A Self-organizing Method for Robot Navigation based on Learned Place and Head-direction cells

Xiaomao Zhou*†, Cornelius Weber†, Stefan Wermter†
*College of Automation
Harbin Engineering University, Harbin, China
xiaomaozhou26@gmail.com
†Department of Informatics
University of Hamburg, Hamburg, Germany
{weber, wermter}@informatik.uni-hamburg.de

***Abstract*—This paper describes a neural model for a robot learning spatial knowledge and navigating on learned place and head-direction (HD) cell representations. The place and HD cells, which are trained through unsupervised slow feature analysis (SFA) from sequences of visual stimuli, provide positional and directional information for navigation. Based on the ensemble activity of place cells, the robot learns a topological map of the environment through extracting the statistical distribution of the place cell activities covering the traversable areas and realizes self-localization based on the map. The robot's heading direction, which is encoded by the HD cells, works as a control signal to adjust its behavior. Action representations supporting state transitions are learned through memorizing the same movement from a previous phase where an experimenter drives a robot to explore an environment. Given reward signals spreading from a target location along the topological map, the robot can reach the goal in a reward-ascending way. This work intends to build a practical navigation system by simulating animals' hippocampal cell firing activities on a robot platform using its self-contained sensor. Experimental results from simulation demonstrate that our system navigates a robot to the desired position smoothly and effectively.**

***Index Terms*—Place cell, Head-direction cell, Robot navigation, Slow Feature Analysis**

I. Introduction

Endowing mobile robots with the capacities of autonomy and intelligence has attracted substantial attention from different research communities for many years and still poses a challenge in certain real-world environments. It is a fundamental ability for an autonomous robot to complete spatial navigation which usually involves information from external sensory perception and internal path integration. Simultaneous localization and mapping (SLAM) [1], [2], in which a mobile robot incrementally builds a map of the ambient world while exploring and simultaneously keeps track of its position with respect to the derived map, provides a popular approach, but mainly focuses on the environment's metric characteristics. The computational load of building a metric-precise occupancy map and the loss of semantics impose restrictions on the flexibility of further planning. For example, a geometrically precise occupancy map could not tell a user where a desired object is lying.

In contrast, planning based on a cognitive map demonstrates a high flexibility. Instead of representing all the world's details, which would be redundant, a cognitive map enables to interpret the world on a higher abstract level and represents an environment succinctly but without loss of important information. This can make planning much easier and human-interpretable. For example, a person can easily find a route between two positions by referring to their related positions in a sketch map, without knowing their precise positions. In biology, a cognitive map is regarded as a mental representation of an environment which abstractly encodes information about spatial positions together with local attributes which vary in the form of environmental features, spatial relationships or events [3]. Enabling robots to navigate based on a cognitive map can contribute to both biology and robotics research [4], [5].

Advances in neuroscience have long suggested that cognitive maps are neurally instantiated by animals' hippocampus and related structures [6], where certain types of neuronal populations were found to exhibit space-related firing properties, e.g., place, HD, grid and border cells [7], [8], [9]. Although the precise mechanisms of hippocampal learning need to be further studied, many approaches have been proposed to successfully explain the formation of hippocampal representations, among which some are intended to solve spatial learning for robot navigation [10], [11]. Generally, these models can capture some aspects of spatially selective firing patterns on different levels of abstraction, using different input types and forming mechanisms. In terms of neural computation, the slowness principle has been argued as a fundamental principle for hippocampal learning during navigation [12]. Based on this principle, a hierarchical Slow Feature Analysis (SFA) network can result in the self-organization of certain hippocampal cell types such as place and HD cells through learning from visual stimuli in an unsupervised way [13].

Obtaining hippocampal activities does not mean that these representations can be easily interpreted to explain their contributions to navigation tasks. For instance, a place field observed from an animal's brain readings only codes for a specific location to which a particular place cell responds, but such sparse firing pattern needs to be interpreted properly in order to support a continuous navigation task. Besides, except for spatial metric information, spatial cognition also requires a real-world understanding of world contexts, i.e., objects and events that an agent perceives in an environment. The spatial representations originating from hippocampal cells must incorporate the knowledge about the relationships with the ambient world and lead to the behavioral changes of an agent. To this aim, a reinforcement learning mechanism has been used in goal-oriented navigation tasks where place cell activities are mapped to action cells through reward learning, generating a navigational map denoting the direction to the desired goal from each position [14], [15]. However, this approach only considers the relationship between the current position and a goal position, ignoring state transitions between neighboring positions, and requires re-learning whenever a goal position changes. For state transition learning, [16], [17] present a neural network to learn the state-action associations to determine the actions leading to any adjoining position. And [18] proposed sensory-motor learning between each performed transition and the corresponding movement, where the movement is encoded in a neural field providing the current heading direction. They adopt the same concept to associate each transition with an appropriate moving direction.

