Learning Places in Newly Explored Environments

Verena Vanessa Hafner
AI Lab, Dept. of Information Technology, University of Zurich
Winterthurerstr. 190, 8057 Zurich, Switzerland
vhafner@ifi.unizh.ch

Abstract

This paper presents a computational model of cognitive maps for navigation, which is implemented on a mobile robot. The model is based on a self-organising neural network which creates a topological map of the environment as explored by the robot. The map neurons can be seen as 'place cells' which are inspired by the discovery of real place cells in the rat's hippocampus. To include metric information, the topological map is complemented with a physical force model. The performance of the model is tested both in simulation and on the mobile robot 'Samurai'. The robot is equipped with a compass as well as an omnidirectional camera, and continuously produces a low-resolution, orientation-invariant place vector. Using only this place vector and the compass information, a consistent map of a newly explored environment can be produced.

1. Introduction

Robot navigation is a relatively simple task if the environment is known in advance and does not change. However, this assumption is not fulfilled in most realistic applications. Especially when robots are used as a tool to model biological navigation, whether of humans or of animals, the test environment should yield conditions similar to the real world. The simplest way to navigate only includes reactive responses. More sophisticated navigation strategies definitely include learning, and consequently memory. One method of such memory structures are cognitive or mental maps, which act as a representation of the environment that could also be coupled with actions. The term 'cognitive map' has already been introduced by Tolman in 1948 (Tolman, 1948).

Electrophysiological investigations during behavioural experiments revealed some interesting correlations: In 1976, cells were detected in the rat's hippocampus which preferably fired when the rat was in a particular portion of its environment, but were largely independent of its orientation and actual view (O'Keefe, 1976). The area in the environment where one particular cell has the highest activation is called the place field of this cell. Another type of cells were found which preferably fired when the rat's head was turned in a certain direction, regardless of its position in the environment (Taube et al., 1990).

This paper presents a model of learning places which is inspired by those discoveries. The cognitive map that represents places and their relations is modelled by a self-organising neural network. The information content of the cognitive maps is enhanced by applying a physical force model. Both the self-organising neural network model and the physical force model are introduced and explained in detail in section 2. The performance of the model is tested in simulation in section 3.1. Section 3.2 describes the implementation of the model on the mobile robot 'Samurai' in experiments under 'real world' conditions. Finally, the results are discussed in section 4.

2. Models

In this section, the models for the creation of cognitive maps are explained. The type of input is kept general and will be specified in more detail when the model is applied to exploration tours, both in simulation (section 3.1) and in robot experiments (section 3.2).

2.1 Neural Network Model

The neural network that is used to represent the cognitive map is a modified version of the Kohonen self-organising map (Kohonen, 1982). The main difference between this model and the standard Kohonen self-organising map is that no assumption about the spatial position of the neurons in the cognitive map is made, and therefore only the connection weights of the winning neuron are updated. As there has not yet been found any obvious correlation between the spatial arrangement of place cells in the hippocampus and the spatial arrangement of their place fields, this change seems more appropriate. Another important change to the standard Kohonen network is the enforcement of the weights between the current winner neuron and the winner neuron from one time-step before. This action is based on the continuous movement of the agent in its environment. It allows for the creation of a topological map without any association between place cells and spatial coordinates.

The neural network consists of an input layer $\mathbf{f}$ =

$(f_1, ..., f_I)$ with I neurons and an output layer $\mathbf{o} = (o_1, ..., o_J)$ with J neurons, which are fully interconnected by weight vectors $\mathbf{r}_j, (j \in \{1, ..., J\})$. The output layer (or map layer) can be seen as the topological representation of the cognitive map whose nodes are the neurons and whose edges store the connection weights α_{jk} and angles $\rho_{jk}, (j, k \in \{1, ..., J\})$.

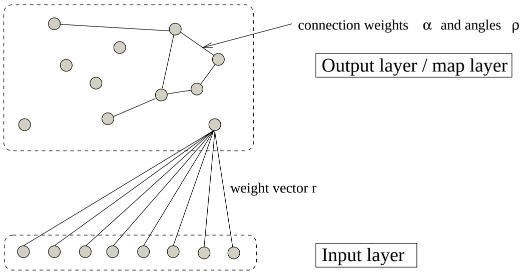

Figure 1: Structure of the neural network. The output layer represents the cognitive map. The connections between the map layer neurons contain connection weights and angle information.

The activation a_k^t for each neuron o_k in the map layer is calculated as a sigmoid function g applied to the sum x_k^t of four different terms, $a_k^t = g(x_k^t) = (1 + e^{-x_k^t})^{-1}$ with $x_k^t = s_k^t + c_k^t + l_k^t - \theta$
The output neuron o_j with the strongest activation is deemed the winner neuron. Its activation will be set to one. The other neurons keep their activation value which lies between 0 and 1 due to the sigmoid function.

The terms, which will be described in more detail in the following subsections, are the feature similarity s_k, the connectedness c_k, the movement values l_k and a fixed threshold θ. The threshold θ is subtracted to shift the input of the sigmoid function more towards zero, which results in a greater variety of possible activation values. The value of θ has to be adjusted manually according to the input types.

