View–based cognitive mapping and path planning

Bernhard Schölkopf and Hanspeter A. Mallot
Max–Planck–Institut für biologische Kybernetik
72076 Tübingen Germany

Present addresses of the authors:

Bernhard Schölkopf
Max-Planck-Institut für biologische
Kybernetik
Spemannstr. 38
72076 Tübingen, Germany
http://www.kyb.tuebingen.mpg.de

Hanspeter A. Mallot
Cognitive Neuroscience Lab
Department of Zoology
Auf der Morgenstelle 28
72076 Tübingen, Germany
http://www.uni-tuebingen.de/cog

Abstract

This paper presents a scheme for learning a cognitive map of a maze from a sequence of views and movement decisions. The scheme is based on an intermediate representation called the *view graph*, whose nodes correspond to the views while the labelled edges represent the movements leading from one view to another. By means of a graph theoretical reconstruction method, the view graph is shown to carry complete information on the topological and directional structure of the maze. Path planning can be carried out directly in the view graph without actually performing this reconstruction. A neural network is presented that learns the view graph during a random exploration of the maze. It is based on an unsupervised competitive learning rule translating temporal sequence (rather than similarity) of views into connectedness in the network. The network uses its knowledge of the topological and directional structure of the maze to generate expectations about which views are likely to be encountered next, improving the view recognition performance. Numerical simulations illustrate the network's ability for path planning and the recognition of views degraded by random noise. The results are compared to findings of behavioural neuroscience.

1   Introduction

1.1   Navigation and cognitive maps

Navigation can be defined as the planning and travelling of a path in space. It is a complex task consisting of several different levels. In course control, obstacles must be avoided, passive deviations (e.g. by wind) must be compensated for, etc. If we navigate through a town, say, this level is concerned with avoiding bumping into houses and cars. The cognitive level becomes important if we have a goal. We then need to take path decisions at each crossing we encounter (cf. Tani and Fukumura 1994, and the "levels of competence" of Brooks 1987). In open environments, we do have more freedom to change paths. However, it is still a good strategy to use salient landmarks to define places where to decide on the future path.

We use the term *map behaviour* to refer to behaviours like free exploration, repeating previously travelled routes, making use of knowledge about directions and/or distances to certain places (possibly using path integration and compass directions), finding detours around obstacles, approaching a goal from a novel starting point, finding a path between two arbitrary points of the environment, or optimizing the path with respect to "costs" like distance, safety, energy consumption. A *cognitive map* is a neural mechanism which supports navigation and orientation tasks similar to the ones we can solve by using a real map of the environment (see Gallistel 1990). To this end, it must contain two types of information, concerning (a) the recognition of places and (b) the spatial relations between them (including movements joining them). Information (a) can be included by storing representations of environmental cues in the map, like street names or landmarks. Examples of (b) are adjacency of places (subway plan), metrical information about possible paths (street plan), or about the complete environment (hiking map). With information type (a), the map helps one to determine one's own position, while (b) can be used for *path planning*.

We include these two types of information in a *view graph*. We will argue that this structure — containing information about views, their mutual topological relationships and movements linking the views — is a sufficient representation of spatial knowledge. In learning the view graph, the observer is in general facing the problem of selecting the views to be remembered. In a maze, however, a canonical set of views exists which ought to be used: every time one comes to a branching place, a decision needs to be taken — therefore, we can simply use the views of those places as perceived on approaching them through the corridors.

This article addresses the question how a cognitive map could be constructed which supports appropriate behaviour in the perception–action–cycle sketched in Fig. 1. We first introduce the concept of a view graph and demonstrate that it contains sufficient information to reconstruct the topological and directional structure of the maze. A simple scheme for path–planning on the view graph (rather than on the reconstructed maze) is also included. We then present a neural network model which is able to learn this view graph, and a mechanism which uses this information for path planning. Results of simulation runs are provided, and some these
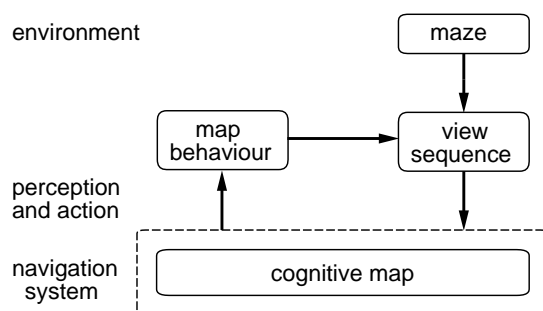

Figure 1: The perception–action–cycle for a navigation system.

dataare used to discuss exploration strategies that can be derived from our theory. Finally, we relate our simulation results to findings of behavioural neuroscience.

1.2 The role of hippocampal "place cells" in spatial orientation

As the network model is inspired by the findings on hippocampal "place cells" in rodents, we will briefly review these findings. The hippocampus is usually considered to be of importance for both the rapid acquisition of new memories and the consolidation and transformation of these memories into long–term representations in the neocortex (see McClelland, McNaughton and O'Reilly 1994). As spatial orientation depends on rapidly learning particular combinations of cues, this view is consistent with the important role that the hippocampus is thought to play in spatial navigation tasks. In rodents, hippocampal complex–spike cells have been recorded which encode the spatial location of the animal by correlating their firing rate to the position of the animal ("place cells"); they mainly fire if the animal is located in their respective "place fields"; often they additionally code direction and speed of movement (McNaughton, Barnes and O'Keefe 1983, Eichenbaum and Cohen 1988). For the definition of a place field for a given cell, visual cues seem to be more important than olfactory and auditory ones (O'Keefe 1979). However, once a place field is established, it persists after removal of cues (O'Keefe and Speakman 1987) — the neural activity represents not only sensory input but also internal states related to the cognitive map. The latter is supported by the finding that place cell firing may also code the animal's position in the near future (Muller and Kubie 1989). Unlike topology preserving maps in other parts of the brain (in the visual cortex, the somatosensory cortex, or the auditory cortex), the mapping from the hippocampus into space is not continuous since nearby cells can have place fields far apart: O'Keefe and Speakman (1987) rule out significant positive interrelations between the spatial firing rates of nearby cells in most parts of the hippocampus. In fact, a small group of neighbouring cells sufficed to represent most of the environment.

Most of the place cells are located in the hippocampal subfields CA3 and CA1. It is known that at least within CA3 there are extensive excitatory lateral interactions. Moreover, there is strong evidence for synaptic weight changes in form of long term potentiation in the hippocampus, and hippocampal circuits do change as a result of exploratory activity in rats. This supports the idea that the hippocampus might incorporate some kind of associative memory (for a review see McNaughton and Nadel, 1990). The view that the hippocampus deals with sequential aspects of associations has been proposed by Braitenberg and Schüz (1983).

Very little is known on how this spatial information could actually be used for navigation purposes. The "functional fields" discussed by Eichenbaum, Wiener, Shapiro and Cohen (1989) which represent non–spatial along with spatial information might indicate a possible interface between spatial knowledge and action. Movement–related activity has for example been found in the posterior parietal cortex: cells have been recorded which exhibit an elevation of firing rate during and after specific locomotion modes, such as right turns (McNaughton et al. 1994).

As we shall see in the next section, many models for orientation assume that learning takes place with respect to a goal that is to be approached afterwards. This does not seem to be the case for rodents: O'Keefe (1991) reports that the hippocampal representation of an environment is not altered by relocation of the goal. The component of the navigation system which decides where to go to seems to be located outside the hippocampus.

1.3 Some models for spatial orientation

Out of the vast literature on models for spatial orientation (for a review, see O'Keefe 1991), we mention only a few.[1] The models differ in how much importance they assign to the following

---

[1]In particular, we focus on the acquisition of spatial knowledge by *learning*. Recently, interesting examples of map behaviour have also been produced using evolutionary methods (see Floreano and Mondada, 1994).

issues:

1. The closeness of the model to neurobiological data (Does it reproduce place cell firing? What kind of learning rules does it employ?)

2. The explanatory power with respect to behavioural data (Does the model reproduce place field persistence at cue removal, and invariance of place fields under goal relocation?)

3. The possible applicability in robotics (Does navigation to a new goal require new training? Is the model robust enough to cope with noisy sensory input?).

We next discuss key ideas developed in spatial orientation, along with some exemplary models.

1.3.1   Associations between sensory input and motor commands

The stimulus–response approach to animal behaviour suggests to implement navigation systems as functions which map the current sensory input to unique motor commands (cf. the associative search model of Barto and Sutton 1981; and the model of Nehmzov and McGonigle 1994, using supervised learning). However, suppose there are two locations where the sensory inputs of such a navigation system happen to be identical. Then the above function yields identical motor commands, even if the two different locations call for different movements to be executed. This problem can be avoided by allowing the system to have internal states which include knowledge about the context in which sensory inputs are perceived.

1.3.2   Goal–directed navigation

Suppose somebody organises a party, inviting guests from different parts of town. The street plan sent out with the invitation would probably include different ways leading to the venue, but only few (if any) redundant paths connecting different points of the surroundings, in order not to confuse the visitors. This map is then easy to use, but it works only for navigation to the party venue (and points that happen to lie on the way). For cognitive maps, the situation is similar: if the navigation goal is already known during map learning, then the problem of using the map becomes easier, but map applicability is restricted.

Tani and Fukumura's (1994) robot navigation model learns sets of paths leading to a common goal. However, the navigation to another goal then requires new training, even if the new goal has been seen before. The snapshot model of Cartwright and Collett (1987) generates goal–directed motion in a model bee in the following way: far away from the hive, the bee uses a map built on remembered snapshots, each one with a vector specifying the direction and the distance of the hive. Close to the hive, the bee calculates a homing vector from the difference between current retinal image and stored snapshots of the goal. Burgess, Recce and O'Keefe (1994) use a sophisticated mechanism of how to generate the direction of the goal from sensory input consisting of tuning curve responses to distances of environmental cues. They argue that the θ rhythm in the hippocampal EEG could provide a frame of reference for the extraction of information from the firing phase of place cells. In their model, phases code for angular positions of cues, and cells downstream of the hippocampus read a hippocampal map in order to provide a population vector which allows navigation to a previously encountered reward site.

1.3.3   Associations between locations

The idea that animals project a *means-end field* (a directed graph with means and goal-objects as vertices and means-end relations as edges) on an environment dates back to Tolman (1932). McNaughton (1989), who proposed to store possible place transitions in an association matrix, with the places linked by movements. He further suggests that this matrix could be coded with "compound place/movement representations", which allow the prediction of future places,

given a certain movement. To our knowledge, no simulation of this model has been published so far. In a related scheme presented by Schmajuk and Thieme (1992) associations between neighboring places are learned with a neural network in order to build a topological map for path planning. Unlike McNaughton, Schmajuk and Thieme introduce a distinction between places and views. They conceive of places as regions in a maze and of views as sights available from places. Their network, however, assumes a one-to-one correspondence between places and views; a place is required to correspond to the same view no matter from which neighboring place the network sees it.[2]

Although very useful in certains environments, topological maps cannot account for all forms of animal map behavior. In the following paragraphs, we discuss additional information that can be used.

