}$$

The transformation is applied to each session independently, to eliminate any systematic error due to relative movement between the position-measuring equipment and the experimental environment between sessions.

Henceforth, variables x_i and y_i, without primes, will be used to refer to the transformed positions.

We calculate the locations of the spikes of a cell by linear interpolation in the position samples at the recorded spiking times. We will denote the interpolated location for the spike at time τ_j by (ξ_j, χ_j).

A speed filter is applied to the data. We calculate the average speed over a 1-second (25-sample) interval around each sample by dividing the total distance covered in this interval by the total time. This can be written as

$$v_i = \frac{\sum_{k=-12}^{12} \Delta d_{i+k}}{\sum_{k=-12}^{12} \Delta t_{i+k}} \ , \tag{4.6}$$

with

$$\begin{aligned} \Delta d_i &= \frac{1}{2}\Bigg(\sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \\ &\qquad + \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \Bigg) \ , \\ \Delta t_i &= \frac{1}{2}\left(t_{i+1} - t_{i-1}\right) \ . \end{aligned} \tag{4.7}$$

In the analysis, we only consider samples and spikes for which the current average speed v_i exceeds 2.0 cm/s.

The environment B is divided into 42×42 identical square bins B_{ij} of width $\delta = L \ / \ 42 \approx 3.57$ cm:

$$\begin{aligned} B_{ij} &= I_i \times I_j \ , \quad i, j \in \{1, \dots, 42\} \ , \\ I_i &= \ [(i-1)\delta - L/2, i\delta - L/2) \ . \end{aligned} \tag{4.8}$$

We use this binning to create a histogram S_{ij} by counting the number of spikes recorded within each cell. We also calculate the total time T_{ij} spent by the rat in each bin during a recording session. The histograms can be defined as

$$\begin{aligned} S_{ij} &= \sum_k \mathbf{1}\left((\xi_k, \chi_k) \in B_{ij}\right) \ , \\ T_{ij} &= \sum_i \Delta t_k \mathbf{1}\left((x_k, y_k) \in B_{ij}\right) \ . \end{aligned} \tag{4.9}$$

The spike and time histograms are filtered to smooth out noisy variations on small length scales. The average firing rate f_{ij} in each bin is the defined as then quotient of the smoothed spike and time histograms,

$$f_{ij} = \frac{(h * S)_{ij}}{(h * T)_{ij}} \ . \tag{4.10}$$

Here, $*$ is the discrete convolution operator, and h denotes the smoothing filter kernel. We use a Gaussian kernel of standard deviation 3.3 cm in terms of physical dimensions, which translates to a discrete gaussian kernel of standard deviation $\sigma = 3.3 \text{ cm} \ / \ \delta = 0.924$. The firing rate is left undefined for indices k, l where $(h * S)_{ij} = (h * T)_{ij} = 0$.[1]

A typical firing rate map as computed by this procedure is visualized in Figure 4.1.

Autocorrelogram peaks

We define the cross-correlogram of two firing rate maps as the Pearson product-moment correlation coefficient of the overlapping part of the maps at each available relative displacement. For rate maps f and g, this can be written as

$$r_{ij} = \frac{\sum \left(f_{kl} - \bar{f}_{ij}\right)\left(g_{k+i,l+j} - \bar{g}_{ij}\right)}{\sqrt{\sum \left(f_{kl} - \bar{f}_{ij}\right)^2 \sum \left(g_{k+i,l+j} - \bar{g}_{ij}\right)^2}} \ , \tag{4.11}$$

[1] If we find bins where $(h * T)_{ij} = 0$ while $(h * S)_{ij}$ is finite, we realize that we have done something wrong.

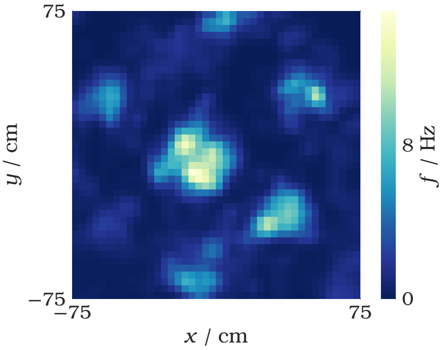


Figure 4.1: Spatial firing rate map of a grid cell. This map was computed from the raw data visualized in Figure 2.1, following the steps explained in the text.

with

$$\begin{aligned}\bar{f}_{ij} &= \frac{1}{n}\sum f_{kl}\,, \\ \bar{g}_{ij} &= \frac{1}{n}\sum g_{k+i,l+j}\,.\end{aligned} \tag{4.12}$$

All sums are taken over the indices k, l where both f_{kl} and $g_{k+i,l+j}$ are defined, and n is the number of such indices. Note in particular that the averages in Equation (4.12) are also only taken over overlapping bins, hence the definition of $\bar{f}_{ij}$ depends on g and vice versa.

The autocorrelogram of a rate map is the cross-correlogram of the map with itself. The firing rate autocorrelogram of a grid cell will contain a characteristic ring of six peaks surrounding the central peak, corresponding to the periodicity in the spatial firing pattern. We assign coordinates to these peaks by first identifying the connected region around each peak where $r_{ij} > r_{\text{min}}$. Here r_{min} is a threshold used to separate peaks from each other and the background in the autocorrelogram. A default value of $r_{\text{min}} = 0.2$ is used for most cells, but for some cells this parameter must be tuned manually in order to successfully isolate a region around all six peaks of interest. A peak's location $\boldsymbol{p}$ was then defined as the center of mass of the corresponding region,

$$\boldsymbol{p}_k = \frac{\sum r_{ij}\boldsymbol{x}_{ij}}{\sum r_{ij}}\,, \tag{4.13}$$

where $\boldsymbol{x}_{ij}$ is the vector pointing from the center of the autocorrelogram to the center of the bin indexed by i,j, and the sum is taken over the indices i,j corresponding to the connected region around the peak in question. The index k identifies the peak among the six, and runs from 1 to 6 with the peaks sorted by positive angle with the x axis. Because an autocorrelgram must be symmetric under under reflection through the center, the peak vectors satisfy

$$\boldsymbol{p}_{k+3} = -\boldsymbol{p}_k \ , \quad k = 1, 2, 3 \ . \tag{4.14}$$

Projection to Bravais lattice

The pattern created by the firing fields can be characterized by the primitive vectors of the Bravais lattice approximated by the pattern. The natural candidates for primitive vectors are the peak vectors $\boldsymbol{p}_k$. However, there are (up to reflection) three unique peak vectors, and we only need two primitive vectors. For consistency, we must be able to take any pair of unique peak vectors as primitive vectors, and have the others be ordinary lattice vectors in accordance with Equation (A.1). In particular, we must have

$$\boldsymbol{p}_k = \boldsymbol{p}_{k-1} + \boldsymbol{p}_{k+1} \ , \tag{4.15}$$

where the indices are understood to wrap around cyclically. If this is satisfied, any pair of unique vectors among the $\boldsymbol{p}_k$ can be used as primitive vectors.

In practice, the peak vectors extracted from autocorrelograms are not guaranteed to satisfy this constraint exactly. We have therefore developed a projection from the space of arbitrary peak vectors $\boldsymbol{p}_k$ to the space of vectors satisfying Equation (4.15).

Considering only the three unique vectors $\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3$ and making use of the Equation (4.14), the condition from Equation (4.15) can be written

$$\begin{aligned} \boldsymbol{p}_1 &= \boldsymbol{p}_2 - \boldsymbol{p}_3 \ , \\ \boldsymbol{p}_2 &= \boldsymbol{p}_1 + \boldsymbol{p}_3 \ , \\ \boldsymbol{p}_3 &= \boldsymbol{p}_2 - \boldsymbol{p}_1 \ . \end{aligned} \tag{4.16}$$

Even if not satisfied, these conditions provide two candidates for each of the three lattice vectors that the $\boldsymbol{p}_i$ presumably approximate. For example, the lattice vector approximated by $\boldsymbol{p}_2$, should ideally be equal to both $\boldsymbol{p}_2$ and $\boldsymbol{p}_1 + \boldsymbol{p}_3$, and we can consider these our two candidates for this vector. If we let $\boldsymbol{a}_i$ be the lattice vector approximated by $\boldsymbol{p}_i$, and assume that it can be expressed as a weighted average of our two candidates, the transformation from peak vectors $\boldsymbol{p}_i$ to lattice vectors $\boldsymbol{a}_i$ can be written in matrix form as

$$\begin{pmatrix} \boldsymbol{a}_1 \\ \boldsymbol{a}_2 \\ \boldsymbol{a}_3 \end{pmatrix} = P \begin{pmatrix} \boldsymbol{p}_1 \\ \boldsymbol{p}_2 \\ \boldsymbol{p}_3 \end{pmatrix} , \tag{4.17}$$

with

$$P = \frac{1}{\lambda + 1} \begin{pmatrix} \lambda & 1 & -1 \\ 1 & \lambda & 1 \\ -1 & 1 & \lambda \end{pmatrix} . \tag{4.18}$$

For the $\boldsymbol{a}_i$ to satisfy Equation (4.16) for any $\boldsymbol{p}_i$, it is both necessary and sufficient that P is a projection matrix, that is, $P^2 = P$. This requirement gives the unique solution $\lambda = 2$. Thus, a transformation from arbitrary peak vectors $\boldsymbol{p}_i$ to lattice vectors $\boldsymbol{a}_i$ can be written

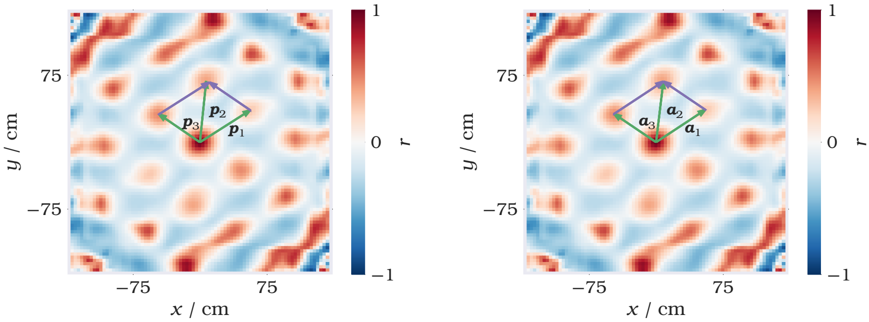
$$\begin{pmatrix} \boldsymbol{a}_1 \\ \boldsymbol{a}_2 \\ \boldsymbol{a}_3 \end{pmatrix} = \frac{1}{3} \begin{pmatrix} 2 & 1 & -1 \\ 1 & 2 & 1 \\ -1 & 1 & 2 \end{pmatrix} \begin{pmatrix} \boldsymbol{p}_1 \\ \boldsymbol{p}_2 \\ \boldsymbol{p}_3 \end{pmatrix} . \tag{4.19}$$

The effect of the projection is illustrated in Figure 4.2. It should be noted that this projection assigns equal importance to all peak vectors $\boldsymbol{p}_i$, regardless of the strength of the corresponding peak in the autocorrelogram. The possibility of generalizing the projection by weighting the peak vectors, e.g. by the peak amplitude, has not been considered.