2.1.1 Feature Similarity s_k

The feature similarity is a measure of resemblance between the input view and an already encountered view. It does not take any context into account and is comparable with simple pattern recognition. s_k is calculated as the product of the input vector $\mathbf{f}$ and the weight vector $\mathbf{r}_k$ connecting $\mathbf{f}$ to a neuron o_j in the output layer, $s_k^t = \sum_{i=1}^{I} (f_i^t r_{ik}^t)$. Since $\mathbf{f}$ as well as $\mathbf{r}_k$ have been normalised, s_k reaches its maximum value when $\mathbf{f} = \mathbf{r}_k$.

The weights $\mathbf{r}_k$ of the winner neuron are updated according to the Kohonen learning rule

$$\mathbf{r}_k^t = \mathbf{r}_k^{t-1} + \delta(\mathbf{f}^t - \mathbf{r}_k^{t-1})$$

where δ is a learning constant. The weight vectors are then normalised to constant length. This training law gradually aligns the weight vectors $\mathbf{r}_k$ of winning neurons o_k with the normalised input vectors $\mathbf{f}$.

2.1.2 Connectedness c_k

The connectedness is used to incorporate temporal as well as spatial context into the neural network. Neurons with a strong connection to the previous winner neuron are more likely to win than other neurons. The connection value α_{jk} between the winner neuron o_k and the last winner neuron o_j increases according to the following formula:

$$\alpha_{jk}^t = \alpha_{jk}^{t-1} + \gamma(\alpha_{max} - \alpha_{jk}^{t-1}) \tag{1}$$

where γ is a learning constant and α_{max} is a constant maximum connection weight. This learning rule can be seen as a variation of the standard Hebbian learning rule.

A weight decay has been applied which decreases all connection weights between map layer neurons by a factor proportional to their actual weight. In the activation function, the connectedness c_k^t of each neuron o_k in the output layer to neurons o_j is weighted with the last activation a_j^{t-1} of the neurons.

$$c_k^t = \frac{1}{J} \sum_{j=1}^{J} a_j^{t-1} \alpha_{jk}^t$$

2.1.3 Movement Value l_k

The movement value contains additional angle information of the travelled path. Its use is motivated by the discovery of head direction cells in the rat's brain. The movement angle between two nodes is stored in ρ_{jk} and is updated each time step when o_k is the winner neuron after o_j by bisecting the stored angle ρ_{jk}^{t-1} and the input angle ρ_{inp}^t. The movement value m_{jk} for each connection is calculated as

$$m_{jk} = \cos\left(|\rho_{jk} - \rho_{inp}|\right).$$

The value for $|\rho_{jk} - \rho_{inp}|$ lies between 0 and 2π which assigns m_{jk} a value between -1 and 1.

In the activation function, the movement value of the connection between neuron o_k and each neuron o_j in the output layer is weighted with the last activation a_j^{t-1} of the neurons.

$$l_k^t = \frac{1}{J} \sum_{j=1}^{J} a_j^{t-1} m_{jk}^t$$

Having introduced these terms, the activation function can be expanded to

$$g(x_k^t) = g(\sum_{i=1}^{I} f_i^t r_{ik}^t + \frac{1}{J} \sum_{j=1}^{J} (a_j^{t-1}(\alpha_{jk}^t + m_{jk}^t)) - \theta)$$

2.2 Physical Force Model

So far, the cognitive map does not contain any explicit metric information. However, the angle information stored in the map layer connections help transforming the topological into a metric map. We arrange the map layer of the neural network into a graph $G = (V, E)$, where the set of vertices V represents the map layer nodes, and the set of edges E represents the connections between nodes.

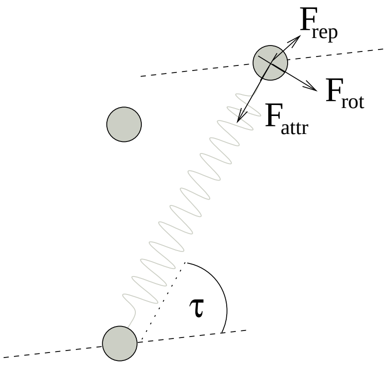

Figure 2: Three different forces can be applied to the nodes: an attractive force (springs), a repulsive force (charges) and a rotational force (local magnetic field applied to the edges).

The nodes are imagined as repulsive charges and the edges as springs between them. The spring constant has been chosen to be the same for all springs, since ideally the place neurons are equally distributed on the plane. Note that the neural network model does not contain any explicit metric information about the position of the map neurons.

Let $\delta(v, w) \in \mathrm{R}^2$ be the distance vector between node v and node w and $d(v, w) = ||\delta(v, w)||$ its Euclidean distance. Between each pair of nodes, there is a force $F(v, w) = F_{rep}(v, w) + F_{attr}(v, w)$, the sum of a repulsive and an attractive force. The repulsive force is caused by the charges and the attractive force is caused by the springs. If the two nodes are not connected by a spring, then $F_{attr}(v, w) = 0$.