1.3.4   Metric maps, compass orientation, and path integration

Metric maps (e.g., Prescott 1994) can be used to derive predictions about short-cut or straight-line routes across unexplored terrain, using vector algebra. In nontrivial mazes, these predictions are less reliable, as the unexplored terrain might contain unexpected wall.[3] Nevertheless, in the case where old paths are blocked, metric maps could serve as conjectures about possible other ways, guiding detour behavior. In experiments using a 4-arm maze, Cheng and Gallistel (1984) showed that rats rely on metrical properties of the layout of a set of landmarks visible from the whole maze. In the model of O'Keefe (1991), a rat builds an allocentric (i.e., world-centered) map of its environment by constructing an absolute direction and an absolute reference point from the overall slope and the centroid of a cue configuration, respectively. O'Keefe (1991) suggests that the necessary vector calculations could be done with neuronal sinusoids in the θ system, representing the angle and the distance to a cue by the phase and the amplitude of a sine wave.

Through path integration, bees and ants are able to keep track of their position relative to their starting point. To this end, they use skylight compasses for measuring directions (see Wehner & Menzel, 1990). In a rat navigation model, Touretzky, Wan and Redish (1994) combine visual input, path integration and vestibular information to construct place cells coding combinations of internal and external states.

1.3.5   The present approach: Learning the view graph

The present model combines ideas from two different sources: First, it has been inspired by models memtioned in section 1.3.3 (McNaughton, 1989; Schmajuk & Thiems, 1992). Second, it has been influenced by recent work on view-based object recognition (e.g., Bülthoff & Edelmann, 1992; we shall retrun to this point in section 6.4). We hope that joining these ideas has allowed us to clarify the distinction between direction-dependent views (i.e., the sensory input of the system) and the places inferred from them. The following point characterize our approach:

- The present model gives a formal account of the relationship between views and places. For a maze environment with local views, we show that the view graph is sufficient for navigation.

- The input in our model is constituted by sequences of views and movements. Thus, movements are not treated as mere reactions to sensory input but are an essential part of the information forming the cognitive map.

---

[2]The distinction between places and views that we will introduce in the present work differs from the one of Schmajuk and Thieme in that we do not assume this one-to-one correspondence. As described later, we think of places as constructed from several views of a maze junction.

[3]Dartsch and Mallot (1995) recently reported that in mazes humans tend to use topological distance measures rather than metrical ones.
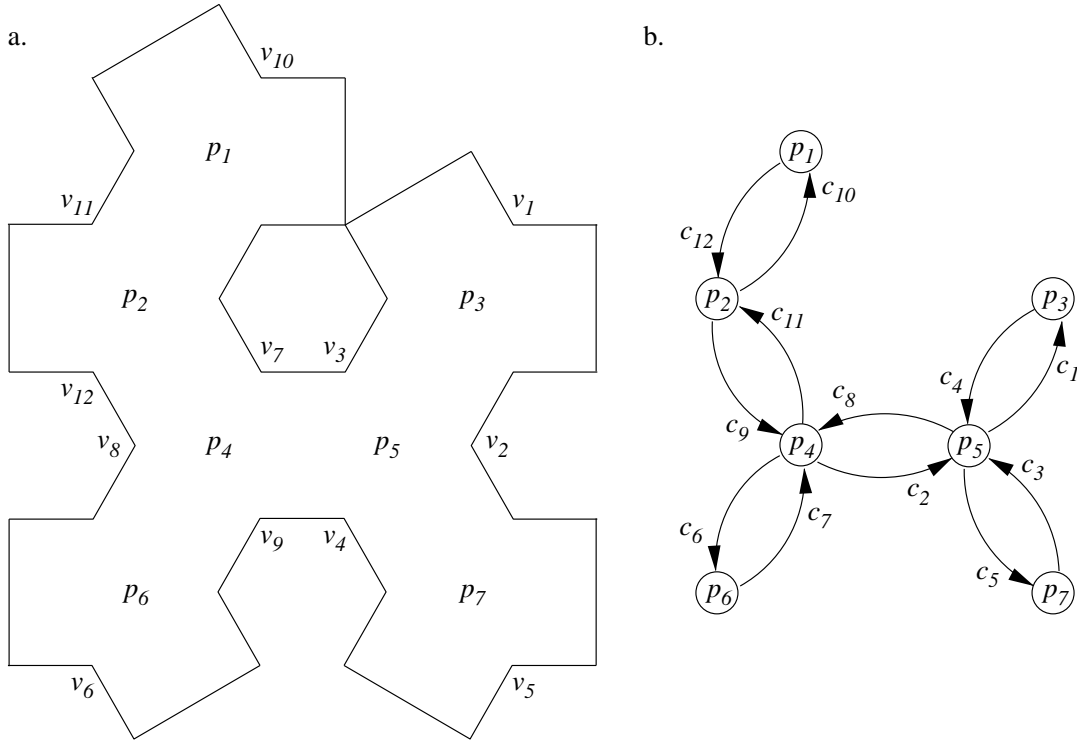
Figure 2: (a.) Simple maze with places p_i ($i = 1, \ldots, 7$) and views v_j ($j = 1, \ldots, 12$). One can think of the views as visual input available during approaching a junction through the corresponding corridor (e.g. pictures attached to the maze walls in the direction of heading, or visual information gathered along the corridor). (b.) Directed place graph of the maze, with corridors c_j corresponding to the views v_j.

- Our model exhibits a number of properties similar to findings of behavioural neuroscience, including directional specificity of place cells, the use of stored map information to support view recognition, and the fact that map learning is independent from future goal locations.

2 Places, views, and movements

2.1 Place graph and view graph

Consider a simple maze composed of *places* $p_1, \ldots, p_n$ and *corridors* $c_1, \ldots, c_m$. One way to think of the topological structure of this maze is a graph where the places are the vertices and the corridors are the edges: a *place graph* (Fig. 2). We consider all corridors to be directed but allow for the existence of two reversely directed corridors between any two vertices.

When walking through a maze, the explorer executes a sequence of *movements* to encounter a sequence of *views*. We can think of this exploration as a walk on the *view graph*. The edges of the view graph indicate temporal coherence: two views are connected if and only if they can be experienced in immediate temporal sequence. Each edge of the view graph is labelled by the egocentric movement the explorer had to execute in order to perceive the corresponding view sequence. While the edges make up the *topological* structure of the view graph, the movement labels represent its (local) *directional* structure.

2.2 Recovering places from views

Concerning the layout of the maze, we make two assumptions: First, that we deal with connected mazes consisting of at least two places. Second, we assume that there is a one–to–one correspondence between directed corridors and views, with each view attached to the wall
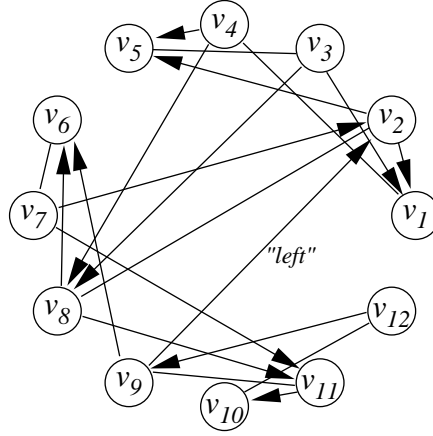
Figure 3: View graph of the place graph in Fig. 2. The views v_i correspond one–to–one to the corridors c_i (the directed edges) in the place graph. Lines without arrow–heads represent two reversely directed edges. Each directed edge carries a unique movement label (only one of them is displayed). Of course, the graph can be displayed simpler (with less crossings), however this could obscure the fact that only the connectivity structure and the movement labels are relevant.

opposite of the entry of the corresponding corridor into a place. As a consequence of this one–to–one correspondence, all views are distinguishable. Concerning the maze exploration, we assume the following: views can either be approached by walking towards them through the corresponding corridor (we will refer to this as *locomotion*) or by *turning* to them within a place, and it is always remembered how the views have been approached. Therefore, one will have the possibility to use selectively subsets of the view sequences encountered during exploration. In this paper we will mainly make use of those sequences of views that can be generated through locomotion,[4] unless explicitly stated otherwise.

In Fig. 3, the view graph (containing all possible view sequences generated by locomotion) is displayed. The vertices of the view graph are the views v_p, which by assumption can be identified with the corridors in the place graph. We denote the start and target place of a view v by $P_{out}(v)$ and $P_{in}(v)$, respectively. As an example, from Fig. 2 we can see that $P_{out}(v_{10}) = p_2$ and $P_{in}(v_{10}) = p_1$. Of course, the functions P_{in} and P_{out} are not known when exploring the maze. As we do not consider turning, P_{out} is well–defined; moreover the locomotion implies that $P_{out}(v)$ and $P_{in}(v)$ are adjacent places. Hence two views v_p and v_q are connected if and only if $P_{in}(v_q) = P_{out}(v_p)$. We denote by $A = (a_{pq})$ the *adjacency matrix* of the view graph:

$$a_{pq} = \begin{cases} 1 & \text{if } P_{in}(v_q) = P_{out}(v_p) \\ 0 & \text{otherwise} \end{cases} \qquad (1)$$

In terms of graph theory, the view graph generated from the locomotion view sequences is the *interchange graph* (e.g., Wagner 1970) of the place graph.

By means of the *successor set* $\mathcal{S}(v_p)$ of v_p, i.e. the set of views that can follow v_p,

$$\mathcal{S}(v_p) = \{v_r | a_{rp} = 1\}, \qquad (2)$$

we can reconstruct *places* from the view graph. The idea is that two views lead to the same place if and only if they have the same successor set, and that successors sets which are not identical must be disjoint (proved in Appendix A), a fact that can be used to derive information from incomplete view graphs (see section 5.2). The number of elements of the successor set

---

[4]Therefore the theory developed in the next section largely also applies to the case where the views are constituted by visual input gathered during approaching a junction through the corresponding corridor. In this case, turning toward views is impossible.
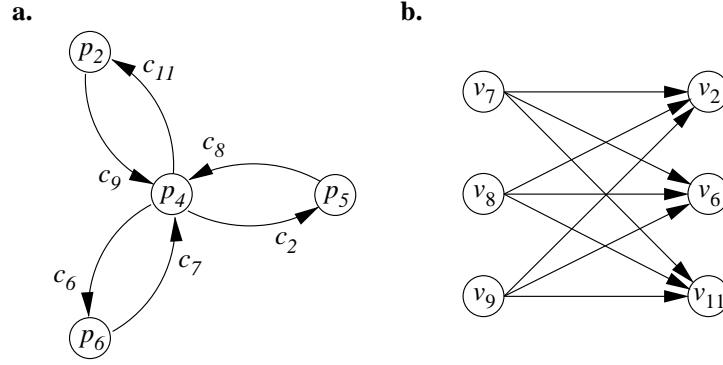
Figure 4: a.: Subgraph of the place graph in Fig. 3; b.: Subgraph of the view graph associated to p_4 of the place graph. It is a complete bipartite graph which is known to be non–planar (see Section 2.3).

of a view leading to place p is called the *out-degree* of p (the number of exits); the number of views leading to p is called its *in-degree.*

2.3   Is the view graph planar?

If we want to learn view graphs with a neural network, it is important to find a network topology in which they can be embedded. Given a graph, we can ask the question whether it can be represented without singularities or crossings in the plane (i.e., whether the graph is *planar*). This question bears relevance for the kind of neural network topology that can be used to learn view graphs: if we use a 2–dimensional grid network where connections are allowed to evolve only to the 4 nearest neighbours, the network connectivity graph will always be planar. Consequently, if we wanted to represent a view graph by the network connectivity graph, it would have to be planar as well. For mazes with bidirectional corridors, the criterion stated below tells us that if some place is lying at the intersection of three or more corridors (which is generally the case for mazes—for example places 4 and 5 in Fig. 2), then the view graph (in our case, Fig. 3) will not be planer.