For implementation purposes, it may be desirable to define the analogous projection using all six $\boldsymbol{p}_i$, such that it immediately provides six corresponding $\boldsymbol{a}_i$. This projection can be written as follows:

$$\begin{pmatrix} \boldsymbol{a}_1 \\ \boldsymbol{a}_2 \\ \boldsymbol{a}_3 \\ \boldsymbol{a}_4 \\ \boldsymbol{a}_5 \\ \boldsymbol{a}_6 \end{pmatrix} = \frac{1}{6} \begin{pmatrix} 2 & 1 & -1 & -2 & -1 & 1 \\ 1 & 2 & 1 & -1 & -2 & -1 \\ -1 & 1 & 2 & 1 & -1 & -2 \\ -2 & -1 & 1 & 2 & 1 & -1 \\ -1 & -2 & -1 & 1 & 2 & 1 \\ 1 & -1 & -2 & 1 & 1 & 2 \end{pmatrix} \begin{pmatrix} \boldsymbol{p}_1 \\ \boldsymbol{p}_2 \\ \boldsymbol{p}_3 \\ \boldsymbol{p}_4 \\ \boldsymbol{p}_5 \\ \boldsymbol{p}_6 \end{pmatrix} . \tag{4.20}$$

This projection is the one that is applied to the data below.



(a) Before projection

(b) After projection

Figure 4.2: Spatial autocorrelograms of the firing rate from Figure 4.1, with peak vectors $\boldsymbol{p}_i$ superimposed as green arrows in (a), and lattice vectors $\boldsymbol{a}_i$ obtained from Equation (4.19) superimposed as green arrows in (b). The purple arrows show $\boldsymbol{p}_1$ ($\boldsymbol{a}_1$) drawn from the end of $\boldsymbol{p}_3$ ($\boldsymbol{a}_3$) and vice versa, clearly showing that $\boldsymbol{p}_2 \neq \boldsymbol{p}_1 + \boldsymbol{p}_3$, while $\boldsymbol{a}_2 = \boldsymbol{a}_1 + \boldsymbol{a}_3$.

For the remainder of the text we write the components of a lattice vector $\boldsymbol{a}_i$ as $(a_{x,i}, a_{y,i})$ in Cartesian coordinates and as (l_i, β_i) in polar coordinates.

Deformation ellipse

If the firing rate pattern constituted a perfect hexagonal lattice, we would expect the lattice vectors $\boldsymbol{a}_i$ to be equal in length and have a uniform angular spacing of 60° between neighboring vectors, spreading nicely out on a circle. This idealization does not hold for a general grid cell, and Stensola et al. introduced the practice of fitting an ellipse through the autocorrelogram peaks and use its eccentricity to quantify the deformation of the firing rate pattern from perfect hexagonality [9]. They use the ellipse fitting method proposed by Fitzgibbon, Pilu, and Fisher, minimizing an algebraic distance between the points and the ellipse [19]. We would like to take the opportunity to warn against the statement and implementation of this method used in the original article due to its lack of well-posedness and numerical instability, and rather point the reader to the reformulation of the method by Halíř and Flusser in [20]. However, in the following we will show that approximate fitting algorithms are unnecessary when fitting ellipses through autocorrelogram peaks, since the ellipse fitting problem has an exact solution for points satisfying the reflection symmetry from Equation (4.14).

A general conic section can be written as an implicit quadratic equation in two variables:

$$Ax^2 + 2Bxy + Cy^2 + 2Dx + 2Ey + F = 0 \ . \tag{4.21}$$

Clearly, scaling the six parameters A, B, C, D, E, F by a common factor will not change the conic, leaving five degrees of freedom for size, shape and orientation. The conic will be an ellipse if $B^2 - AC < 0$.

For N points (x_i, y_i) , $i = 1, \dots, N$, define the $N \times 6$ matrix M as

$$M = \begin{pmatrix} x_1^2 & 2x_1y_1 & y_1^2 & 2x_1 & 2y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_N^2 & 2x_Ny_N & y_N^2 & 2x_N & 2y_N & 1 \end{pmatrix} \tag{4.22}$$

Collecting the parameters of a conic in a vector $\boldsymbol{\theta} = (A, B, C, D, E, F)^T$, the condition that the conic passes through all N points can be written as the matrix equation $M\boldsymbol{\theta} = 0$. In general, this will be an overdetermined system of equations, and this is the starting point for several ellipse fitting algorithms [19–21].

However, if the rank of M is 5, its null space must be one-dimensional, giving a unique solution for $\boldsymbol{\theta}$ up to a scalar factor, and thus a unique conic. We can then solve for $\boldsymbol{\theta}$ by performing a singular value decomposition of M, and $\boldsymbol{\theta}$ will be the right-singular vector corresponding to the vanishing singular value.

If we construct the matrix M for the six points $\boldsymbol{a}_i = (a_{x,i}, a_{y,i})$ and recall the reflection symmetry from Equation (4.14), we may write it in block form as

$$M = \begin{pmatrix} A & B + C \\ A & -B + C \end{pmatrix} \ , \tag{4.23}$$

which is row equivalent to

$$M' = \begin{pmatrix} A & C \\ 0 & B \end{pmatrix} \ . \tag{4.24}$$

Here,

$$\begin{aligned} A &= \begin{pmatrix} a_{x,1}^2 & 2a_{x,1}a_{y,1} & a_{y,1}^2 \\ a_{x,2}^2 & 2a_{x,2}a_{y,2} & a_{y,2}^2 \\ a_{x,3}^2 & 2a_{x,3}a_{y,3} & a_{y,3}^2 \end{pmatrix}, \\ B &= \begin{pmatrix} 2a_{x,1} & 2a_{y,1} & 0 \\ 2a_{x,2} & 2a_{y,2} & 0 \\ 2a_{x,3} & 2a_{y,3} & 0 \end{pmatrix}, \\ C &= \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}. \end{aligned} \tag{4.25}$$

Provided that $\boldsymbol{a}_1$, $\boldsymbol{a}_2$ and $\boldsymbol{a}_3$ are not all parallel, we see that $\operatorname{rank}(B) = 2$. As long as *none* of them are parallel, we also find that A is full-rank (the easiest way to show this is to consider the linear independence of the column vectors of A), such that $\operatorname{rank}([A \;\; C]) = 3$. Thus, $\operatorname{rank}(M') = \operatorname{rank}(M) = 5$, and using the prescription above we can solve for the conic section passing through the vectors $\boldsymbol{a}_i$ from a grid cell autocorrelogram. Note that this derivation only relies on the reflection symmetry from Equation (4.14), so the conic can just as easily be fitted through the unprojected peak vectors $\boldsymbol{p}_i$.[2] Figure 4.3 shows an ellipse fitted through the lattice vectors in an autocorrelogram.

The condition $\operatorname{rank}(M) = 5$ does not guarantee that the fitted conic will be an ellipse, but intuitively one would expect this to be the case whenever the polygon created by connecting the points $(a_{x,i}, a_{y,i})$ in counterclockwise order is convex. This will always the case in grid cell autocorrelograms.

The parameter vector $\boldsymbol{\theta} = (A, B, C, D, E, F)^T$ specifies an ellipse (or in general, a conic section) in the language of analytic geometry. An equivalent parametrization of ellipses that is often easier to reason about consists of the canonical parameters (x_c, y_c, a, b, θ), where (x_c, y_c) are the coordinates of the center of the ellipse, a and b are the lengths of the semi-major and semi-minor axes, respectively, and $\theta \in [-\pi/2, \pi/2)$ is the angle between the major axis and the x-axis. Defining the translated

[2] What we have discovered here is really that six points with pairwise reflection symmetry through a common center, or equivalently, a center and three peripheral points, uniquely specify a conic section, as long as none of the points lie on the same line through the center.

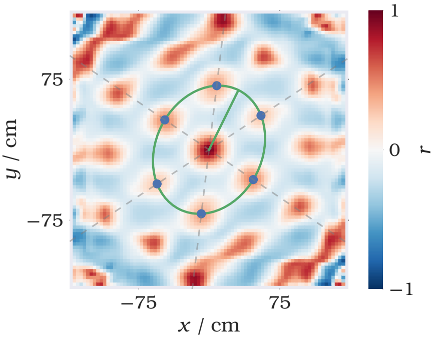


Figure 4.3: Spatial autocorrelogram of the firing rate from Figure 4.1, with the lattice vectors $\boldsymbol{a}_i$ shown as blue points, and the deformation ellipse fitted through them shown in green. The semi-major axis of the is also drawn, to clearly show the centering and orientation of the ellipse.

and rotated coordinates

$$\begin{aligned} x' &= (x - x_c)\cos\theta + (y - y_c)\sin\theta \ , \\ y' &= -(x - x_c)\sin\theta + (y - y_c)\cos\theta \ , \end{aligned} \tag{4.26}$$

the ellipse satisfies

$$\frac{x'^2}{a^2} + \frac{y'^2}{b^2} = 1 \ . \tag{4.27}$$

Deriving the transformation between the two parametrizations is straightforward, if somewhat tedious, see e.g. [22].

The eccentricity of an ellipse is defined as

$$\varepsilon = \sqrt{1 - \frac{b^2}{a^2}} \ . \tag{4.28}$$

The eccentricity quantifies the elongation of the ellipse, with $\varepsilon = 0$ corresponding to a circle and $\varepsilon \to 1$ being the limit where the ellipse turns into a parabola.

Grid scale

In principle, two primitive vectors such as $\boldsymbol{a}_1, \boldsymbol{a}_2$ are necessary and sufficient to fully specify the lattice approximated by a grid cell firing pattern. However, in practice we sometimes prefer to separate the characterization into a single dimensionful scale parameter, the *grid scale* or *grid spacing*, which we denote l, and a set of dimensionless shape and orientation parameters.

Stensola et al. defines l as the average of the spacing along the three independent directions in the lattice, that is, as the average of the lengths l_i of the lattice vectors $\boldsymbol{a}_i$ [9]. Here, we adopt a slightly different definition by setting $l = \sqrt{ab}$, the geometric mean of the semi-major and semi-minor axes of the fitted ellipse. This definition is motivated by the fact that the area of an ellipse is πab, making $\sqrt{ab}$ the radius of a circle with the same area as the fitted ellipse. This is motivated by the recent evidence that the distortion from perfect hexagonality in grid cell firing patterns is consistent with applying a shear mapping along the walls in the environment [8]. Since shear mappings are area-preserving, they leave $l = \sqrt{ab}$ invariant. The definition of grid spacing presented here may therefore capture the characteristic firing field spacing of a grid cell in a more fundamental and environment-independent way than any alternatives.