In this paper, we develop a neural-inspired model that contains not only hippocampal learning but also spatial learning and behavior-based robot navigation. The neural navigation mechanism is inspired by the work of Yan et al. [19], where, the explicit position and direction information was obtained using a ceiling camera. In contrast, our work uses no external sensors and extracts metric information from the firing activities of a population of location- or direction-related neurons which are generated using a robot-mounted camera. Moreover, during robot navigation, the position and head direction are represented by the ensemble activity of place and HD cell activity respectively, without any specific position or direction value given in a reference coordinate. The cognitive map also contains action information relating to state transitions which is first recorded during the map-building phase and then is recalled for similar transitions in a future phase. In principle, our model can be extended to incorporate context understanding where the local view information in the world can be learned through feature extraction and assigned to the corresponding spatial representation in the map. With the proposed approach, our model can navigate a robot to a desired position of choice smoothly by using its own camera only.

II. Methods

Our framework consists of two parts. As shown in Figure 1, the first part contains two pretrained networks, place and HD cell networks, to generate place and HD cell activities based on visual images only. These networks are trained according to the slowness principle [12] in SFA and output activities encoding a certain position and direction respectively. As the basis of our model, this part works as a preprocessing step to extract positional and directional information from visual images. The second part is constructed on top of the first part and interprets the encoded metric information from it to enable a robot to perform navigation. It consists of: (1) two spatial learning networks that encode information about the distinct spatial representation from place and HD cells respectively. The first network (PC-GWR) learns the structure of the environment from place cell activities which are provided by the trained place cell network. Meanwhile, the second network (HD-GWR) learns from HD cell activities to represent the robot's current heading direction; (2) an action memory for storing the action representations for state transitions. When the robot executes a state transition and its state representation in the PC-GWR changes, the corresponding action supporting this movement will be encoded in the action memory. This action will be recalled when the same transition needs to be done and serves as the input to the action layer; (3) an action layer (dynamical neural field (DNF)) that generates smooth action commands for robot controlling. During movement, the emerged action input signals are aggregated on the action layer and are then integrated by the DNF to provide a smooth control command to adjust the robot's behaviors to approach a target position. For future work, we plan to provide the robot with knowledge of the world information, e. g., the knowledge of what objects can be perceived at a certain position, so that by showing it a picture of an object, the robot will know where to find it [20].

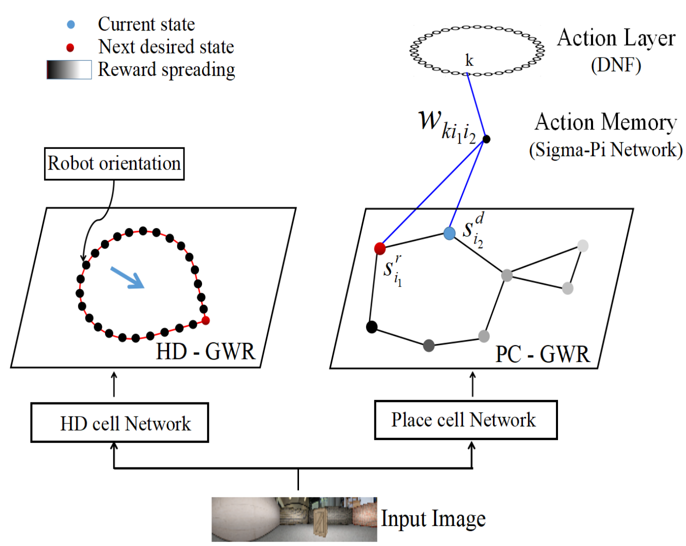


Fig. 1: The architecture of the proposed model. The place and HD cell networks on the lower part provide the basic positional and directional information. The upper part works on this information to enable goal-directed navigation.