The attractive force is computed according to Hooke's law:

$$F_{attr}(v, w) = \gamma_{attr} \delta(v, w)$$

and the repulsive force is set to be proportional to the inverse square of the distance,

$$F_{rep}(v, w) = -\gamma_{rep} \frac{\delta(v, w)}{d(v, w)^3} = -\gamma_{rep} \frac{\delta(v, w)}{||\delta(v, w)||^3},$$

where γ_{rep} and γ_{attr} are free parameters to adjust the forces.

At the beginning, all the nodes get assigned random x and y values on the Euclidean plane. In each step, the forces for each node are calculated and the nodes are moved towards the direction of the according force vector by the amount the force indicates.
The graph is balanced when the sum of forces between all nodes

$$\sum_{v, w \in V, w \neq v} (F_{rep}(v, w) + F_{attr}(v, w))$$

is minimal and therefore the potential energy has reached a minimum. The graph is perturbed from time to time to avoid getting stuck in stuck in local minima.

The algorithm for a fast and stable solution comes from Fruchterman and Reingold (Fruchterman and Reingold, 1991) and is already implemented in the graph algorithms library of LEDA (LEDA, 1998). It is a modification of the original spring-embedder model of Eades (Eades, 1984) and works in analogy to forces in natural systems.

This algorithm does not solve the edge orientation problem yet, since it only implements spring and repulsive forces. Therefore, we modified Fruchterman's spring algorithm by adding rotational forces $F_{rot}(v, w)$ to the springs. These forces take into account the angle information between connected map layer neurons in the neural network (see figure 2).

$$F_{rot}(v, w) = \gamma_{rot} \tau(v, w) d(v, w)^2 \delta_{\perp}(v, w)$$

where $\tau(v, w)$ denotes the angle between the current and the stored edge orientation and $\delta_{\perp}(v, w)$ denotes the unit vector normal to $\delta(v, w)$ whose cross product with the preferred edge direction vector does not contain any negative components[1]. Since $\tau(v, w) = \pi - \tau(w, v)$, $F_{rot}(v, w) = -F_{rot}(w, v)$.

2.3 Related Work

There is a vast literature on navigation systems (for a thorough review see for example Trullier et al. (Trullier et al., 1997)). We mention only a few that are closely related to our work:

Schölkopf and Mallot (Schölkopf and Mallot, 1995) developed a view-based mapping and path planning system for mazes with a one-to-one correspondence between directed corridors and views, using a self-organising map. This self-organising map is similar to the map in our model in its basic structure, the environment however is restricted since the positions of the places were predefined. The quality of the resulting maps has been tested by a goal finding task. Map quality measurement by performing specific goal finding tasks is feasible for mazes with a deterministic number of routes, but turns out to be very difficult for open environments.

[1] In R^2 there are two possibilities for a unit vector normal to $\delta(v, w)$.

A model by Franz and Mallot (Franz et al., 1998b) was used to explore open environments and build graph-like structures. It has been implemented on a mobile robot, however it was not based on a self-organising map.

Self-organising feature maps for navigation tasks were also used by Owen and Nehmzow (Owen and Nehmzow, 1997), who were associating recognised places with actions which allowed route learning. The method has been implemented on a mobile robot using sonar as well as infrared sensors.

A recent computational model of the rat's hippocampus has been presented by Arleo and Gerstner (Arleo and Gerstner, 2000). They are modelling an assembly of place cells coding for the current position of an agent. Both external (visual input) and internal, self-generated information (idiothetic information) is used, as in our model. The movement direction is coded in four neurons standing for north, west, east and south. An interesting aspect is the active exploration technique which leads to a better distribution of the exploration tour. The second part of their work implements goal oriented behaviour.

Goal oriented behaviour has also been implemented by Trullier and Meyer (Trullier and Meyer, 1998). A topological representation with a population of place cells as a directed graph has been used in their model. Sequences of places were learned in a continuous environment with obstacles. It was assumed that place fields are uniformly distributed in the environment.

3. Experiments

Experiments have been performed in simulation (section 3.1) as well as on a mobile robot (section 3.2).

3.1 Simulation

A simple two-dimensional environment has been chosen for this experiment. It is restricted by surrounding walls and contains some convex obstacles (see figure 3). The exploration is performed by a circular robot, equipped with eight distance sensors positioned equidistantly around its body. They allow for a very low resolution one-dimensional omnidirectional 'view'. The sensors have a maximum range of 15 times the robot's body size. The sensor reading can be adjusted according to the current bearing of the robot. The choice of distances as a sensor input has not been chosen for its biological plausibility, but rather for its simplicity and for fulfilling a requirement such as continuity, which is essential for the map building. Using a surround 'view' instead of directional views roughly agrees with the visual field of the rat, which covers 320°.