4.3 Clustering into modules

Grouping grid cells into modules is performed by embedding the cells as points in a high-dimensional space using a suitable set of lattice parameters, henceforth referred to as the *features* of the cell, and applying a clustering algorithm to this collection of points. It was shown by Stensola et al. that grid cells in the same module are similar in both grid scale, orientation of grid axes, and grid distortion as parametrized by the distortion ellipse [9]. In the language used here, this article defined the features of a cell as the vector

$$X = (l_1, l_2, l_6, \beta_1, \beta_2, \beta_6, \varepsilon \cos 2\theta, \varepsilon \sin 2\theta)^T \, . \tag{4.29}$$

Here, we define the feature array slightly differently, namely

$$X = (w_l \log l, a_{x,1}/l, a_{y,1}/l, a_{x,2}/l, a_{y,2}/l, a_{x,3}/l, a_{y,3}/l)^T \, , \tag{4.30}$$

where w_l is a parameter that is yet to be determined. The rationale for choosing this feature array can be summed up as follows:

- The analysis in [9] demonstrates that the width of the distribution of grid scales within a module is approximately constant on a logarithmic axis. This is not very surprising – it is equivalent to stating that width of the distribution is proportional to the mean, or alternatively that the *relative* width is constant, since the general quantity that is the same for all modules is $\log(\langle l\rangle + \Delta l) - \log(\langle l\rangle) = \log(1 + \Delta l/\langle l\rangle)$. Therefore, the contribution from the grid scale to the distance between two cells is best measured on a logarithmic axis, motivating the use of $\log l$ as one of the features.

- For the distance between cells to be independent of measurement units, all features – or at least, all feature differences – must be non-dimensional. This is satisfied for the logarithmic grid scale since $\log(l_a) - \log(l_b) = \log(l_a/l_b)$, and is obviously true for the rest of the features. If we had used features of different dimensionalities, their relative weight in cell-to-cell distances would depend on the choice of units.

- There is no way to know the relative importance of unrelated groups of features, and hence we should in general introduce an unknown parameter for every extra group of parameters to determine their relative weight in cell-to-cell distances. For example, the orientation angles β_i and the ellipse parameters ε, θ both characterize scale-independent properties of the cell firing pattern, but they are not obviously comparable, and one would in general have to introduce a parameter to weight them against each other. By using $(a_{x,i}/l, a_{y,i}/l)$ instead, all scale-independent properties are taken into account simultaneously via the projection of the three unique lattice vectors of a cell onto the ”unit ellipse” (the distortion ellipse of the cell, scaled down to the same area as the unit circle). The difference between the projected vectors increases with increasing difference in both orientation and deformation, but in a way that makes the two kinds of differences naturally comparable. This only leaves one undetermined parameter, w_l, determining the relative weight of grid scale difference and difference in scale-independent properties.

With this feature array, the distance between two cells a and b, $\|X_a - X_b\|$, is therefore completely independent of measurement units, not biased towards considering smaller cells more similar than larger cells, and can be tuned across the whole spectrum of relative importance between grid scale and scale-independent properties using a single weighting parameter w_l.

As in [9], clusters of cells in feature space are identified as grid cell modules. Two different clustering algorithms are used to find clusters. Brief descriptions of the algorithms and how they are applied follow.

The kmeans++ algorithm

The kmeans++ algorithm is a variation of the standard k-means algorithm, using a seeding technique for initializing the centroids [23].

Given a set of n data points in $\{x_i\}_{i=1}^{n} \subset \mathbb{R}^d$, the objective of the k-means clustering problem is to choose k centroids $\{c_j\}_{j=1}^{k} \subset \mathbb{R}^d$ minimizing the potential function, or *inertia*,

$$\varphi = \sum_{i=1}^{n} \min_{j \in \{1,\ldots,k\}} \|x_i - c_j\|^2 \ . \tag{4.31}$$

Each centroid c_j of the final solution defines one cluster C_j containing all the x_i closer to this centroid than any other. The standard algorithm used to find an approximate solution to this problem works as follows:

1. Choose k random points in $\mathbb{R}^d$ to use as centroids $\{c_j\}$.
2. Define the clusters $\{C_j\}$ such that C_j contains the set of x_i closer to c_j than to any other centroid. Let n_j be the number of points in C_j.
3. Redefine the centroids $\{c_j\}$ as

   $$c_j = \frac{1}{n_j} \sum_{x \in C_j} x \ , \tag{4.32}$$

   that is, the center of mass of the points in C_j.
4. Repeat 2 and 3 until convergence.

The algorithm is only guaranteed to converge to a local minimum of φ.

The kmeans++ algorithm replaces step Item 1 with a seeding technique which randomly selects the centroids c_j one by one from the points x_i, assigning probabilities to the candidate points in such a way as to ensure that φ is already relatively low from the beginning. Empirically, the kmeans++ algorithm is found to be a significant improvement on the standard algorithm, in the sense that the average final value of φ over many runs on the same set of points is lower.

The mean shift algorithm

Given a set of n data points in $\{x_i\}_{i=1}^n \subset \mathbb{R}^d$, the mean shift algorithm works by computing a centroid candidate c_j using the following iterative procedure [24]:

1. Choose a random point in $\mathbb{R}^d$ as initial guess c_j.

2. Compute the mean shift vector

   $$m(c_j) = \frac{\sum_{i=1}^n x_i k(x_i - c_j)}{\sum_{i=1}^n k(x_i - c_j)} , \tag{4.33}$$

   where k is the same box kernel as used for estimating the pair correlation function, given in Equation (3.40). The kernel depends on one parameter, the bandwidth h, which can optionally be estimated from the data.

3. Update the centroid by translating with the mean shift vector:

   $$c_j \leftarrow c_j + m(c_j) . \tag{4.34}$$

   The centroid c_j is thus updated to be the mean of the samples within distance h from its previous location.

4. Repeat 2 and 3 until convergence.

The algorithm is performed for a large number of different starting positions, and in the end duplicate centroids that have converged to the same location are eliminated. The set of points that were at some step found within the kernel for *all* sequences that converged to the same final centroid constitute the cluster associated with this centroid.

Two notable differences between the mean shift algorithm and the k-means/kmeans++ algorithm are:

- The k-means algorithm takes the number of clusters as parameter, while the mean shift algorithm discovers the number of clusters to make and may thus be used without knowing how many clusters to expect. If an automatic estimator of the bandwidth is used, the mean shift algorithm may even be completely non-parametric.
- While the k-means algorithm forces every point to be a member of a cluster, the mean shift algorithm may generate outliers. Thus, the resulting clusters only contain points that actually belong to a region of high point density.

The latter property is the main reason for using the mean shift algorithm in addition to the k-means algorithm: it allows us to check whether any features observed in the grid phases or other properties of a module are dominated by cells on the fringes of the modules, for which the assignment to a particular module might be ambiguous, or if the observed features persist in the absence of these cells.

For both algorithms, the implementation from the Python package *scikit-learn* is used [25]. The desired number of clusters is known to be 4 from [9]. For the kmeans++ algorithm, the most optimal result (lowest φ) of 10 runs is selected. This procedure is repeated 10 times, and if the most optimal result is not identical in all 10 cases, the parameter w_l is adjusted in search of a more stable separation of the clusters. The clustering is accepted when the best result of 10 runs is identical 10 times in a row, and this results strikes a good balance between scale and orientation/distortion clustering. Setting $w_l = 1.20$ turns out to accomplish this.

For the mean shift algorithm, the bandwidth h is tuned to produce 4 clusters containing as many points as possible, minimum of 4 points each (clusters with 3 or fewer points are merged with the outliers). Setting $h = 0.16$ does the job.

4.4 Determining the grid phase

So far, analyses of grid cell phases have usually considered the relative phases between pairs of cells [7, 10, 11]. The relative phase between two cells can be defined as the location of the peak closest to the center in the cross-correlogram of their firing rates. Here, however, we want to map the absolute phase of all cells in a module with respect to a common reference, so that we can employ the full machinery of point process statistics in our analysis. For each module, we therefore construct the firing rate map of a “template cell” representing the average firing pattern of all the cells in the module, and compute the cross-correlogram of all cells with respect to this template.

The template cell of a module

Let a module contain n cells indexed by $c = 1, \dots, n$, and write the lattice vectors for the cells as $\boldsymbol{a}_k^c$, $k = 1, \dots, 6$. The template lattice of the module is then defined as $\boldsymbol{a}_k^t = \sum_c \boldsymbol{a}_k^c / n$.[3]

For each cell in the module, a two-dimensional Gaussian is fitted to the central peak in the autocorrelogram in order to characterize the average shape of the firing fields of the cell. A 2D Gaussian of arbitrary amplitude can in general be written

$$f(\boldsymbol{x}) = \frac{A}{2\pi\sqrt{\det(\boldsymbol{\Sigma})}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}-\boldsymbol{\mu})\right) , \tag{4.35}$$

where A is a scale factor equal to the value of the integral of the Gaussian over all of space, $\boldsymbol{\mu} = (\mu_x, \mu_y)^T$ is the mean that the peak of the Gaussian is centered around, and $\boldsymbol{\Sigma}$ is the covariance matrix, which characterizes the spread of the peak in different directions and can be any symmetric positive-definite matrix. A 2×2 symmetric positive-definite matrix can be defined by three scalar parameters via orthogonal diagonalization,

$$\Sigma = QDQ^T , \tag{4.36}$$

where

$$\begin{aligned} Q &= \begin{pmatrix} \cos\gamma & \sin\gamma \\ -\sin\gamma & \cos\gamma \end{pmatrix} , \\ D &= \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} , \end{aligned} \tag{4.37}$$

and $\lambda_1, \lambda_2 > 0$ and $\gamma \in [-\pi/2, \pi/2)$. We can thus write the Gaussian as a function of its argument and the six parameters defining it: $f(\boldsymbol{x}; A, \mu_x, \mu_y, \lambda_1, \lambda_2, \gamma)$.

[3] This is the first occasion where we actually *need* the true lattice vectors $\boldsymbol{a}_k$. We would not be able to construct a lattice of firing fields for the template cell using autocorrelogram peak vectors $\boldsymbol{p}_k$.

The central peak in the autocorrelogram is defined in the same way as the ring of six peaks, as the connected region in the center of the autocorrelogram with values r_{ij} larger than the autocorrelation criterion r^c_{min} (which is 0.2 for most cells). The Gaussian is fitted by the method of least squares, minimizing the cost function

$$S = \sum (r_{ij} - f(\boldsymbol{x}_{ij}; A, \mu_x, \mu_y, \lambda_1, \lambda_2, \gamma))^2 \ , \tag{4.38}$$

subject to the mentioned constraints on λ_1, λ_2 and γ, as well as the obvious constraint $A > 0$. The sum is taken over the bins belonging to the connected region around the center peak. We already know that we will find $\mu_x = \mu_y = 0$ due to the central symmetry of the autocorrelogram, and for the present purpose the scale factor A is not of further interest. We thus consider the shape of the firing field of cell c fully defined by the fitted covariance matrix $\boldsymbol{\Sigma}^c$.