A. Modelling Place and HD Cells

Throughout the work, the position and head direction information is represented by the ensemble activity of place and HD cells respectively. To model these activities, this work uses SFA to train two networks generating place cells and HD cells respectively. SFA is an unsupervised learning algorithm working on image sequences. For an image sequence encoding an agent's spatial movement, taking advantage of the inherently existing temporal coherence among time-depended sensory input, SFA together with an additional sparse coding step [13] can calculate a distributed representation of the agent's position and orientation, which resembles the firing pattern found in hippocampal cells.

Since the SFA learning representation highly depends on the input statistics [13], for intentional modelling of different cell types, we drive the robot to actively explore the environment with its visual system and simultaneously train two networks using image sequences collected during different movement phases. For instance, during forward movements, the robot continuously changes its position and rarely adjusts its direction. The emerging slow features during these phases will compactly encode the robot's direction and can be learned for modelling HD cells. Similarly, the data during rotational movements is used to train the place cell network. After training, the learned representations from these two networks highly resemble the firing patterns found in the place cells and head-direction cells respectively. The response of a network to a single image which is captured at a certain position to a certain direction will approximate the place cell activity at that position or the HD cell activity to that direction. For details on how to train these two networks, please refer to our previous work [21].

B. Spatial Learning based on Place Cells

Since place and HD cells are obtained by unsupervised learning, their activities encode no predefined relationship to a real-world position. Since the learned place cells encode positions in an environment by their firing preference to different locations, to interpret the encoded positional information, a GWR network [22] is used which learns the internal relationship of these activities. For learning from place cell activities covering an area, a GWR network (PC-GWR) can result in a map approximating the topology of the explored space, where GWR nodes represent spatial positions, and connections represent relations between positions. Starting with two random nodes, the map grows incrementally when trying to capture the distribution of the place cell activities. During learning, neurons and connections are allocated or updated dynamically, and will also be deleted when meeting certain criterions. The learning algorithm is shown in **Algorithm 1**.

After learning, the network consists of a set A of nodes, each associated with feature vectors v, and a set N of connections to describe the relations between each connected nodes pair in A. Given a robot's current position, which is represented by an ensemble activity of place cells, the robot can be localized by simply calculating the best matching node in the PC-GWR network. As shown in Figure 1, the robot's current position is represented by the red neuron in the PC-GWR.

C. Orientation Representation based on HD Cells

Since knowing a robot's orientation is important for controlling its movement, in this work we represent the robot's heading direction using learned HD cells, which have demonstrated strong selectivity to directions and large invariance to positions. We use a second GWR network (HD-GWR) to learn from HD cell activities covering a robot's entire angle range of $360°$. The GWR learning algorithm is the same as in the previous section, except that the input now is a robot's HD cell activities. Typically, a ring-form topology of the HD-GWR arises in our experiment. With this GWR network, although the HD cell activities do not indicate an explicit direction value due to unsupervised learning, a robot's current orientation can be represented by the best matching node in the map. Note that this HD-GWR also plays an important role in action representation learning and robot control. We allow the HD-GWR network to grow up to a size of 36 in an attempt to represent the direction in $10°$ increments. An illustration can be seen in Figure 1, where the robot's current orientation is denoted by the best matching neuron (red node) in the HD-GWR (the red circle around the robot's current state).

Algorithm 1 Growing-When-Required Network [22]

1: Start with two neurons n_1 and n_2 with random weights w_{n_1} and w_{n_1}.
2: Generate an input signal ζ (place cell activity vector) according to the place cell network.
3: Find the nearest neuron s and second-nearest neuron t according to the distance from the input: $\|\zeta - w_i\|$
4: If there is no connection between s and t, create it. Otherwise, reset the age of this connection to zero.
5: Calculate the activity of each neuron i :
   $s_i = exp(-\|\zeta - w_i\|/2\delta^2)$
6: If $s_s <$ activity threshold a_T and firing counter $h_s <$ firing threshold h_T, insert a new neuron as follows:
   - Add a new neuron r halfway between the best matching neuron and current input: $w_r = (w_s + \zeta)/2$
   - Insert connections between s and r and t and r
   - Remove the connection between s and t

