AVAILABLE AT
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®

Neural Networks 17 (2004) 5–27

Neural Networks

www.elsevier.com/locate/neunet

Self-organising continuous attractor networks with multiple activity packets, and the representation of space

S.M. Stringer[a], E.T. Rolls[a,*], T.P. Trappenberg[b]

[a]*Department of Experimental Psychology, Centre for Computational Neuroscience, Oxford University, South Parks Road, Oxford OX1 3UD, UK*
[b]*Faculty of Computer Science, Dalhousie University, 6050 University Avenue, Halifax, NS, Canada B3H 1W5*

Received 21 February 2003; accepted 13 June 2003

Abstract

'Continuous attractor' neural networks can maintain a localised packet of neuronal activity representing the current state of an agent in a continuous space without external sensory input. In applications such as the representation of head direction or location in the environment, only one packet of activity is needed. For some spatial computations a number of different locations, each with its own features, must be held in memory. We extend previous approaches to continuous attractor networks (in which one packet of activity is maintained active) by showing that a single continuous attractor network can maintain multiple packets of activity simultaneously, if each packet is in a different state space or map. We also show how such a network could by learning self-organise to enable the packets in each space to be moved continuously in that space by idiothetic (motion) inputs. We show how such multi-packet continuous attractor networks could be used to maintain different types of feature (such as form vs colour) simultaneously active in the correct location in a spatial representation. We also show how high-order synapses can improve the performance of these networks, and how the location of a packet could be read by motor networks. The multiple packet continuous attractor networks described here may be used for spatial representations in brain areas such as the parietal cortex and hippocampus.

Keywords: Continuous attractor neural networks; Multiple activity packets; Spatial representation; Idiothetic inputs; Path integration

1. Introduction

'Continuous attractor' neural networks are neural networks which are able to maintain a localised packet of neuronal activity representing the current state of an agent in a continuous space, for example head direction or location in the environment, without external sensory input (Amari, 1977; Taylor, 1999). They are useful in helping to understand the representation of head direction (Redish, Elga, & Touretzky, 1996; Skaggs, Knierim, Kudrimoti, & McNaughton, 1995; Stringer, Trappenberg, Rolls, & de Araujo, 2002b; Zhang, 1996), place (Redish & Touretzky, 1998; Samsonovich & McNaughton, 1997; Stringer, Rolls, Trappenberg, & de Araujo, 2002a), and in the primate hippocampus, spatial view (Stringer, Rolls, & Trappenberg, 2003a). Continuous attractor networks use excitatory recurrent collateral connections between the neurons to reflect the distance between the neurons in the state space (e.g. head direction space) of the agent. Global inhibition is used to keep the number of neurons in a bubble of activity relatively constant. In the applications of continuous attractor networks discussed above, where a network is required to represent only a single state of the agent (i.e. head direction, place or spatial view), it is appropriate for the continuous attractor networks to support only one activity packet at a time.

In this paper we propose that continuous attractor networks may be used in the brain in an alternative way, in which they support multiple activity packets at the same time. The stability of multiple activity packets in a single network has been discussed previously by, for example, Amari (1977) and Ermentrout and Cowan (1979). Ermentrout and Cowan (1979) analysed neural activity in a two-dimensional (2D) network, demonstrating the existence of a variety of doubly periodic patterns as solutions to the field equations for the net activity. Amari (1977) considered a continuous attractor neural network in which the neurons are mapped onto a one-dimensional (1D) space x, where there are

* Corresponding author. Tel.: +44-1865-271348; fax: +44-1865-310447.
E-mail address: edmund.rolls@psy.ox.ac.uk (E.T. Rolls).
Web pages: www.cns.ox.ac.uk

0893-6080/$ - see front matter © 2003 Elsevier Ltd. All rights reserved.
doi:10.1016/S0893-6080(03)00210-7

short range excitatory connections and longer range inhibitory connections between the neurons. If two activity packets are stimulated at separate locations in the same continuous attractor network, then the two packets may interact with each other. The neurons in the second packet will receive an input $s(x)$ from the first packet. In this case the second activity packet moves searching for the maximum of $s(x)$. The effect of the second packet on the first one is similar. Depending on the precise shape of the synaptic weight profile within the network, the effect of this interaction may be to draw the two packets together or repel them. If the two activity packets are far enough apart, then the gradient of the function $s(x)$ may be close to zero, and the two packets will not affect each other (Amari, 1977). However, in this paper we investigate a more general situation in which a single continuous attractor network can maintain multiple packets of activity simultaneously, where individual packets may exist in different feature spaces. We show how such multi-packet continuous attractor networks could be used to maintain representations of a number of different classes of feature (such as particular line shapes and colour) simultaneously active in the correct location in their respective feature spaces, where such feature spaces might correspond to the egocentric physical space in which an agent is situated.

The above proposal is somewhat analogous to that described by Recce and Harris (1996). These authors developed a model that learned an egocentric map of the spatial features in a robot's environment. During navigation through the environment, the representations of the spatial features were used in conjunction with each other. That is, in a sense, the representations of the different spatial features were co-active in working memory. This provided a robust basis for navigation through the environment. However, the model developed by Recce and Harris (1996) was not a biologically plausible neural network model. In the work presented here, our aim is to develop biologically plausible neural network models that are capable of the simultaneous representation of many spatial features in the environment. The underlying theoretical framework we use to achieve this is a continuous attractor neural network which has been trained to encode multiple charts. Previous investigations with multiple activity packets in a multichart neural network have been described by Samsonovich (1998), who showed that multiple 'discordant' activity packets may co-exist and move independently of one another in such a network. Samsonovich (1998) also reported that the network could support activity packets simultaneously active on different charts. However, the simulation results shown in the paper were restricted to the case of multiple activity packets co-active on the *same* chart.

To elaborate, each neuron in the network might represent a particular class of feature (such as a straight edge, or red colour) in a particular egocentric location in the environment. Thus, each class of feature is represented by a different set of neurons, where each of the neurons responds to the presence of a feature at a particular location. The different sets of neurons that encode the different features may have many cells in common and so significantly overlap with each other, or may not have cells in common in which case each neuron will respond to no more than one feature. For each type of feature, the ordering in the network of the neurons that represent the location of the feature in space is random. Therefore, each separate feature space has a unique ordering of neurons which we refer to as a 'map'. However, all the feature maps are encoded in the same network. The presence of a particular feature at a particular location in the environment is represented by an activity packet centred at the appropriate location in the map which is associated with that particular feature. The network we describe can maintain representations of a particular combination of, e.g. colour features in given relative egocentric spatial positions in the appropriate maps, and simultaneously maintain active another combination of, e.g. shape features in given relative spatial positions. Considered in another way, the network can model several different state spaces, with no topological relation between the positions of the neurons in the network and the location that they encode in each state space. The topology within each state space is specified by the connection strengths between the neurons, with each synapse representing the distance that two neurons are apart from each other in the state space. In the example above, one state space might represent the egocentric location of a straight edge in the physical environment, and another state space might represent the egocentric location of the colour red in the physical environment. In this example, all the state spaces are mapped onto the same physical environment, but this need not be the case.

Moreover, in this paper we show how the absolute spatial locations of the packets can be moved together (or independently) within the separate feature spaces using, for example, an idiothetic (self-motion) signal. Furthermore, the locations of the activity packets in the separate feature maps can be kept in relative alignment during movement of the agent with respect to, for example, an object which consists of a combination of features. We thus note that this architecture has implications for understanding feature binding. Since the network is able to maintain and update the representations of many different (e.g. shape) features simultaneously (which implies binding) using an idiothetic signal, this means that the network is able to maintain a full three-dimensional (3D) representation of the spatial structure of an agent's environment, even as the agent moves within its environment in the absence of visual input.

2. Model 1: network model with low order synapses

In this section we present Model 1, which is a continuous attractor network that is able to stably maintain simultaneously active the representations of multiple features each one of which is in its own location. The model allows the relative spatial location of each feature to be fixed

relative to each other, in which case the agent can be thought of as moving through a fixed environment. The model also allows for the case where each feature can move to different locations independently. What characterises a packet of neuronal activity is a set of active neurons which together represent a feature in a location. A set of simultaneously firing activity packets might be initiated by a set of visual cues in the environment. In this paper we go on to show how these representations may be updated by idiothetic (self-motion, e.g. velocity) signals as the agent moves within its environment in the absence of the visual cues. Model 1 is able to display these properties using relatively low order synapses, which are self-organised through local, biologically plausible learning rules. The architecture of the network described is that shown in Fig. 2. The weights learned in the network are different from those that we have considered previously (Stringer et al., 2002b) in that more than one topological space is trained into the synapses of the network, as will be shown in Figs. 6 and 7.

2.1. The neural representation of the locations of multiple features with respect to a stationary agent

In this section we demonstrate how Model 1 is able to stably maintain the representations of multiple features after the visual input has been removed, with the agent remaining stationary within its environment. The reduced version of Model 1 used to evaluate this contains a network of feature cells, which receive inputs from initiating signals such as visual cues in the environment and are connected by the recurrent synaptic weights w^{RC}. In the light, individual feature cells i are stimulated by visual input I_i^V from particular features μ in the environment, with each feature in a particular position with respect to the agent. Then, when the visual input is removed, the continued firing of the feature cells continues to reflect the position of the features in the environment with respect to the agent. The spaces represented in the attractor are continuous in that a combination of neurons represents a feature, and the combination can move continuously through the space bringing into activity other neurons responding to the same feature but in different locations in the space. The connectivity that provides for this continuity in the spaces is implemented by the synaptic weights of the connections between the neurons in the continuous attractor.

2.1.1. The dynamical equations governing the network of feature cells

The behaviour of the feature cells is governed during testing by the following 'leaky-integrator' dynamical equations. The activation h_i^F of a feature cell i is governed by the equation

$$\tau \frac{dh_i^F(t)}{dt} = -h_i^F(t) + \frac{\phi_0}{C^F} \sum_j (w_{ij}^{RC} - w^{INH}) r_j^F(t) + I_i^V, \quad (1)$$

where we have the following terms. The first term, $-h_i^F(t)$, on the right-hand side of Eq. (1) is a decay term. The second term on the right-hand side of Eq. (1) describes the effects of the recurrent connections in the continuous attractor, where r_j^F is the firing rate of feature cell j, w_{ij}^{RC} is the recurrent excitatory (positive) synaptic weight from feature cell j to cell i, and w^{INH} is a global constant describing the effect of inhibitory interneurons.[1] The third term, I_i^V, on the right-hand side of Eq. (1) represents a visual input to feature cell i. When the agent is in the dark, then the term I_i^V is set to zero. Lastly, τ is the time constant of the system.

The firing rate r_i^F of feature cell i is determined from the activation h_i^F and the sigmoid transfer function

$$r_i^F(t) = \frac{1}{1 + e^{-2\beta(h_i^F(t) - \alpha)}}, \quad (2)$$

where α and β are the sigmoid threshold and slope, respectively.

2.1.2. Self-organisation of the recurrent synaptic connectivity in the continuous attractor network of feature cells

We assume that during the initial learning phase the feature cells respond to visual input from particular features in the environment in particular locations. For each feature μ, there is a subset of feature cells that respond to visual input from that feature. The subset of feature cells that respond (whatever the location) to a feature μ is denoted by Ω^μ. The different subsets Ω^μ may have many cells in common and so significantly overlap with each other, or may not have cells in common in which case any particular feature cell will belong to only one subset Ω^μ. (We note that a feature cell might also be termed a feature–location cell, in that it responds to a feature only when the feature is in a given location.)

After each feature μ has been assigned a subset Ω^μ of feature cells, we evenly distribute the subset Ω^μ of feature cells through the space x^μ. That is, the feature cells in the subset Ω^μ are mapped onto a regular grid of different locations in the space x^μ, where the feature cells are stimulated maximally by visual input from feature μ. However, crucially, in real nervous systems the visual cues for which individual feature cells fire maximally would be determined randomly by processes of lateral inhibition and competition between neurons within the network of feature cells. Thus, for each feature μ the mapping of feature cells through the space x^μ is performed randomly, and so the topological relationships between the feature cells within each space x^μ are unique. The unique set of topological relationships that exist between the subset Ω^μ of feature cells that encode for a feature space x^μ is a map. For each

[1] The scaling factor (ϕ_0/C^F) controls the overall strength of the recurrent inputs to the continuous attractor network of feature cells, where ϕ_0 is a constant and C^F is the number of synaptic connections received by each feature cell from other feature cells.

feature μ there is a unique map, i.e. arrangement of the feature cells in the subset Ω^μ throughout the location space x^μ where the feature cells respond maximally to visual input from that feature.

The recurrent synaptic connection strengths or weights w_{ij}^{RC} from feature cell j to feature cell i in the continuous attractor are set up by associative learning as the agent moves through the space as follows:

$$\delta w_{ij}^{RC} = k^{RC} r_i^F r_j^F, \tag{3}$$

where δw_{ij}^{RC} is the change of synaptic weight and k^{RC} is the learning rate constant.[2] This rule operates by associating together feature cells that tend to be co-active, and this leads to cells which respond to the particular feature μ in nearby locations developing stronger synaptic connections.

In the simulations performed below, the learning phase consists of a number of training epochs, where each training epoch involves the agent rotating with a single feature μ present in the environment during that epoch. During the learning phase, the agent performs one training epoch for each feature μ in turn. Each training epoch with a separate feature μ builds a new map into the recurrent synaptic weights of the continuous attractor network of feature cells, corresponding to the location space x^μ for the particular feature.

2.1.3. Stabilisation of multiple activity packets within the continuous attractor network

The following addition to the model is not a necessary component, but can help to stabilise activity packets. As described by Stringer et al. (2002b), the recurrent synaptic weights within the continuous attractor network will be corrupted by a certain amount of noise from the learning regime. This in turn can lead to drift of an activity packet within the continuous attractor network. Stringer et al. (2002b) proposed that in real nervous systems this problem may be solved by enhancing the firing of neurons that are already firing. This might be implemented through mechanisms for short-term synaptic enhancement (Koch, 1999), or through the effects of voltage dependent ion channels in the brain such as NMDA receptors. In the models presented here we adopt the approach proposed by Stringer et al. (2002b), who simulated these effects by adjusting the sigmoid threshold α_i for each feature cell i as follows. At each timestep $t + \delta t$ in the numerical simulation we set

$$\alpha_i = \begin{cases} \alpha^{HIGH}, & \text{if } r_i^F(t) < \gamma \\ \alpha^{LOW}, & \text{if } r_i^F(t) \geq \gamma \end{cases} \tag{4}$$

where γ is a firing rate threshold. This helps to reinforce the current positions of the activity packets within the continuous attractor network of feature cells. The sigmoid slopes are set to a constant value, β for all cells i. We employ the above form of non-linearity described by Eq. (4) to stabilise each activity packet in the presence of noise from irregular learning, and to reduce the effects of interactions between simultaneously active activity packets.

2.1.4. Simulation results with a stationary agent

In the simulations performed in this paper, we simulate the agent rotating clockwise, and the position of each feature μ with respect to the agent in the egocentric location space x^μ is in the range 0–360°.