The local criterion for non–planarity is: *If a graph contains a vertex p for which both in–degree(p) and out–degree(p) are greater than 2, its interchange graph is not planar.* This is true since the interchange graph contains a non–planar subgraph $K_{3,3}$ (Fig. 4) associated to p in the following way: it consists of 3 of the edges (views) $i_1, \ldots, i_3$ leading to p, fully connected to 3 edges $o_1, \ldots, o_3$ starting at p (successor views). Note that by the assumptions about the in– and out–degree of p there exist at least 3 of either. The complete bipartite graph $K_{3,3}$ is one of the so–called Kuratovski graphs, and its non–planarity can be proved using Euler's polyhedron equation (see Wagner, 1970).

2.4   Two extensions

2.4.1   Turning within places

If we include the view sequences labelled by a turn within a place into the view graph, we get the following:

- The view graph will contain rather than equal the interchange graph of the place graph.

- The subgraph of the view graph containing only the view sequences labelled by a turn has a disconnected structure, with the components being undirected graphs representing maze places. Thus we have yet another means of reconstructing places from views.

2.4.2   A complementary notion of places

In Sect. 2.2, we have introduced a notion of places as collections of views with the same successor set. The views then are perceived on arriving at a maze junction. An alternative way is to think of the views as belonging to the same place as those that can be seen on looking from the place into the corridors leading off. This would amount to saying that the views have the same predecessor sets. Our theory would look almost the same.

Interestingly, this dual situation can also be generated by walking backward in a maze. This suggests that our scheme could be extended such that it allows walking both forward and backward without substantial changes in the theory. A view sequence corresponding to a backward walk through a maze can be generated simultaneously with the forward sequence by looking back after each step. If we walk through a foreign city, say, we can use backtracking behaviour of this kind to make sure that we will later find our way back.

2.5   Labelling view graphs with movements

From the place graph drawing in Fig. 2, egocentric movement directions can be assigned to each transition $c_i \to c_j$. In the case of our maze where each junction stands for a ternary decision, we only need *go left*, *go right*, and *go back* as locomotion labels and *turn left, turn right* as turn labels. In the adjacency matrix, Eq. 1, movement labels can be included by asigning the appropriate movement type rather than the number 1 to the coefficients a_{pq}. Note that all edges starting at a given vertex will have different labels. The sequence of movement labels constitutes the path travelled through the maze.

2.6   Reconstruction of the directional structure of the maze

In Sect. 2.2, we have explained how to find the views belonging to the same place. This way, we were able to reconstruct the topological structure of the maze, i.e. the place graph. The movement information included in the view graphs can be used to reconstruct also the local directional structure of the places (i.e. the ordering of the views within a place, and the movement decisions necessary to get to a certain neighbouring place); the reconstruction method is described in Appendix 2. Knowledge of both the topology of the maze and the local directional structure of the places then amounts to knowing the maze structure up to the level of detail that we are dealing with in this paper (not including distance information, or bended corridors etc.).

In Sect. 2.2, we stated a constraint on the topology of view graphs, namely that successors are either disjoint or identical. There exists a similar constraint on the movement labels, allowing to infer unknown movement labels from incompletenesses in the (labelled) view graph: *neither does a transition matrix column contain two identical entries, nor does a row of the matrix* (proved in Appendix C).

2.7   Path planning

The problem of path planning is the following: *Given a view graph whose edges are labelled by movement decisions, and two arbitrary different vertices s and g of the graph. How can we find a sequence of movement decisions belonging to a path of minimal length, joining s to g?* To construct such a sequence, we use the following algorithm:

- For all possible movements μ (labels of outward edges) starting from s, do:

    - simulate execution of μ
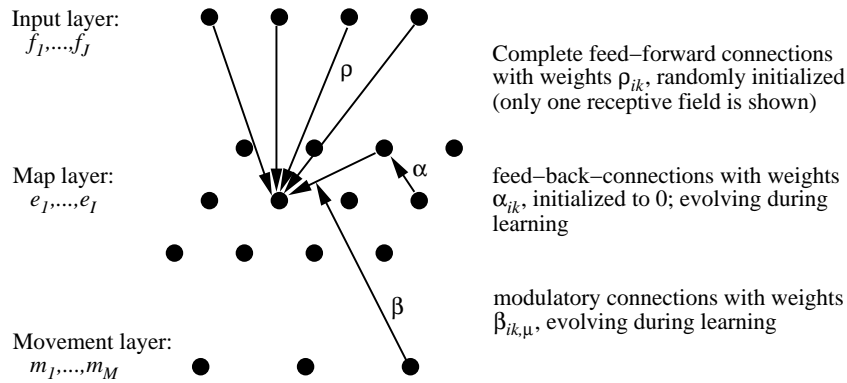   - mark the reached point

Figure 5: Sketch of the network topology. The J input units feed via weights ρ_{ij} into I map units which are interconnected via feedback weights α_{ik}. Through the ρ_{ij}, the network learns the views, and via the α_{ik}, the maze topology is learnt. Movement labels are learnt in terms of modulatory connections from the movement layer to map layer connections.

- — until g is marked: mark all vertices which can be reached from any marked vertex. The recursion depth of this procedure gives the distance to the goal.

- Add the μ for which the goal distance was minimal to the movement decision sequence.

- Simulate execution of this movement, and take the reached vertex as the new starting point s.

- If the new s is not equal to g, go back to the beginning.

3   A neural network model learning the view graph

3.1   Self–organizing sequence map

We construct a neural network consisting of three layers called input layer, map layer, and movement layer (Fig. 5). In order to support orientation behaviour in a maze, the network must acquire information concerning three problems:

P1 Each view must be identified and associated with a particular location in the map. In our network, this is achieved by a set of input weights ρ_{ij}. After learning, perception of a view will be represented by activity in the associated map unit, i.e. the unit whose input weights are most closely tuned to the presented view.[5]

P2 The topology of the maze must be learnt. This is accomplished by developing weights α_{ik} connecting units within the map layer. Weights between units will be assigned according to the temporal sequence of the views represented by these units.

P3 Path planning amounts to planning movement sequences which lead to a desired view. Therefore the network must store knowledge about which movement decisions are necessary to generate certain view sequences. This is done by developing modulatory connections from the movement layer to the map layer connections.

---

[5]This map unit could be thought of as a directional place cell, as the ones recorded from rat hippocampus by McNaughton, Barnes and O'Keefe (1983). McNaughton (1989) argues that most place cells are probably sensitive to direction.
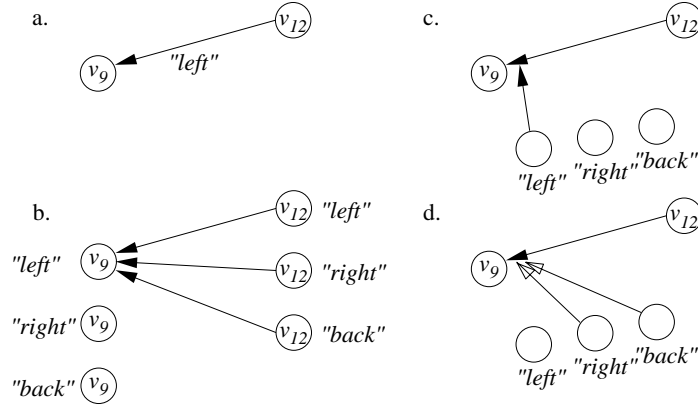
Figure 6: Movement information in the network. If in the place graph p_2 has been approached through c_{12}, the movement decision necessary to get to p_4 (through c_9) is *(go) left*. In the view graph, this can be coded by assigning a label to the corresponding edge (a.). Alternatively, we can triple the number of vertices of the view graph and assign the labels to the vertices (b.), or we label the edges by some kind of external markers pointing on them (c.,d.). Case c. is used in the network model.

While the view recognition part of the proposed network is reminiscent of a standard Kohonen map (Kohonen 1982), there are three important differences: First, nearness in the map corresponds to temporal adjacency, not to featural similarity. In fact, similar views can occur at great distances in the maze and must not be confused. Second, distance in the map is measured as the minimal number of synapses that must be passed between two units (the "combinatorial distance" in graph theory), rather than using a metric on a predefined space with a given dimensionality. Therefore the topological structure of the resulting "map" is not limited (see Martinetz and Schulten, 1994). This is desirable since view graphs need not be planar, as shown in Sect. 2.3. Third, in addition to learning topology, the network also learns about means of reaching neighbouring feature vectors (i.e., views) by actions.

3.2   Description of the model

3.2.1   Network structure

Topology.   The network consists of an *input layer* (J units) with activity variables f_j^t, a *map layer* (I units) with activity variables e_i^t and thresholds θ_i^t, and a *movement layer* (M units) with activity variables m_μ^t; superscripts denote time steps (Fig. 5). The input layer is fully connected to the map layer via the input weights ρ_{ij}^t. For each map unit i, the weight vector $\mathbf{r}_i := (\rho_{i1}, ..., \rho_{iJ})$ is called its "receptive field". As a consequence of the Cauchy–Schwarz inequality, the map unit will be activated most strongly by an input vector parallel to its receptive field.[6] Within the map layer, synaptic weights α_{ik}^t are considered which are initialized to zero but may evolve during learning. The units of the movement layer develop modulatory synapses to the map layer connections. These synapses are later used to facilitate specific connections.

---

[6]The Cauchy-Schwarz inequality, $|\sum_{j=1}^J \rho_{ij} f_j| \le \sqrt{\sum_{j=1}^J \rho_{ij}^2} \sqrt{\sum j = 1^J f_j^2}$ (e.g., Friedberg, Insel & Spence 1979), states that the covariance between two vectors (left side) is less than or equal to the product of the square roots of the variances of the involved vectors (right side), Equality of both sides (i.e. the largest possible value of covariance) occurs if $(\rho_{i1}, \dots, \rho_{iJ})$ and $(f_1, \dots f_J)$ are parallel (i.e. if they differ at most by a constant factor). If input vectors and receptive fields are normalized, map units will thus be activated most strongly by input vectors equalling their receptive fields.

Modulatory movement connections. McNaughton (1989) suggested that a neural "compound place/movement representation" could serve to implement a structure like the view graph. If we consider M different movements, this would amount to multiplying the number of vertices of the view graph by M, and coding the movements by the choice between M labelled vertices that each edge can lead to (cf. Fig. 6b). Adopting this strategy for the network should lead to similar results as we will presently describe — it would simply require the inclusion of efference copies of motor signals and some minor changes in the sequence learning.

We chose a different implementation: Instead of tripling the number of vertices, we introduce three movement units, standing for left, right, and back. During the learning, they develop links which have influence on specific map layer connections representing the view graph edges, rather than on map layer units. Mechanisms like that are known to exist in the brain; among them are presynaptic inhibition and facilitation, and inhibition at single spines of pyramidal cells (see White, 1989). Alternatively, one could think of the map layer units as being composed of groups of cells, with different axons leading to different other groups. In this case, the specificity required can be accomplished by postsynaptic connections. In any case, we will refer to these specific links as *modulatory* connections.

For reasons of modelling simplicity, we chose the connections to be of a facilitating type (cf. Fig. 6c): activation of a movement unit facilitates all synapses belonging to a certain movement type. it should however be noted that the mechanism can equally well be implemented with inhibitory connections (cf. Fig. 6d).