   Else, i.e., no new neuron is added, adapt the positions of the best matching neuron s and its neighbours i:
   $\Delta w_s = \epsilon_b \cdot h_s \cdot (\zeta - w_s)$
   $\Delta w_i = \epsilon_n \cdot h_i \cdot (\zeta - w_i)$
   where $0 < \epsilon_n < \epsilon_b < 1$ are learning rates and h_s is the value of the firing counter for node s.
7: Age connections with an end at s: $age_{(s,i)} = age_{(s,i)} + 1$
8: Reduce the firing counters of neuron s and its neighbours:
   $h_s(t) = h_0 - \frac{S(t)}{\alpha_b}(1 - e^{(-\alpha_b t/\tau_b)})$
   $h_i(t) = h_0 - \frac{S(t)}{\alpha_n}(1 - e^{(-\alpha_n t/\tau_n)})$
9: Remove all connections with ages larger than a_{max} and remove neurons without connections.
10: If the stopping criterion is not yet fulfilled, go to step 2.

D. Global Path Planning

A robot's navigation path can be regarded as a combination of many small transitions in which a robot performs each transition selectively from its current state to the next directly connected one and repeats this process until reaching the desired goal. For a goal-directed navigation task where the robot's current and target state representations in the PC-GWR are known, it then needs to select the immediate next state among the many connected ones. For this, we assign each node in the PC-GWR a reward value. During spreading, the reward signal starts with an initial reward r from a known target state g ($r_g(t=0)=r$) and decreases iteratively to the nearest neighboring node i (within a dynamically updated neighborhood list nl) with a discount factor $\lambda < 1$:

$$r_j(t+1) = \lambda r_i(t), \text{ for } j \in \text{nl(t)} \; and \; r_j(t) < r_i(t) \tag{1}$$

For each iteration, the neighborhood list nl will be updated as follows:

$$\begin{aligned} n' \leftarrow i \;\; & \text{if i connects with neuron } j \in nl(t) \\ & \text{and } r_i(t+1) < r_i(t),\; i \notin nl(t) \\ nl(t+1) = n' & \end{aligned} \tag{2}$$

Assuming the robot's current position is represented by node i_1 in the PC-GWR, we define that the desirability activity $s^d_{i_2}$ of its neighboring node i_2 is calculated as follows:

$$s^d_{i_2} = s^r_{i_1} r_{i_2} \tag{3}$$

where $s^r_{i_1}$ is the map activity of neuron i_1 in the PC-GWR (see **Algorithm 1**).

During navigation, the robot will choose its next immediate state by selecting the one with the largest desirability activity s^d among its neighboring ones.

E. Local Action Learning

Performing state transitions requires executing appropriate actions, for example, a robot needs to turn left in order to get to its left area. In this work, we use the robot's heading direction, which is represented by the HD-GWR, as a control signal.

1) Sigma-Pi network: For action learning, a Sigma-Pi network [23] is used to learn the action information associated with the state transition. The Sigma-Pi network has two dimensions of input neurons which receive the neuron activities of a robot's current and next desired state in the PC-GWR respectively, and outputs the action code to the action layer one layer upstream. The action information is stored in the Sigma-Pi connections between pairs of PC-GWR neurons and the action layer, using second-order weights $\{w_{ki_1i_2}\}$. Assume that the robot moves from the current state i_1 whose activity is $s^r_{i_1}$ to a target state i_2 with an activity $s^d_{i_2}$. The input to the k-th neuron in the action layer is computed as:

$$I_k = w_{ki_1i_2} s^r_{i_1} s^d_{i_2} \tag{4}$$

where k represents the number of neurons in the action layer.

Training the Sigma-Pi network requires the robot to execute every state transition at least once beforehand so that the executed action during each transition can be stored. For this, we need to drive the robot to explore the traversable areas safely during or after map building. When the robot moves around and its spatial representation in the PC-GWR changes from i_1 to i_2, its moving direction during this transition is indicated by the HD-GWR network whose activity will be associated with this state transition. During future navigation, whenever the same transition is needed, i.e., given the same current state and next desired state, this action will be recalled to support this movement. The connection weights $\{w_{ki_1i_2}\}$ are trained as follows:

(1) In the PC-GWR, when the robot moves straight from node i_1 to its neighboring node i_2 with its orientation pointing forward, record the current HD-GWR activity as ζ_{HD} and find the most active node n_p in the HD-GWR network.