The average of n independent multivariate random variables has covariance matrix equal to the average of the covariance matrices of each of the variables. In analogy with this, we define the shape of the firing field of the template cell using the average of the covariance matrices of the firing fields of the individual cells,

$$\boldsymbol{\Sigma}^t = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{\Sigma}^c \ . \tag{4.39}$$

We construct the firing rate map of the template cell by placing a Gaussian with covariance matrix $\boldsymbol{\Sigma}^t$ at each of the lattice points defined by the lattice vectors $\boldsymbol{a}^t_k$. In doing this, we do not have to confine the rate map to the experimental environment. In fact, we can benefit greatly from extending the firing pattern significantly, as this allows us to compute cross-correlograms between the real cell firing rates and the template firing rate without having the number of overlapping bins decrease as the displacement increases. The template firing rate map is therefore constructed in an environment that is four times as long and wide as the experimental environment, thus being divided into 168 × 168 bins. An example of a template firing rate is shown in Figure 4.4.

Defining the grid phase

We define the grid phase of a cell by computing the cross-correlogram between its firing rate and the template firing rate, and locating the

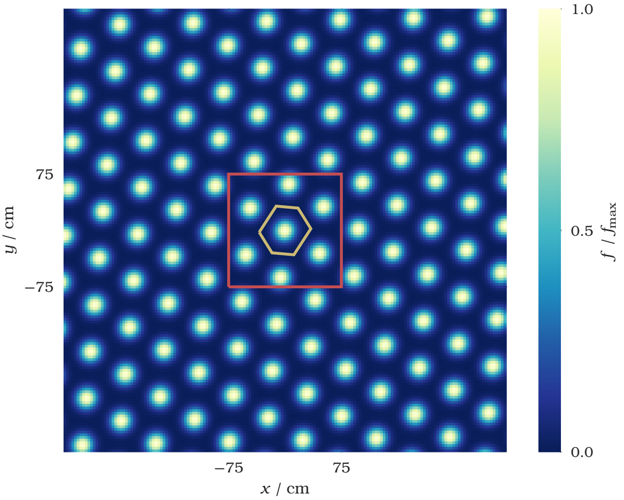


Figure 4.4: Template firing rate for the k-means module that the cell from Figure 2.1 belongs to. The edges of the actual experimental environment are shown in red, and the boundary of the Voronoi cell of the lattice defining the firing pattern is shown in yellow. The extension of the template firing pattern beyond the experimental environment lets cross-correlograms between real rate maps and the template be computed without reduction in the number of overlapping bins at all relevant displacements.

peak closest to the center of the correlogram, labeled $\boldsymbol{\delta}^c = (\delta_x^c, \delta_y^c)$ for cell c. The peaks are located in the same way as the autocorrelogram peaks, by computing the center of mass of the connected regions of values higher than the correlation criterion r_{min} for the cell. Because the template firing rate by definition satisfies perfect lattice periodicity and extends such that decline in the number of overlapping bins is avoided, the cross-correlogram must satisfy the same periodicity as the template, and hence the peak closest to the center must lie within the Voronoi cell of the lattice.[4] Thus, the template cell provides a periodically tiling window in which the grid phases are defined, without requiring additional steps such as renormalizing the phases individually to fit

[4] We could also have used another unit cell of the lattice and arrived at the exact same results, but using the Voronoi cell makes it easy to pick the desired peak in the each cross-correlogram: it is the one closest to the center.

into a common window, due to the slightly varying grid scale of the different cells in the module, as was done for the relative phases in [10].

Computing the phases of all the cells in a module thus results in a spatial point pattern that is periodic in nature. From Chapter 3, we know a great deal about how to analyze such patterns.

4.5 Analyzing the phase patterns

We want to investigate whether the assumption that the grid cells in a module cover the space of possible phases uniformly is consistent with experimental data. We may thus take as our null hypothesis that the phase patterns are uniform and independently distributed set of points, or in the language of point process statistics, that they were generated by a periodic binomial process. To investigate this hypothesis, we use the L-test from Section 3.8 with the L-function estimator for periodic point patterns defined by Equations (3.37) and (3.38). We generate 1000 simulations of uniformly and independently distributed points in the same window as the phases to determine the p-value for the observed phase pattern under the null hypothesis. The value for r_{min} used in all tests is $r_{\text{min}} = 1.05/(r_{\text{max}} * \lambda)$, a value that is found by trial and error and seems to work well.

To gain additional understanding of the characteristics of the phase pattern, we also plot the L-function and pair correlation function of the patterns, overlaid on simulation envelopes generated with $alpha = 0.05$, such that they contain the central 95 % of the distribution of function values at each value of r. The bandwidth in the estimator for the pair correlation function is set to $h = 0.2/\sqrt{\lambda}$.

To get a visual indication of the distribution of firing fields in the module, in a way that does not depend on our definition of the grid phase, we define the *stacked firing rate* as

$$s_{ij} = \frac{1}{n} \sum_{c=1}^{n} \frac{f_{ij}^{c}}{\overline{f}^{c}} , \tag{4.40}$$

where f_{ij}^{c} is the firing rate map of cell c, and $\overline{f}^{c}$ is its mean. Thus, the stacked firing rate is defined, bin for bin across the experimental environment, as the average of all the firing rates of the cells from the module in that bin, normalized such that the mean firing rate of each cell is 1.

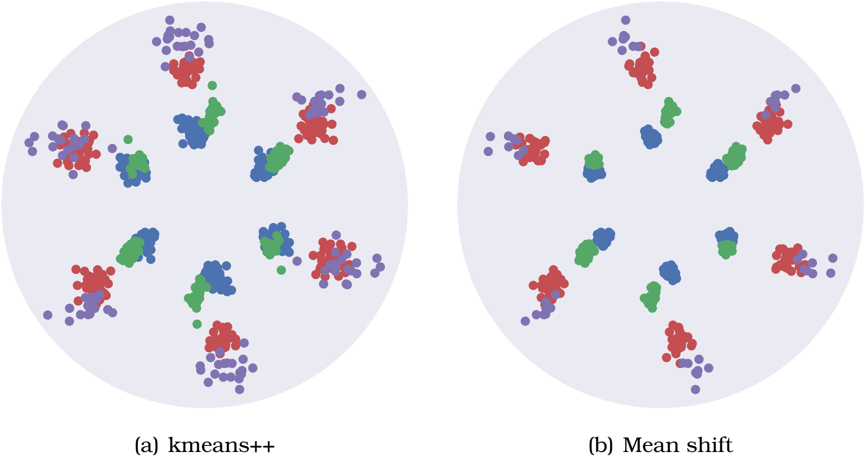
The pair correlation function and stacked firing rate are helpful aids in the discussion of the results, but we will not perform any quantitative analysis based on either of them.

5

Results

The results of the analysis of the phase pattern in each module will be presented using a similar set of figures for each module, starting from Figure 5.2. The explanation for the figures is therefore stated here, leaving the caption for comments pertaining to each particular figure:

- In the phase pattern plots, the blue dots show the grid phases, the black line is the boundary of the phase space window, and the green dots show the periodic extension of the pattern.
- The L-function plots are based on 1000 simulated patterns. The blue curve and shaded area show the mean and central 95 % envelope of the L-function estimators from the simulations, and the green curve shows the estimator from the phase pattern. The dashed red line shows the theoretical value $L(r) = r$. The yellow vertical lines show the values of r_{min} and r_{max} used in Equation (3.42).
- The test statistic plots are based on the same 1000 simulated patterns. The blue curve and shaded area show a kernel density estimate and histogram of the of the test statistic τ from the simulations, and the green line shows the value of τ from the phase pattern. The p-value of τ given the simulated distribution is stated in the caption to each figure.
- In the pair correlation plots, the blue curve, shaded area and green curve have the same meaning as in the L-function plot, only replacing the L-function estimator with the pair correlation function estimator. The dashed red line shows the theoretical value $g(r) = 1$.


(a) kmeans++

(b) Mean shift

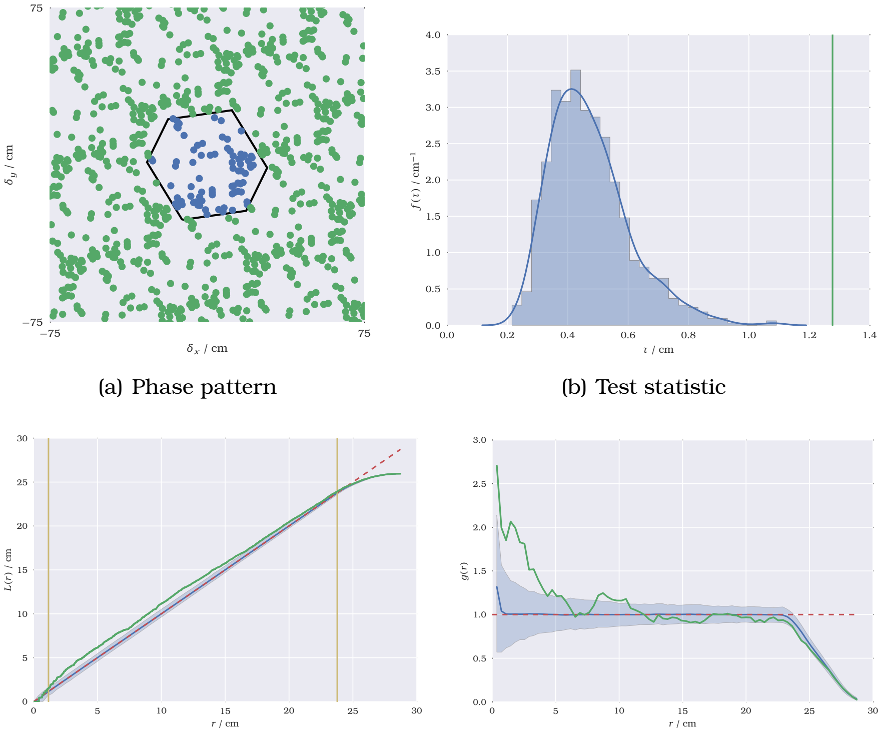
Figure 5.1: Result of clustering the grid cells into modules. (a): The location of all six lattice vectors for all the cells in the whole dataset, color coded by the module the cell was assigned to. The modules are number 1–4 from the smallest scale to the largest scale. (b): The location of all six lattice vectors for the cells in the dataset that were assigned to a module by the mean shift algorithm, color coded by the module the cell was assigned to. The modules are numbered 1–4 from the smallest scale to the largest scale.

Table 5.1: Properties of modules resulting from *k*-means clustering. The mean grid scale is defined as the grid scale of the template firing pattern of the module.