3.2.2   Input sequences

The network receives two input sequences, the movement decisions μ^t ($\mu^t \in \mathcal{M} := \{go\ left,\ go\ right, go\ back\}$) and the input views, modeled as vectors[7] $\mathbf{f}^t = (f_1^t, ..., f_J^t)$, $t \in \mathbf{N}$. The sequence of movement decisions is represented by a sequence of activity patterns $(m_1^t, ..., m_M^t)$ in the movement layer such that at each time step the unit corresponding to the currently performed movement is active; one entry m_μ^t is 1 and the other ones are 0. In the simulations, both sequences are generated from a random walk on the view graph of a maze. All edges leading off a certain vertex are traversed with equal probability (i.e. all available movements are chosen with equal probability). In our simulations, we constructed a fixed J-dimensional vector for each view of the maze in the following way: We assigned random values between 0 and 1 to its entries and then normalized the vector (in the view of footnote 6, longer vectors would otherwise systematically lead to higher map layer unit activities then would shorter vectors). This vector is fed to the input layer of the network each time the random walk passes by.

3.2.3   Activation dynamics

The activity of the map layer units is given by

$$e_i^t = g\left(-\theta_i^{t-1} + \sum_{j=1}^{J} \rho_{ij}^t f_j^t + \sum_{k=1}^{I} \alpha_{ik}^t e_k^{t-1}\right), \tag{3}$$

where $g : \mathbf{R} \to [0,1]$ denotes the logistic function. Let $w(t)$ denote the index of the most active unit (the winner unit) at time t, i.e.: $e_{w(t)}^t = \max_i\{e_i^t\}$. It represents the view currently perceived in the maze. The effects of incoming information are biased by an intrinsic term (the second sum in Eq. 3) such that the current winner is likely to be a unit connected to the last winner (i.e., the one with the highest value of e_k^{t-1}) via a strong weight α_{ik}^t. As we shall

---

[7]These vectors can be thought of as the result of preprocessing applied to the actual (spatiotemporal) visual input.

see later, the learning dynamics are such that knowledge that two views can be perceived in sequence will be represented by nonzero weights α_{ik}^t.

The modulatory synapses can be used to facilitate connections that are associated with particular movements μ by replacing α in Eq. 3 by $\alpha_{\text{fac.}}$:

$$(\alpha_{ik}^t)_{\text{fac.}} = \alpha_{ik}^t + (1 - \alpha_{ik}^t)\phi \sum_{\mu \in \mathcal{M}} \beta_{ik,\mu}^t m_\mu^t. \tag{4}$$

Here ϕ is a facilitation factor between 0 and 1 and the $\beta_{ik,\mu}^t$ is the facilitation weight of movement unit μ to the map layer connection α_{ik}. The simulations in section 4.2 will show that using $\alpha_{\text{fac.}}$ (with $\phi > 0$) instead of α improves the view-recognition performance of the cognitive map.

3.2.4   Learning dynamics

Receptive fields. The input weights ρ_{ij} are randomly initialized to values between 0 and 1, with a subsequent Euclidean normalization of each receptive field. The receptive field of the winner unit approaches the presented input via the competitive learning rule with learning rate λ_1 (see Kohonen 1982)

$$\mathbf{r}_{w(t)}^{t+1} = \frac{\mathbf{r}_{w(t)}^t + \lambda_1 \mathbf{f}^t}{\|\mathbf{r}_{w(t)}^t + \lambda_1 \mathbf{f}^t\|}. \tag{5}$$

The input weights of the other units remain unchanged.

Intrinsic map layer weights. Connections within the map layer are established between the last two winners, representing a transition between the last two views in the maze:

$$\alpha_{w(t),w(t-1)}^t = (1 - \lambda_2)\alpha_{w(t),w(t-1)}^{t-1} + \lambda_2 \alpha_{max} \tag{6}$$

Here, λ_2 is a learning rate and α_{max} an upper limit for the weights.

Map layer thresholds. Due to the intrinsic term in the activation dynamics, the network could converge to a state where just two strongly connected map units are the winner units in all view presentations. In order to overcome this problem, the thresholds of winner units increase according to

$$\theta_{w(t)}^t = (1 - \lambda_3)\theta_{w(t)}^{t-1} + \lambda_3 \theta_{max}, \tag{7}$$

with λ_3 being the learning rate and θ_{max} being the maximal threshold. Initially, all thresholds are set to the same value θ_{init}.

Movement layer modulatory weights. The modulatory weights (Eq. 4) are adjusted as follows: let $\beta_{ik,\mu}$ be the weight of the modulatory synapse from the movement unit number μ ($\mu \in \mathcal{M}$) to the map layer connection α_{ik}. Then the rule reads

$$\beta_{ik,\mu}^t = \begin{cases} 1 & \text{if} \quad \mu \text{ was the movement associated} \\ & \quad\;\; \text{with the last increase of } \alpha_{ik} \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

Using this rule, the sum in Equation 4 in effect consists of a most one nonzero term. Given a movement μ^t that has been executed, of all movement layer activities only m_μ^t will be nonzero; the weight α_{ik}^t will be facilitated only if $\beta_{ik,\mu}^t$ is nonzero, too (i.e., if the corresponding modulatory connection exists).

3.3   Path planning with movement units

Once the cognitive map has been learned (e.g., using the random exploration described in the previous section), it can be used to plan paths to arbitrary goals. We implemented path planning in close analogy to the algorithm described in Sect. 2.7. Note, however, that the combinatorial explosion that the algorithm presented in Sect. 2.7 seems to imply for the path

planning problem does not take place in the network model: the latter in principle executes the time–consuming computations in parallel (albeit we simulated it on sequential hardware).

To find a path joining two arbitrary points of the map, a procedure is carried out which one could call *mental exploration*: first, the movement units facilitate certain map layer connections in order to infer the neighbours reachable via certain movements. Then, activity is spread within the map layer to determine the combinatorial distance to the goal. This way, it is tested for all neighbours how close they are to the goal, and it is decided to go to the closest one. The comparison of expectations evoked by alternatives can also be found in animals: Tolman refers to it as "vicarious trial–and–error–behavior" (1932, quoted from Schmajuk and Thieme, 1992, who also use a path-planning mechanism inspired by this idea).

In more detail, the algorithm is the following:

1. Initially, the goal view g is fed on the input layer to find out which unit represents it.

2. For all movement types $\mu \in \mathcal{M}$, we do the following:

   - feed the start view s on the input layer, to activate the map layer. Set the activity of the winner unit to 1.

   - set the activity of movement unit μ to 1, with the effect that certain connections get facilitated (see Eq. 4)

   - let the activity spread within the map layer for one time step, and determine the winner unit (which ought to be the unit representing the view reached by μ)

   - if the winner activity is below a certain threshold ν, it is concluded that μ is impossible from s,
     otherwise all other unit activities are set to 0, and we measure the time it takes for the activity to spread to the goal unit

   - reset all movement unit activities to 0

3. Pick the μ for which the above time was minimal (in case of ambiguities, choose one of the respective μs at random).

4. Execute the movement, and take the reached vertex as the new starting point s.

5. If the new s is not equal to g, go back to the beginning of the algorithm (with s as the new starting point).

Note that the path planning algorithm requires another neural system which makes use of the map contained in the network. The same seems to hold for rat navigation: as mentioned in Sect. 1.2, relocating the goal does not change the hippocampal representation of the environment. This suggests that to navigate to a goal, another neural system must interact with the hippocampal map (cf. O'Keefe, 1991). This system is not explicitly modelled here. The operations which it has to carry out in our model (as stated above), however, are not altogether unrealistic. The movement units can be thought of as representing movements at a higher level, not distinguishing between different ways of locomotion.[8]

4   Simulations

4.1   Learning the topology

In this section, we will show that the sequence–learning rule of Eq. 6 can be used to recover the view graph topology from the sequence of views encountered during a random walk through the

---

[8]MacFarlane (1930) found that rats could transfer spatial knowledge acquired in swimming through a maze to running tasks in the same maze.
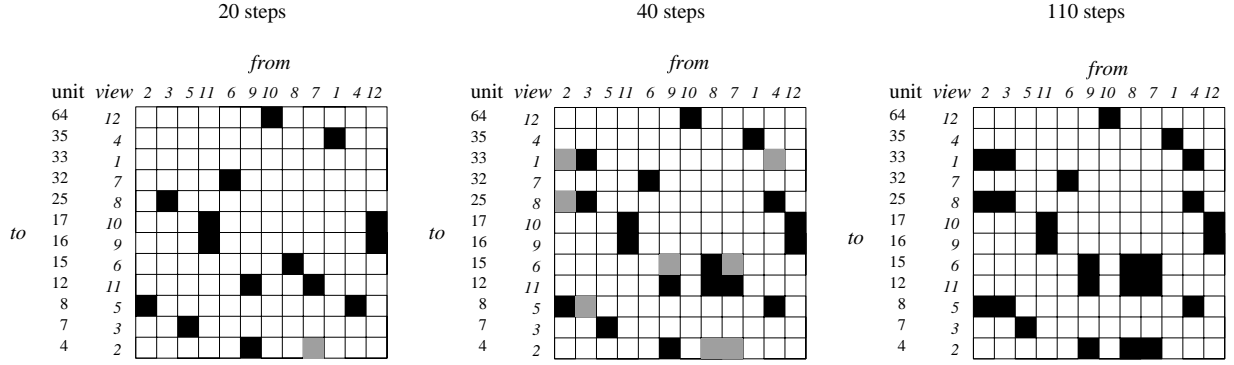
Figure 7: Intrinsic weights α_{ik} of the network (i.e. adjacency matrix of the learnt view graph) after 20, 40 and 110 learning steps. Identical columns in the matrices indicate views belonging to the same places. Since the theory predicts that non–orthogonal columns must be equal, the weight matrix after 40 steps already contains as much information as the one after 110 steps. Black squares represent positive entries, grey ones signify weights that are predicted by the theory but have not been learnt yet. Columns and rows with no positive entries have been left out in the display.

maze. The simulations consisted of learning phases and testing phases, during which weight learning and threshold increase had been disabled. In practise this need not be the case — the map can be built and read out at the same time. In the testing phases, it is tested whether the winner units in the map layer correctly represent the currently perceived views.

As a simple example consider the view graph shown in Fig. 3. To learn this graph, we used a map layer with $I = 64$ units and an input layer consisting of $J = 20$ units (the movement layer was not needed in this simulation). With parameter values $\lambda_1 = \lambda_2 = 0.26, \lambda_3 = 0.02, \alpha_{max} = 0.2, \theta_{init} = 2.5, \theta_{max} = 2.9$, and $\phi = 0$, we found that after 110 random learning steps the view graph was completely learnt in the sense that the map layer connectivity graph (of the nonzero weights) was isomorphic to the view graph (Fig. 7). The map layer weights α_{ij} depend on the frequency with which the corresponding edge of the view graph has been traversed during learning.

Convergence of the network and the goodness of the estimated view graph topology can be judged by the following two conditions:

1. The combinatorial distance (on the map layer connectivity graph) between winners in subsequent time steps should be 1.