The dynamical equations (1) and (2) given above describe the behaviour of the feature cells during testing. However, we assume that when visual cues are available, the visual inputs I_i^V dominate all other excitatory inputs driving the feature cells in Eq. (1). Therefore, in the simulations presented in this paper we employ the following modelling simplification during the initial learning phase. During the learning phase, rather than implementing the dynamical equations (1) and (2), we train the network with a single feature μ at a time, and set the firing rates of the feature cells within the subset Ω^μ according to Gaussian response profiles as follows. During training with a feature μ, each feature cell i in the subset Ω^μ is randomly assigned a unique location $x_i^\mu \in [0, 360]$ in the space x^μ, at which the feature cell is stimulated maximally by the visual input from the feature. Then, during training with each different feature μ, the firing rate r_i^F of each feature cell i in the subset Ω^μ is set according to the following Gaussian response profile

$$r_i^F = e^{-(s_i^F)^2/2(\sigma^F)^2}, \tag{5}$$

where s_i^F is the distance between the current egocentric location of the feature x^μ and the location at which cell i fires maximally x_i^μ, and σ^F is the standard deviation. For each feature cell i in the subset Ω^μ, s_i^F is given by

$$s_i^F = \text{MIN}(|x_i^\mu - x^\mu|, 360 - |x_i^\mu - x^\mu|). \tag{6}$$

Experiment 1: the stable representation of two different features in the environment with a stationary agent. The aim of experiment 1 is to demonstrate how a single continuous attractor network can stably support the representations of two different types of feature after the visual input has been removed, with the agent remaining

[2] The associative Hebb rule (3) used to set up the recurrent weights leads to continual increase in the weights as learning proceeds. To bound the synaptic weights, weight decay can be used in the learning rule (Redish & Touretzky, 1998; Zhang, 1996). The use of a convergent learning rule for the recurrent weights within continuous attractor networks has also been demonstrated by Stringer et al. (2002b), who normalised the synaptic weight vector on each neuron continuously during training. In the current research, we did not examine weight normalisation in more detail, but more simply after training set the lateral inhibition parameter w^{INH} to an appropriate level so that the firing of only a small proportion of the neurons in the network could inhibit all of the other neurons from firing. This leads to small packets of activity being stably maintained by the network.

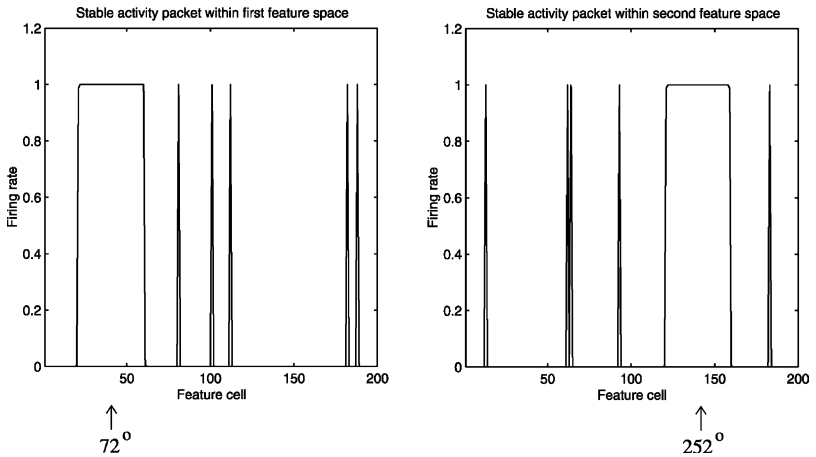

Fig. 1. Experiment 1. Simulation of Model 1 with two activity packets active in two different feature spaces, x^{μ} and x^{ν}, in the same continuous attractor network of feature cells. The figure shows the steady firing rates of the feature cells within the two feature spaces after the external visual input has been removed and the activity packets have been allowed to settle. In this experiment the two feature spaces significantly overlap, i.e. they have feature cells in common. The left plot shows the firing rates of the subset of feature cells Ω^{μ} belonging to the first feature space x^{μ}, and the right plot shows the firing rates of the subset of feature cells Ω^{ν} belonging to the second feature space x^{ν}. In the plot on the left the feature cells have been ordered according to the order they occur in the first feature space, and in the plot on the right the feature cells have been ordered according to the order they occur in the second feature space. In each plot there is a contiguous block of active cells which represents the activity packet within that feature space. In addition, in each plot there is also noise from the activity packet which is active in the other feature space.

stationary within its environment.[3] This is done by performing simulations of Model 1 with two activity packets active in two different feature spaces, x^{μ} and x^{ν}, in the same continuous attractor network of feature cells. The network of feature cells thus represents the presence of two different types of feature, μ and ν, in the environment.

For experiment 1 the network is trained with two features, μ and ν. The continuous attractor network is composed of 1000 feature cells. In the simulations presented here, 200 of these cells are stimulated during the learning phase by visual input from feature μ. This subset of 200 feature cells is denoted Ω^{μ}, and it is this subset of cells that is used to encode the location space for feature μ. Similarly, a further 200 feature cells are stimulated during the learning phase by visual input from feature ν. This subset of 200 feature cells is denoted Ω^{ν}, and it is this subset of cells that encodes the location space for feature ν. For experiment 1 the two subsets, Ω^{μ} and Ω^{ν}, are chosen randomly from the total network of 1000 feature cells, and so the subsets significantly overlap. During the learning phase, the subset Ω^{μ} of feature cells is evenly distributed along the 1D space x^{μ} (and correspondingly for the Ω^{ν} cells in the x^{ν} space). The training is performed separately with 10 revolutions for each of the two spaces.

After the training phase is completed, the agent is simulated (by numerical solution of Eqs. (1) and (2)) for 500 timesteps with visual input available, with the agent remaining stationary, and with features μ and ν present in the environment. There is one occurrence of feature μ at $x^{\mu} = 72°$, and one occurrence of feature ν at $x^{\nu} = 252°$. Next the visual input was removed by setting the I_i^{V} terms in Eq. (1) to zero, and the agent was allowed to remain in the same state for another 500 timesteps. This process leads to a stable packet of activity at $x^{\mu} = 72°$ represented by the feature cells in the subset Ω^{μ} (Fig. 1, left), and a stable packet of activity at $x^{\nu} = 252°$ represented by the feature cells in the subset Ω^{ν} (Fig. 1, right). In the plot on the left the feature cells have been ordered according to the order they occur in the first feature space, and a stable activity packet in this space is demonstrated. In the plot on the right the feature cells have been ordered according to the order they occur in the second feature space, and a stable activity packet in this second space is confirmed. The two activity packets were perfectly stable in their respective spatial feature spaces, with no change even over much longer simulations.

2.2. Updating the neural representations of the locations of the features with idiothetic inputs when the agent moves

In the model described above, we considered only how the continuous attractor network of feature cells might stably maintain the representations of the locations of features as the agent remained stationary. In this section we address the issue of path integration. That is, we show how the representations of the locations of the features within the network might be updated by idiothetic (self-motion) signals as the agent moves within its environment. This is

[3] In experiment 1 we used the following parameter values. The parameter governing the response properties of the feature cells during learning was $\sigma^{\mathrm{F}} = 10°$. A further parameter governing the learning was $k^{\mathrm{RC}} = 0.001$. The parameters governing the leaky-integrator dynamical equations (1) and (2) were $\tau = 1.0$, $\phi_0 = 300\,000$ and $w^{\mathrm{INH}} = 0.0131$. The parameters governing the sigmoid activation function were as follows: $\alpha^{\mathrm{HIGH}} = 0.0$, $\alpha^{\mathrm{LOW}} = -20.0$, $\gamma = 0.5$, and $\beta = 0.1$. Finally, for the numerical simulations of the leaky-integrator dynamical equation (1) we employed a Forward Euler finite difference method with a timestep of 0.2.

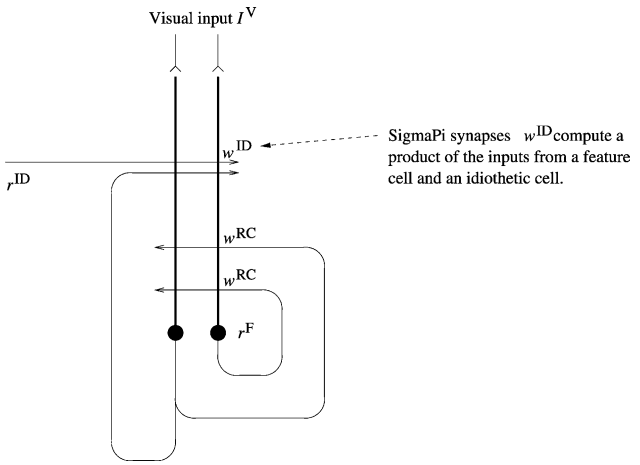

Fig. 2. Network architecture for continuous attractor network Model 1, including idiothetic inputs. The network is composed of two sets of cells: (i) a continuous attractor network of feature cells which encode the position and orientation of the features in the environment with respect to the agent, and (ii) a population of idiothetic cells which fire when the agent moves within the environment. When the agent is in the light, the feature cells are stimulated by visual input I^V. In Model 1 there are two types of modifiable connection: (i) recurrent connections (w^{RC}) within the layer of feature cells, and (ii) idiothetic Sigma–Pi connections (w^{ID}) to feature cells from combinations of idiothetic cells (clockwise rotation cells for the simulations presented here) and feature cells.

an important problem to solve in order to explain how animals can perform path integration in the absence of visual input. The issue also emphasises the continuity of each of the spaces in the continuous attractor, by showing how each packet of activity can be moved continuously.

The full network architecture of Model 1, now including idiothetic inputs, is shown in Fig. 2. The network is composed of two sets of cells: (i) a continuous attractor network of feature cells which encode the position and orientation of the features in the environment with respect to the agent, and (ii) a population of idiothetic cells which fire when the agent moves within the environment. (In the simulations performed below, the idiothetic cells are in fact a population of clockwise rotation cells.) For Model 1, the Sigma–Pi synapses connecting the idiothetic cells to the continuous attractor network use relatively low order combinations of only two pre-synaptic cells. (Background to the proposal we develop here is provided by Stringer et al., 2002b.)

The network of feature cells receives Sigma–Pi connections from combinations of feature cells and idiothetic cells, where the idiothetic cells respond to velocity cues produced during movement of the agent, such as in this paper clockwise rotation. (The velocity cues could represent vestibular and proprioceptive inputs produced by movements, or could reflect motor commands.)

2.2.1. The dynamical equations of Model 1 incorporating idiothetic signals to implement path integration

The behaviour of the feature cells within the continuous attractor network is governed during testing by the following leaky-integrator dynamical equations. The activation h_i^F of a feature cell i is governed by the equation

$$\tau \frac{dh_i^F(t)}{dt} = -h_i^F(t) + \frac{\phi_0}{C^F} \sum_j (w_{ij}^{RC} - w^{INH}) r_j^F(t) + I_i^V + \frac{\phi_1}{C^{F\times ID}} \sum_{j,k} w_{ijk}^{ID} r_j^F r_k^{ID}. \quad (7)$$

The last term on the right-hand side of Eq. (7) represents the input from Sigma–Pi combinations of feature cells and idiothetic cells, where r_k^{ID} is the firing rate of idiothetic cell k, and w_{ijk}^{ID} is the corresponding overall effective connection strength.[4]

Eq. (7) is a general equation describing how the activity within a network of feature cells may be updated using inputs from various kinds of idiothetic cells. In the simulations presented later, the only movement performed by the agent is clockwise rotation, and in principle only a single idiothetic cell is needed in the model to represent this movement (although in the brain such a movement would be represented by a population of cells). However, the general formulation of Eq. (7) can be used to incorporate inputs from various other kinds of idiothetic (self-motion) cells, for example, forward velocity cells. These cells fire as an animal moves forward, with a firing rate that increases monotonically with the forward velocity of the animal. Whole body motion cells have been described in primates (O'Mara, Rolls, Berthoz, & Kesner, 1994). In each case, however, the idiothetic signal must represent a velocity signal (speed and direction of movement) rather than say acceleration.

2.2.2. Self-organisation of synaptic connectivity from the idiothetic cells to the network of feature cells

At the start of the learning phase the synaptic weights w_{ijk}^{ID} may be set to zero. Then the learning phase continues with the agent rotating with the feature cells and idiothetic cells firing according to the response properties described above. The synaptic weights w_{ijk}^{ID} are updated at each timestep according to a trace learning rule

$$\delta w_{ijk}^{ID} = k^{ID} r_i^F \bar{r}_j^F r_k^{ID}, \quad (8)$$

where δw_{ijk}^{ID} is the change of synaptic weight, r_i^F is the instantaneous firing rate of feature cell i, $\bar{r}_j^F$ is the trace value (temporal average) of the firing rate of feature cell j, r_k^{ID} is the firing rate of idiothetic cell k, and k^{ID} is the learning rate. The trace value $\bar{r}^F$ of the firing rate of a feature cell is a form

[4] The scaling factor $\phi_1/(C^{F\times ID})$ controls the overall strength of the idiothetic cell inputs, where ϕ_1 is a constant and $C^{F\times ID}$ is the number of connections received by each feature cell from combinations of feature cells and idiothetic cells. We note that ϕ_1 would need to be set in the brain to have a magnitude which allows the actual head rotation cell firing to move the activity packet at the correct speed, and that this gain control has some similarity to the type of gain control that the cerebellum is believed to implement for the vestibulo-ocular reflex (Rolls & Treves, 1998).

of temporal average of recent cell activity given by

$$\bar{r}^{\mathrm{F}}(t+\delta t) = (1-\eta)r^{\mathrm{F}}(t+\delta t) + \eta \bar{r}^{\mathrm{F}}(t) \quad (9)$$

where η is a parameter set in the interval $[0,1]$ which determines the contribution of the current firing and the previous trace. The trace learning rule (8) involves a product of three firing rate terms on the right-hand side of the equation. The general form of this three-term rule was originally developed by Stringer et al. (2002b) for path integration in a 1D network of head direction cells. However, a simpler form of trace learning rule, involving only two firing rate terms, has been previously used as a biologically plausible learning rule for invariant object recognition (Földiák, 1991; Rolls & Deco, 2002; Wallis & Rolls, 1997).

During a training epoch with a feature μ, the trace learning rule (8) operates as follows. As the agent rotates, learning rule (8) associates an earlier activity pattern within the network of feature cells (representing an earlier location of the feature with respect to the agent), and the co-firing of the idiothetic cells (representing the fact the agent is rotating clockwise), with the current pattern of activity among the feature cells (representing the current location of the feature with respect to the agent). The effect of the trace learning rule (8) for the synaptic weights w_{ijk}^{ID} is to generate a synaptic connectivity such that, during testing without visual input, the co-firing of a feature cell j, and the idiothetic cells, will stimulate feature cell i where feature cell i represents a location that is a small translation in the appropriate direction from the location represented by feature cell j. Thus, the co-firing of a set of feature cells representing a particular feature in a particular location, and the idiothetic cells, will stimulate the firing of further feature cells such that the pattern of activity within the feature cell network that represents that feature evolves continuously to track the true location of the feature in the environment.

Stringer et al. (2002b) showed that a continuous attractor network of the same form as implemented here can perform path integration over a range of velocities, where the speed of movement of the activity packet in the continuous attractor network rises approximately linearly with the firing rate of the idiothetic cells.