(2) According to ζ_{HD} and n_p, create a circular bump of activation on the HD-GWR with the same size and bump direction, except that this new bump has a circular normal distribution starting from the peak node n_p. The robot's current direction is then represented by a population code of all the active neurons in this bump, rather than by a single neuron n_p. Since the HD cell network might be not trained to cover all the states in the environment, this ensemble representation can mitigate the systematic error from it. This also enables a more efficient learning in the action learning phase where all these neurons' activities contribute to the action representation. The circular bump presents a firing pattern symmetrical around the node n_p and the activity of the k-th neuron in the activation bump coding for the current orientation (represented by node n_p) is calculated as follows:

$$p_k = \frac{e^{Kcos(\frac{k \cdot 10 \cdot \pi}{180} - \frac{n_p \cdot 10 \cdot \pi}{180})}}{2\pi J_0(k)} \tag{5}$$

where K is a constant and $J_0(k)$ is the modified Bessel function of order 0:

$$J_0(k) = \frac{1}{\pi} \int_0^{\pi} e^{kcos(\theta)} d\theta \tag{6}$$

(3) The second order weights $w_{ki_1i_2}$ are learned using the current bump activity p_k which is associated with the current state transition:

$$\delta w_{ki_1i_2} = \eta (p_k - w_{ki_1i_2}) \tag{7}$$

where η is a constant learning rate. This results in the weights $w_{ki_1i_2}$ trained to encode a robot's heading direction during state transitions, which then compute the input I_k to the NDF network. Since the rebuilt bump activation is symmetrical to the node representing the robot's current orientation, this guarantees that the bump generated on the DNF also points to the direction denoted by the same node.

2) DNF network: The dynamical neural field (DNF) network is a model inspired by the neural dynamics in the cortical tissues [24] and has been proven an efficient method to generate dynamical behaviors in robotics [25], [26]. For input with a distributed representation, it is able to generate stable output signals by fusing relevant information and canceling noise, which is important for noisy input or in real-world conditions. In this work, a DNF is used to integrate action coding from the Sigma-Pi network and to produce the suggested moving direction for the next step. It instantiates the suggested orientation in a similar way as the HD-GWR network represents the robot's orientation. The DNF is arranged in a one-dimensional ring-form with k=36 neurons, which is the same size as the HD-GWR. In the DNF, each neuron has a membrane potential representing its activity and lateral connections with neighboring neurons. Given stimuli from the Sigma-Pi network, the DNF can stabilize itself by interacting with neighboring neurons and generates an activation bump that denotes the next desired moving direction. The bump's direction depends on the recently received input signals.

For each neuron k in the DNF, its membrane potential μ_k, given fixed lateral connections n_{kj} with a neighbour neuron j, is updated as follows [27]:

$$\tau \Delta \mu_k = -\mu_k + \sum_{j=1}^{k} n_{kj} f(\mu_j) + I_k + h \tag{8}$$

where h is a rest potential, τ represents the decay rate of the neural activity, and I_k denotes the neuron's input stimulus from the Sigma-Pi network. The network implements local cooperation between neighboring neurons k and j and far-range competition. To this aim, lateral connection weights n_{kj} are modeled by a Gaussian-function with negative offset as follows:

$$n_{kj} = \beta e^{-\frac{(k-j)^2}{2\sigma^2}} - c \tag{9}$$

where β is a constant scaling factor, σ^2 determines the width of local cooperation, k, j index the positions of neurons and c is a constant value. The function $f(\mu)$ is a sigmoid transfer function of a single neuron with a constant offset g:

$$f(\mu_k) = \frac{1}{1 + e^{-(\mu_k - g)}} \tag{10}$$

Here the activity bump of the DNF represents the suggested direction for reaching the next state. The way the DNF represents the direction is similar to the way the HD-GWR represents the direction, which enables the robot to reach the desired direction by minimizing the discrepancy between their firing patterns.

3) Rotational action network: Now, the robot's current orientation is represented by the HD-GWR, while its next desired heading direction is represented by the DNF. Since the HD-GWR and the DNF represent the direction in a similar way, for a movement, a robot just needs to turn into a direction so that the HD-GWR has a similar firing pattern as the DNF, especially, the bumps of the HD-GWR and the DNF should point to the same direction.