2. The number of connections in the map layer should not exceed the number of edges in the view graph.

We call the frequency of cases satisfying this condition (in a test run of 200 steps) the *neighbourhood preservation rate* (NPR). As an NPR of 1 could be trivially achieved by simply fully connecting the map layer, this criterion alone does not suffice—hence, condition 2. As these two conditions would not generally exclude the case of a map of few units, each specialized to more than one view, it is moreover worth noticing that the network does converge to a state where there is a one–to–one correspondence between different winner units and views.

Table 1 shows the simulation results. After 110 time steps, the network has converged to the correct view graph. Sect. 5 will show that the maze topology can already be inferred from the weight matrix after 40 learning steps.

To find out how the network performance scales to larger mazes, we tested view graph learning in a maze with 24 views (amounting to twice as many as in the maze used above). From the results, the following can be said:

Table 1: View graph topology convergence in the simulation. NPR: Neighbourhood preservation rate (cf. Sect. 4.1).

Learning time	0	10	20	30	50	70	90	110	130	ideal
NPR (in %)	0	40	60	73	77	83	87	100	100	100
No. of connections	0	9	14	17	19	22	24	26	26	26

- Due to the random–walk exploration, learning times grew stronger than proportionally with the maze size (to about 300 steps). One goal in developing more clever exploration strategies is to make learning times scale not more than proportionally with the maze size.

- Using the same network dimensions as above, 300 learning steps yielded a neighbourhood preservation rate (NPR) of 95%.

To show that our network model also works if we allow for turning movements, we used a maze consisting of 6 places on the corners of a hexagon. Into each place, we put 3 views, reachable by turning and/or locomotion, yielding 60 view graph edges. With a map layer of 121 units, a set of movement units coding for *turn left*, *turn right*, *go left* and *go right*, and, correspondingly, a modified exploration random walk, we found that our network was able to learn this view graph perfectly within 300 learning steps (learning parameters as above). In this case, places can also be reconstructed from the fact that they correspond to maximal undirected subgraphs of the view graph. If the exploration has not exhausted all possible paths of the maze, the complete view graph is not known yet. However, we can use the fact that in this case any two bidirectionally connected views must belong to the same place.

4.2 Expectation effects

4.2.1 Enhancement of view recognition through the use of topological and movement information

After learning, the views and the structure of the view graph (topology and movement labels) are jointly encoded in the weights of the network. The map and movement layer weights reflect the graph structure, and the receptive fields of the map layer units are tuned to the views. This structured encoding of views enables the network to generate *expectations* about which views are likely to be seen next, given the previous view (and possibly also the movement carried out). To show how this can improve classification in unclear cases, we conducted the following experiment: First, the network went through the learning procedure as described in Sect. 4.1. Various amounts of random noise were then added to the input view vectors and the relative contribution of the various network features to its performance was measured as the NPR (Fig. 8):

1. In the first case (+ in Fig. 8), we switched off all topological feed–back by resetting the map layer activation to zero at the beginning of each step (i.e. by dropping the intrinsic term in Eq. 3). In this case, only the recognition weights ρ_{ij} can account for the performance.

2. If map layer weights α are used but the movement facilitation is switched off (i.e. $\phi = 0$ in Eq. 4), an improvement of about 3.6 dB in required signal–to–noise ratio as compared to case 1 is found (Δ in Fig. 8). This is due to the bias produced by the intrinsic map layer weights towards the perception of views which are known to be adjacent to the previous view (*topological biasing*).
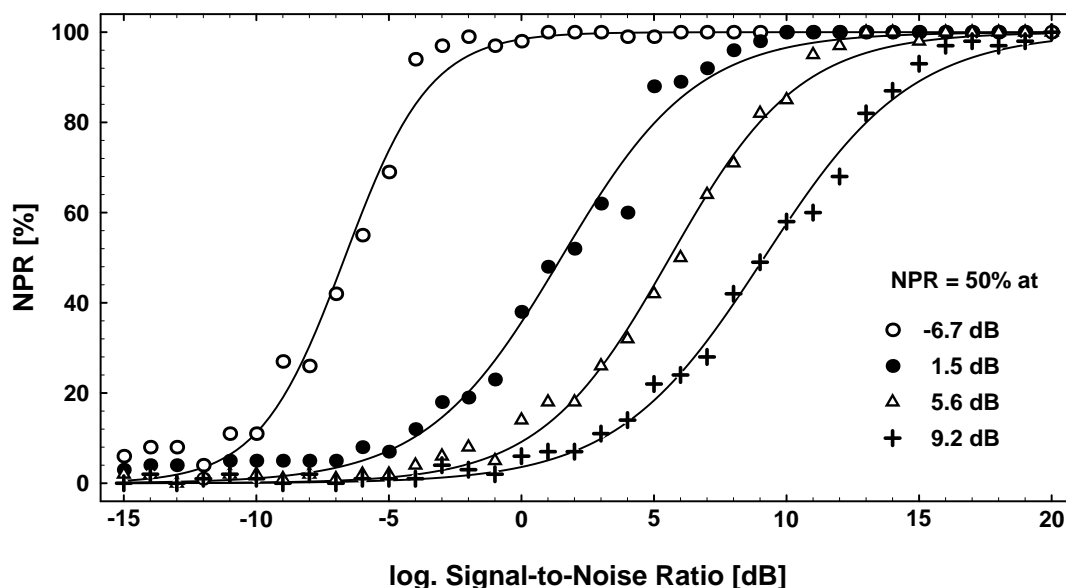
Figure 8: Neighbourhood preservation rates (NPR) measured in 200 testing steps for different amounts of Gaussian noise added to the input views. Network dimensions and maze as above, learning time 110 steps. +: intrinsic connections cut. Δ: intrinsic connections without movement facilitation (topological biasing). •: intrinsic connections with movement facilitation. ○: additionally, the winner unit activity is set to one in each time step. The curves are logistic functions fitted to the data. The relative shift of the curves, i.e. the improvements achieved by adding the different features are given by the differences of the 50%–thresholds shown to the right.

3. In the previous case, topological expectations biased a small number of units. Movement facilitation additionally biases a set of units which are known to be reachable by a certain movement decision (from somewhere). The intersection of the two biased sets of units only contains *one* unit, which then is very likely to become winner even if we add considerable noise. If we include movement facilitation by setting $\phi = 0.5$ in Eq. 4 (• in Fig. 8), the NPR curve is shifted by another 4.1 dB compared to case 2 (*movement expectations*).

4. The network performance can be further improved by setting the winner unit activity to 1 at each time step, thereby increasing the bias effect (○ in Fig. 8). We find an additional shift of 8.2 dB.

4.2.2 Orientation in the dark

Any position on the view graph can be fully determined by the movement decisions that have been taken from a defined initial position. In fact, rats seem to be able to perform a similar task: McNaughton and Nadel (1990) describe an experiment where place cell firing in the dark was investigated, in an environment previously explored. In one condition, the light was switched off before the rats were placed in the environment. Then place cell firing specificity was found to be disrupted. In another condition, the light was switched off after the rats were placed in the environment, giving them a chance to determine their position using vision. In this case, place cell firing specificity was maintained, suggesting that the rats used their spatial knowledge and information about egomotion to update the place cell firing pattern, starting from an initial calibration.

Because our network model produces topological and movement expectations, it can qualitatively reproduce this effect. To show this, we proceeded as follows:

1. The network went through the learning procedure (see Sect. 4.1 and 3.2.1)

2. We erased all views to be seen in the maze

3. For 200 random movement steps (starting from the position where the learning had stopped), the map layer connections corresponding to the respective movement were facilitated according to Eq. 4 (with $\phi = 0.2$), activity spreaded (generating topological biasing), the new winner unit was determined, and its activation was set to 1.

The result was that *without* any visual input, the same units were most active that had been most active in the case where the views could be seen. In control experiments (designed to account for the case where the light had been switched off before the rat was placed in the environment), we changed the position after learning, with a random resetting of map layer activity. Not surprisingly, what we found in this case was that spatial specificity of winner units was completely lost. However, the network always came up with a winner unit sequence which was at least internally consistent: winner units of subsequent time steps where always connected (i.e. the NPR was 100%).

4.3 Finding paths between arbitrary views

Using the path planning mechanism of Sect. 3.3 and the 12–view maze, we tested the network's path finding performance for all possible combinations of start and goal views. With a facilitation factor $\phi = 0.2$ and an activity threshold $\nu = 0.09$, the network (trained for 110 steps, as above) always found the shortest path (nothing can be said about metrical distances as our model does not include metrical information). This also proves that apart from the topology (Sect. 4.1), the network has learnt the movement labels of the view graph. If the network is trained for only 40 steps, it does nevertheless always find a way; in more than half of the cases it is the shortest way.

It is worth pointing out that the network finds paths that it has never completely travelled along during learning. However, the paths always consist of subpaths which have been seen during learning.[9] Paths which include travelling along corridors which are genuinely new cannot be planned in mazes, since it is not possible to have the necessary knowledge about unexplored parts of the maze.

Testing the same network on a 24–view maze maze, we found that path planning was impaired: only in 76% of the combinatorially possible cases did the network find a path; an optimal path was found in 27% of the cases. There are two ways how to solve this problem (Fig. 9):

- Using easily distinguishable views: instead of the 24 random views, we used the canonical basis vectors of $\mathbf{R}^{24}$ (in this case, the size of the input layer had to be increased to 24, too). 400 learning steps then resulted in an NPR of 100% and perfect path planning performance.

- Substantially increasing the network dimensions: with a 256 unit map layer, we achieved a NPR of 98%, and in 90% of all possible cases an optimal path was found.

The first point indicates that the network has no problems with the increase in complexity of the topological structure of mazes. Therefore the problem must lie in the view classification. Indeed, the second point shows that an enlarged network, having a denser distribution of receptive fields, can cope with it.

---

[9]According to Schmajuk and Thieme (1992), integration of separately learned pieces of information is the basis for detour behavior (cf. Tolman, 1932).

% of paths

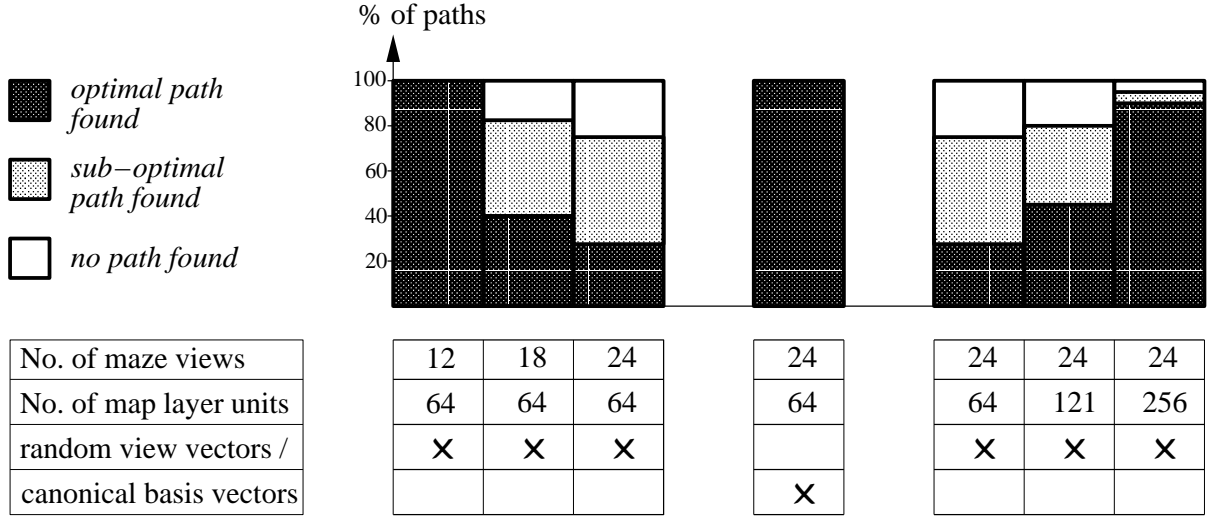| No. of maze views	12	18	24	24	24	24	24
No. of map layer units	64	64	64	64	64	121	256
random view vectors /	✗	✗	✗		✗	✗	✗
canonical basis vectors				✗			

Figure 9: Path planning performance for different mazes (left hand side) and different network sizes (right hand side). In the middle, it can be seen that if we use canonical basis vectors as view vectors (in this case, view classification becomes trivial), even the small network can cope with the bigger maze. *No path found* denotes the case in which no path has been found which is shorter than three times the maximal combinatorial distance between views of the respective maze.