2.2.3. Simulation results with a moving agent

In this section we present numerical simulations of Model 1 with a moving agent, in which the locations of the activity packets within the network of feature cells must be updated by idiothetic signals. The simulations are for a case where the idiothetic training signal is the same for the different feature spaces represented in the network. This achieves the result that the different features move together as the agent moves, providing one solution to the binding problem, and indeed showing how the features can remain bound even despite a transform such as spatial translation through the space.

Experiment 2: moving the representation of two identical features at different locations in the environment as an agent moves. It is well known that the representation of two identical objects is a major issue in models of vision (Mozer, 1991). The aim of experiment 2 is to demonstrate how a single continuous attractor network can represent two identical features at different locations in the environment, and update these representations as the agent rotates.[5] This is done by performing simulations of Model 1 with two activity packets active at different locations in the same feature space x^{μ} in the continuous attractor network of feature cells. In this situation the network of feature cells represents the presence of the same feature at different locations in the environment relative to the agent.

For this experiment the network is trained with only a single feature μ. The continuous attractor network is composed of 1000 feature cells. In this experiment a subset of 200 feature cells, denoted Ω^{μ}, is stimulated during training in order to encode the location space for feature μ. For each feature cell i in the subset Ω^{μ} there is a unique location of the feature μ within its space x^{μ} for which the feature cell is stimulated maximally. During the learning phase, the agent rotates clockwise for 10 complete revolutions with visual input available from feature μ present in the environment. The learning phase establishes a set of recurrent synaptic weights between the feature cells in the subset Ω^{μ} that allows these cells to stably support activity packets in the feature space x^{μ} represented by these cells.

After the training phase was completed, the agent was simulated (by numerical solution of Eqs. (2) and (7)) for 500 timesteps with visual input available, with the agent remaining stationary, and with two occurrences of feature μ in the environment. There was one occurrence of feature μ at $x^{\mu} = 72°$, and another occurrence of feature μ at $x^{\mu} = 252°$. While the agent remained in this position, the visual input terms I_i^{V} for each feature cell i in Eq. (7) were set to a Gaussian response profile identical (except for a constant scaling) to that used for the feature cells during the learning phase given by Eq. (5). (When there is more than one feature present in the environment, the term I_i^{V} is set to the maximum input from any one of the features.) The visual input was then removed by setting the I_i^{V} terms in Eq. (7) to zero, and the agent was allowed to remain in the same direction for another 500 timesteps. The activity for the next 200 timesteps is shown at the beginning of Fig. 3, and it is clear that two stable packets of activity were maintained in

[5] In experiment 2 we used the following parameter values. The parameter governing the response properties of the feature cells during learning was $\sigma^{\mathrm{F}} = 10°$. Further parameters governing the learning were $\eta = 0.9$, $k^{\mathrm{RC}} = 0.001$ and $k^{\mathrm{ID}} = 0.001$. The parameters governing the leaky-integrator dynamical equations (2) and (7) were $\tau = 1.0$, $\phi_0 = 300\,000$, $\phi_1 = 70\,000$ and $w^{\mathrm{INH}} = 0.0143$. The parameters governing the sigmoid activation function were as follows: $\alpha^{\mathrm{HIGH}} = 0.0$, $\alpha^{\mathrm{LOW}} = -20.0$, $\gamma = 0.5$, and $\beta = 0.1$. Finally, for the numerical simulations of the leaky-integrator dynamical equation (7) we employed a Forward Euler finite difference method with a timestep of 0.2.

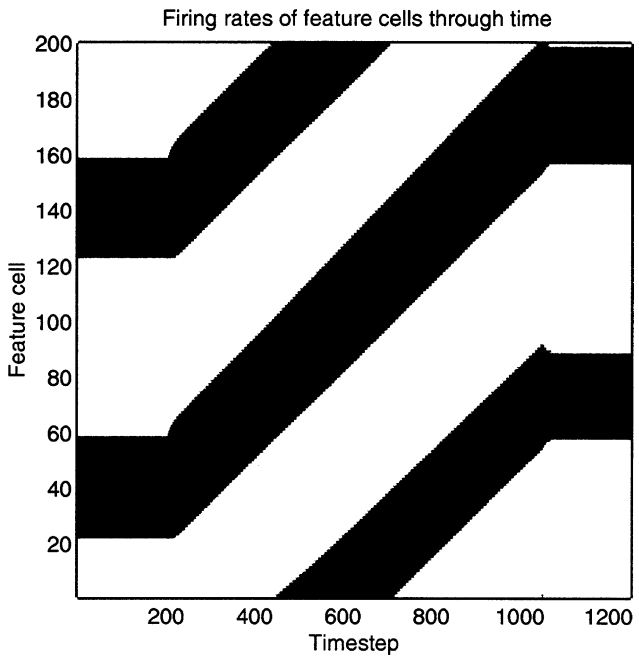

Fig. 3. Experiment 2. Simulations of Model 1 with two activity packets active at different locations in the same feature space x^{μ} in the continuous attractor network of feature cells. The network thus represents the presence of the same feature at different locations in the environment relative to the agent. The figure shows the firing rates (with high rates represented by black) of the feature cells through time, where the feature cells have been ordered according to the order they occur in the feature space x^{μ}. The plot shows the two activity packets moving through the feature space x^{μ}.

this memory condition at the locations (of $x^{\mu} = 72°$ and 252°) where they were started. Next, in the period 201–1050 timesteps in Fig. 3 the agent rotated clockwise (for a little less than one revolution), and the firing of the idiothetic clockwise rotation cells (set to 1) drove the two activity packets through the feature space x^{μ} within the continuous attractor network. From timestep 1051 the agent was again stationary and the two activity packets stopped moving.

From these results we see that the continuous attractor network of feature cells is able to maintain two activity packets active at different locations in the same feature space x^{μ}. Furthermore, as the agent moves, the network representations of the egocentric locations of the features may be updated by idiothetic signals.

However, it may be seen from Fig. 3 that when the two packets begin to move, one activity packet grew a little in size while the other activity packet shrank. In other simulations it was found that during movement one activity packet can die away altogether, leaving only a single activity packet remaining. This effect was only seen during movement, and was due to the global inhibition operating between the two activity packets. Thus the normal situation was that the network remained firing stably in the state into which it was placed by an external cue; but when the idiothetic inputs were driving the system, some of the noise introduced by this was able to alter the packet size.

The shape of the activity packets shown in Fig. 3 are relatively binary, with the neurons either not firing or firing fast. The degree to which the firing rates are binary vs graded is largely determined by the parameter w^{INH} which controls the level of lateral inhibition between the neurons. When the level of lateral inhibition is relatively high, the activity packets assume a somewhat Gaussian shape. However, as the level of lateral inhibition is reduced, the activity packets grow larger and assume a more step-like profile. Furthermore, the non-linearity in the activation function shown in Eq. (4) also tends to make the firing rates of the neurons somewhat binarised. The contributions of both factors have been examined by Stringer et al. (2002b). In the simulations described in this paper a relatively low level of inhibition was used in conjunction with the non-linear activation function, and this combination led to step-like profiles for the activity packets. However, in further simulations we have shown that the network can support multiple activity packets when the firing rates are graded, although keeping the network in a regime where the firing rates are relatively binary does contribute to enabling the network to keep different activity packets equally active. Although the network operates best with a relatively binary firing rate distribution, we note that the network is nevertheless a continuous attractor in that all locations in the state space are equally stable, and the activity packet can be moved continuously throughout the state space.

Experiment 3: updating the representation of two different features in the environment using non-overlapping feature spaces. The aim of experiment 3 is to demonstrate how a single continuous attractor network can represent two different types of feature in the environment, and update these representations as the agent rotates.[6] This is done by performing simulations of Model 1 with two activity packets active in two different feature spaces, x^{μ} and x^{ν}, in the same continuous attractor network of feature cells. The network of feature cells thus represents the presence of two different types of feature, μ and ν, in the environment.

The whole experiment was run similarly to experiment 2, except that the network was trained with two features, with 200 of the cells assigned to the subset Ω^{μ} that represents feature μ, and 200 of the cells assigned to the subset Ω^{ν} that represents feature ν. For experiment 3 the two subsets, Ω^{μ} and Ω^{ν}, did not overlap, that is, the two subsets did not have any cells in common. During the first learning stage the network was trained with feature μ, and then during the second learning stage the network was trained with feature ν.

The results from experiment 3 are shown in Fig. 4. The left plot shows the firing rates of the subset of feature cells Ω^{μ} that encode the location space x^{μ} for feature μ, and the right plot shows the firing rates of the subset of feature cells Ω^{ν} that encode the location space x^{ν} for feature

[6] The model parameters used for experiment 3 were the same as those used for experiment 2, except for $\phi_1 = 200\,000$ and $w^{\mathrm{INH}} = 0.0191$.

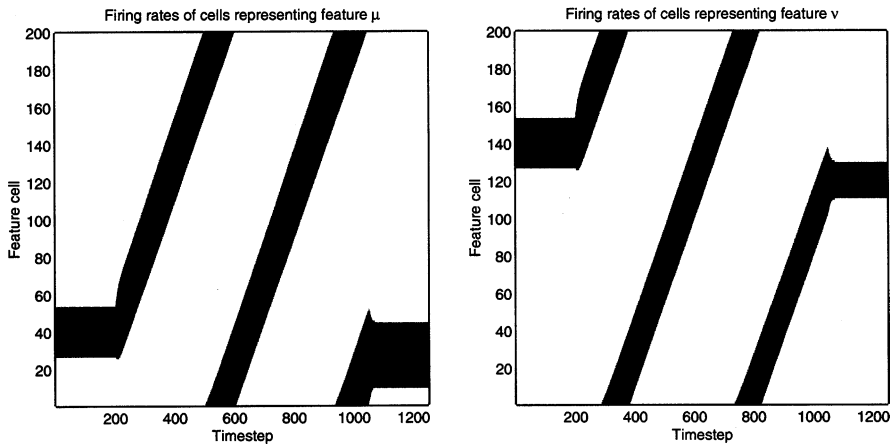

Fig. 4. Experiment 3. Simulation of Model 1 with two activity packets active in two different feature spaces, x^{μ} and x^{ν}, in the same continuous attractor network of feature cells which has global inhibition. The network thus represents the presence of two different types of feature, μ and ν, in the environment. In this experiment the two feature spaces do not have any feature cells in common. The left plot shows the firing rates of the subset of feature cells Ω^{μ} belonging to the first feature space x^{μ}, and the right plot shows the firing rates of the subset of feature cells Ω^{ν} belonging to the second feature space x^{ν}. Furthermore, in the plot on the left the feature cells have been ordered according to the order they occur in the first feature space, and in the plot on the right the feature cells have been ordered according to the order they occur in the second feature space. Thus, the left and right plots show the two activity packets moving within their respective feature spaces.

ν. Furthermore, in the plot on the left the feature cells have been ordered according to the order they occurred in the feature space x^{μ}, and in the plot on the right the feature cells have been ordered according to the order they occurred in the second feature space x^{ν}. Thus, the left and right plots show the two activity packets moving within their respective feature spaces. From timesteps 1 to 200 the agent is stationary and the two activity packets do not move. From timesteps 201 to 1050, the agent rotates clockwise and the idiothetic inputs from the clockwise rotation cells drives the two activity packets through their respective feature spaces within the continuous attractor network. From timestep 1051 the agent is again stationary and the two activity packets stop moving. From these results we see that the continuous attractor network of feature cells is able to maintain activity packets in two different feature spaces, x^{μ} and x^{ν}. Furthermore, as the agent moves, the network representations of the egocentric locations of the features may be updated by idiothetic signals.

Experiment 4: updating the representation of two different features in the environment using overlapping feature spaces. In experiment 4 we demonstrate how a continuous attractor network can represent two different features in the environment using two different overlapping feature spaces, and update these representations as the agent rotates. In this case the continuous attractor network stores the feature spaces of two different features μ and ν, where the subsets of feature cells used to encode the two spaces x^{μ} and x^{ν} have a number of cells in common. This is the most difficult test case, since using overlapping feature spaces leads to significant interference between co-active representations in these different spaces. Experiment 4 was composed of two parts, 4a and 4b. In experiment 4a we used the same size network as was used for experiment 3, whereas for experiment 4b the network was five times larger in order to investigate the effects of increasing the number of neurons.

Experiment 4a was run similarly to experiment 3, with a network of 1000 feature cells, and where each of the subsets Ω^{μ} and Ω^{ν} contained 200 cells.[7] However, for experiment 4a the two subsets Ω^{μ} and Ω^{ν} were chosen randomly from the total network of 1000 feature cells, and so the subsets significantly overlapped. That is, the two subsets had approximately 40 cells in common.

The results from experiment 4a are shown in Fig. 5. The left plot shows the firing rates of the subset of feature cells Ω^{μ} that encode the location space x^{μ} for feature μ, and the right plot shows the firing rates of the subset of feature cells Ω^{ν} that encode the location space x^{ν} for feature ν. From these results we see that the continuous attractor network of feature cells is able to maintain activity packets in two different feature spaces, x^{μ} and x^{ν}. Furthermore, as the agent moves, the network representations of the egocentric locations of the features may be updated by idiothetic signals. However, experiment 4a showed two effects that were not present in experiment 3. Firstly, because the two feature spaces have cells in common, each feature space contains noise from the firing of cells in the activity packet present in the other feature space. This shows as random cell firings in each of the two spaces. Secondly, the activity packets in each of the two feature spaces are both distorted due to the interference between the two spaces. The gross distortion of the two packets was only seen during movement. However, although the two packets were able to influence each other through global inhibition, the distortion of the two activity packets was primarily due

[7] The model parameters used for experiment 4a were the same as those used for experiment 2, except for $\phi_1 = 200\,000$ and $w^{\mathrm{INH}} = 0.0131$.

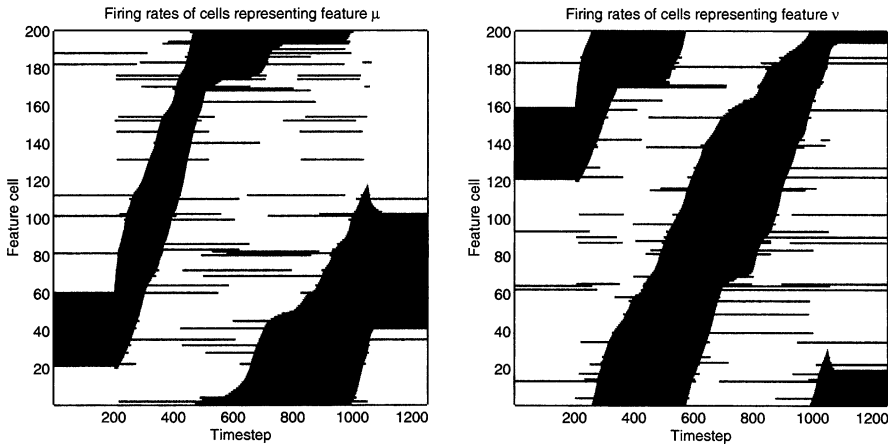

Fig. 5. Experiment 4a. Simulation of Model 1 with two activity packets active in two different feature spaces, x^{μ} and x^{ν}, in the same continuous attractor network of feature cells. Conventions as in Fig. 4. In this experiment the two feature spaces significantly overlap, i.e. they have feature cells in common so that there is some interference between the activity packets. Nevertheless, path integration in each of the spaces is demonstrated.

to excitatory connections that existed between the neurons in the two packets.