For this aim, we trained an MLP network, as shown in Figure 2, which generates an appropriate turning command. It inputs the robot's current direction (coming from the HD-GWR network) together with the desired direction (coming from the DNF) and outputs an angle the robot should rotate in order to reach the desired orientation from the current orientation. These two activation bumps representing the robot's current and next desired directions, respectively, have the same size and activation coding. We train this network as a regression model and the output angle could be positive, negative or zero representing turning right, turning left or not turning. For training this network, we point the robot to a random direction

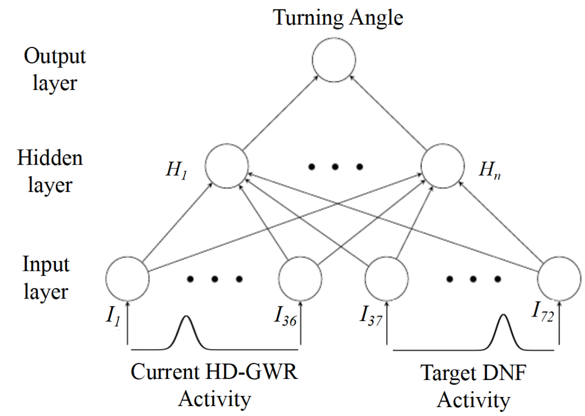


Fig. 2: Illustration of the MLP network which provides the robot's turning commands. The input consists of the robot's current HD-GWR activity and target DNF activity. The output is an angle value representing turning left, keeping still or turning right.

θ_1 and record its current HD-GWR activity, and then calculate a DNF bump representing the desired direction θ_2 according to previous sections. These two bump activations are fed as input to train the MLP to produce the required turn angle, which for training is completed geometrically. This is the only supervised learning component in the model.

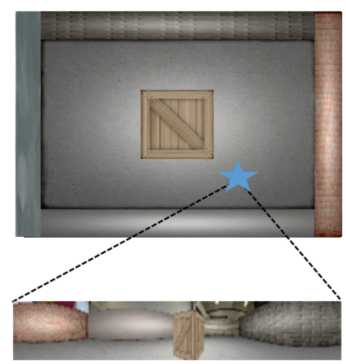
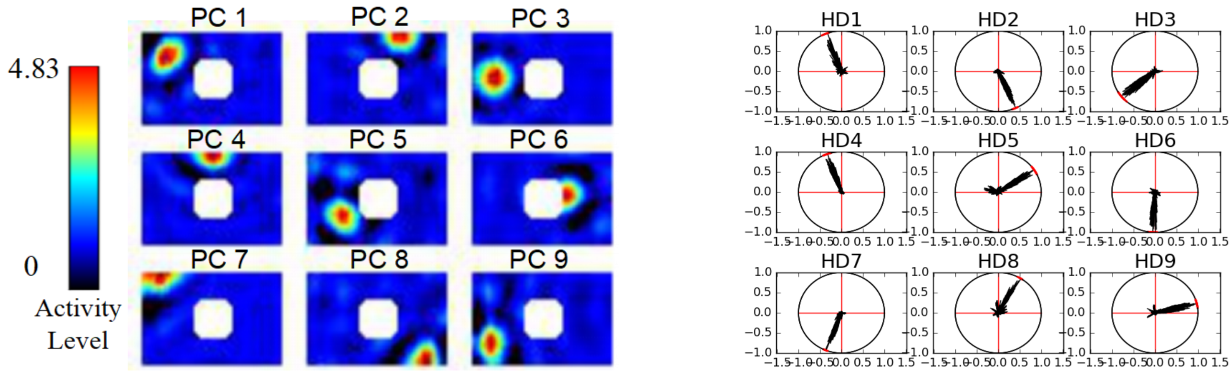


Fig. 3: An overhead view of the rendered RatLab environment. Below is an image seen by the robot at the current position (indicated by the star).

During movement, whenever a direction adjustment is needed, the robot can quickly calculate an appropriate turning command by inputting the two bumps of activities.

III. Experimental Results

To evaluate the proposed model, we first need to obtain the place and HD cells that our system builds upon. We did this in a virtual-reality environment, RatLab [28], which provides an easy way to generate data for SFA training. RatLab was originally designed for experiments on learning place and HD cells from visual images. It simulates a rat exploring an environment and offers a set of choices on environmental varieties and movement patterns. It is also a good candidate



(a) (b)

Fig. 4: Firing patterns of learned place and HD cells to different positions or directions. (a) Spatial firing patterns of 9 representative place cells. The activation level of the firing activity is obtained by averaging over all orientations and represented by the color. The white square represents an obstacle in the environment. (b) Polar plots showing the orientation firing patterns of 9 representative HD cells, obtained by averaging over all positions.

to replicate experiments in robotic scenarios. Figure 3 shows a screenshot of the environment constructed in this work.