The exploration scenario obeys the following rules: the agent explores the environment by moving through it randomly but smoothly. Obstacle avoidance is per-

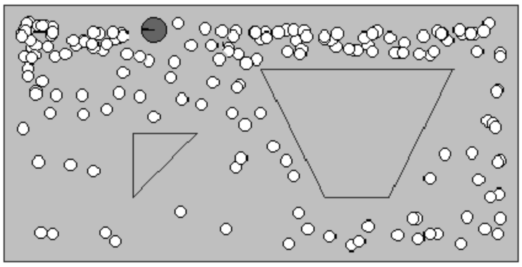

Figure 3: The simulated environment after an exploration tour of the robot. The white circles mark the position where the sensors have been read.

formed by reading the three front distance sensors and turning accordingly. Every few time steps, the agent transfers its sensor readings to the neural network presented in section 2.. These readings include the current compass orientation. Using this method, the cognitive map is gradually generated during exploration.

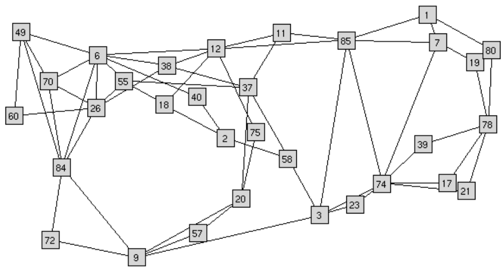

Figure 4: This figure shows a cognitive map resulting from the tour pictured in figure 3. The nodes represent the center of the place fields and the edges connections between the places. Only neurons with at least one connection to another neuron are displayed.

The neural network parameters have been optimised experimentally: The learning constant γ_r for the weights between input and map layer, the learning constant for the map layer weights γ_α and the maximal value for the map layer connection weights α_{max} were all set to 0.3. The activation function threshold θ was set to 2.5.

3.1.1 Analysis of the Place Fields

In figure 5, two typical place field pictures for different numbers of map layer neurons are shown. Noticeable in both pictures is the increased number of place fields around the objects. This phenomenon can be explained by the sensor input changing more frequently at convex corners. The effects, however, might lead an outside observer to the conclusion that the robot is more interested in objects than in open spaces.

If one of the place cells dies, the map is only slightly disturbed in its topology. Other neurons, ideally those with neighbouring place fields, will take the part of the

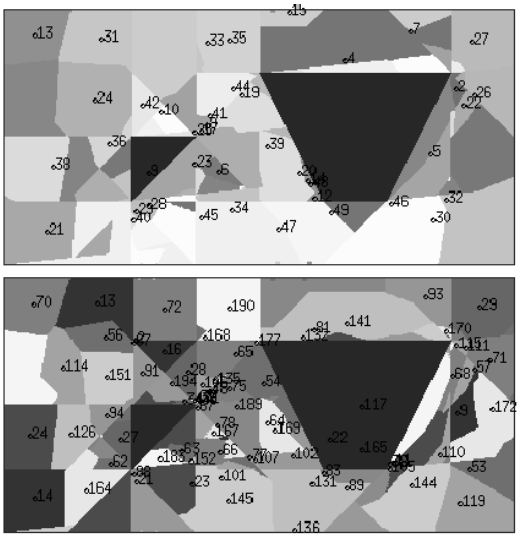

Figure 5: Distribution and shape of the place fields. The grey value of each pixel shows the index of the neuron with the highest activation for sensor measurements taken at this place. Picture a) shows the typical place fields for a neural network with 50 map layer neurons, picture b) uses 200 map layer neurons.

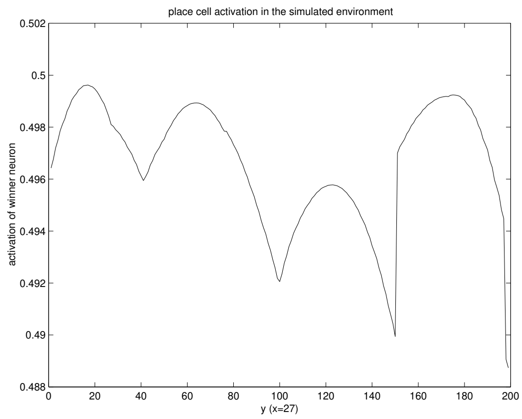

Figure 6: Winner neuron activities on a vertical section through the place fields in the environment. The simulated place fields show Gaussian activity distribution as in rat's place fields.

dead cell (no 'Grandmother' cell effect). This follows directly from the definition of a place field.

The distribution of the winner neuron activities for particular places is shown in figure 6. Each place field can be recognised as a small activity hill. Place fields measured in rats look similar to the place fields in these experiments. They are usually a few times the animal's body size, circular, and show firing rates which decrease similarly to Gaussian distributions the further away the rat is from the mass centre of the place field.

3.2 *Robot Experiments*

To show that the model not only works in theory or with highly regular input data, but is also able to produce stable real world maps, it has been tested on a mobile robot.

3.2.1 *Hardware*

The mobile robot 'Samurai'[2] is equipped with an omnidirectional camera system as well as a magnetic field compass. The omnidirectional camera system consists of a CCD camera pointing towards a convex mirror (Chahl and Srinivasan, 1997). It provides omnidirectional visual information between 30° above and 58° below the horizon. (see figure 7).