5   Developing exploration strategies

5.1   The completion matrix as an indicator which places to explore

The result that successors are either identical or disjoint (Sect. 2.2) can be used to find out where further exploration might be needed. The successor of view v_p (Eq. 2) corresponds to the pth column of the adjacency matrix A; its size is the number of ones in that column. The fact that views of the same place have identical successors implies that A can have only up to n (the number of places) different columns $\mathbf{a}_i$ $(i = 1, \ldots, n)$. Two columns $\mathbf{a}_i$ are either identical or orthogonal, since successors are either identical or disjoint. Therefore, after permuting the view indices in a way that identical columns stand next to each other, the symmetrical matrix $A^\top A$ will have block structure with n blocks each corresponding to one place of the maze. In

$$(A^\top A)_{ik} = \sum_j (A^\top)_{ij}(A)_{jk} = \sum_j (A)_{ji}(A)_{jk} = (\mathbf{a}_i, \mathbf{a}_k), \tag{9}$$

the scalar product $(\mathbf{a}_i, \mathbf{a}_k)$ can either equal 0 (outside of the blocks) or, if $\mathbf{a}_i = \mathbf{a}_k$, the number of ones in $\mathbf{a}_i$. This number is constant inside each block and equals the out–degree of the corresponding place. The size of the blocks equals the frequency of a particular column (successor set) in a matrix (i.e. the in–degree of the corresponding place).

If A is interpreted as a transition matrix, $A^\top A$ describes a step forward followed by a step backward. Starting with some view, this movement will reach all views leading to the same place. We call $A^\top A$ the *completion matrix* of the view graph (cf. Fig. 10). Note that each block of the completion matrix corresponds to one of the complete bipartite subgraphs discussed in Section 2.3.

After incomplete exploration, A does still have some incorrect zero entries. However, this incompleteness is easily visible from the completion matrix: from the weight matrix after t learning steps, we derive an estimate of the completion matrix A of the view graph in four steps. First, note that only the rows and columns corresponding to one of the winner units will contain entries different from zero. We delete the all–zero rows and columns. In the example of Fig. 7, we obtain an 12×12 matrix since one winner unit has evolved for each view. Second, we
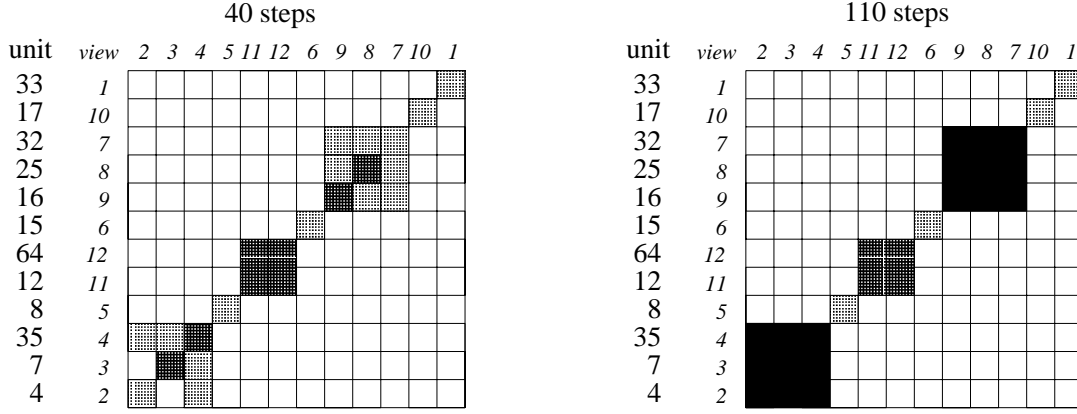
Figure 10: Completion matrices $\tilde{A}^\top \tilde{A}$ derived from the weight matrices after 40 and 110 learning steps (cf. Fig. 7). Grey–levels correspond to the values 0 (white) to 3 (black), unit numbers denote units in the map layer. The block structure is clearly visible after 40 steps, the underlying maze can be inferred already from this stage.

apply the Heaviside–function to the remaining weights such that all entries different from zero are set to one. Third, we permute the columns (and the corresponding rows) in such a way that non–orthogonal ones stand next to each other. The resulting matrix is denoted by $\tilde{A}^t$. Fourth, we compute the completion matrix $(\tilde{A}^t)^\top \tilde{A}^t$. The estimated completion matrices $(\tilde{A}^{40})^\top \tilde{A}^{40}$ and $(\tilde{A}^{110})^\top \tilde{A}^{110}$ are shown in Fig. 10. The block structure is already fully developed after 40 learning steps. However, the inhomogeneities within the blocks indicate the need for further exploration, as discussed in the next section.

5.2 Deriving hypotheses from the incomplete view graph

An exploration scheme that can specifically look for left-out views uses the fact that (complete) successor sets are either disjoint or identical. As an example, consider the weight matrices presented in Fig. 7. The exploration is complete, if all views and all possible transitions between views have been encountered: in our case, after 110 steps. After 40 steps, the transition $v_7 \rightarrow v_6$ has not yet been seen. From the incompleteness in the block containing v_7, v_8, v_9 (see Fig. 10), however, the explorer can see that something is wrong with view v_7. Does it belong to the same place as v_8? A look at the weight matrix (Fig. 7, after 40 steps) reveals the problem: both views can be followed by view v_{11}, but only v_8 has been followed by v_6 so far. As a possible exploration strategy, one can explicitly search for the transition $v_7 \rightarrow v_6$.

If the transition really does not exist, v_7 and v_8 cannot belong to the same place. This in turn implies that one of the links $v_8 \rightarrow v_{11}$ and $v_7 \rightarrow v_{11}$ has been learnt errouneously. A possible reason for this could be that there are two slightly different views, one in the successor of v_7 and one in the successor of v_8, whose differences had been neglected in earlier explorations. Therefore one should go there again, to have a closer look. If this does not reveal any difference in the two views, one could distinguish the two views using the internal history information, i.e. their position relative to other recognized views (see Sect. 6.3).

We can take these methods which were derived from constraints on possible view graph topologies one step further and also recover movement information from an incomplete view graph. This information permits the extraction of the complete labelled view graph from the incomplete transition matrix displayed in Fig. 11, similar to the process of extracting the topology of the view graph from the incomplete weight matrix discussed previously.

from

					b						
							b				
▦	r										
			b								
▦	l							r			
			r							b	
			b							l	
					▦	l	▦				
				b	r	l					
r	▦							l			
		b									
				l	▦						

from

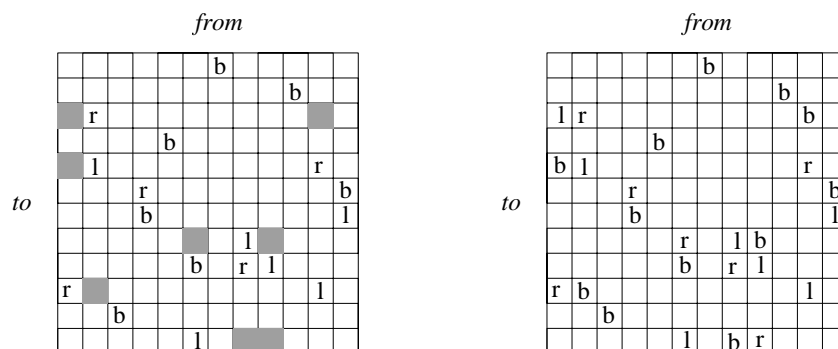|					b						
							b				
l	r							b			
			b								
b	l							r			
		r								b	
		b								l	
					r		l	b			
					b		r	l			
r	b							l			
		b									
					l		b	r			

Figure 11: Left: incomplete transition matrix, corresponding to the graph learnt in 40 steps, cf. Fig. 7. Grey boxes represent entries that must be non–empty for topological reasons, as argued in Fig. 7. The fact that rows and columns of the transition matrix may only contain each label once (Sect. 2.6) allows us to substitute the grey boxes by unique movements. Thus we get the transition matrix of the complete view graph (right).

6 Discussion

6.1 Characterization of our approach

6.1.1 Representation of spatial knowledge

What do cognitive maps look like? We have tried to presuppose as little as possible about the answer to this question — the basic building blocks of our model are simple *sequences of views and movement decisions*. Nevertheless, we were able to show that our simple map already suffices for path planning. This way we have demonstrated that it is not necessary to build up something which resembles the explored environment itself. Our theoretical reconstruction of the maze from the view graph has shown that the view graph carries sufficient information about the maze. However, the neural network does not have to reconstruct the maze, it just has to support appropriate behaviour in the perception–action–cycle, shown in Fig. 12. Compared to Fig. 1, we have now displayed the interior structure of our navigation system, including mechanisms for expectations and mental exploration.

Perception is a process which uses sensory input and internal states to produce new internal states — it does *not* establish any simple relationship to the world. Accordingly, when we used the term *representation* in this paper, we did not refer to a direct correspondence between the world and states within the system.[10] We use it in a more pragmatic sense: by *representations* we denote internal structures which allow us to *act* successfully. "If an organism's behavior can be shown empirically to be affected by past experience, then some representation of that experience must exist. Simply positing the existence of a representation, however, explains nothing; the nature of that particular representation, rather than the fact of its existence, is what provides a kernel of an explanation, and is the proper target for dispute." (Roitblat 1982)

6.1.2 Why use a maze?

The reasons why we chose a maze for the study of cognitive mapping are the following simplifications: First, mazes provide a canonically given set of landmarks which need to be remembered in order to support map behaviour: the views of junctions. Second, there is a finite set of discrete movement possibilities (in our case, *left, right, back*). This way, we were able to focus on the topological and directional structure of the map. In more complex environments, providing