A. Modelling Place and HD Cells

For modeling place and HD cells, we developed separate cell types simultaneously from one exploration, through restricting the SFA learning to different movement phases of the spatial exploration. In this work, we train 30 place cells and 50 HD cells whose ensemble activity encodes the spatial position and direction respectively. We use more HD cells in an attempt to represent the direction more precisely since the direction is more important during movement. Figure 4(a) shows that the learned place cells only fire at certain positions in the environment and different cells fire at different positions. Figure 4(b) shows that the firing activity of learned HD cells becomes significantly active only when its head direction points to a certain direction.

B. Spatial Learning with Hippocampal Cells

At the beginning of the map building, a PC-GWR network starts with two random neurons. When driven to explore the traversable area, where each position is encoded by an ensemble activity of learned place cells, the activities along the trajectory are fed to the PC-GWR. The PC-GWR grows automatically in order to sufficiently represent the input space, until the exploration ends. Through learning the internal relationships of these place cell activities, the resultant PC-GWR network gives rise to a topological map of the explored area (see Figure 5).

Learning from HD cell activities results in a ring-shape HD-GWR network, as shown in Figure 6. Its ring topology accords with a circular orientation space. As we can see, the robot's current direction is represented by the most active neuron among the 36 HD-GWR neurons.

C. Goal-directed Navigation

As shown in Figure 7, the robot can localize itself by selecting the best matching neuron (the blue neuron) in the PC-GWR network. For a given goal (the red neuron), a reward signal spreads from the goal node towards neighboring nodes until covering the whole map. The gray-scaled brightness indicates the value of the reward. The brighter the color is, the higher the reward. Based on the current position representation and reward signals, the robot can calculate the next desired state by finding the neighboring node with the highest activity. The current and next desired states then coactivate the Sigma-Pi network to produce an action code for the DNF, which generates the desired moving direction. As shown in Figure 7, the red circle surrounding the robot represents the neuron activities of the DNF, where its activation bump denotes the desired orientation. Upon reaching the next state, the robot's current state representation in the PC-GWR and its next desired state will change accordingly. Its moving direction also changes for the next state transition. This process repeats until the robot reaches the target position.

Note that the robot's trajectory does not exactly overlap with the connections in the map and this discrepancy sometimes may lead to a failure (not shown). This is due to the fact that each neuron in the PC-GWR represents an area rather than a single position. The robot will consider itself already reaching a position even though it just starts entering this area and neglects that there is still a distance between the robot's position and the center of the area. Then the robot starts changing its direction to execute the next state transition. This also explains why the final position of the robot does not overlap with the goal position in the PC-GWR (shown in Figure 7(c)). A distributed representation [30] could mitigate this effect.

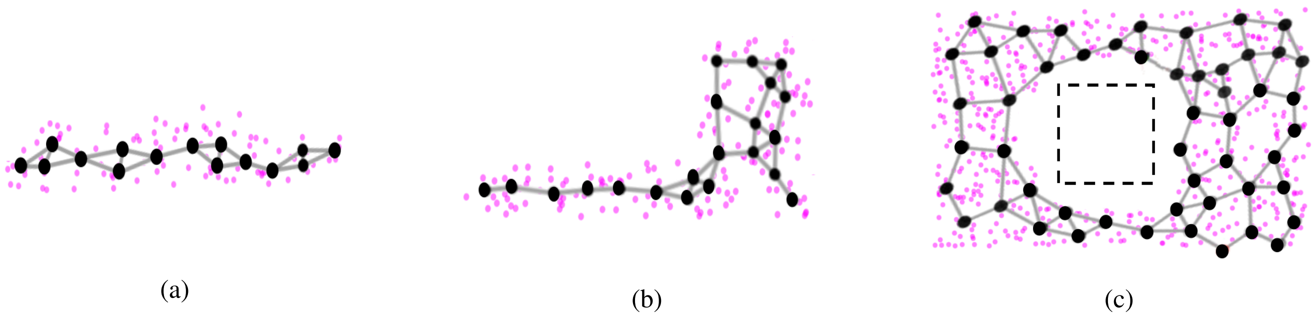


Fig. 5: PC-GWR learning. The PC-GWR starts with two random neurons and grows (a) and updates its neurons and connections (b) based on received place cell activities when a robot explores the environment, resulting in a topological map of the explored area (c). The pink dots represent the input space of place cell activities over the area that the robot has experienced so far. Since the place cell activity is high-dimensional, the map and dots are projected for visualization into a 2D space via multidimensional scaling (MDS) [29].