Fig. 6 shows the learned recurrent synaptic weights w_{ij}^{RC} between feature cells in experiment 4a. The left plot of Fig. 6 shows the recurrent weights w_{ij}^{RC} between the feature cells in the subset Ω^{μ} which encodes the first feature space x^{μ}. For this plot the 200 feature cells in the subset Ω^{μ} are ordered according to their location in the space x^{μ}. The plot shows the recurrent weights from feature cell 99 to the other feature cells in the subset Ω^{μ}. The graph shows an underlying symmetric weight profile about feature cell 99, which is necessary for the recurrent weights to stably support an activity packet at different locations in the space x^{μ}. However, in this experiment cell 99 was also contained in the subset Ω^{ν} which encoded the second feature space x^{ν}. Thus, there is additional noise in the weight profile due to the synaptic weight updates associated with the second feature space, between feature cell 99 and other feature cells encoding the second feature space x^{ν}. The right plot of Fig. 6 shows the recurrent weights w_{ij}^{RC} between the feature cells in the subset Ω^{ν} which encodes the second feature space x^{ν}. For this plot the 200 feature cells in the subset Ω^{ν} are ordered according to their location in the space x^{ν}. The plot shows the recurrent weights from feature cell 97 to the other feature cells in the subset Ω^{ν}. The right plot for the second feature space shows similar characteristics to the left plot.

Fig. 7 shows the learned idiothetic synaptic weights w_{ijk}^{ID} between the idiothetic (rotation) cells and feature cells in experiment 4a. The left plot of Fig. 7 shows the idiothetic weights w_{ijk}^{ID} between the rotation cell k and the feature cells in the subset Ω^{μ} which encodes the first feature space x^{μ}. For this plot the 200 feature cells in the subset Ω^{μ} are

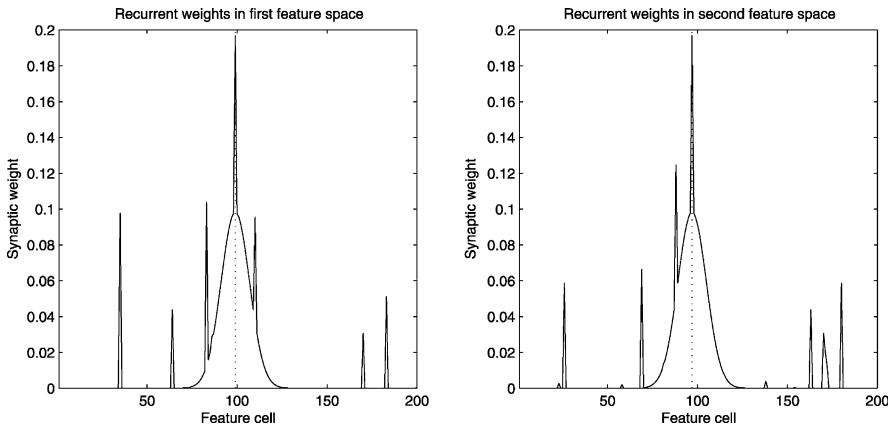

Fig. 6. Learned recurrent synaptic weights between feature cells in experiment 4a. The left plot shows the recurrent weights w_{ij}^{RC} between the feature cells in the subset Ω^{μ} which encodes the first feature space x^{μ}. For this plot the 200 feature cells in the subset Ω^{μ} are ordered according to their location in the space x^{μ}. The plot shows the recurrent weights from feature cell 99 to the other feature cells in the subset Ω^{μ}. The graph shows an underlying symmetric weight profile about feature cell 99, which is necessary for the recurrent weights to stably support an activity packet at different locations in the space x^{μ}. However, in this experiment feature cell 99 was also contained in the subset Ω^{ν} which encoded the second feature space x^{ν}. Thus, there is additional noise in the weight profile due to the synaptic weight updates associated with the second feature space, between feature cell 99 and other feature cells encoding the second feature space x^{ν}. The right plot shows the recurrent weights w_{ij}^{RC} between the feature cells in the subset Ω^{ν} which encodes the second feature space x^{ν}. For this plot the 200 feature cells in the subset Ω^{ν} are ordered according to their location in the space x^{ν}. The plot shows the recurrent weights from feature cell 97 to the other feature cells in the subset Ω^{ν}. The right plot for the second feature space shows similar characteristics to the left plot.

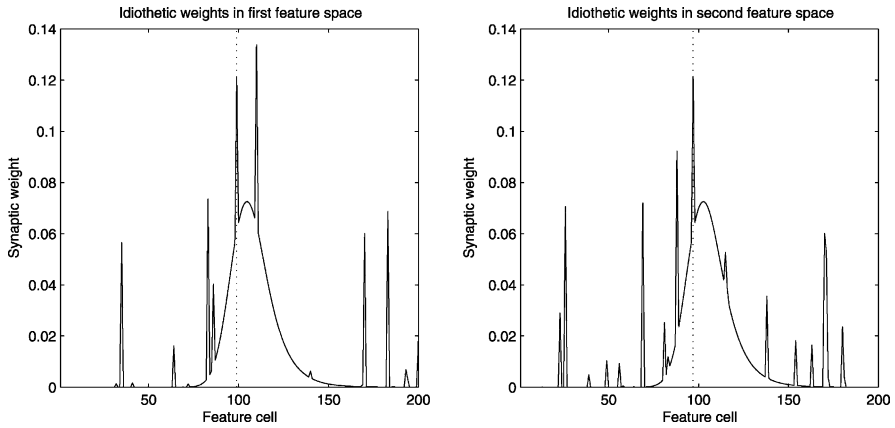

Fig. 7. Learned idiothetic synaptic weights between the idiothetic (rotation) cells and feature cells in experiment 4a. The left plot shows the idiothetic weights w_{ijk}^{ID} between the rotation cell k and the feature cells in the subset Ω^{μ} which encodes the first feature space x^{μ}. For this plot the 200 feature cells in the subset Ω^{μ} are ordered according to their location in the space x^{μ}. The plot shows the idiothetic weights from the rotation cell and feature cell 99 to the other feature cells in the subset Ω^{μ}. The graph shows an underlying asymmetric weight profile about cell 99, which is necessary for the idiothetic weights to shift an activity packet through the space x^{μ}. However, in experiment 4a feature cell 99 was also contained in the subset Ω^{ν} which encoded the second feature space x^{ν}. Thus, there is additional noise in the weight profile due to the synaptic weight updates associated with the second feature space. The right plot shows the idiothetic weights w_{ijk}^{ID} between the rotation cell k and the feature cells in the subset Ω^{ν} which encodes the second feature space x^{ν}. For this plot the 200 feature cells in the subset Ω^{ν} are ordered according to their location in the space x^{ν}. The plot shows the idiothetic weights from the rotation cell and feature cell 97 to the other feature cells in the subset Ω^{ν}. The right plot for the second feature space shows similar characteristics to the left plot.

ordered according to their location in the space x^{μ}. The plot shows the idiothetic weights from the rotation cell and feature cell 99 to the other feature cells in the subset Ω^{μ}. The graph shows an underlying asymmetric weight profile about cell 99, which is necessary for the idiothetic weights to shift an activity packet through the space x^{μ} (Stringer et al., 2002b). From the idiothetic weight profile shown in the left plot of Fig. 7, it can be seen that the co-firing of the rotation cell k and feature cell 99 will lead to stimulation of other feature cells that are a small distance away from feature cell 99 in the space x^{μ}. This will lead to a shift of an activity packet located at feature cell 99 in the appropriate direction in the space x^{μ}. In this way, the asymmetry in the idiothetic weights is able to shift an activity packet through the space x^{μ} when the agent is rotating in the absence of visual input. However, in experiment 4a feature cell 99 was also contained in the subset Ω^{ν} which encoded the second feature space x^{ν}. Thus, there is additional noise in the weight profile due to the synaptic weight updates associated with the second feature space. The right plot of Fig. 7 shows the idiothetic weights w_{ijk}^{ID} between the rotation cell k and the feature cells in the subset Ω^{ν} which encodes the second feature space x^{ν}. For this plot the 200 feature cells in the subset Ω^{ν} are ordered according to their location in the space x^{ν}. The plot shows the idiothetic weights from the rotation cell and feature cell 97 to the other feature cells in the subset Ω^{ν}. The right plot for the second feature space shows similar characteristics to the left plot.

In experiment 4b we investigated how the network performed as the number of neurons in the network increased.[8] This is an important issue given that recurrent networks in the brain, such as the CA3 region of the hippocampus, may contain neurons with many thousands of recurrent connections from other neurons in the same network. Experiment 4b was similar to experiment 4a, except that for experiment 4b the network contained five times as many neurons as in experiment 4a. For experiment 4b the network was composed of 5000 feature cells, with each of the feature spaces represented by 1000 feature cells chosen randomly. It was found that as the number of neurons in the network increased there was less interference between the activity packets, and the movement of the activity packets through their respective spaces was much smoother. This can be seen by comparing the results shown in Fig. 8 for the large network with those shown in Fig. 5 for the smaller network. It can be seen that, in the small network, the size of the activity packets varied continuously through time. In further simulations (not shown) this could lead to the ultimate extinction of one of the packets. However, in the large network, the activity packets were stable. That is, the size of the activity packets remained constant as they moved through their respective feature spaces. This effect of increasing the number of neurons in the network is analysed theoretically in Section 5 and Appendix A. This important result supports the hypothesis that large recurrent networks in the brain are able to maintain multiple activity packets, perhaps representing different features in different locations in the environment.

From experiment 4 we see that one way to reduce interference between activity packets in different spaces is to increase the size of the network. In Section 3, we describe another way of reducing the interference between simultaneously active packets in different feature spaces, using higher order synapses.

[8] The model parameters used for experiment 4b were the same as those used for experiment 4a, except for $w^{\mathrm{INH}} = 0.0146$.

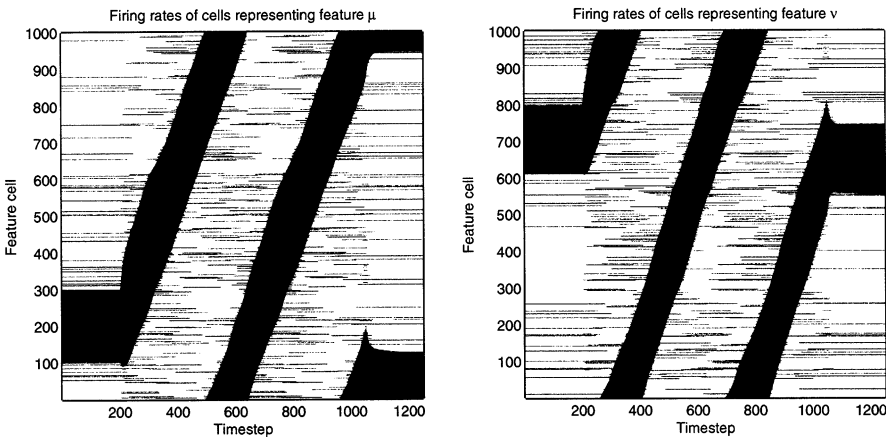

Fig. 8. Experiment 4b. Simulation of Model 1 with two activity packets active in two different feature spaces, x^{μ} and x^{ν}. Experiment 4b was similar to experiment 4a, except that for experiment 4b the network contained five times as many neurons as in experiment 4a. For experiment 4b the network was composed of 5000 feature cells, with each of the feature spaces represented by 1000 feature cells chosen randomly. As the number of neurons in the network increases the movement of the activity packets through their respective spaces is much smoother, which can be seen by comparing the results shown here with those shown in Fig. 5 for the smaller network.

3. Model 2: Network model with higher order synapses

In Model 2 the recurrent connections within the continuous attractor network of feature cells employ Sigma–Pi synapses to compute a weighted sum of the products of inputs from other neurons in the continuous attractor network. In addition, in Model 2 the Sigma–Pi synapses connecting the idiothetic cells to the continuous attractor network use even higher order combinations of pre-synaptic cells.

The general network architecture of Model 2 is shown in Fig. 9. The network architecture of Model 2 is similar to Model 1, being composed of a continuous attractor network of feature cells, and a population of idiothetic cells. However,

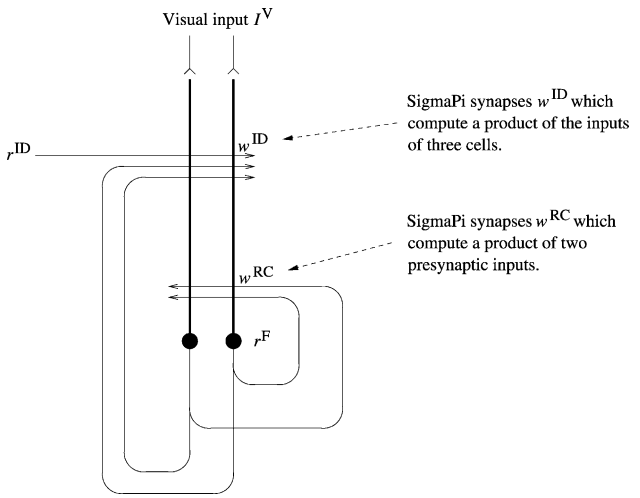

Fig. 9. General network architecture for continuous attractor network Model 2. The network architecture of Model 2 is similar to Model 1, being composed of a continuous attractor network of feature cells, and a population of idiothetic cells. However, Model 2 uses Sigma–Pi recurrent synaptic connections w^{RC} within the continuous attractor network, and higher order Sigma–Pi idiothetic synaptic connections w^{ID} to feature cells from combinations of idiothetic cells and feature cells.

Model 2 combines two presynaptic inputs from other cells in the attractor into a single synapse w^{RC}; and for the idiothetic update synapses combines two presynaptic inputs from other cells in the continuous attractor with an idiothetic input in synapses w^{ID}. The synaptic connections within Model 2 are self-organised during an initial learning phase in a similar manner to that described above for Model 1.

3.1. The dynamical equations of Model 2

The behaviour of the feature cells within the continuous attractor network is governed during testing by the following leaky-integrator dynamical equations. Model 2 is introduced with synapses that are only a single order greater than the synapses used in Model 1, but in principle the order of the synapses can be increased. In Model 2 the activation h_i^{F} of a feature cell i is governed by the equation

$$\tau\frac{\mathrm{d}h_i^{F}(t)}{\mathrm{d}t} = -h_i^{F}(t) + I_i^{V} + \frac{\phi_0}{C^{F}}\sum_{j,m} w_{ijm}^{RC}(r_j^{F}(t)r_m^{F}(t)) - \frac{\phi_0}{C^{F}}\sum_{j} w^{INH} r_j^{F}(t) + \frac{\phi_1}{C^{F\times ID}}\sum_{j,m,k} w_{ijmk}^{ID}(r_j^{F} r_m^{F} r_k^{ID}). \quad (10)$$

The effect of the higher order synapses between the neurons in the continuous attractor is to make the recurrent synapses more selective than in Model 1. That is, the synapse w_{ijm}^{RC} will only be able to stimulate feature cell i when both of the feature cells j,m are co-active. Each idiothetic connection also involves a high order Sigma–Pi combination of two pre-synaptic continuous attractor cells and one idiothetic input cell. The effect of this is to make the idiothetic synapses more selective than in Model 1. That is, the synapse w_{ijmk}^{ID} will only be able to stimulate feature cell i when both of the feature cells j,m and the idiothetic cell k are co-active. The firing rate r_i^{F} of

feature cell i is determined from the activation h_i^{F} and the sigmoid function (2).