---

[10]In navigation, a representation of the latter kind could be a set of 3–D objects which are placed at certain positions in some sort of internal euclidean space, with a perceptual means of determining one's own position within this space.
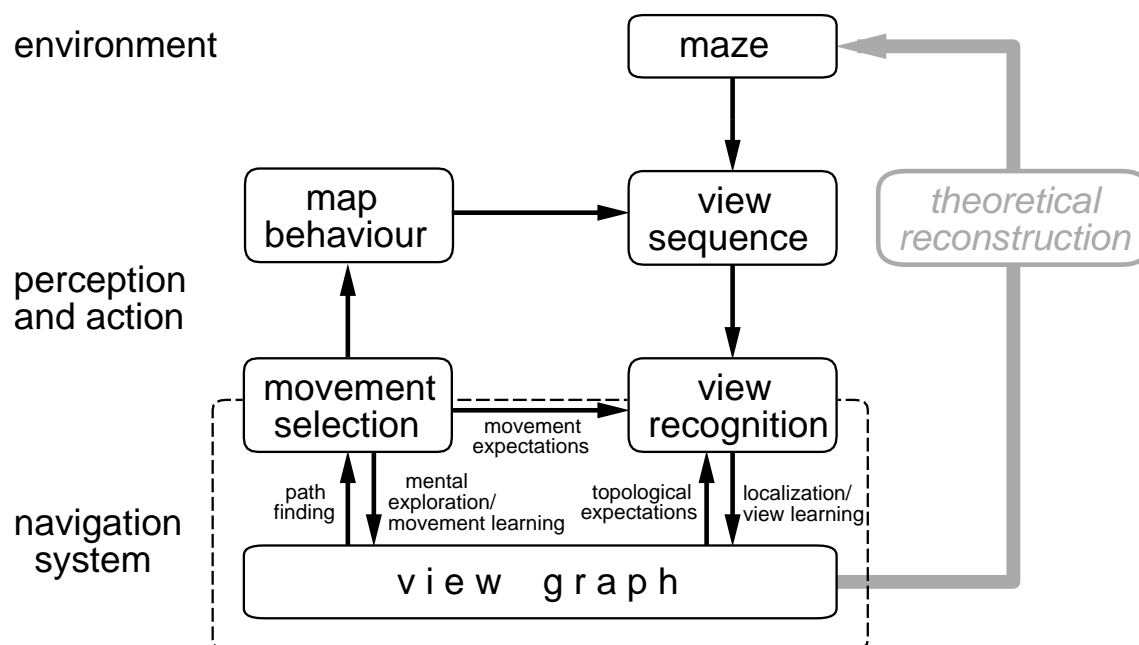
Figure 12: Overview of the approach presented in this paper. Notice the feedback to the view recognition through expectations. The grey arrow represents the maze reconstruction from the view graph (sections 2.2 and 2.6). This reconstruction is theoretically possible, but it needs not be carried out by the explorer. If we switch off the light, cutting the downward input to the view recognition, *expectations* alone are sufficient to update the map layer activity correctly (see Sect. 4.2.2).

richer sensory information than just a sequence of views, more sophisticated maps (including metrical information, for example) could be constructed.

6.1.3 Comparison to other models

The dichotomy between places and views is central to navigation (e.g., Gallistel, 1990; Schmajuk & Thieme, 1992). To our knowledge, the present model proposes a novel way of establishing a relationship between places and (direction-dependent) views by means of the successor set. To this end, it was necessary to make some simplifying assumption, most notably that we only study cognitive maps of mazes. Our studies support the view that for the navigation in a maze, the learning of views and their spatial relations (including movements linking them) is sufficient. Places can be reconstructed — all the necessary information exists in the view graph — but it is not necessary to do so.

Unlike many other models, where the goal location is already known in the learning phase, our model is not restricted concerning the paths it can find after learning. Our path planning mechanism uses all paths that have been seen during exploration, and novel combinations of them, to find paths between arbitrary points of the mapped environment.

Like some of the models in the literature (Hetherington and Shapiro 1993, Touretzky, Wan and Redish 1994,) our model uses internal information to produce expectations. This aspect deserves further discussion and therefore it will be dealt with it in the next section.

In our model, the purpose of the movement units is not simply to obey commands mechanically. They rather carry spontaneity themselves: they trigger mental exploration, and they facilitate certain map layer connections, generating movement expectations.

6.2   Behavioural data

6.2.1   Expectations

In Sect. 4.2.1 shown that the view graph actually codes something more complex than just views — it rather encodes a combination of perceived and (from topology or movement) expected views, thereby increasing map quality (in terms of neighbourhood preservation). As mentioned in Sect. 1.2, hippocampal place cells seem to have similar properties: their place fields persist when cues are removed. This could be explained as a pattern completion of an associative memory. However, the rat's ability to maintain internal position knowledge without any visual input requires the use of knowledge about the map (as explained in Sect. 4.2.2). In Sect. 4.2.2, we showed how expectation mechanisms can help understanding these effects. The feedback of internal information into the processing of external input (pattern/landmark recognition) is an interesting example of how different sources of information can interact in order to stabilize the information processing. Note, moreover, that for this integration and allothetic information, it was crucial to have some kind of synchronization of two time scales: the one in which the movement takes place, and the neural network simulation time scale. This could be viewed as a crude example of the synchronization necessary for a path integration system to work properly.

Map quality could be enhanced further by exploiting information between the map and motor levels. For instance, directly returning to a place from a neighboring place provides an additional view of the original place.[11]

6.2.2   Free exploration

As we have pointed out in Sect. 1.2, the hippocampal representation of an environment is invariant under relocation of the goal. This is mirrored in our model as the fact that during exploration, the map is learnt in a way which is not tailored to specific future goal locations. In this sense, the network performs a free exploration.

6.2.3   Predictions concerning passive exploration

Held (1965) has shown that humans must actively interact with the environment to develop sensory–motor coordination. If we simulate passive exploration in our network by not learning the modulatory connections from the movement units, i.e. by using the views but not the movement inputs, we get the following:

- sequential relations about views are still learnt

- path planning becomes practically impossible, since it is no longer learnt which movements are necessary to produce specific view sequences

- movement expectation effects vanish, with a degradation in view recognition

In ongoing work carried out at our laboratory (Dartsch and Mallot, 1995) these predictions are tested in psychophysical experiments using a maze simulated by interactive computer graphics.

6.3   Disambiguating identical views at different locations

In our scheme, the views associated with different vertices of the view graph were assumed to be different.[12] If we want to relax this assumption and still want the view graph to contain

---

[11]This is related to the ability to compute "movement equivalences", as discussed by McNaughton (1989): one parietal cortex neuron in a rat selectively responded to left turns; it did, however, also respond to right turns exceeding $180°$.

[12]In the simulations, this difference came up naturally by using randomly generated view vectors: given that they were lying on the sphere in 20–dimensional space, they are most likely to be different.

sufficient information for both path planning and theoretical place graph reconstruction, we have to find a way to disambiguate identical views at different locations.

Expectations generated by information about either topology or movements can be used to this end: a simple example of the first kind is that an explorer knows that walking one step takes one to a location different from the one where one started. This way, two neighbouring identical views can be distinguished. From then on, they will get connected to different other views, which means that they will be embedded into different contexts. An example which uses both kinds of expectations is the following: making some prior assumptions about the maze layout, e.g. that the corridors are reasonably straight, enables one to conclude that a movement sequence *go left, go right* cannot take one back to the start location, and therefore even if the views are identical they should be distinguished.

Related ideas have been put forward by Tani and Fukumura (1994). They use history information in constructing a mapping which assigns unique movement commands to sequences of sensorial input rather than to the sensorial input at a single point of time.

6.4   Relationship to object recognition

Unlike topographical or topological place maps of the environment, our approach uses views as primitives. This bears some resemblance of *view–based object recognition*: in 3–D object recognition, one can either favour the approach that we keep 3–D models of objects in our brains, which we match to perceived views in recognition tasks by mental rotation, or a view–based approach, which interpolates between a number of stored 2–D views to match the perceived one (for a review, see Bülthoff and Edelman, 1992). Similar to our view–based model, Edelman and Weinshall (1991) have proposed a scheme connecting different views of a 3D object in a graph.

Mental rotation is usually considered to support the 3–D model approach. The present work, however, could help explaining mental rotation without explicit 3–D models: if we build a cognitive map consisting of different views of an object and label the view graph connections not by movements through a maze but by object rotations,[13] we could then use the map to produce rotated views of the objects by generating movement expectations and topological expectations (see Sect. 4.2.1).

7   Conclusion

We have argued that the *view graph* is a concept which is both more natural and more powerful with respect to its theoretical content than the place graph of a maze: First, exploration of a maze produces a sequence of views, and not of places. Second, the view graph contains all the information of the place graph. Third, two views can be connected by a unique movement decision, whereas for two places, this does not work (this allowed adding movement information for path planning).

We have proposed an artificial neural network which in simulations was able to learn the view graph from a sequence of views and movements generated by a random walk through an unknown maze.

To show that the view graph does not only contain all the information necessary to reconstruct the maze structure but that it is also possible to make use of this information, we have presented a simple path planning mechanism which enabled the network reliably to find optimal paths between arbitrary points of a previously explored maze.

The model produces a number of results which resemble findings of behavioural neuroscience, namely directional specificity of place cells, use of stored map knowledge to update the

---

[13]Alternatively, we could label them by active movements of a perceiver around the object, with the perceiver permanently looking at the object.

place cell firing pattern, and invariance of the map under goal relocations.

Ongoing work includes the implementation of the neural network algorithm on a mobile miniature robot (Schlkopf, Georg & Mallot 1995). In this case, the view vectors are constructed from the time course of two infrared sensor readings recorded while traversing a corridor (cf. footnote 4). Moreover, we plan to extend the framework presented here to more general navigation problems: in an open environment, a landmark extraction procedure needs to be carried out first, to define a set of views to be remembered (cf. the "snapshots" of Cartwright and Collett 1987). The learning of topological relations between these can then be accomodated in our scheme as in the maze case.

Appendix A

Reconstructing the place graph from the view graph topology (Sect. 2.2)

The following statements are equivalent:

$$\mathcal{S}(v_p) = \mathcal{S}(v_q) \tag{10}$$

$$\mathcal{S}(v_p) \cap \mathcal{S}(v_q) \neq \emptyset. \tag{11}$$

$$P_{in}(v_p) = P_{in}(v_q) \tag{12}$$

To prove this equivalence, we show that

$$(10) \Longrightarrow (11) \Longrightarrow (12) \Longrightarrow (10).$$

First we substitute from Eqs. 2, 1 and obtain

$$\mathcal{S}(v_p) = \mathcal{S}(v_q)$$
$$\Longleftrightarrow \quad \{v_r | a_{rp} = 1\} = \{v_s | a_{sq} = 1\}$$
$$\Longleftrightarrow \quad \{v_r | P_{in}(v_p) = P_{out}(v_r)\} = \{v_s | P_{in}(v_q) = P_{out}(v_s)\}.$$

Using the one–to–one correspondence between views and directed corridors, this is equivalent to

$$\{c_r | P_{in}(v_p) = P_{out}(v_r)\} = \{c_s | P_{in}(v_q) = P_{out}(v_s)\}.$$

Based on our assumption about the maze layout the last two sets cannot be empty, which amounts to saying that the target places of c_p and c_q have at least one common outward corridor, implying Eq. 11. This in turn implies the identity of the two places, Eq. 12. On the other hand, the latter implies immediately equality of the successors, Eq. 10, completing the proof.

Although we cannot recover the functions P_{in} and P_{out} explicitly, the equivalence of Eqs. 10–12 shows that the equivalence relation defined by the equality of successors partitions the set of all views into n subsets corresponding to the places $\{p_i | i \leq 1 \leq n\}$. These are the vertices of the recovered graph. The corridors can be found by inspection of the view graph.