The robot is equipped with a compass that consists of two orthogonally adjusted flux gate sensors which produce output voltages V_x^a and V_y^a, which vary as sine and cosine functions of the compass angle θ. The angle θ of the compass (and therefore the robot) is measured by taking the arctangent of the quotient of the A/D converted output voltages V_x and V_y.

$$\theta = \arctan \frac{V_x}{V_y}$$

This allows for rather stable values, which however are strongly influenced by power lines etc. and do not always represent the true global compass orientation accurately.

3.2.2 *Visual Processing*

Instead of distance information like in the simulation, visual information has been chosen in the robot experiments. Visual input can be regarded as a biologically plausible tool when performing navigation tasks. Visual navigation usually distinguishes place based navigation from directional view based navigation. In our case, place based navigation has been chosen for the following reasons: views of places have not to be transformed long windedly into descriptions of places, but present them almost directly. It requires less memory to store one place view instead of several directional views. The visual field of many animals, in contrast to humans, spans almost 360° (320° for rats). It seems that navigation strategies can be kept simpler if the field of views are larger. Places can be remembered by orientation invariant place vectors when using direction adjusted place views, directional views are not orientation invariant. Omnidirectional cameras with similar mirror techniques we applied have already been used by (Chahl and Srinivasan, 1997), (Franz et al., 1998a), (Srinivasan et al., 1997), (Möller et al., 1998) and (Yagi and Yachida, 1991).

[2]Samurai is produced by Neuronics, Zurich

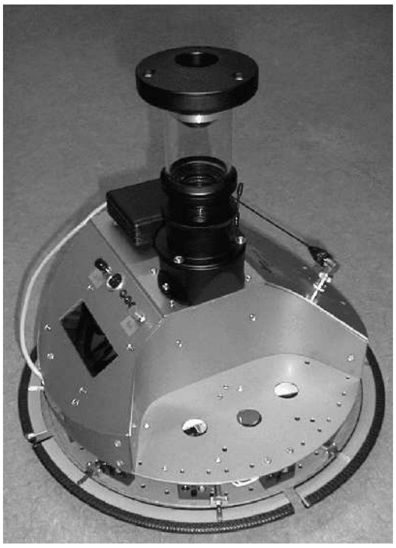

Figure 7: The Samurai robot, equipped with an omnidirectional camera and a magnetic compass.

The visual input is processed in several steps to produce a low-resolution, rotation-invariant output that is continuous in space and time. No explicit feature recognition has to be performed, neither in the simulation nor in the robot experiments. This has the advantage of the processing being faster and more reliable. Moving objects, for which the model has not been tested yet however, might prove to be problematic.

Figure 8: The environment as seen through the camera.

First, the original camera image from figure 8 is tranformed into polar coordinates. The size of the range below and above the horizon is adjustable. Next, the image is vertically averaged and a horizontal Gaussian filter is applied to smoothen the image (which is now 1D, see figures 9 and 10). To account for the current robot orientation, this image line is rotated by a value proportional to the compass angle. The data is sub-sampled to an output vector of 16 elements, which can also be called a place vector. This choice has been made, since the dimensionality d of the output vector has to lie in a certain range: if d is too large, errors in the compass reading become more problematic, additionally the learning time of the neural network increases. If d is too small, the standard deviation of the Gaussian filter is not large enough to have the desired effect of creating continuous place vectors over time, which results in unstable place fields.

The neural network parameters used are the same as in the simulated environment apart from the input vector dimension, which is 16 instead of eight, and the threshold θ, which is 1.0 instead of 2.5. Input to the neural network are the place vectors gained through visual processing as well as the current compass value θ.

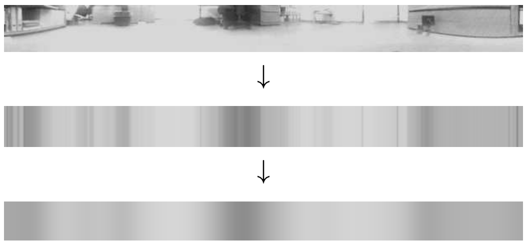

Figure 9: The polar mapping of the camera image (top), which is then vertically averaged (center) and smoothened (bottom).

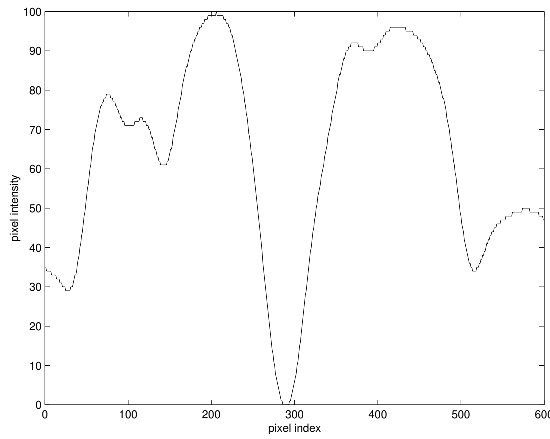

Figure 10: Intensity curve of the vertically averaged, smoothened and normalised polar image section.

3.2.3 Environment

All robot experiments have been performed in a rectangular office room (approximately 6x4 meters) containing three desks as well as shelves. Numerous items were lying on the desks as well as on the floor. The light source consisted of three neon lamps at the ceiling. The robot has been steered manually through the room for exploration using a remote control. During this tour, the place vectors to be processed by the neural network have been recorded.