The recurrent synaptic weights within the continuous attractor network of feature cells are self-organised during an initial learning phase in a similar manner to that described above for Model 1. For Model 2 the recurrent weights w_{ijm}^{RC}, from feature cells j, m to feature cell i may be updated according to the associative rule

$$\delta w_{ijm}^{\mathrm{RC}} = k^{\mathrm{RC}} r_i^{\mathrm{F}} r_j^{\mathrm{F}} r_m^{\mathrm{F}} \tag{11}$$

where $\delta w_{ijm}^{\mathrm{RC}}$ is the change of synaptic weight and k^{RC} is the learning rate constant. This rule operates by associating the co-firing of feature cells j and m with the firing of feature cell i. This learning rule allows the recurrent synapses to operate highly selectively in that, after training, the synapse w_{ijm}^{RC} will only be able to stimulate feature cell i when both of the feature cells j, m are co-active.

The synaptic connections to the continuous attractor network of feature cells from the Sigma–Pi combinations of idiothetic (or motor) cells and feature cells are self-organised during an initial learning phase in a similar manner to that described above for Model 1. However, for Model 2 the idiothetic weights w_{ijmk}^{ID} may be updated according to the associative rule

$$\delta w_{ijmk}^{\mathrm{ID}} = k^{\mathrm{ID}} r_i^{\mathrm{F}} \bar{r}_j^{\mathrm{F}} \bar{r}_m^{\mathrm{F}} r_k^{\mathrm{ID}} \tag{12}$$

where $\delta w_{ijmk}^{\mathrm{ID}}$ is the change of synaptic weight, r_i^{F} is the instantaneous firing rate of feature cell i, $\bar{r}_j^{\mathrm{F}}$ is the trace value (temporal average) of the firing rate of feature cell j, etc., r_k^{ID} is the firing rate of idiothetic cell k, and k^{ID} is the learning rate. The trace value $\bar{r}^{\mathrm{F}}$ of the firing rate of a feature cell is given by Eq. (9). During a training epoch with a feature μ, the trace learning rule (12) operates to associate the co-firing of feature cells j, m and idiothetic cell k, with the firing of feature cell i. Thus, learning rule (12) operates somewhat similar to learning rule (8) for Model 1 in that, as the agent rotates, learning rule (12) associates an earlier activity pattern within the network of feature cells (representing an earlier location of the feature with respect to the agent), and the co-firing of the idiothetic cells (representing the fact the agent is rotating clockwise), with the current pattern of activity among the feature cells (representing the current location of the feature with respect to the agent). However, learning rule (12) allows the idiothetic synapses to operate highly selectively in that after training, the synapse w_{ijmk}^{ID} will only be able to stimulate feature cell i when both of the feature cells j, m and idiothetic cell k are co-active.

3.2. Simulation results with Model 2

Experiment 5: representing overlapping feature spaces with higher order synapses. The aim of experiment 5 is to demonstrate that the higher order synapses implemented in Model 2 are able to reduce the interference between activity packets which are simultaneously active in different spaces.[9] Experiment 5 is run similarly to experiment 4. That is, experiment 5 involves the simulation of Model 2 with two activity packets active in two different feature spaces, x^{μ} and x^{ν}, in the same continuous attractor network of feature cells. In this experiment the two feature spaces have the same degree of overlap as was the case in experiment 4. For experiment 5, due to the increased computational cost of the higher order synapses of Model 2, the network was simulated with only 360 feature cells. Each of the two feature spaces, x^{μ} and x^{ν}, was represented by a separate subset of 200 feature cells, where two subsets were chosen such that the two feature spaces had 40 feature cells in common. This overlap between the two feature spaces was the expected size of the overlap in experiment 4, where there were a total of 1000 feature cells, and each of the two feature spaces recruited a random set of 200 cells from this total.

The results of experiment 5 are presented in Fig. 10, and these results can be compared to those shown in Fig. 5 for Model 1. The left plot shows the firing rates of the subset of feature cells Ω^{μ} belonging to the first feature space x^{μ}, and the right plot shows the firing rates of the subset of feature cells Ω^{ν} belonging to the second feature space x^{ν}. It can be seen that with the higher order synapses used by Model 2, the activity packets in the two separate feature spaces are far less deformed. In particular, over the course of the simulation, the activity packets maintain their original sizes. This is in contrast to experiment 4, where one packet became larger while the other packet became smaller. Hence, with the higher order synapses of Model 2, there is much less interference between the representations in the two separate feature spaces.

4. How the representations of multiple features within a continuous attractor network may be decoded by subsequent, e.g. motor systems

In this section we consider how subsequent, for example motor, systems in the brain are able to respond to the representations of multiple features supported by a continuous attractor network of feature cells. The execution of motor sequences by the motor system may depend on exactly which features are present in the environment, and where the features are located with respect to the agent. However, for both models 1 and 2 presented in this paper, if multiple activity packets are active within the continuous attractor network of feature cells, then the representation of each feature will be masked by the 'noise' from the other active representations of other features present in the environment, as shown in Fig. 5. In this situation, how can subsequent motor systems detect the representations of individual features? What we propose in this section is that

[9] The model parameters used for experiment 5 were the same as those used for experiment 2, except for $\phi_0 = 108\,000\,000$, $\phi_1 = 43\,200\,000$ and $w^{\mathrm{INH}} = 0.129$.

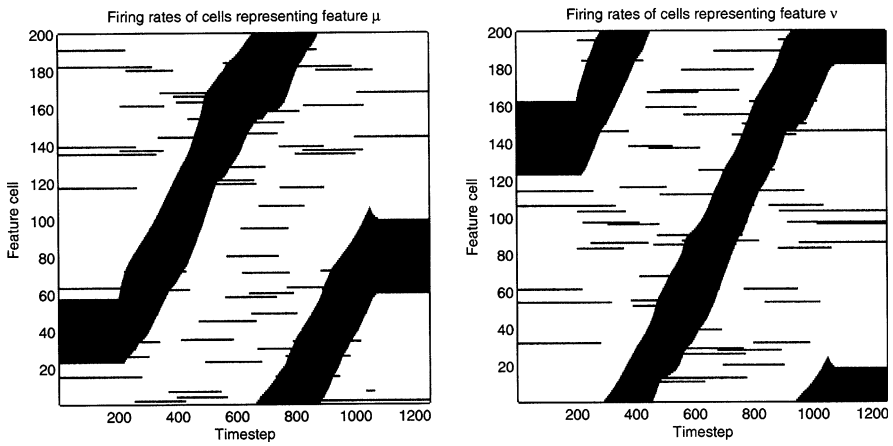

Fig. 10. Experiment 5. Simulation of Model 2 with two activity packets active in two different feature spaces, x^{μ} and x^{ν}, in the same continuous attractor network of feature cells. In this experiment the two feature spaces significantly overlap, and have the same degree of overlap as in experiment 4. This experiment is similar to experiment 4, except here we implement Model 2 with higher order synapses instead of Model 1. The results presented in this figure should be compared to those shown in Fig. 5 for Model 1. It can be seen that with the higher order synapses used by Model 2, there is much less interference between the representations in the two separate feature spaces.

a pattern associator would be able to decode the representations in the continuous attractor network and would have the benefit of reducing noise in the representation. (The operation and properties of pattern association networks are reviewed by Hertz, Krogh, & Palmer, 1991; Rolls & Treves, 1998; Rolls & Deco, 2002.)

The way in which the decoding could work is shown in Fig. 11, which shows the network architecture for Model 1 augmented with a pattern associator in which the neuronal firing could represent motor commands. During an initial motor training phase in the light, the feature cells in the continuous attractor are stimulated by visual input I^{V}, the motor cells are driven by a training signal t, and the synapses w^{M} are modified by associative learning. Then, after the motor training is completed, the connections w^{M} are able to drive the motor cells to perform the appropriate motor actions. We have described elsewhere a network which enables motor cells to be selected correctly by movement selector cells (Stringer et al., 2003b), and this could be combined with the architecture shown in Fig. 11 to allow the motor cells activated to depend on both the feature representation in the continuous attractor and on the desired movement. During the learning, the synaptic weights w_{ij}^{M} from feature cells j to motor cells i are updated at each timestep according to

$$\delta w_{ij}^{\mathrm{M}} = k^{\mathrm{M}} r_i^{\mathrm{M}} r_j^{\mathrm{F}}. \tag{13}$$

4.1. Simulation results with a network of motor cells

The motor activity of the agent was characterised by an idealised motor space y. We defined the motor space y of the agent as a toroidal 1D space from $y = 0$ to 360. This allowed a simple correspondence between the motor space y of the agent and the feature spaces. Next, we assumed that each motor cell fired maximally for a particular location in the motor space y, and that the motor cells are distributed evenly throughout the motor space y.

Experiment 6: how motor cells respond to individual representations within the network of feature cells. In experiment 6 we demonstrate that the motor network is able to respond appropriately to the representation of a particular feature μ in the continuous attractor network of feature cells, even when the representation of feature μ is somewhat masked by the presence of noise due to the representation of another feature ν in the same continuous attractor network.[10] Experiment 6 was similar to experiment 4, except here we augment the Model 1 network architecture to include a network of motor cells, as described above.

For experiment 6, the continuous attractor network of feature cells is trained with two features μ and ν in an identical manner to that described above for experiment 4. The motor network was trained as follows. The motor network contains 200 motor cells. During the first stage of learning, while the continuous attractor network of feature cells was being trained with feature μ, the network of motor neurons was trained to perform a particular motor sequence. The learned motor sequence was simply $y = x^{\mu}$. That is, the motor neurons learned to fire so that the activity packet within the motor network mirrors the location of the activity packet in the feature space x^{μ}. While the agent ran through the motor sequence associated with feature μ during the first stage of the training phase, the synaptic weights w_{ij}^{M} are updated according to Eq. (13). During the second stage of training, in which the feature cells were trained with feature ν, the motor cells did not fire, and so the synaptic weights w_{ij}^{M} do not change. Then, after training, the goal was for the network to repeat the motor sequence with $y = x^{\mu}$, even when there is a representation of the second feature ν also

[10] The model parameters used for experiment 6 were the same as those used for experiment 2, except for $\phi_1 = 200\,000$ and $w^{\mathrm{INH}} = 0.0131$.

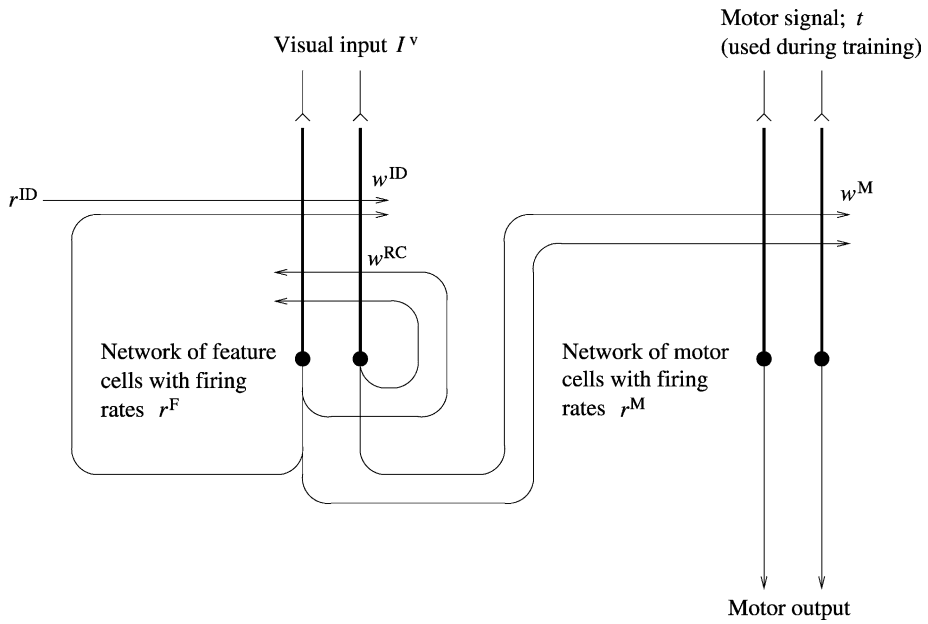

Fig. 11. Network architecture for Model 1 augmented with an additional decoder network of e.g. motor cells. The network is composed of three sets of cells: (i) a continuous attractor network of feature cells, (ii) a population of idiothetic cells which fire when the agent moves within the environment, and (iii) a population of motor cells which represent the motor activity of the agent. During the initial motor training phase in the light, the feature cells are stimulated by visual input I^V, and the motor cells are driven by a training signal t. There are three types of modifiable synaptic connection: (i) Recurrent connections (w^{RC}) within the layer of feature cells, (ii) Idiothetic Sigma–Pi connections (w^{ID}) to feature cells from combinations of idiothetic cells (clockwise rotation cells for the simulations presented here) and feature cells, and (iii) Associative connections (w^M) to the motor cells from feature cells.

present in the network of feature cells. In this case, the network of motor cells must be able to ignore the irrelevant representation in the second feature space x^ν.

Results from experiment 6 are shown in Figs. 5 and 12, which show the firing rates of the feature cells and motor cells through time. In Fig. 12 the motor cells have been ordered according to the order they occur in the motor space y. There is a single activity packet in the motor network which tracks the location of the activity packet in the feature space x^μ represented by the continuous attractor network of feature cells. This may be seen by comparing the results in Fig. 12 with those shown in the left plot of Fig. 5. However, the pattern of activity in the motor network does not contain the noise that is present in the feature space x^μ due to the representation of feature ν. Thus, the motor network is able to filter out the noise due to the representations of irrelevant features. This means that the motor network performs the motor sequence correctly in that, at each stage of the sequence, the correct motor neurons fire, but with no firing of the other neurons in the motor network.

Experiment 6 demonstrates that the motor system is able to detect and respond appropriately to the representation of a particular feature μ in the continuous attractor network of feature cells, even when the representation of feature μ is somewhat masked by the presence of noise due to the representations of other features in the same continuous attractor network. This is because, if the firing threshold is set to a relatively high value for the motor cells, the motor cells will ignore random 'salt and pepper' noise (due to activity packets in overlapping feature spaces) in the feature

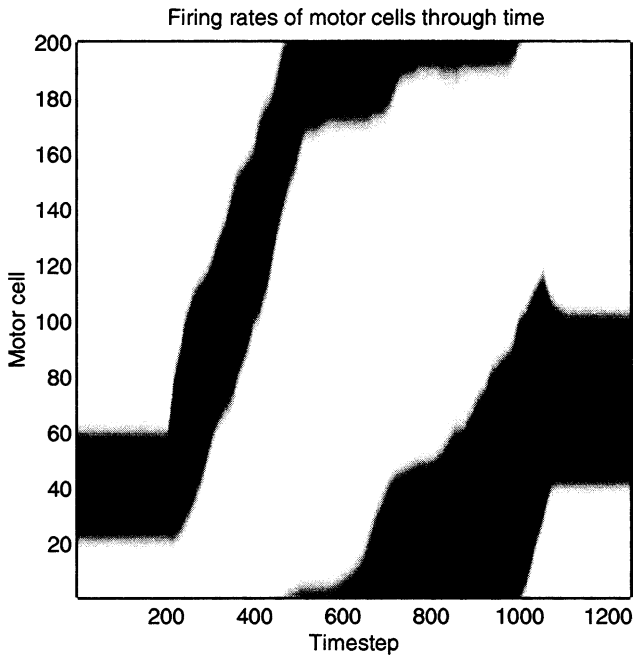

Fig. 12. Experiment 6. Simulation of the network architecture of Model 1 augmented by the addition of a network of motor cells. The simulation of the continuous attractor network of feature cells is performed in an identical manner to experiment 4, with two activity packets active in two different overlapping feature spaces, x^μ and x^ν. This figure shows the firing rates of the linked network of motor cells through time, where the motor cells have been ordered according to the order they occur in the motor space y. There is a single activity packet in the motor network which tracks the location of the activity packet in the feature space x^μ represented by the continuous attractor network of feature cells. However, the pattern of activity in the motor network does not contain the noise that is present in the feature space x^μ due to the representation of feature ν.

space x^{μ} they have learned to respond to, and will only respond to the presence of a genuine contiguous activity packet in the space x^{μ}.