Module	Number of cells	Mean grid scale /cm
1	80	49.4
2	38	59.2
3	38	94.1
4	20	108.2

5.1 Modules from *k*-means clustering

The result of the kmeans++ clustering is shown in Figure 5.1(a). The clustering provides four modules with the properties described in Table 5.1. The analysis of the grid phases in these modules is presented in Figures 5.2 to 5.9.



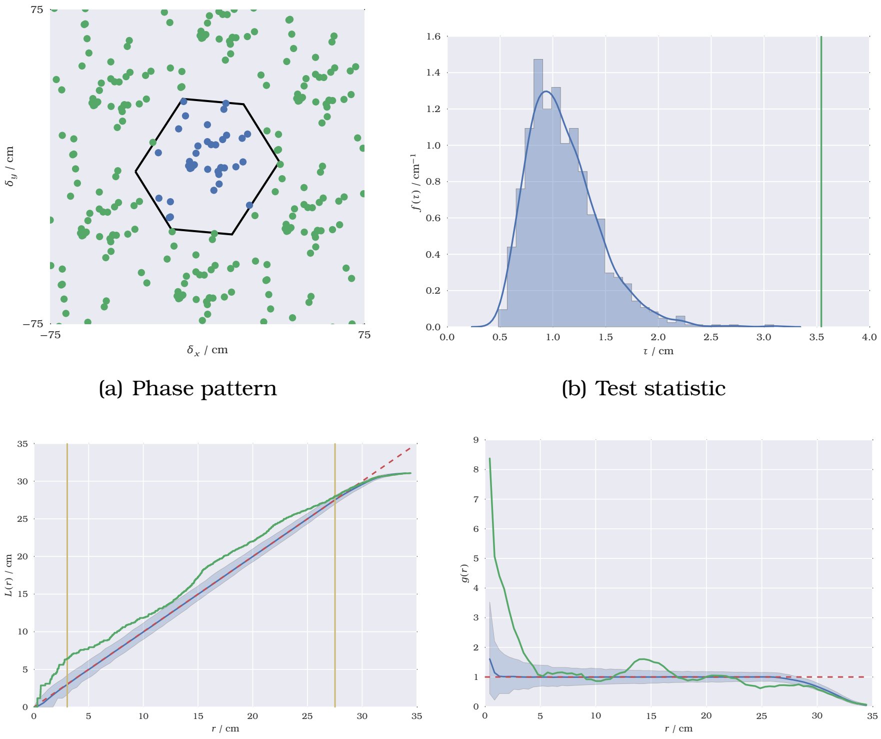
(a) Phase pattern

(b) Test statistic

(c) L-function

(d) Pair correlation function

Figure 5.2: **Phase pattern analysis, k-means module 1.** (a): The phase pattern has a much higher density of phases in the south and east of the window than in the north and west. (b): The p-value of τ computed from the pattern is 0.0, pointing strongly towards rejection of the periodic binomial process. (c): The L-function estimator from the phase pattern lies above the simulation envelope for almost all values of r, pointing towards a clustered pattern. (d): The pair correlation function indicates a strong tendency towards clustering at the smallest length scales, with an additional significant peak around 10 cm.



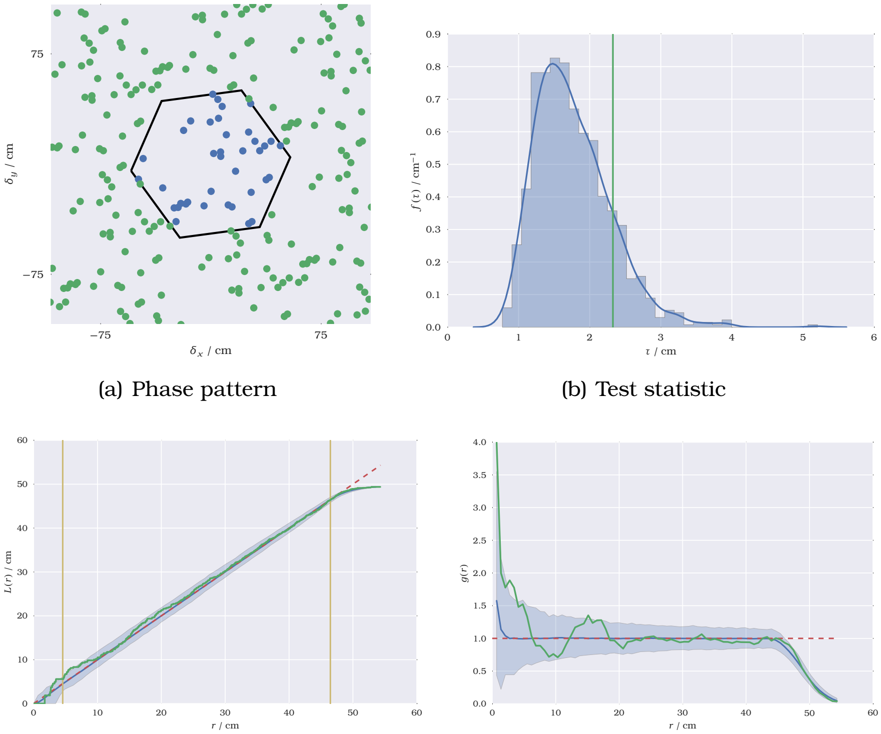
(a) Phase pattern

(b) Test statistic

(c) L-function

(d) Pair correlation function

Figure 5.3: **Phase pattern analysis, k-means module 2.** (a): The phase pattern has a tight cluster near the center and almost no points towards the southwest. (b): The p-value of τ computed from the pattern is 0.0, pointing strongly towards rejection of the periodic binomial process. (c): The L-function estimator from the phase pattern lies above the simulation envelope for almost all values of r, pointing towards a clustered pattern. (d): The pair correlation function indicates a strong tendency towards clustering at the smallest length scales, with an additional significant peak around 15 cm.



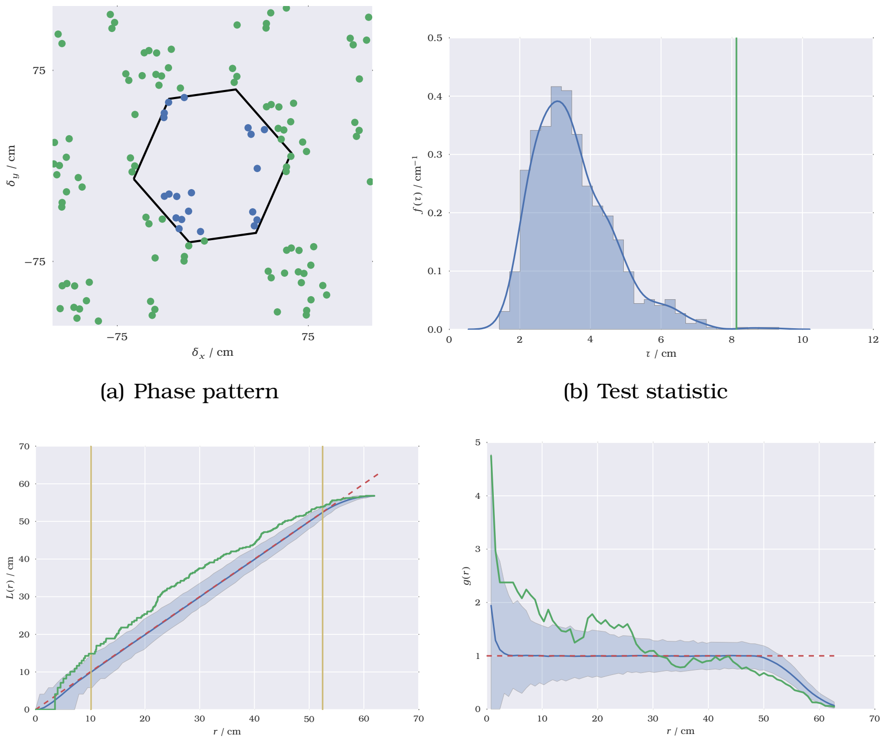
(a) Phase pattern

(b) Test statistic

(c) L-function

(d) Pair correlation function

Figure 5.4: **Phase pattern analysis, k-means module 3.** (a): The phase pattern is quite evenly spread out, but thinner in the northwest of the window than elsewhere. (b): The p-value of τ computed from the pattern is 0.139. This does not support rejecting the periodic binomial process at any relevant significance level. (c): The L-function estimator from the phase pattern lies more or less inside the simulation envelope for almost all values of r. This is highly consistent with the periodic binomial process. (d): The pair correlation function generally varies within the range of what one could expect for a periodic binomial process, although it tends towards the clustered end of the spectrum at the shortest length scales.



(a) Phase pattern

(b) Test statistic

(c) L-function

(d) Pair correlation function

Figure 5.5: **Phase pattern analysis, k-means module 4.** (a): The phases are only found close to the corners of the window, but the periodic extension reveals that they are all more or less part of the same cluster, with empty space in between. (b): The p-value of τ computed from the pattern is 0.003. This supports rejecting of the periodic binomial process at common significance levels such as $\alpha = 0.05$ and $\alpha = 0.005$. (c): The L-function estimator from the phase pattern lies above the simulation envelope for almost all values of r between r_{min} and r_{max}, pointing towards a clustered pattern. (d): The pair correlation function indicates a tendency towards clustering from the smallest length scales up to around 25 cm.

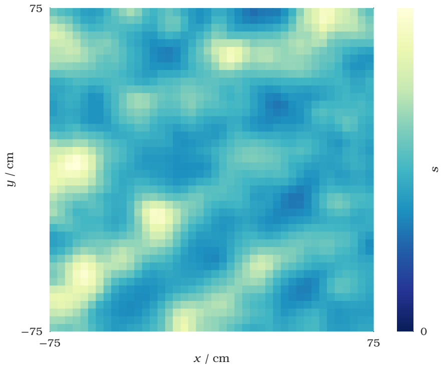


Figure 5.6: **Stacked firing rate, *k*-means module 1.** The regions of highest activity seem to form parallel lines along one of the grid axes of the module. The regions of lowest firing rate almost form a grid pattern themselves, possibly indicating that the module lacks cells in a particular region of phase space. In general, it is easy to recognize common features in this plot and Figure 5.2(a), as we could reasonably expect.

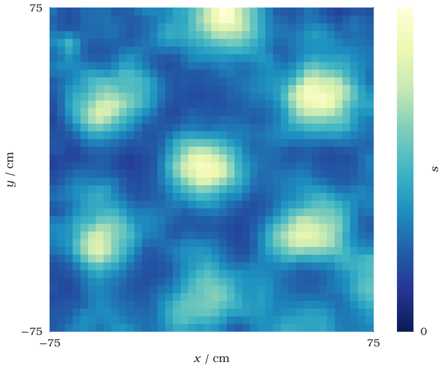


Figure 5.7: **Stacked firing rate, *k*-means module 2.** The regions of highest activity form an almost perfect grid cell pattern, possibly indicating that the module is dominated by cells in a particular region of phase space.