3.2.4 Rotation Invariance

In figure 11, the 16 sensor values are plotted for the robot rotating on place. Sensor values for each time step are connected by a line. Note that the neighbourhood relationship between adjacent pixels will not be conserved when the sensor values are fed into the neural network.

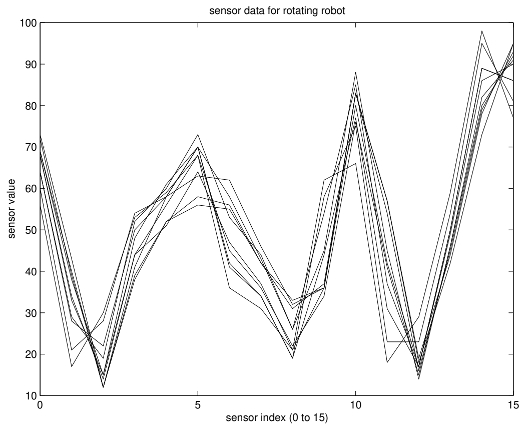

Figure 11: Output from the visual processing system while the robot was rotating at the same place.

In figure 12, the 16 sensor values are plotted for the robot moving from one side of the room to the other in a straight line. Note that the sensor input changes quite fast, however the continuity from step to step is still apparent.

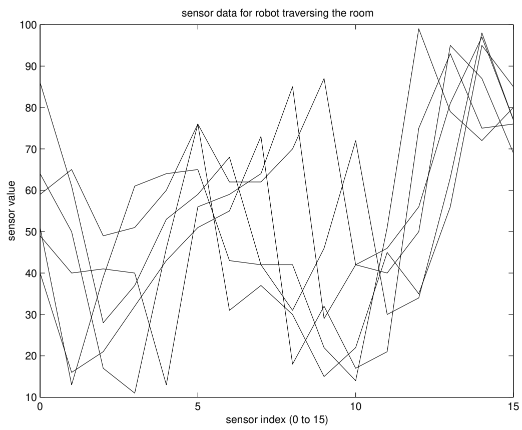

Figure 12: Output from the visual processing system while the robot was moving in a straight line from one side to the other side of the room.

3.2.5 Evaluation of the Topological Structure

We distinguish between different exploration tours performed in the same room described above. The first and simplest one is a movement forth and back along a large shelf with eventual rotations at the turning points (tour1). The second exploration tour (tour2) has been chosen to be a movement around an object (a coloured cardboard box) placed in the middle of the room. Both clockwise and counterclockwise movements have been performed. The distance to the object has been kept smaller than 1.5 meters. The third one (tour3) is a random tour through the room, avoiding obstacles.

A typical place cell graph for each tour is depicted in figure 13. The number of nodes is in accordance with the

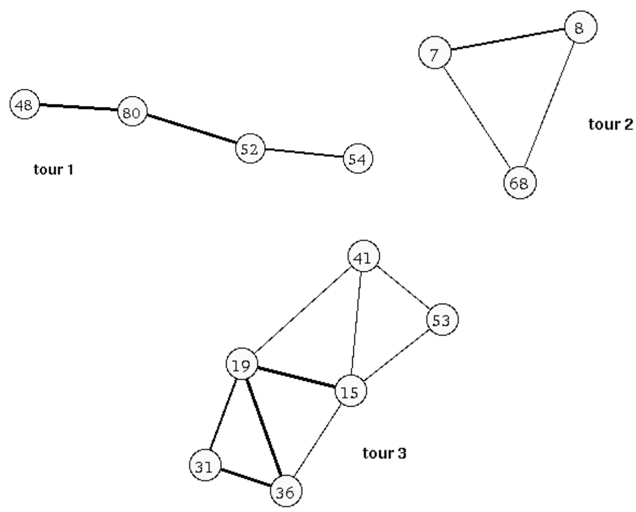

Figure 13: Maps for different exploration tours in an office room. The parameters for all three maps were the same (100 map neurons). The strength of the connection weights is shown as the thickness of the drawn edges.

area of space that has been explored. The connectivity is clear and usually results in planar maps. The place fields have the size of a few times the robot's body size.

Even when exactly the same sensor recordings are fed into the neural network, the resulting maps can have different topology as well as different angle values. The reason lies in the random initialisation of the neural network weights. For two different random exploration tours with the robot, it is very unlikely to receive exactly the same maps. However, each map converges to a stable state during learning. As there is no such thing like 'the right map', maps with different topology can have the same functionality for navigation in the same environment.

3.2.6 Structure of the Place Fields

In the robot experiments, it is more difficult to find appropriate map quality evaluation methods than in simulation. The picture of place fields that can be measured is coarser and not as accurate as in the simulation, mostly due to the noise that influences the camera picture as well as the position and the compass information of the robot.