Finally, in further simulations (not shown here) it was found that the performance of the motor network could be improved further by implementing even higher order synapses w^{M}, where the pre-synaptic input terms involved a product consisting of the firing rates of two feature cells.

5. Theoretical analysis of interaction between activity packets in different feature spaces

In Section 1 we reviewed the theoretical results of Amari (1977), which describe how multiple activity packets interact in a single feature space encoded in a continuous attractor network. As the numerical results in this paper have shown that multiple activity packets can co-exist in different feature spaces in the same network, we have developed a mathematical analysis of this scenario. The analysis examines how activity packets in different feature spaces, encoded within the same continuous attractor network, affect each other. The mathematical details of the analysis are provided in Appendix A, and the main points are summarised here.

The analysis provides an explanation for the numerical findings of experiments 4a and 4b, where it was found that as the number of neurons in the network increased, the activity packets moved through the two feature spaces more stably. That is, the shape and size of the activity packets remained constant through time. Both the numerical and the analytic results support the proposal that recurrent networks in the brain, in which there are tens of thousands of connections to each neuron (Rolls & Treves, 1998), are able to represent and track multiple spatial features in the manner described in this paper.

The analysis begins (Appendix A.1) by examining the interaction between activity packets in the discrete model described in Section 2, in which the feature spaces are represented discretely by a large number of neurons. A core assumption of this analysis is that the different feature spaces are encoded by different random orderings of the feature cells. (Evidence for this in a brain area such as the CA3 region of the rat hippocampus is provided in Section 6.) Given these random orderings of the neurons in the different feature spaces, an activity packet in one space will appear as noise in another space. The random ordering requires the analysis to be performed in a discrete system (Zhang, 1996). The analysis (Appendix A.1) shows that, as the number of feature cells N^{F} in the network increases, the input to an activity packet in a first feature space x^{μ} from the activity packet in a second feature space tends towards a continuous function $s(x,t)$. The function $s(x,t)$ takes the form of a fixed wave profile which follows the activity packet profile in the first feature space x^{μ}. We show in Appendix A.2 that this form of interaction between the packets in the different feature spaces is required for the packets to not destabilise each other.

In Appendix A.2 we use a continuous neural field model in which there is a single activation function $h(x,t)$ which is a continuous function of space x and time t. This framework permits us to apply the analytical methods of Amari (1977). An equation is derived which gives the speed of any location of the activity profile (or packet) $h(x,t)$ in the space x^{μ}. It is shown that, for the system to settle into a fixed activity wave (i.e. where the shape and size of the activity packet is constant) moving through the space x^{μ}, the input function $s(x,t)$ (from an activity packet in a second space) must take the form of a fixed wave profile which follows the activity profile $h(x,t)$ in the space x^{μ}. This form of interaction between the packets, which is required for the packets not to destabilise each other, is shown to be true for large systems in Appendix A.1. Any random perturbations from such an ideal input wave $s(x,t)$ from the second packet (such as might occur in small networks) that occur during the time evolution of the system will result in different speeds for different parts of the activity profile in the space x^{μ}, and hence lead to deformation of the shape of the packet in this space. This analysis thus provides theoretical insight into the numerical results of experiments 4a and 4b. That is, as the size of the network increases, there is less time variation in the shape and size of the activity packets in the different spaces produced by the interactions between the packets in the different feature spaces in the same network.

6. Discussion

In this paper we have explored how continuous attractor neural networks can be used to support multiple packets of neuronal activity, and how these separate representations may be simultaneously updated by external inputs to the continuous attractor network, where such external inputs might represent, for example, idiothetic inputs or motor efference copy. To achieve this, the continuous attractor networks presented here make use of (i) a recurrent synaptic connectivity that encodes each activity packet on a separate map, and (ii) higher order Sigma–Pi synapses to enable the idiothetic input to move the activity packets. The networks also benefit from a non-linear neuronal activation function that could be implemented by NMDA receptors. An important property of the models presented in this paper is that they are self-organising with biologically plausible learning rules. However, the framework developed here is quite flexible. For example, although in the simulations presented above the features moved in tandem, it is possible for the external signals to learn to drive some but not other activity packets, and to drive the activity packets at differing speeds from one another. Hence, we propose that such multi-packet continuous attractor networks may play an important role in various aspects of brain function.

The simulations described in this paper showed that when more than one activity packet is active in the network, the activity packets may under some conditions be stable. One of the conditions is a bounded non-linear transfer function is used. Considering stationary activity packets, two packets of activity in a single feature space remain separate and stable if the activity packets are far enough apart, as shown by Amari (1977) and summarised in Section 1. This is true even when the activity packets are of unequal size. Further, as demonstrated in this paper, two stationary activity packets can remain stable even if they share active neurons, but the activity packets are in different feature spaces. Next we consider the situation when the agent is moving and path integration is being performed with moving activity packets. In the case of two activity packets moving in one feature space, the activity packets may interfere with each other, with one activity packet growing while the other packet shrinks. These effects were observed in experiment 2, the results of which are shown in Fig. 3. When the activity packets are moving in different overlapping feature spaces, then the packets may interfere more severely. This effect was observed in experiment 4a, the results of which are shown in Fig. 5. As the overlap between two feature spaces increases, the activity packets in the two spaces interfere with each other more and more. This can be seen by comparing Figs. 4 and 5, which show results with zero overlap and an overlap of approximately 40 cells between the feature spaces, respectively. The nature of this interaction was analysed in Section 5 and Appendix A, where it was shown that the interference between co-active packets in different feature spaces reduces as the size of the network increases. This analysis was supported by the results of experiment 4b, in which increasing the size of the network reduced the interference between the activity packets in the two different feature spaces, as shown in Fig. 8. Thus we have shown in this paper that two activity packets in different feature spaces can both be moved successfully by path integration, and result in persistent separate non-moving activity packets when the idiothetic movement-inducing signal is removed. In simulations of continuous attractor networks with the sigmoidal transfer function (2), the level of activity in the network due to both the size and number of activity packets could be controlled by the level of lateral inhibition between the neurons w^{INH}.

The importance of a bounded non-linear transfer function for the findings just summarised is supported by the following results not described elsewhere in this paper. In attractor networks governed by Eq. (1), but trained with a small set of random binarised activity patterns, it was found that with the sigmoidal transfer function (2), which remains bounded as $h \rightarrow \infty$, stable multiple activity patterns were supported. However, when the sigmoid function was replaced with a threshold linear function, the network was unable to support multiple activity patterns. With the threshold linear transfer function, the firing rates of the neurons in a single pattern always grew large enough to suppress the other neurons in the network. Hence, only a single pattern could be supported by the network. However, this limitation of the threshold linear transfer function could be remedied by introducing an upper bound on the neuronal firing rates. When the firing rates were clipped to a maximum of 1, the network was once more able to support multiple patterns. Thus, these simulations suggest that in order for the network to support multiple patterns, the form of the transfer function must ensure that the neuronal firing rates are bounded as the activation h_i increases (Treves, 1991). Providing the transfer function was bounded, the number of activity patterns stably supported by the network could be controlled by the level of lateral inhibition between the neurons w^{INH}.

The concept of storing multiple maps in a single continuous attractor network has been investigated previously by Battaglia and Treves (1998) and Samsonovich and McNaughton (1997), who used the term 'chart'. (See also Redish (1999) for a detailed discussion of the multiple chart hypothesis and relevant experimental data.) The concept of a chart comes from neurophysiological studies of place cells in rats, which respond when the rat is in a particular place. When moved to a different environment, the relative spatial locations in which neurons fire appear to be different, and hence the concept arises that these hippocampal place cells code for different charts, where each chart applies in a different spatial environment (Wilson & McNaughton, 1993). However, within any one environment or chart, there would be only one activity packet representing the current location of the animal. (Only one activity packet is required in these models (Battaglia & Treves, 1998; Samsonovich & McNaughton, 1997) because the rat cannot be in two places at once.) In contrast, in the continuous attractor model described in this paper, the concept is that multiple maps can have simultaneously active packets of activity. Indeed, we adopt a more abstract viewpoint, in which different maps may be thought of as distinct spaces in which a representation specific to that space may move continuously. Thus, in the models presented in this paper, rather than using different charts to represent different environments, we use each map to represent the space through which the representation of an individual feature may move continuously to represent the changing position of the feature with respect to the agent. The system we describe can maintain several packets of activity simultaneously because each activity packet receives support from the other neurons in the same space in the continuous attractor. As shown in the simulations, individual neurons can be in more than one of the spaces.

A key question is how many feature spaces may be learned by a continuous attractor network before the network reaches its loading capacity. This question has been investigated in detail by Battaglia and Treves (1998), and summarised by Rolls, Stringer, and Trappenberg (2002b) and Tsodyks (1999). The maximum number of

feature spaces (charts) that may be stored is

$$N_{\text{feature spaces}} \sim -C/\log(a_{\text{m}}), \quad (14)$$

where C is the average number of recurrent synaptic contacts per cell, and a_{m} is the map sparseness which is related to the size of a typical activity packet relative to the size of the entire feature space.

The system we describe enables the representations in the different maps to be moved together or separately by the same or different idiothetic inputs. This provides one solution to the binding problem, in that features in different classes (and hence maps) can be moved together by, for example, a single idiothetic input.

In this paper the theory of multi-packet continuous attractor networks was developed in the context of how a brain might represent the 3D structure of an animal's environment. This involves the representation of many independent features and their individual spatial relationships to the agent. After the agent has learned an alphabet of features through early visual experience, when the agent is exposed to a single snapshot view of a new environment (not encountered during training), a full 3D representation of the new environment is initiated in the network of feature cells. Then, when the visual input is removed, the representation can be maintained and updated as the agent moves through its environment in the absence of visual input. Hence, the models presented in this paper can perform path integration using only a single view of a new environment. Another application of this class of model is to the situation when more than one spatial location must be simultaneously and independently represented, as occurs for example when one moving object may collide with or miss another moving object. To the extent that this can be represented in allocentric space, this is likely to involve the hippocampus (or structures that receive from it), and the hippocampus does not use topological mapping in the brain (O'Keefe & Conway, 1978), so that the individual representations of allocentric space would overlap.

The network described here is able to learn how to move the activity packets in their egocentric feature spaces given any kind of vestibular velocity signal (e.g. clockwise or anti-clockwise rotation, forward velocity, etc. or perhaps some form of motor efference copy). It should not matter, for example, that rotations in 3D space do not commute, unlike rotations in 1D space. To see this, consider the following situation. If a clockwise rotation of the agent results in the relative change in position of a feature μ from egocentric location A to location B1, and a further upward turn of the agent results in the relative movement of the feature μ from location B1 to location C1, then these transitions are what the network would learn. Furthermore, if an upward turn of the agent results in the relative change in position of feature μ from egocentric location A to location B2, and a further clockwise rotation of the agent results in the relative movement of the feature μ from location B2 to location C2, then these transitions would also be learned by the network. The network would be capable of learning both transition sequences, and replaying either sequence using only the relevant vestibular signals.

A key problem with current models of hippocampal place cells (Redish, 1999) is the inability of the representations supported by these neural networks to provide a basis for planning novel routes through complex environments full of obstacles. Current models of place cells assume a single activity packet in a 2D layer of place cells, where the cells are simply mapped onto the floor of the containment area. However, such a representation merely locates the agent in a 2D space, and cannot provide information about the full 3D structure of the surrounding environment, which would be necessary for planning a novel route along paths and past obstacles, etc. However, in the models developed in this paper, we address how the full 3D structure of the surrounding environment might be represented in a continuous attractor network, and how this representation may be updated through idiothetic signals or motor efference copy. Only such a representation of the full 3D structure of the agent's environment will provide a robust basis for planning novel routes in complex, cluttered environments.

The concept introduced in this paper may be relevant to understanding the visuo-spatial scratchpad thought to implement a representation of spatial positions of several objects in a scene (Rolls & Arbib, 2003). Consider the output of the inferior temporal cortex (IT), which provides a distributed representation of an object close to the fovea under natural viewing conditions (Rolls et al., 2002a). The representation of this object in different positions in egocentric space would be learned by a continuous attractor network of the type described in this paper by combining this output of IT with a signal (present in the parietal cortex) about the position of the eyes (and head, etc). For each egocentric position, the network would have an arbitrary set of neurons active that would represent the object in that position. As the agent moved, the relation between the idiothetic self-motion signals and the object input would be learned as described here. Each object would be trained in this way, with a separate 'object' space or chart for each object. After training, eye movements round the scene would establish the relative positions of objects in the scene, and after this, any idiothetic self-motion would update the positions of all the objects in egocentric space in the scene.

There is some evidence from cue rotation experiments that different representations can be simultaneously active in the rat hippocampus (Wiener, Korshunov, Garcia, & Berthoz, 1995), and if so, the simultaneously active representations could be based on processes of the type studied in this paper. There is also evidence for multiple representations in the rat hippocampus from experiments in which individual visual cues in the rat's environment are moved. In such experiments (Tanila, Shapiro, & Eichenbaum, 1997a; Tanila, Sipila, Shapiro, & Eichenbaum, 1997b), the activity of many cells followed either the distal

or local cue sets, while other cells encoded specific subsets of the cues. In particular, it was demonstrated that some hippocampal place cells encode the egocentric location of a number of different subsets of environmental stimuli with respect to the rat (for a review, see Shapiro & Eichenbaum, 1999). These experiments suggest that the rat hippocampus may support multiple independent activity packets that represent different aspects of the spatial structure of the environment, with individual place cells taking part in more than one representation. Moreover, the fact that spatial cells in the hippocampus are not arranged in a topographic map suggests that the feature spaces are encoded by different random orderings of the cells. Whether it is the hippocampus or some other brain region that maintains a dynamical representation of the full 3D structure of the agent's environment to provide a robust basis for planning novel routes in complex, cluttered environments remains to be shown.

Acknowledgements

This research was supported by the Wellcome Trust, by the Human Frontier Science Program the MRC (by a Programme Grant to E. T. Rolls), and by the MRC Interdisciplinary Research Centre for Cognitive Neuroscience.

Appendix A. Mathematical analysis of interaction between activity packets

A.1. Interaction between activity packets in the discrete model

In this section we examine how activity packets in different feature spaces interact in the discrete model described in Section 2. Let us consider a fully connected continuous attractor neural network composed of N^{F} feature cells, which has been trained to encode two different feature spaces, x^{μ} and x^{ν}, where each of the feature spaces is encoded by a different random ordering of the N^{F} feature cells. In the following analysis, the feature spaces x^{μ} and x^{ν} we consider are 1D closed spaces from 0 to 2π radians, as implemented in the numerical simulations described above.