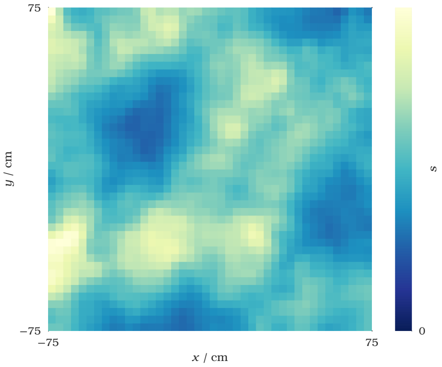


Figure 5.8: **Stacked firing rate, k-means module 3.** As in Figure 5.6, we see a grid pattern made up of regions of low average firing rate, possibly indicating that this module lacks cells in a particular region of phase space.

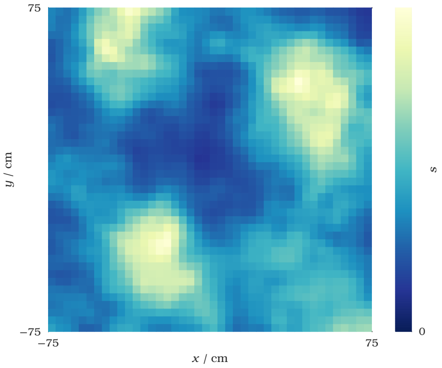


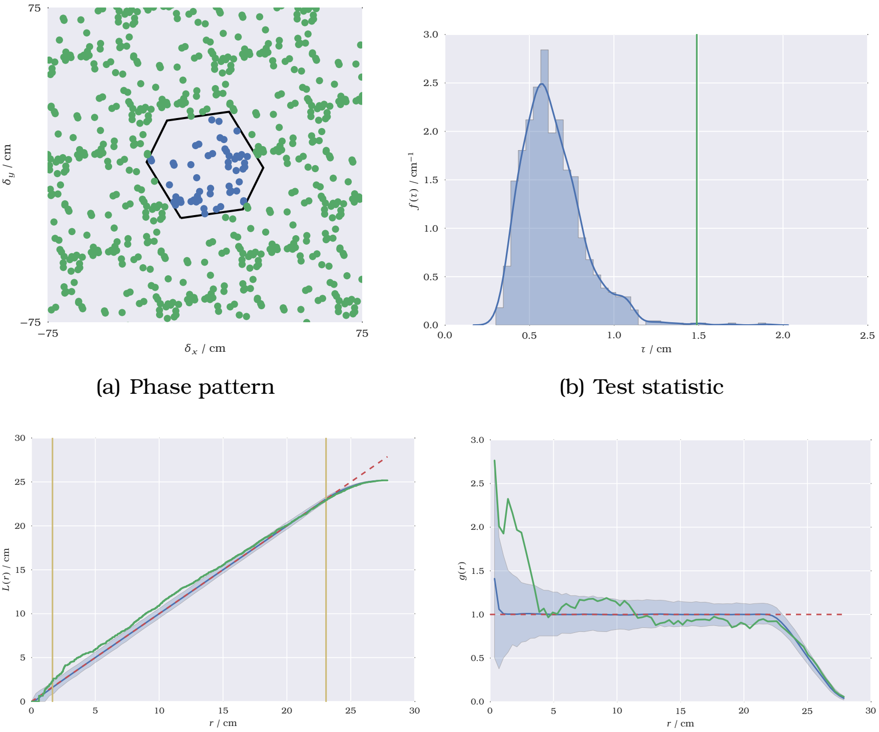
Figure 5.9: **Stacked firing rate, k-means module 4.** Like in Figure 5.7, the regions of high activity in this module form a grid pattern, possibly indicating that this module is dominated by cells in a particular region of phase space.

Table 5.2: Properties of modules resulting from mean shift clustering. The mean grid scale is defined as the grid scale of the template firing pattern of the module.

Module	Number of cells	Mean grid scale /cm
1	55	48.0
2	31	59.7
3	31	94.8
4	8	112.2

5.2 Modules from mean shift clustering

The result of the mean shift clustering is shown in Figure 5.1(b). The clustering provides four modules with the properties described in Table 5.2. The analysis of the grid phases in these modules is presented in Figures 5.10 to 5.17.



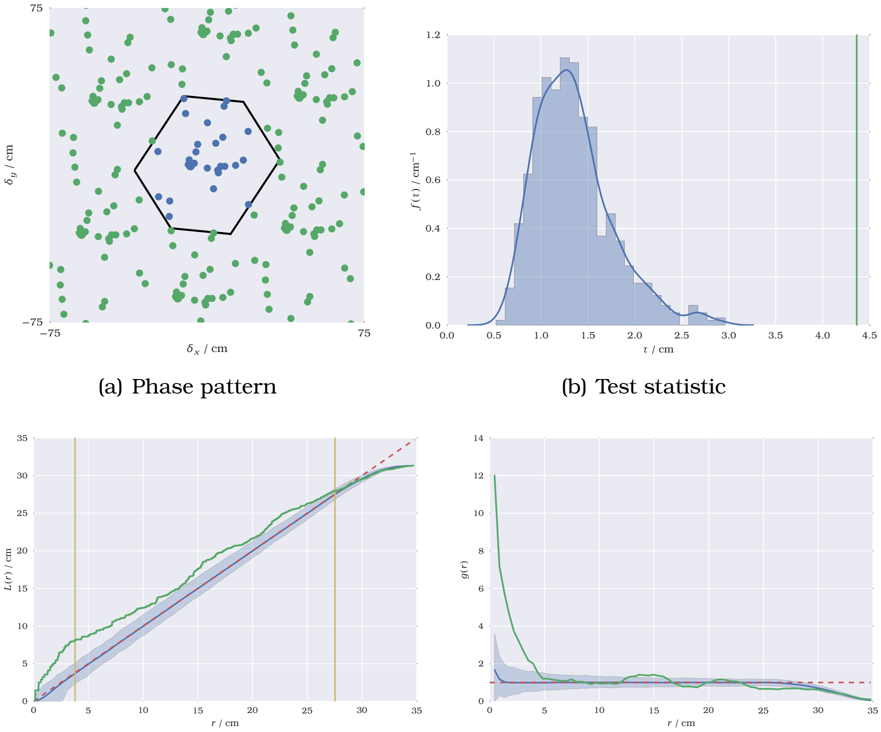
(a) Phase pattern

(b) Test statistic

(c) L-function

(d) Pair correlation function

Figure 5.10: **Phase pattern analysis, mean shift module 1.** (a): The phase pattern looks similar to Figure 5.2, but is certainly less dense. (b): The p-value of τ computed from the pattern is 0.004. pointing strongly towards rejection of the periodic binomial process. This supports rejecting of the periodic binomial process at common significance levels such as $\alpha = 0.05$ and $\alpha = 0.005$. (c): The L-function estimator from the phase pattern lies just above the simulation envelope for almost the first half of r-values, pointing towards a clustered pattern. (d): The pair correlation function indicates a strong tendency towards clustering at the smallest length scales.



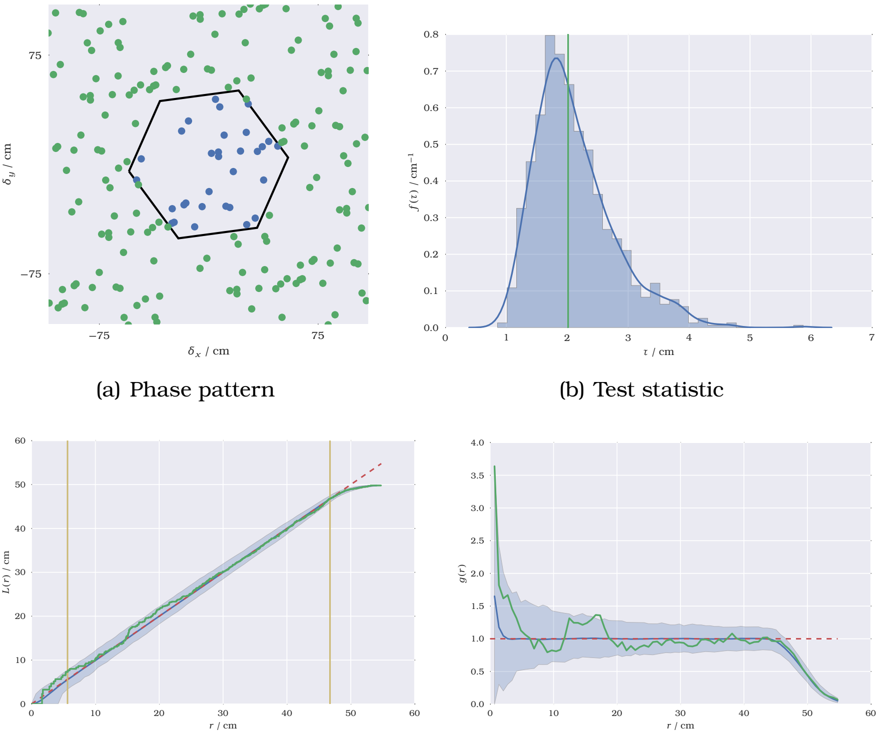
(a) Phase pattern

(b) Test statistic

(c) L-function

(d) Pair correlation function

Figure 5.11: **Phase pattern analysis, mean shift module 2.** (a): Qualitatively, this phase pattern looks very similar to the one in Figure 5.3. (b): The p-value of τ computed from the pattern is 0.0, pointing strongly towards rejection of the periodic binomial process. (c): The L-function estimator from the phase pattern lies above the simulation envelope for almost all values of r, pointing towards a clustered pattern. (d): The pair correlation function indicates a strong tendency towards clustering at the smallest length scales.