Appendix B

Reconstruction of the directional structure of the maze

For this reconstruction, we only need the view graph edges corresponding to locomotions. Let $\{v_1, \ldots, v_n\}$ be a set of views that have been found to belong to the same place. Then for any v_i ($i = 1, \ldots, n$), we can draw a Y–shaped graph Y_i with the view in the centre and the view
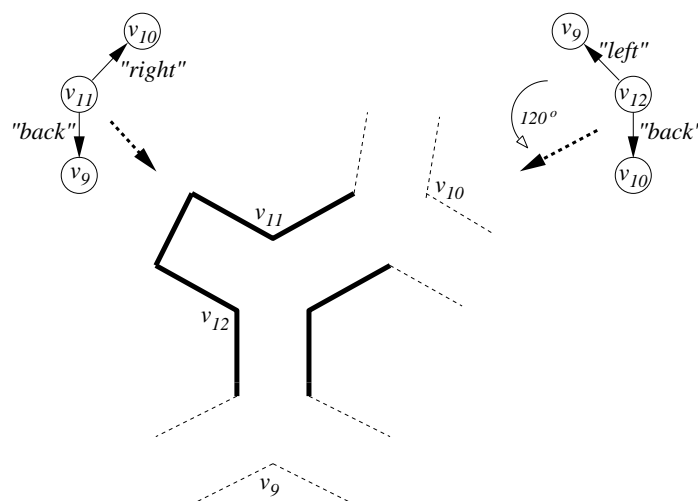
Figure 13: Reconstruction of the directional structure of the maze. Suppose v_{11}, v_{12} are the views belonging to a certain place (in fact they are taken from one place of our place graph above), and the views reached from v_{11}, v_{12} by going left/right/back are known. By displaying them in Y–shaped graphs, containing directional information, the ordering of the views in the maze place can be determined from the angle of the rotation necessary to superimpose identical successor views. At the bottom, the reconstructed maze building block is shown, with inferred dead end and view placement (think of the views as pictures which can be seen if the place is approached from a suitable corridor).

reached by going left, right, or back at the appropriate end vertices of the Y.[14] In Fig. 13, the left graph tells us that the local structure of the maze includes a dead–end left of v_{11}, a corridor leading to v_{10} right of v_{11}, and another one leading to v_9 which can be traversed by going back from v_{11}. By equality of successors we know that v_{12} belongs to the same place as v_{11}. From the fact that v_9 is reached by going left from v_{12} we can infer that v_{12} is attached to the wall left of v_{11}, i.e. that it can be reached from v_{11} by turning left.

We see that this procedure allows the extraction of some allocentric information (in form of an elementary Y–shaped maze building block) from the egocentric view graph labels and the views. If we made some prior assumptions about the regularity of the maze layout, namely that all the turning angles are 60° and all corridors are straight, we could take this even further and reconstruct an allocentric frame of reference including the relative compass directions of all corridors.

Appendix C

A constraint on the labels of a view graph

Neither a transition matrix column nor a row of the matrix contains two identical entries.

Starting from a given view, different locomotion or turning movements take one to different views (by assumption, all views are different — see Sect. 2.2), hence all edges starting at that view will have different labels. This tells us that in a transition matrix with the movements as entries (cf. McNaughton 1989), columns do contain each movement only once. What about the rows of the transition matrix? Can a view be reached from different views by identical movements? If we have reached the view by turning, we can simply reverse the turn to get the unique view where the original turn started from. If we have reached it by locomotion, we

---

[14]This holds for the case that the maze consists of ternary junctions, as in the examples used here. In general, the shape of the graph may have any number of endings, not just 3, as the Y–graph has. The method described here works, as long as the movement labels carry sufficient information to distinguish the exits.

must have come from a neighbouring place, as there is a one–to–one correspondence between views and directed corridors. We assume that the maze junctions are such that between any two views fixed to the wall there is a corridor leading off, and vice versa (this holds for the maze in Fig. 2). Then we can conclude that in order to get from the different views of the neighbouring place to our view, different movements have to be executed.

Acknowledgements

References

[1] A. G. Barto and R. S. Sutton. Landmark learning: an illustration of associative search. *Biological Cybernetics*, 42:1 – 8, 1981.

[2] V. Braitenberg and A. Schüz. Some anatomical comments on the hippocampus. In W. Seifert, editor, *Neurobiology of the hippocampus*. Academic Press, London, 1983.

[3] R. Brooks. Autonomous mobile robots. In W. E. L. Grimson and R. S. Patil, editors, *AI in the 1980s and beyond*. The MIT Press, Cambridge, Ma., 1987.

[4] H. H. Bülthoff and S. Edelman. Psychophysical support for a two–dimensional view interpolation theory of object recognition. *Proceedings of the National Academy of Sciences, USA*, 89:60 – 64, 1992.

[5] N. Burgess, M. Recce, and J. O'Keefe. A model of hippocampal function. *Neural Networks*, 7:1065 – 1081, 1994.

[6] B. A. Cartwright and T. S. Collett. Landmark maps for honeybees. *Biological Cybernetics*, 57:85 – 93, 1987.

[7] K. Cheng and C. R. Gallistel. Testing the geometric power of an animal's spatial representation. In H. L. Roitblat, T. B. Bever, and H. S. Terrace, editors, *Animal Cognition*. Lawrence Erlbaum Associates, Hilldale, NJ, 1984.

[8] S. Dartsch and H. A. Mallot. Psychophysical measurement of distance in cognitive maps: metric vs. topological organization. In N. Elsner and R. Menzel, editors, *Learning and Memory. Proc. 23. Göttingen Neurobiol. Conf. Vol I*, page 47. G. Thieme Verlag, Stuttgart, 1995.

[9] S. Edelman and D. Weinshall. A self–organizing multiple–view representation of 3D objects. *Biological Cybernetics*, 64:209 – 219, 1991.

[10] H. Eichenbaum and N. J. Cohen. Representation in the hippocampus: what do hippocampal neurons code? *Trends in Neurosciences*, 11:244 – 248, 1988.

[11] H. Eichenbaum, S. I. Wiener, M. L. Shapiro, and N. J. Cohen. The organization of spatial coding in the hippocampus: A study of neural ensemble activity. *The Journal of Neuroscience*, 9:2764 – 2775, 1989.

[12] D. Floreano and F. Mondada. Autonomous and self-sufficient: Emergent homing behaviour in a mobile robot. Technical Report R94.141, LAMI, Swiss Federal Institute of Technology, 1994.

[13] S. H. Friedberg, A. J. Insel, and L. E. Spence. *Linear Algebra*. Prentice Hall, Englewood Cliffs, NJ, 1979.

[14] C. R. Gallistel. *The organization of learning*. The MIT Press, Cambridge, MA, USA, 1990.

[15] R. Held. Plasticity in sensory-motor systems. *Scientific American*, 213:84 – 94, 1965. (Cited after E. B. Goldstein (1984), *Sensation and Perception*. Belmont, CA: Wadsworth).

[16] P. A. Hetherington and M. L. Shapiro. A simple network model simulates hippocampal place fields: Ii, computing goal-directed trajectories and memory fields. *Behavioral Neuroscience*, 107:434 – 443, 1993.

[17] T. Kohonen. Self–organized formation of topologically correct feature maps. *Biological Cybernetics*, 43:59 – 69, 1982.

[18] D. A. MacFarlane. The role of kinesthesis in maze learning. *University of California Publications in Psychology*, 4:277 – 305, 1930. (Cited after H. Roitblat (1994), Mechanism and process in animal behavior. In D. Cliff, P. Husbands, J.-A. Meyer and S. W. Wilson (Eds.), *From animals to animates 3*. Cambridge, MA: The MIT Press.).

[19] T. M. Martinetz and K. Schulten. Topology representing networks. *Neural Networks*, 7:507 – 522, 1994.

[20] J. L. McClelland, B. L. McNaughton, and R. C. O'Reilly. Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory. Technical Report PDP.CNS.94.1, Carnegie Mellon University, Pittsburgh, PA, 1994.

[21] B. L. McNaughton. Neuronal mechanisms for spatial computation and information storage. In L. Nadel, L. A. Cooper, P. Culicover, and R. M. Harnish, editors, *Neural Connections, Mental Computation*. MIT Press, London, 1989.

[22] B. L. McNaughton, C. A. Barnes, and J. O'Keefe. The contributions of position, direction, and velocity to single unit activity in the hippocampus of freely moving rats. *Experimental Brain Research*, 52:41 – 49, 1983.

[23] B. L. McNaughton, S. J. Y. Mizumori, C. A. Barnes, B. J. Leonard, M. Marquis, and E. L. Green. Cortical representation of motion during unrestrained spatial navigation in the rat. *Cerebral Cortex*, 4:27 – 39, 1994.

[24] B. L. McNaughton and L. Nadel. Hebb-Marr networks and the neurobiological representation of action in space. In M. A. Gluck and D. E. Rumelhart, editors, *Neuroscience and Connectionist Theory*. Lawrence Erlbaum Associates, Hillsdale NJ, 1990.

[25] R. U. Muller and J. L. Kubie. The firing of hippocampal place cells predicts the future position of freely moving rats. *Journal of Neuroscience*, 9:4101 – 4110, 1989.

[26] U. Nehmzow and B. McGonigle. Achieving rapid adaptation in robots by means of external tuition. In D. Cliff, P. Husbands, J.-A. Meyer, and S. W. Wilson, editors, *From Animals to Animates 3*. MIT Press, Cambridge, MA, 1994.

[27] J. O'Keefe. A review of hippocampal place cells. *Progress in Neurobiology*, 13:419 – 439, 1979.

[28] J. O'Keefe. The hippocampal cognitive map and navigational strategies. In J. Paillard, editor, *Brain and Space*, pages 273 – 295. Oxford University Press, Oxford, 1991.

[29] J. O'Keefe and A. Speakman. Single unit activity in the rat hippocampus during a spatial memory task. *Experimental Brain Research*, 68:1 – 27, 1987.

[30] T. Prescott. Spatial learning and representation in animats. In D. Cliff, P. Husbands, J.-A. Meyer, and S. W. Wilson, editors, *From Animals to Animates 3*. MIT Press, Cambridge, MA, 1994.

[31] H. L. Roitblat. The meaning of representation in animal memory. *Behavioural & Brain Sciences*, 5:353 – 406, 1982.

[32] N. A. Schmajuk and A. D. Thieme. Purposive behaviour and cognitive mapping: a neural network model. *Biological Cybernetics*, 67:165 – 174, 1992.

[33] B. Schölkopf, P. Georg, and H. A. Mallot. Perception and action in view–based maze navigation. *Perception*, 24(Suppl.):95, 1995.

[34] J. Tani and N. Fukumura. Learning goal-directed sensory-based navigation of a mobile robot. *Neural Networks*, 7:553 – 563, 1994.

[35] E. C. Tolman. *Purposive behavior in animals and men*. The Century Co., New York, 1932.

[36] D. S. Touretzky, H. S. Wan, and A. D. Redish. Neural representation of space in rats and robots. In J. M. Zurada, R. J. Marks, and C. J. Robinson, editors, *Computational Intelligence: Imagining Life. Proceedings of the symposium held at the 1994 IEEE World Congress on Computational Intelligence*, Piscataway, NJ, 1994. IEEE Press.

[37] K. Wagner. *Graphentheorie*. Bibliographisches Institut, Mannheim, Wien, Zürich, 1970.

[38] R. Wehner and R. Menzel. Do insects have cognitive maps? *Annual Review of Neuroscience*, 13:403 – 414, 1990.

[39] E. L. White. *Cortical Circuits*. Birkhäuser, Boston, Basel, Berlin, 1989.