To be able to evaluate the place fields in the robot experiments, some measurements had to be taken: The room has been classified into a grid of 15 times eight squares (120 in total), each covering $625cm^2$. The robot has been placed at the center of each square and the corresponding camera snapshot has been stored. After the network has been trained with a random walk, the place vectors of these 120 snapshots were fed into the neural network. The activation of each map layer cell in the case of the place field picture was recorded by only using the feature similarity term without the connectedness and movements values, since the order of the snapshots was not taken into account. The indices of the winner neurons have been recorded and were used to create a place field map of the environment representation. Examples are shown in figure 14.

Despite the low resolution of the place field picture, one can clearly see the place fields which have the size of a few times the robot's body size. The size distribution of the place fields also takes care of the fact that two chairs and several folders were standing in the area of the upper two corners. The winner place cells are changing more frequently at these locations. Another effect to be noticed is the occasional occurance of multiple place fields. This effect has also been observed in rats.

Multiple place fields are problematic for the visualisation of the place cell graph, since one cell with multiple place fields represents two or more regular cells. The number of connections for this cell is increased, and its spatial position gained from the physical force model lies somewhere between the positions of its place field centers. This can lead to non-planar graphs. A practical solution to this problem is to only include those edges whose strength lies beyond a certain threshold (see figure 14).

4. Discussion

4.1 Convergence

During learning, the activation of the winner neuron converges to a certain value. We are feeding the recorded place vectors of the same run several times consequently into the neural network. In doing so, the result of the data is more comparable, additionally, it also resembles the playing back of neural states in the hippocampus during REM sleep (Wilson and McNaughton, 1994). In figure 15, the activity of the winner neuron is plotted over a run of 100 place vector readings. The dotted line shows the first iterations of the run, the solid line shows the 100^{th} iteration. The certainty of being at a certain place increases over time.

In figure 16, ten learning runs of the same exploration tour through the room but with different random starting weights are plotted. All of them are converging, however they are not converging to the same value.

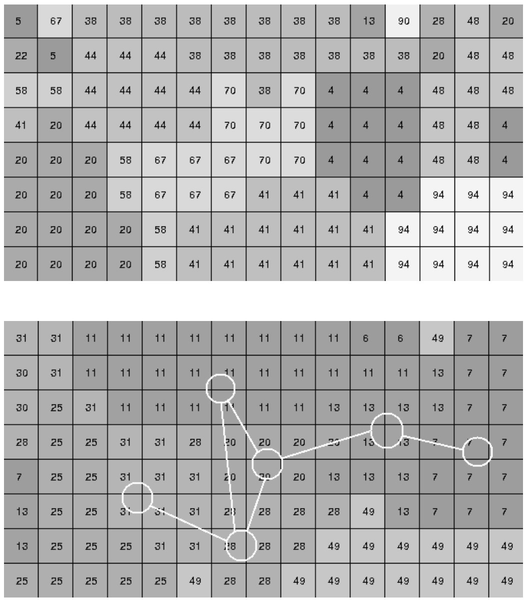

Figure 14: This figure shows two examples of place fields of the environment representation after a random walk. A place field in these figures consists of directly connected squares with the same index. The indices represent the winner neurons. The neural network of the first picture started with 100 map layer neurons, the second with 50. In the second place field picture, the graph connections exceeding a certain connection strength threshold were inserted into the diagram.

4.2 Implementation of Goal Finding

Future work will include the implementation of a goal finding mechanism. Finding a known goal could be realized by letting the agent 'think' of the goal place, which triggers one place cell to fire with the highest activation and win. After this activation spread on other connected nodes until it has reached the current node where the agent 'thinks' it currently is, it just has to walk into the direction where the activation came from. The spread of activation can be modulated by the connection weights in a way that allows stronger connections pass the activation faster. The problem with this approach is that there needs to be a control system outside of the map, which performs some sort of *mental exploration*.

4.3 Conclusion

A computational neural network model has been presented and tested in simulation as well as on a mobile robot. The model creates sufficiently stable maps, which can be used for navigation purposes. The topology of the map does not change after a few iterations of an exploration tour. The main advantage of this model to some existing models is that it does not require any metric data of the robot's actual position. The model ex-

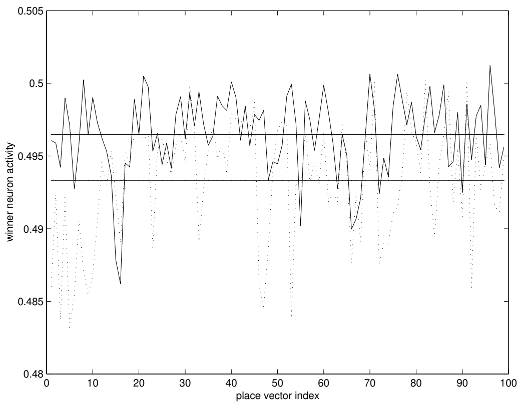

Figure 15: Winner neuron activations for an exploration tour with the robot. The dotted line shows the first iterations of the run, the solid line shows the 100^{th} iteration. The straight lines are the average activation of the winner neuron over each iteration.