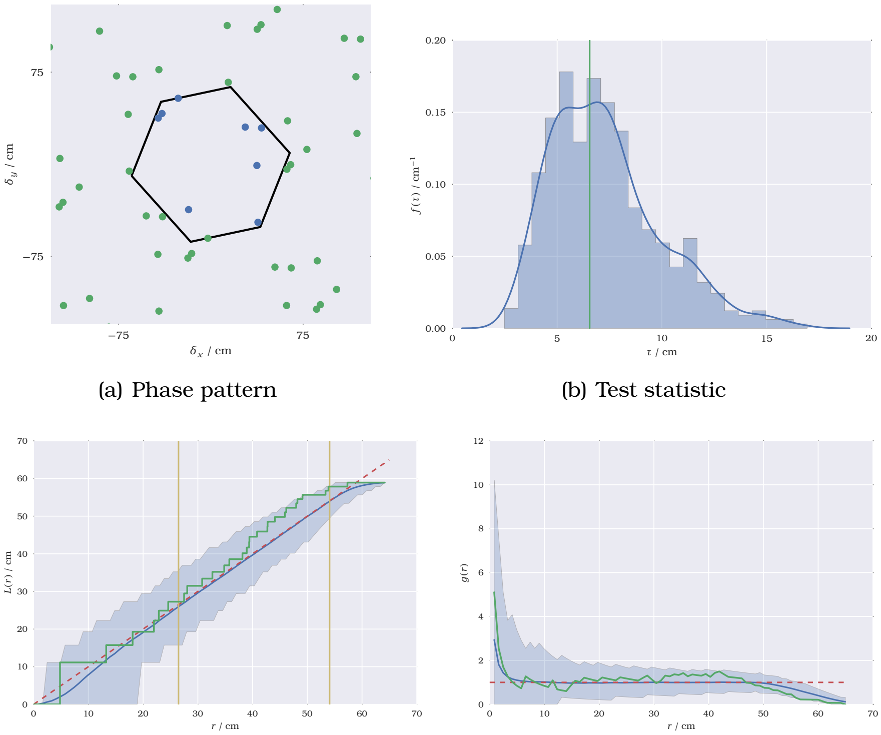
(a) Phase pattern

(b) Test statistic

(c) L-function

(d) Pair correlation function

Figure 5.12: **Phase pattern analysis, mean shift module 3.** (a): The phase pattern is even more evenly spread out than in Figure 5.4. (b): The p-value of τ computed from the pattern is 0.481, placing right in the middle of the simulated patterns, and giving no reason to reject the periodic binomial process. (c): The L-function estimator from the phase pattern lies inside the simulation envelope for all values of r. This is fully consistent with the periodic binomial process. (d): The pair correlation function generally varies within the range of what one could expect for a periodic binomial process.



(a) Phase pattern

(b) Test statistic

(c) L-function

(d) Pair correlation function

Figure 5.13: **Phase pattern analysis, mean shift module 4.** (a): The eight phases from this module seem to stick close to the boundary of the window, but look fairly evenly spread out when taking the periodic extension into account. (b): The p-value of τ computed from the pattern is 0.57. This does not suggest rejection of the periodic binomial process. (c): The L-function estimator from the phase pattern lies within the simulation envelope for all values of r, fully consistent with a periodic binomial process. (d): The pair correlation function indicates a very close correspondence between this pattern and what we could expect from the periodic binomial process.

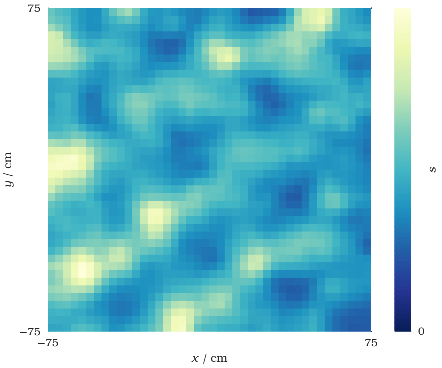


Figure 5.14: **Stacked firing rate, mean shift module 1.** The lines of high activity from Figure 5.6 are still present, and the grid pattern formed by low-activity regions is even more pronounced.

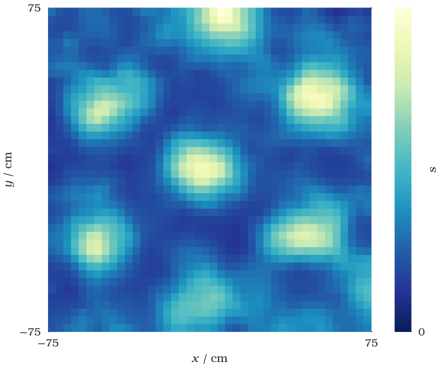


Figure 5.15: **Stacked firing rate, mean shift module 2.** Like in Figure 5.8, we see an almost perfect grid cell pattern formed by the regions where the module as a hole is most active.

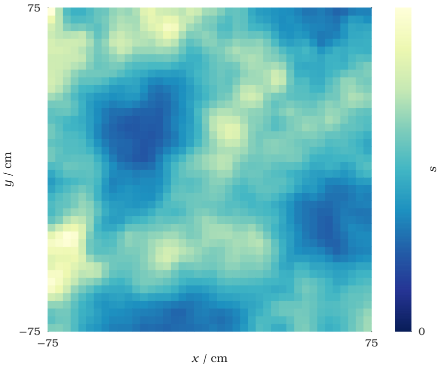


Figure 5.16: **Stacked firing rate, mean shift module 3.** As in Figure 5.8, we see a grid pattern made up of regions of low average firing rate.

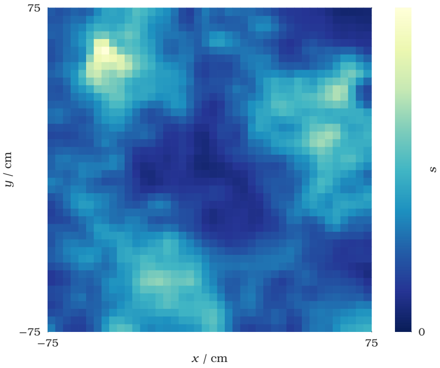


Figure 5.17: **Stacked firing rate, mean shift module 4.** Like in Figure 5.9, the regions of high activity in this module seem to form a grid pattern.

6 Discussion

The results presented in the previous chapter provide a very strong indication that the phases of grid cells in a module are not necessarily uniformly distributed. On the contrary, they show that there exist modules with an abundance of cells with similar phases. In particular, the analysis presented in Figures 5.2 and 5.3 present extremely convincing arguments for rejecting the assumption of uniformity for modules 1 and 2 in this dataset – for these modules, the test statistic quantifying the deviation from uniformity virtually goes through the roof! Moreover, the similar findings for the mean shift clustered versions of the modules, presented in Figures 5.10 and 5.15, show that the conclusion is independent of the particular details of the clustering used and which modules fringe cells are assigned to – the degree of non-uniformity is very similar in the hard core of archetypical cells of each module, and in the modules in their most inclusive definition.

For modules with larger scale, the situation is slightly different. In this dataset, no evidence against uniformity is discovered in module 3, and for module 4 the evidence differs between the two types of clustering (however, the failure to provide evidence against uniformity in the mean shift clustered module 4 should perhaps not be taken too seriously – as evidenced by the very wide envelope in Figure 5.13(c), we should not really expect a statistically significant rejection of anything with only 8 samples). Hence, the main finding in this analysis is not that grid cell modules *must* have non-uniform coverage of phase space. The significant finding is rather that *there exist* grid cell modules with non-uniform coverage of phase space.

6.1 Implications

If this conclusion stands up to scrutiny, it will have considerable implications for the credibility of different models for grid cell networks. As explained in Chapter 2, models based only on continuous attractor networks need uniform phase coverage to create grid cell-like dynamics. Such models will therefore have to be modified to accommodate the new evidence. At the same time, the relative standing and plausibility of alternatives such as adaptation models may strengthened.

Since the findings here only suggest that grid cell modules *can* have non-uniform phase space coverage, not that they *must*, there is no immediate need for models that explicitly predict non-uniform phase space coverage – they only have to *allow* it. Further analysis using data from several animals will be needed to formulate more precise model requirements.

6.2 Possible confounders

The analysis leading to these results is based on certain assumptions. Crucially, we have trusted that Stensola et al. were successful in eliminating all duplicates of the same cell from the data in their original analysis [9]. Including multiple recordings of the same cell in the same environment would obviously lead to phase space clustering and might contribute towards a result similar to the one observed here.

We have also assumed the absence of any topographical organization of phases that could be resolved by the multisite tetrode recordings, such that the dataset can be regarded a random and independent sample from the total grid cell population without bias towards particular phases. If, instead, cells recorded on the same tetrode tended to be close in phase, this assumption would not be justified, and we would not be able to draw conclusions about the distribution of phases in the total grid cell population based on our result. Experimental evidence supports our assumption by indicating that grid phase lacks topographical organization [3, 7, 10].

Finally, we have assumed that the absolute grid phase is comparable over multiple recording sessions spanning several months. Previous results suggest that absolute phase is less stable than pairwise relative phase across multiple recordings [10]. However, unless the variations in absolute phase turn out to be in the form of systematic drift in some direction, this should not be a problem – random perturbations of absolute phases should not increase the probability of observing phase clustering when comparing cells from different sessions.

6.3 Possible structure in the phase distribution

If the conclusion is drawn that the distribution of grid phases may be non-uniform, the natural follow-up is to ask whether one can identify any particular kind of structure in the distribution. We will not attempt a rigorous approach to this question here, but based on the phase patterns and stacked firing rates we can speculate a little bit and perhaps stumble upon some fruitful topics for further research.

A striking feature of the stacked firing rate in Figure 5.6, which is also obvious in the corresponding phase pattern in Figure 5.2(a) when one knows what to look for, is the organization of high activity in bands parallel to one of the grid orientation axes, with corresponding bands of low activity in between. Similar features are evident in the phase patterns for all modules: a band of low phase density strikes through each phase pattern perpendicular to one pair of window edges, and thus parallel to a grid orientation axis. This may indicate a systematic anisotropy wherein the phase distribution is uniform in a dimension parallel to one of the grid axes, and non-uniform in the perpendicular direction. The anisotropy may perhaps even be related to the distortion or shearing of the grid, since for both modules 1 and 2, the band of low phase density runs perpendicular to the direction in which the window, and hence the grid pattern, is most stretched.

A completely different kind of structure, only relevant for modules with large grid scale, can be guessed from the stacked firing rate Figure 5.9 and the phase pattern Figure 5.5(a). We see that the module has low activity in the center of the environment, and correspondingly all grid phases are found close to the window boundary. But for grid cells with scales this large, having a firing field at the center of the environment would also mean that this would be the the only firing field located within the environment, while cells with other phases easily fit three fields in the box. If the dynamics of grid cell networks favorize configurations where cells have more than one firing field within the environment, this would naturally lead to a non-uniform phase distribution for modules with large scale.

6.4 Comparison with previous results

This is not the first time the distribution of grid cell phases is investigated, and both Hafting et al. and Yoon et al. have failed to find evidence for non-uniformity [7, 10].

An obvious difference between this work and the previous results is the size of the dataset. For example, Yoon et al. assessed the grid phase distribution based on 223 cell pairs, while here, the 80 cells in module 1 alone give 6320 cell pairs. It is therefore not surprising that we are able to draw new conclusions from this dataset.

Methodological differences are also likely to play a part. Hafting et al. test the uniformity in one dimension of the absolute value of the relative phase between pairs of cells, what we would write $|\boldsymbol{\delta}^c - \boldsymbol{\delta}^d|$. But the uniformity of these quantities is not equivalent to the uniformity of $\boldsymbol{\delta}^c$ in two dimensions. The latter condition would imply a greater number of large absolute phase displacements compared to small.