In the absence of visual input, the activations of the feature cells are governed by

$$\tau\frac{\mathrm{d}h_i^{\mathrm{F}}(t)}{\mathrm{d}t} = -h_i^{\mathrm{F}}(t) + \frac{\phi_0}{N^{\mathrm{F}}}\sum_j (w_{ij}^{\mathrm{RC}} - w^{\mathrm{INH}})r_j^{\mathrm{F}}(t) + \frac{\phi_1}{N^{\mathrm{F}}N^{\mathrm{ID}}}\sum_{j,k} w_{ijk}^{\mathrm{ID}} r_j^{\mathrm{F}} r_k^{\mathrm{ID}}, \quad \text{(A1)}$$

where N^{ID} is the number of idiothetic cells. Since we are assuming the network has been trained to encode two feature spaces x^{μ} and x^{ν} (where the synaptic weight updates for the two feature spaces have been combined additively), Eq. (A1) may be rewritten as

$$\tau\frac{\mathrm{d}h_i^{\mathrm{F}}(t)}{\mathrm{d}t} = -h_i^{\mathrm{F}}(t) + \frac{\phi_0}{N^{\mathrm{F}}}\sum_j (w_{ij}^{\mathrm{RC},\mu} + w_{ij}^{\mathrm{RC},\nu} - w^{\mathrm{INH}})r_j^{\mathrm{F}}(t) + \frac{\phi_1}{N^{\mathrm{F}}N^{\mathrm{ID}}}\sum_{j,k} (w_{ijk}^{\mathrm{ID},\mu} + w_{ijk}^{\mathrm{ID},\nu}) r_j^{\mathrm{F}} r_k^{\mathrm{ID}}, \quad \text{(A2)}$$

where $w_{ij}^{\mathrm{RC},\mu}$ are the components of the excitatory recurrent weights associated with the first feature space x^{μ}, $w_{ij}^{\mathrm{RC},\nu}$ are the components of the excitatory recurrent weights associated with the second feature space x^{ν}, $w_{ijk}^{\mathrm{ID},\mu}$ are the components of the idiothetic weights associated with the first feature space x^{μ}, and $w_{ijk}^{\mathrm{ID},\nu}$ are the components of the idiothetic weights associated with the second feature space x^{ν}.

If we combine the recurrent and idiothetic terms for each feature space, Eq. (A2) may be rewritten as

$$\tau\frac{\mathrm{d}h_i^{\mathrm{F}}(t)}{\mathrm{d}t} = -h_i^{\mathrm{F}}(t) + \frac{1}{N^{\mathrm{F}}}\sum_j W_{ij}^{\mu} r_j^{\mathrm{F}}(t) + \frac{1}{N^{\mathrm{F}}}\sum_j W_{ij}^{\nu} r_j^{\mathrm{F}}(t), \quad \text{(A3)}$$

where the terms W_{ij}^{μ} and W_{ij}^{ν} are resultant weights which encode the two feature spaces x^{μ} and x^{ν}, respectively, and which are given by

$$W_{ij}^{\mu} = \phi_0(w_{ij}^{\mathrm{RC},\mu} - w^{\mathrm{INH}}/2) + \frac{\phi_1}{N^{\mathrm{ID}}}\sum_k w_{ijk}^{\mathrm{ID},\mu} r_k^{\mathrm{ID}}, \quad \text{(A4)}$$

$$W_{ij}^{\nu} = \phi_0(w_{ij}^{\mathrm{RC},\nu} - w^{\mathrm{INH}}/2) + \frac{\phi_1}{N^{\mathrm{ID}}}\sum_k w_{ijk}^{\mathrm{ID},\nu} r_k^{\mathrm{ID}}. \quad \text{(A5)}$$

The resultant weights W_{ij}^{μ} and W_{ij}^{ν} are responsible for supporting activity packets in the two feature spaces x^{μ} and x^{ν}, respectively, and for shifting the activity packets through these spaces when the agent begins to move. Consider the weights W_{ij}^{μ} associated with the first feature space x^{μ}. As can be seen from Eq. (A4), the weights W_{ij}^{μ} are composed of the following two parts. The first part, $\phi_0(w_{ij}^{\mathrm{RC},\mu} - w^{\mathrm{INH}}/2)$, is symmetric in the space x^{μ}, and is responsible for stably supporting the activity packets in the space x^{μ}. The second part, $(\phi_1/N^{\mathrm{ID}})\sum_k w_{ijk}^{\mathrm{ID},\mu} r_k^{\mathrm{ID}}$, is asymmetric in the space x^{μ}, is dependent on the firing rate of the idiothetic cells r_k^{ID}, and is responsible for shifting the activity packets in the appropriate direction through the space x^{μ} when the agent moves.

Let us assume that at some time t the network activity vector $\mathbf{h}^{\mathrm{F}} = [h_1^{\mathrm{F}}, \ldots, h_{N^F}^{\mathrm{F}}]^{\mathrm{T}}$ is composed of two separate components $\mathbf{h}^{\mathrm{F},\mu}$ and $\mathbf{h}^{\mathrm{F},\nu}$, where $\mathbf{h}^{\mathrm{F},\mu}$ represents an activity packet in the space x^{μ}, and $\mathbf{h}^{\mathrm{F},\nu}$ represents an activity packet in the space x^{ν}. That is, for each feature cell i, the cell activation $h_i^{\mathrm{F}}(t)$ is composed as follows

$$h_i^{\mathrm{F}}(t) = h_i^{\mathrm{F},\mu}(t) + h_i^{\mathrm{F},\nu}(t). \quad \text{(A6)}$$

So far we have only defined the vectors $\mathbf{h}^{\mathrm{F},\mu}$ and $\mathbf{h}^{\mathrm{F},\nu}$ as representing static activity packets in the two spaces x^μ and x^ν at some time t. We must also define how the vectors $\mathbf{h}^{\mathrm{F},\mu}$ and $\mathbf{h}^{\mathrm{F},\nu}$ will continue to evolve through time, maintaining the relation (A6), where the evolution of each cell activation, $h_i^{\mathrm{F}}(t)$, continues to be governed by Eq. (A3). We define the evolution of the vectors $\mathbf{h}^{\mathrm{F},\mu}$ and $\mathbf{h}^{\mathrm{F},\nu}$ through time in the following manner.

In Eq. (A3) there are two neuronal interaction terms, $(1/N^{\mathrm{F}})\sum_j W_{ij}^\mu r_j^{\mathrm{F}}(t)$ and $(1/N^{\mathrm{F}})\sum_j W_{ij}^\nu r_j^{\mathrm{F}}(t)$, which contribute to the rate of change of the cell activations $(\mathrm{d}h_i^{\mathrm{F}}(t)/\mathrm{d}t)$. The first term is mediated by the weights W_{ij}^μ which are associated with the first feature space x^μ, and the second term is mediated by the weights W_{ij}^ν which are associated with the second feature space x^ν. Using these associations between weights and feature spaces, we define the temporal evolution of the vectors $\mathbf{h}^{\mathrm{F},\mu}$ and $\mathbf{h}^{\mathrm{F},\nu}$ as follows. We associate the rate of change of cell activations mediated by the weights W_{ij}^μ with the rate of change in the vector $\mathbf{h}^{\mathrm{F},\mu}$. Similarly, we associate the rate of change of cell activations mediated by the weights W_{ij}^ν with the rate of change in the vector $\mathbf{h}^{\mathrm{F},\nu}$. Using these definitions for the temporal evolution of the activation vectors $\mathbf{h}^{\mathrm{F},\mu}$ and $\mathbf{h}^{\mathrm{F},\nu}$, the evolution of the vectors $\mathbf{h}^{\mathrm{F}}$, $\mathbf{h}^{\mathrm{F},\mu}$, and $\mathbf{h}^{\mathrm{F},\nu}$ is governed by the following equations. For each feature cell i we have

$$\frac{\mathrm{d}h_i^{\mathrm{F}}(t)}{\mathrm{d}t} = \frac{\mathrm{d}h_i^{\mathrm{F},\mu}(t)}{\mathrm{d}t} + \frac{\mathrm{d}h_i^{\mathrm{F},\nu}(t)}{\mathrm{d}t} \tag{A7}$$

where

$$\tau\frac{\mathrm{d}h_i^{\mathrm{F},\mu}(t)}{\mathrm{d}t} = -h_i^{\mathrm{F},\mu}(t) + \frac{1}{N^{\mathrm{F}}}\sum_j W_{ij}^\mu r_j^{\mathrm{F}}(t) \tag{A8}$$

and

$$\tau\frac{\mathrm{d}h_i^{\mathrm{F},\nu}(t)}{\mathrm{d}t} = -h_i^{\mathrm{F},\nu}(t) + \frac{1}{N^{\mathrm{F}}}\sum_j W_{ij}^\nu r_j^{\mathrm{F}}(t). \tag{A9}$$

Eqs. (A8) and (A9) are used to define the temporal evolution of the activation vectors $\mathbf{h}^{\mathrm{F},\mu}$ and $\mathbf{h}^{\mathrm{F},\nu}$. Eqs. (A8) and (A9) are coupled through the firing rates $r_j^{\mathrm{F}}(t)$, which reflect the presence of both activity packets, $\mathbf{h}^{\mathrm{F},\mu}$ and $\mathbf{h}^{\mathrm{F},\nu}$. Eq. (A3) may be recovered by summing Eqs. (A8) and (A9).

We now consider how the vector $\mathbf{h}^{\mathrm{F},\mu}$ representing an activity packet in the first feature space x^μ evolves according to Eq. (A8). The firing rates of the feature cells are given by the sigmoid transfer function

$$r_i^{\mathrm{F}} = \frac{1}{1+\mathrm{e}^{-2\beta(h_i^{\mathrm{F}}-\alpha)}} = \frac{1}{1+\mathrm{e}^{-2\beta((h_i^{\mathrm{F},\mu}+h_i^{\mathrm{F},\nu})-\alpha)}}. \tag{A10}$$

However, since the sigmoid function is monotonically increasing, the cell firing rates may be decomposed into

$$r_i^{\mathrm{F}} = r_i^{\mathrm{F},\mu} + r_i^e, \tag{A11}$$

where $r_i^{\mathrm{F},\mu}$ is defined as the cell firing rate that would result from the presence of a single activity packet $\mathbf{h}^{\mathrm{F},\mu}$ in the first feature space x^μ, i.e.

$$r_i^{\mathrm{F},\mu} \equiv \frac{1}{1+\mathrm{e}^{-2\beta(h_i^{\mathrm{F},\mu}-\alpha)}}, \tag{A12}$$

and r_i^e is the component of the cell firing due to the presence of the additional activity packet $\mathbf{h}^{\mathrm{F},\nu}$ in the second feature space x^ν. With a non-linear sigmoid transfer function, it is evident that the noise term r_i^e from the second feature space x^ν depends on the activity $h_i^{\mathrm{F},\mu}$ in the first feature space x^μ.

Substituting Eq. (A11) into Eq. (A8) gives

$$\tau\frac{\mathrm{d}h_i^{\mathrm{F},\mu}(t)}{\mathrm{d}t} = -h_i^{\mathrm{F},\mu}(t) + \frac{1}{N^{\mathrm{F}}}\sum_j W_{ij}^\mu r_j^{\mathrm{F},\mu}(t) + \frac{1}{N^{\mathrm{F}}}\sum_j W_{ij}^\mu r_j^e. \tag{A13}$$

The system of Eq. (A13) governing the evolution of the activity profile $\mathbf{h}^{\mathrm{F},\mu}$ takes the form of a continuous attractor network model encoding a single feature space x^μ, but with additional inputs $(1/N^{\mathrm{F}})\sum_j W_{ij}^\mu r_j^e$ due to the presence of the activity packet $\mathbf{h}^{\mathrm{F},\nu}$ in the other feature space x^ν. The continuous attractor model defined by the system of Eq. (A13) without the input terms $(1/N^{\mathrm{F}})\sum_j W_{ij}^\mu r_j^e$ is identical to that investigated by Stringer et al. (2002b). This system is able to support a stable activity packet in the space x^μ using the symmetric components of the weight matrix W_{ij}^μ, and was shown by Stringer et al. (2002b) to be able to shift the activity packet in the correct direction as the agent begins to move using the asymmetric components of the weight matrix W_{ij}^μ.

When the input terms $(1/N^{\mathrm{F}})\sum_j W_{ij}^\mu r_j^e$ are present in the system of Eq. (A13), the effects of these inputs on the evolution of $\mathbf{h}^{\mathrm{F},\mu}$ may be ameliorated in two ways. Firstly, if the firing rates of the feature cells are somewhat binarised (as in the simulations described in this paper) this may contribute to the stability of the activity packets in each space. This is because shifting the activity packet requires activating new cells maximally, and this creates an energy barrier for the packet to move through. Secondly, the resetting of the sigmoid threshold α according to Eq. (4) helps to stabilise the activity packet. This effect has been explored by Stringer et al. (2002b).

However, even when the input terms $(1/N^{\mathrm{F}})\sum_j W_{ij}^\mu r_j^e$ are relatively large, the effects of these inputs on the evolution of $\mathbf{h}^{\mathrm{F},\mu}$ may be analysed as follows. We consider the situation in which the number of feature cells N^{F} in the network increases to infinity. We assume that the elements of the weight matrices reflect underlying continuous weight profiles $W^\mu(x^\mu, y^\mu)$ which are continuous functions of space x^μ, and that these profiles do not alter as $N^{\mathrm{F}} \to \infty$. Examples of the synaptic weight profiles learned by a relatively small network are shown in Figs. 6 and 7. Each of the plots reflects an underlying continuous weight profile in the relevant feature space, with additional noise due to the synaptic weight updates associated with the other feature space. In

addition, we assume the weight matrix is shift invariant, i.e.

$$W^{\mu}(x_1^{\mu}, x_1^{\mu} + y^{\mu}) = W^{\mu}(x_2^{\mu}, x_2^{\mu} + y^{\mu}) \tag{A14}$$

for $y^{\mu} \in [-2\pi, 2\pi]$.

This implies that the weight profile will be identical for each postsynaptic neuron i. This shift invariance is due to the homogeneous training of the network through all of the locations in each feature space. We also assume the shape of the activity packets in the two spaces x^{μ} and x^{ν} depends only on their respective spaces, and that the shapes of the packets reflect underlying continuous activity profiles which are continuous functions of space x^{μ}, and do not alter as the number of feature cells increases. This means that as $N^{\mathrm{F}} \to \infty$, the activity packets do not shrink relative to their spaces, and are instead represented by a greater density of feature cells.