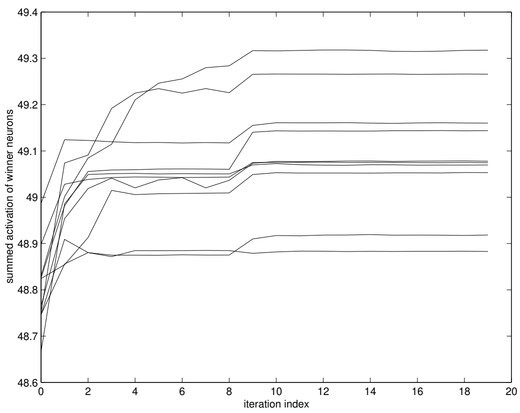

Figure 16: Different winner neuron activations summed for each iteration. The activations are converging to different values for different runs.

ploits the camera data as well as the time and order it is acquired. The required metric information can be extracted in a self-organising manner.

The artificial neural network is inspired by real neuronal architectures. The neurons and synapses in this model are idealised and should not be regarded as biologically realistic neurons and synapses. Direction information is currently stored in synaptic weights, but might also be modelled in a way more similar to head direction cells. This could be done by allowing presynaptic inhibition or facilitation.

So far, only the acquisition of cognitive maps has been investigated, without using these maps for specific tasks. Acquisition and application of maps could be separated phases, but it would also be possible to exploit the already learned map to influence the robot movements during learning.

In summary, a computational model of cognitive maps has been created that can be used for navigation tasks in realistic environments. It has been shown in simulated as well as in robot experiments that a consistent cognitive map can be built during exploration tours. The sensor input for the robot could be kept very low level, no explicit metric information gathering was necessary. The cognitive map has been successfully enhanced with metric information by the physical force model. Learning takes place in an non-supervised manner, which resembles the navigational learning processes in biological systems.

Acknowledgements

Many thanks to Ralf Möller for helpful discussions.

References

Arleo, A. and Gerstner, W. (2000). Spatial Cognition and Neuro-Mimetic Navigation: A Model of Hippocampal Place Cell Activity. *Biological Cybernetics, Special Issue on Navigation in Biological and Artificial Systems (to appear).*

Chahl, J. and Srinivasan, M. (1997). Reflective surfaces for panoramic imaging. *Applied Optics*, 36:8275–8285.

Eades, P. (1984). A heuristic for graph drawing. *Congressus Numerantium*, 42:149–160.

Franz, M., Schölkopf, B., Mallot, H., and Bülthoff, H. (1998a). Where did I take that snapshot? Scene-based Homing by Image Matching. *Biological Cybernetics*, 79:191–202.

Franz, M. O., Schölkopf, B., Mallott, H. A., and Bülthoff, H. H. (1998b). Learning View Graphs for Robot Navigation. *Autonomous Robots*, 5:111–125.

Fruchterman, T. and Reingold, E. (1991). Graph-drawing by force-directed placement. *Software - Practise and Experience*, 21:1129–1164.

Kohonen, T. (1982). Self-organizing formation of topologically correct feature maps. *Biological Cybernetics*, 43(1):59–69.

LEDA (1998). *LEDA, a library of the data types and algorithms of combinatorial computing.* Max-Planck-Institut für Informatik, Saarbrücken.

Möller, R., Lambrinos, D., Pfeifer, R., Labhart, T., and Wehner, R. (1998). Modeling Ant Navigation with an Autonomous Agent. *From Animals to Animats 5: Proceedings of the 5th International Conference on Simulation of Adaptive Behavior*, 5:185–194.

O'Keefe, J. (1976). Place units in the hippocampus of the freely moving rat. *Experimental Neurology*, 51:78–109.

Owen, C. and Nehmzow, U. (1997). Middle scale navigation - a case study. *Proc. AISB 97 workshop on Spatial Reasoning in Animals and Robots.*

Schölkopf, B. and Mallot, H. (1995). View-Based Cognitive Mapping and Path Planning. *Adaptive Behavior*, 3(3):311–348.

Srinivasan, M., Chahl, J., and Zhang, S. (1997). Robot navigation by visual dead-reckoning: inspiration from insects. *International Journal of Pattern Recognition and Artificial Intelligence*, 11:35–47.

Taube, J. S., Muller, R. U., and Ranck, J. B. (1990). Head-direction cells recorded from the postsubiculum in freely moving rats. *Journal of Neuroscience*, 10:420–447.

Tolman, E. C. (1948). Cognitive maps in rats and men. *Psychological Review*, 55:189–208.

Trullier, O. and Meyer, J.-A. (1998). Animat Navigation Using a Cognitive Graph. *From Animals to Animats 5: Proceedings of the 5th International Conference on Simulation of Adaptive Behavior*, 5:213–222.

Trullier, O., Wiener, S. I., Berthoz, A., and Meyer, J.-A. (1997). Biologically based artificial navigation systems: review and prospects. *Progress in Neurobiology*, 51:483–544.

Wilson, J. and McNaughton, B. L. (1994). Reactivation of Hippocampal Ensemble Memories During Sleep. *Science*, 265:676–679.

Yagi, Y. and Yachida, M. (1991). Real-time generation of environment map and obstacle avoidance using omnidirectional image sensor with conic mirror. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 160–165.