Yoon et al. choose to combine pairwise relative phases from many different animals and experiments in their analysis.[1] This is accomplished by projecting all the relative phases into a canonical lattice unit cell that is independent of grid scale, orientation, and deformation. This is an interesting approach if one wants to identify universal structures common in all animals, that scale with grid scale in the phase distribution, such as a tendency of relative phase differences to have a length that is a particular fraction of the grid scale (what we would identify as a bump in the pair correlation functions of all modules), or a particular direction with respect to the grid orientation axes. However, it does not answer the more fundamental question, namely if the grid cells in a module always cover the phase space uniformly or not. In particular, if phases tend to cluster in ways that show little repeatable structure across different animals, combining the relative phases from different animals in one analysis will tend to smooth out the clustering and give an impression of uniformity.

[1] Of course, only pairs of cells from the same animal are used to compute pairwise relative phases.

Hence, the present findings answer a slightly different question than the previous results, and there is no need to consider the new findings and the previous results incompatible.

6.5 Suggestions for further work

Although potentially significant, the results presented here also have a preliminary character: they are based on a single animal, and several objections such as the mentioned confounders can be raised. Further work on this topic should therefore seek to answer some of the following questions:

- Can these results be reproduced with data from other animals?
- Is there some topographical order in the grid phases in this dataset that can explain the observed clustering of grid phases?
- How appropriate is the comparison of cells from different recording sessions in phase space? Can similar results be obtained by only considering cell pairs from the same recording session?
- Is there any identifiable common structure in the distribution of grid phases?

In addition, the L-test should be complemented by another statistical test to confirm the validity of the findings. The theory of point processes has a lot more to offer, for example the J-test [16].

6.6 Conclusion

Using a novel method for computing the absolute spatial phase of a grid cell, and applying tools from the field of point process statistics to analyze the distribution of phases within a module, we have found convincing evidence that not all grid cell modules exhibit uniform coverage of the spatial phase space by their cells. This is a new discovery, which contrasts previous results without being incompatible with them. It has implications of considerable significance for the development of theoretical models for grid cell networks, in particular requiring adjustments to models based on continuous attractor networks.

Appendices

A

Bravais lattices

A Bravais lattice is an infinite set of points arranged in a periodic pattern that looks exactly the same from any of the points [26]. Any such lattice can be characterized by a finite set of vectors known as the *primitive vectors*. There are as many primitive vectors as the dimension of the space in which the pattern is embedded, and each point in the lattice can be written as a sum of an integer multiple of each of the primitive vectors. For example, in two dimensions (which is the case most relevant for firing patterns from grid cells in non-flying animals) a Bravais lattice is defined by two primitive vectors $\boldsymbol{a}_1, \boldsymbol{a}_2$, and the translation between two arbitrary points in the lattice can be written

$$\boldsymbol{R} = n_1 \boldsymbol{a}_1 + n_2 \boldsymbol{a}_2 \ , \tag{A.1}$$

with n_1, n_2 integer. The vectors R are called the *lattice vectors*. Note that there are many possible choices of primitive vectors for the same lattice.

A.1 Primitive unit cells

A primitive unit cell of a Bravais lattice is any region that upon repeated translation with every lattice vector R tiles the space in which the lattice is embedded, without overlap. There are several possible primitive unit cells for a lattice, all having the same area. The two most commonly used are the following:

- *Parallelepiped:* The parallelepiped (or parallelogram in two dimensions) spanned by a set of primitive vectors for the lattice is always a primitive unit cell.

- *Voronoi cell:* The Voronoi cell is the set of points closer to a given lattice point than to any other lattice point. For Bravais lattices this is also known as the Wigner-Seitz cell.

For a two-dimensional Bravais lattice, the Voronoi cell is an irregular hexagon with reflection symmetry through its centroid (except in the case of a rectangular lattice, for which the Voronoi cell is a rectangle). The Voronoi cell and a parallelogram cell for a perfect triangular lattice are shown in Figure A.1.

A.2 Perfect triangular lattices

A perfect triangular lattice is usually characterized by primitive vectors of equal length, $|\boldsymbol{a}_1| = |\boldsymbol{a}_2| = l$, at an angle $\varphi = \pi/3 = 60°$ or $\varphi = 2\pi/3 = 120°$ to each other (these two descriptions are equivalent and interchangeable: if $\boldsymbol{a}_1, \boldsymbol{a}_2$ are primitive vectors for a triangular lattice and have $\varphi = \pi/3$, the vectors $\boldsymbol{a}'_1 = \boldsymbol{a}_1, \boldsymbol{a}'_2 = \boldsymbol{a}_2 - \boldsymbol{a}_1$ are primitive vectors for the same lattice and have $\varphi' = 2\pi/3$). A triangular Bravais lattice is shown in Figure A.1.

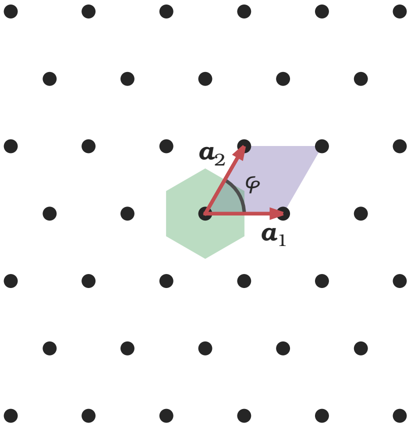


Figure A.1: Finite subset of a triangular Bravais lattice. The arrows show one possible set of primitive vectors for this lattice. The vectors are of equal length, and the angle between them is $\varphi = \pi/3$. The Voronoi cell is shaded green, while the parallelogram unit cell spanned by the primitive vectors is shaded purple.

Bibliography

[1] McNaughton, B. L., Battaglia, F. P., Jensen, O., Moser, E. I., and Moser, M.-B. “Path integration and the neural basis of the ’cognitive map’”. In: *Nature Reviews Neuroscience* 7.8 (Aug. 2006), pp. 663–678.

[2] Moser, E. I., Kropff, E., and Moser, M.-B. “Place cells, grid cells, and the brain’s spatial representation system”. en. In: *Annual Review of Neuroscience* 31 (Jan. 2008), pp. 69–89.

[3] Moser, E. I., Roudi, Y., Witter, M. P., Kentros, C., Bonhoeffer, T., and Moser, M.-B. “Grid cells and cortical representation.” In: *Nature reviews. Neuroscience* 15.7 (July 2014), pp. 466–481.

[4] O’Keefe, J. and Dostrovsky, J. “The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat”. In: *Brain Research* 34.1 (Nov. 1971), pp. 171–175.

[5] O’Keefe, J. “Place units in the hippocampus of the freely moving rat”. In: *Experimental Neurology* 51.1 (Jan. 1976), pp. 78–109.

[6] Fyhn, M., Molden, S., Witter, M. P., Moser, E. I., and Moser, M.-B. “Spatial representation in the entorhinal cortex”. In: *Science* 305.5688 (Aug. 2004), pp. 1258–1264.

[7] Hafting, T., Fyhn, M., Molden, S., Moser, M.-B., and Moser, E. I. “Microstructure of a spatial map in the entorhinal cortex.” In: *Nature* 436.7052 (Aug. 2005), pp. 801–806.

[8] Stensola, T., Stensola, H., Moser, M.-B., and Moser, E. I. “Shearing-induced asymmetry in entorhinal grid cells”. In: *Nature* 518.7538 (Feb. 2015), pp. 207–212.

[9] Stensola, H., Stensola, T., Solstad, T., Frøland, K., Moser, M.-B., and Moser, E. I. “The entorhinal grid map is discretized.” In: *Nature* 492.7427 (Dec. 2012), pp. 72–78.

[10] Yoon, K., Buice, M. A., Barry, C., Hayman, R., Burgess, N., and Fiete, I. R. “Specific evidence of low-dimensional continuous attractor dynamics in grid cells”. en. In: *Nature Neuroscience* 16.8 (Aug. 2013), pp. 1077–1084.

[11] Fyhn, M., Hafting, T., Treves, A., Moser, M.-B., and Moser, E. I. “Hippocampal remapping and grid realignment in entorhinal cortex”. In: *Nature* 446.7132 (Mar. 2007), pp. 190–4.

[12] Amari, S.-i. “Dynamics of pattern formation in lateral-inhibition type neural fields”. In: *Biological Cybernetics* 27.2 (1977), pp. 77–87.

[13] Fuhs, M. C. “A Spin Glass Model of Path Integration in Rat Medial Entorhinal Cortex”. In: *Journal of Neuroscience* 26.16 (Apr. 2006), pp. 4266–4276.

[14] Burak, Y. and Fiete, I. R. “Accurate path integration in continuous attractor network models of grid cells”. In: *PLoS computational biology* 5.2 (Feb. 2009), e1000291.

[15] Kropff, E. and Treves, A. “The emergence of grid cells: Intelligent design or just adaptation?” In: *Hippocampus* 18.12 (Jan. 2008), pp. 1256–1269.

[16] Illian, J., Penttinen, A., Stoyan, H., and Stoyan, D. *Statistical Analysis and Modelling of Spatial Point Patterns*. West Sussex: John Wiley & Sons, Ltd, 2008.

[17] Grünbaum, B. and Shephard, G. C. “Tilings with congruent tiles”. In: *Bulletin of the American Mathematical Society* 3.3 (Nov. 1980), pp. 951–974.

[18] Baddeley, A., Gregori, P., Mateu, J., Stoica, R., and Stoyan, D., eds. *Case Studies in Spatial Point Process Modelling*. New York: Springer New York, 2006.

[19] Fitzgibbon, A., Pilu, M., and Fisher, R. “Direct least squares fitting of ellipses”. In: *Proceedings of 13th International Conference on Pattern recognition*. Vol. 1. IEEE, 1996, pp. 253–257.

[20] Halíř, R. and Flusser, J. “Numerically Stable Direct Least Squares Fitting Of Ellipses”. In: *Proceedings of 6th International Conference in Central Europe on Computer Graphics and Visualization*. Vol. 1. 1998, pp. 125–132.

[21] Kanatani, K., Al-Sharadqah, A., Chernov, N., and Sugaya, Y. “Renormalization Returns: Hyper-renormalization and Its Applications”. In: *Computer Vision – ECCV 2012*. Ed. by Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., and Schmid, C. Vol. 7574. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, pp. 384–397.

[22] Weisstein, E. W. *Ellipse*. From *MathWorld*–A Wolfram Web Resource. URL: http://mathworld.wolfram.com/Ellipse.html (visited on 2015-06-29).

[23] Arthur, D. and Vassilvitskii, S. “k-means++: The Advantages of Careful Seeding”. In: *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, Jan. 2007, pp. 1027–1035.

[24] Comaniciu, D. and Meer, P. “Mean Shift: A Robust Approach Toward Feature Space Analysis”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.5 (May 2002), pp. 603–619.

[25] Pedregosa, F. et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[26] Ashcroft, N. W. and Mermin, N. D. *Solid State Physics*. Brooks Cole, 1976.