Let us assume that the neurons are arranged randomly in the two different feature spaces; that is, the orderings of the neurons in the two spaces are uncorrelated with each other. Consider a stationary activity packet $\mathbf{h}^{\mathrm{F},\mu}$ in the feature space x^{μ}. Since the values of the noise terms r_j^e are dependent on the magnitudes of the activities $h_i^{\mathrm{F},\mu}$ in the first feature space x^{μ}, we obtain the following behaviour as size of the network increases. As $N^{\mathrm{F}} \to \infty$, the input terms $(1/N^{\mathrm{F}}) \sum_j W_{ij}^{\mu} r_j^e$ for each feature cell i situated in the space x^{μ} tend to a continuous function $s(x^{\mu})$, which varies with the activity profile $\mathbf{h}^{\mathrm{F},\mu}$ in the space x^{μ}. Now consider an activity packet $\mathbf{h}^{\mathrm{F},\mu}$ moving through different locations in the feature space x^{μ}. In this case the input function $s(x^{\mu}, t)$ will change through time. However, since the weight matrices W^{μ} are shift invariant, the input function $s(x^{\mu}, t)$ takes the form of a fixed wave profile that follows the activity packet profile $\mathbf{h}^{\mathrm{F},\mu}$ in the space x^{μ}. That is, if we consider an activity packet $\mathbf{h}^{\mathrm{F},\mu}$ moving through the first feature space x^{μ}, then the input $s(x^{\mu}, t)$ at each moving point of the wave profile $\mathbf{h}^{\mathrm{F},\mu}$ remains constant through time. Thus, as the size of the network is increased, the input to an activity packet $\mathbf{h}^{\mathrm{F},\mu}$ in a feature space x^{μ} from the activity packet in the other feature space x^{ν} tends towards a fixed wave profile $s(x^{\mu}, t)$ which follows the movement of the activity packet profile $\mathbf{h}^{\mathrm{F},\mu}$ in the first space x^{μ}. As the number of feature cells increases, Eq. (A13) tends towards

$$\tau \frac{\mathrm{d}h_i^{\mathrm{F},\mu}(t)}{\mathrm{d}t} = -h_i^{\mathrm{F},\mu}(t) + \frac{1}{N^{\mathrm{F}}} \sum_j W_{ij}^{\mu} r_j^{\mathrm{F},\mu}(t) + s(x^{\mu}, t). \tag{A15}$$

As the number of feature cells N^{F} in the network increases, the behaviour of the network equation (A15) tends towards that of a continuous neural field model, in which there is a single activation function $h(x,t)$ which is a continuous function of space x and time t. To complete the analysis of the effects of the input $s(x^{\mu}, t)$ on the evolution of the activity profile $\mathbf{h}^{\mathrm{F},\mu}$, we now turn to the corresponding neural field model, which permits us to apply the analytical methods of Amari (1977).

A.2. Dynamics of activity packets in a continuous neural field model

As the number of feature cells N^{F} in the discrete network model increases, the behaviour of the network equation. (A15) tends towards that of a continuous neural field model of the form

$$\tau \frac{\partial h^{\mathrm{F},\mu}(x^{\mu}, t)}{\partial t} = -h^{\mathrm{F},\mu}(x^{\mu}, t) + \frac{1}{2\pi} \int_0^{2\pi} W^{\mu}(x^{\mu}, y^{\mu}) \times r^{\mathrm{F},\mu}(y^{\mu}, t) \mathrm{d}y^{\mu} + s(x^{\mu}, t), \tag{A16}$$

where $h^{\mathrm{F},\mu}(x^{\mu}, t)$ is the neural field activation function which is a continuous function of space x^{μ} and time t, $W^{\mu}(x^{\mu}, y^{\mu})$ is a continuous weight function that describes the interactions within the neural field $h^{\mathrm{F},\mu}(x^{\mu}, t)$, and $r^{\mathrm{F},\mu}(y^{\mu}, t)$ is the firing rate given by the sigmoid transfer function

$$r^{\mathrm{F},\mu}(x^{\mu}, t) = \frac{1}{1 + \mathrm{e}^{-2\beta(h^{\mathrm{F},\mu}(x^{\mu},t) - \alpha)}}. \tag{A17}$$

Eq. (A16) has an input $s(x^{\mu}, t)$, which is due to the effects of the activity packet in the second feature space x^{ν}, and which takes the form of a wave function that moves with the activity profile $h^{\mathrm{F},\mu}(x^{\mu}, t)$ in the space x^{μ}. The continuous neural field formulation (A16) permits us to apply the analytical methods of Amari (1977). The theory developed in this section is an extension of that developed for an infinite 1D space by Amari (1977), which assumed a step transfer function and binary firing rates. The consideration of a closed finite space permits a more general analysis with arbitrary monotonic (e.g. sigmoid) transfer functions.

We are interested in the conditions on $s(x^{\mu}, t)$ which are necessary for the system (A16) to settle into a fixed wave profile $h^{\mathrm{F},\mu}(x^{\mu}, t)$ which represents a stable activity packet moving through the space x^{μ}. Consider a location $x_p^{\mu}(t)$ which tracks a particular point on the activity wave $h^{\mathrm{F},\mu}(x^{\mu}, t)$ with activation

$$h^{\mathrm{F},\mu}(x_p^{\mu}(t), t) = h_p. \tag{A18}$$

Now consider the state of the system a short time $\mathrm{d}t$ later. We have

$$h^{\mathrm{F},\mu}(x_p^{\mu}(t) + \mathrm{d}x_p^{\mu}, t + \mathrm{d}t) = h_p, \tag{A19}$$

where $x_p^{\mu}(t + \mathrm{d}t) = x_p^{\mu}(t) + \mathrm{d}x_p^{\mu}$. A Taylor expansion of $h^{\mathrm{F},\mu}(x_p^{\mu}(t) + \mathrm{d}x_p^{\mu}, t + \mathrm{d}t)$ about the point $(x_p^{\mu}(t), t)$ gives (following from Amari, 1977)

$$h^{\mathrm{F},\mu}(x_p^{\mu}(t) + \mathrm{d}x_p^{\mu}, t + \mathrm{d}t) = h^{\mathrm{F},\mu}(x_p^{\mu}(t), t) + \frac{\partial h^{\mathrm{F},\mu}(x_p^{\mu}, t)}{\partial x^{\mu}} \mathrm{d}x_p^{\mu} + \frac{\partial h^{\mathrm{F},\mu}(x_p^{\mu}, t)}{\partial t} \mathrm{d}t + \text{higher order terms}. \tag{A20}$$

Substituting Eqs. (A18) and (A19) into Eq. (A20) gives

$$\frac{\partial h^{\mathrm{F},\mu}(x_p^\mu,t)}{\partial x^\mu}\mathrm{d}x_p^\mu + \frac{\partial h^{\mathrm{F},\mu}(x_p^\mu,t)}{\partial t}\mathrm{d}t = 0, \tag{A21}$$

for infinitesimally small $\mathrm{d}x_p^\mu$ and $\mathrm{d}t$. Then from Eq. (A21) we have

$$\frac{\mathrm{d}x_p^\mu}{\mathrm{d}t} = -\frac{\partial h^{\mathrm{F},\mu}}{\partial t}\Big/\frac{\partial h^{\mathrm{F},\mu}}{\partial x^\mu}. \tag{A22}$$

If we define

$$c_p \equiv \frac{\partial h^{\mathrm{F},\mu}(x_p^\mu(t),t)}{\partial x^\mu}, \tag{A23}$$

then substituting Eqs. (A16) and (A23) into Eq. (A22) gives (following from Amari, 1977)

$$\frac{\mathrm{d}x_p^\mu}{\mathrm{d}t} = -\frac{1}{\tau c_p}\left(-h_p + \frac{1}{2\pi}\int_0^{2\pi} W^\mu(x_p^\mu,y^\mu) r^{\mathrm{F},\mu}(y^\mu,t)\mathrm{d}y^\mu + s(x_p^\mu,t)\right). \tag{A24}$$

For the system (A16) to settle into a fixed wave profile $h^{\mathrm{F},\mu}(x^\mu,t)$ moving at constant speed through the feature space x^μ, the velocity of each location $x_p^\mu(t)$ on the activity wave given by Eq. (A24) must be constant through time. We note that, because the weight matrix W^μ is shift invariant, for a moving activity wave $h^{\mathrm{F},\mu}(x^\mu,t)$ the term $\int_0^{2\pi} W^\mu(x_p^\mu,y^\mu) r^{\mathrm{F},\mu}(y^\mu,t)\mathrm{d}y^\mu$ will be constant for each location x_p^μ of the activity wave profile. Therefore, for the velocity $(\mathrm{d}x_p^\mu/\mathrm{d}t)$ of each location $x_p^\mu(t)$ on the activity wave to be constant through time, the input function $s(x_p^\mu,t)$ at each location $x_p^\mu(t)$ of the activity wave must remain equal to some constant s_p through time. This in turn requires that the input function $s(x^\mu,t)$ takes the form of a wave function that moves with the activity profile $h^{\mathrm{F},\mu}(x^\mu,t)$ in the space x^μ. If the input $s(x^\mu,t)$ at a location x_p^μ is randomly perturbed from its constant value s_p, then from Eq. (A24) there will be a perturbation in the velocity of the activation profile at that location, and the shape of the packet will be deformed.

Let us again consider the discrete system analysed above in Appendix A.1. The random perturbations in the input terms $(1/N^{\mathrm{F}})\sum_j W_{ij}^\mu r_j^e$ (from a fixed profile which follows the activity packet profile in the space x^μ) will be largest in small networks, but reduce as the number of feature cells N^{F} in the network increases (as discussed above in Appendix A.1). Hence, this analysis provides an explanation for the numerical results of experiments 4a and 4b. That is, why there is less time variation in the shape and size of the activity packets in the different spaces, as the size of the network increases.

References

Amari, S. (1977). Dynamics of pattern formation in lateral-inhibition type neural fields. *Biological Cybernetics*, *27*, 77–87.

Battaglia, F. P., & Treves, A. (1998). Attractor neural networks storing multiple space representations: a model for hippocampal place fields. *Physical Review E*, *58*, 7738–7753.

Ermentrout, G. B., & Cowan, J. D. (1979). A mathematical theory of visual hallucination patterns. *Biological Cybernetics*, *34*, 137–150.

Földiák, P. (1991). Learning invariance from transformation sequences. *Neural Computation*, *3*, 194–200.

Hertz, J., Krogh, A., & Palmer, R. G. (1991). *Introduction to the theory of neural computation*. Wokingham, UK: Addison Wesley.

Koch, C. (1999). *Biophysics of computation*. Oxford: Oxford University Press.

Mozer, M. C. (1991). *The perception of multiple objects: A connectionist approach*. Cambridge, MA: MIT Press.

O'Keefe, J., & Conway, D. H. (1978). Hippocampal place units in the freely moving rat: why they fire where they fire. *Experimental Brain Research*, *31*, 573–590.

O'Mara, S. M., Rolls, E. T., Berthoz, A., & Kesner, R. P. (1994). Neurons responding to whole-body motion in the primate hippocampus. *Journal of Neuroscience*, *14*, 6511–6523.

Recce, M., & Harris, K. D. (1996). Memory for places: a navigational model in support of Marr's theory of hippocampal function. *Hippocampus*, *6*, 735–748.

Redish, A. D. (1999). *Beyond the cognitive map: From place cells to episodic memory*. Cambridge, MA: MIT Press.

Redish, A. D., Elga, A. N., & Touretzky, D. S. (1996). A coupled attractor model of the rodent head direction system. *Network: Computation in Neural Systems*, *7*, 671–685.

Redish, A. D., & Touretzky, D. S. (1998). The role of the hippocampus in solving the Morris water maze. *Neural Computation*, *10*, 73–111.

Rolls, E. T., Aggelopoulos, N. C., & Zheng, F. (2002a). The receptive fields of inferior temporal cortex neurons in natural scenes. *Journal of Neuroscience*, *23*, 339–348.

Rolls, E. T., & Arbib, M. (2003). *Visual scene perception* (2nd ed.). *Handbook of brain theory and neural networks*, Cambridge, MA: MIT Press, pp. 1210–1251.

Rolls, E. T., & Deco, G. (2002). *Computational neuroscience of vision*. Oxford: Oxford University Press.

Rolls, E. T., Stringer, S. M., & Trappenberg, T. P. (2002b). A unified model of spatial and episodic memory. *Proceedings of The Royal Society B*, *269*, 1087–1093.

Rolls, E. T., & Treves, A. (1998). *Neural networks and brain function*. Oxford: Oxford University Press.

Samsonovich, A (1998). Hierarchical multichart model of the hippocampal cognitive map. *Proceedings of the fifth joint symposium on neural computation* (pp. 140–147).

Samsonovich, A., & McNaughton, B. (1997). Path integration and cognitive mapping in a continuous attractor neural network model. *Journal of Neuroscience*, *17*, 5900–5920.

Shapiro, M. L., & Eichenbaum, H. (1999). Hippocampus as a memory map: synaptic plasticity and memory encoding by hippocampal neurons. *Hippocampus*, *9*, 365–384.

Skaggs, W. E., Knierim, J. J., Kudrimoti, H. S., & McNaughton, B. L. (1995). A model of the neural basis of the rat's sense of direction. In G. Tesauro, D. S. Touretzky, & T. K. Leen (Eds.), (*Vol. 7*) (pp. 173–180). *Advances in neural information processing systems*, Cambridge, MA: MIT Press.

Stringer, S. M., Rolls, E. T., & Trappenberg, T. P (2003a). Self-organizing continuous attractor network models of hippocampal spatial view cells.

Stringer, S. M., Rolls, E. T., Trappenberg, T. P., & de Araujo, I. E. T. (2002a). Self-organizing continuous attractor networks and path integration: two-dimensional models of place cells. *Network: Computation in Neural Systems*, *13*, 429–446.

Stringer, S. M., Trappenberg, T. P., Rolls, E. T., & de Araujo, I. E. T. (2002b). Self-organizing continuous attractor networks and path integration: one-dimensional models of head direction cells. *Network: Computation in Neural Systems*, *13*, 217–242.

Stringer, S. M., Rolls, E. T., Trappenberg, T. P., & de Araujo, I. E. T. (2003b). Self-organizing continuous attractor networks and motor function. *Neural Networks*, *16*, 166–182.

Tanila, H., Shapiro, M., & Eichenbaum, H. (1997a). Discordance of spatial representations in ensembles of hippocampal place cells. *Hippocampus*, *7*, 613–623.

Tanila, H., Sipila, P., Shapiro, M. L., & Eichenbaum, H. (1997b). Brain aging: impaired coding of novel environmental cues. *Journal of Neuroscience*, *17*, 5167–5174.

Taylor, J. G. (1999). Neural bubble dynamics in two dimensions: foundations. *Biological Cybernetics*, *80*, 393–409.

Treves, A. (1991). Are spin-glass effects relevant to understanding realistic autoassociative networks? *Journal of Physics A*, *24*, 2645–2654.

Tsodyks, M. (1999). Attractor neural network models of spatial maps in Hippocampus. *Hippocampus*, *9*, 481–489.

Wallis, G., & Rolls, E. T. (1997). Invariant face and object recognition in the visual system. *Progress in Neurobiology*, *51*, 167–194.

Wiener, S. I., Korshunov, V. A., Garcia, R., & Berthoz, A. (1995). Inertial, substratal and landmark cue control of hippocampal CA1 place cell activity. *European Journal of Neuroscience*, *7*, 2206–2219.

Wilson, M. A., & McNaughton, B. L. (1993). Dynamics of the hippocampal ensemble code for space. *Science*, *261*, 1055–1058.

Zhang, K. (1996). Representation of spatial orientation by the intrinsic dynamics of the head-direction cell ensemble: a theory. *Journal of Neuroscience*, *16*, 2112–2126.