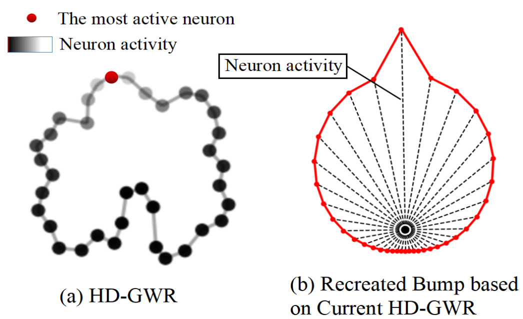


Fig. 6: (a) HD-GWR learning result (visualised by MDS) and (b) the bump created based on its current direction representation. The robot's current direction is represented by the most active neuron (the red node) in the HD-GWR network. (b) A bump (the red circle) is created based on the current HD-GWR network according to equation 5 and the distance from its center to each neuron represents the neuron activity (schematic drawing).

IV. Discussion

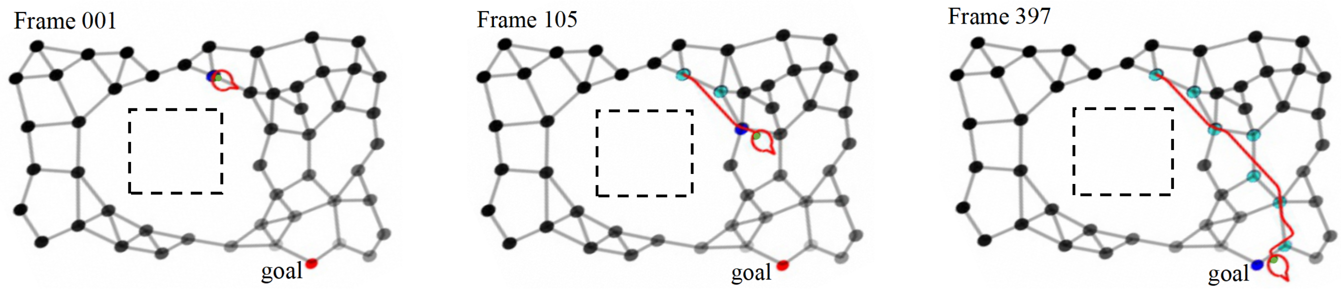
Our navigation system is built on top of place and HD cells which are modeled by SFA. Although an SFA model is able to explain the self-organization of certain hippocampal cell types, it is just a feed-forward memoryless model and contains no recurrent connectivity or path integration mechanisms that are usual in theoretical hippocampal modeling. This means that such a simple mechanism can certainly not replicate all the characteristics of hippocampal firing activities, but it is able to explain the emergence of certain hippocampal cell types through learning directly from visual stimuli. From a practical perspective, this simplicity makes it a convenient model to generate feature responses akin to those of hippocampal cell types in a robotic context. The PC-GWR map is able to capture the topology of an explored area, but it does not record what a robot has seen during exploration. In future, we plan to train a more cognitive map by integrating semantic information to the learned spatial representation, for example, by assigning each node of the PC-GWR with the information of what the robot can see at that position.

A robot could realize the proposed navigation strategy using its own onboard camera, without the necessity of resorting to external sensors, which makes the system easy to be validated on a robot platform. However, this simplicity also brings challenges to the accuracy and robustness of the system. We can improve our system's performance from several aspects. For example, instead of representing the robot's position using a winner-take-all node in the PC-GWR, we can improve this representation by adopting a distribution of related node activations. Another possibility is to resort to other sensors, like depth camera, Laser, etc., to compensate the limits of a navigation system relying only on its own camera. It would be helpful to improve the completeness of our navigation system by fusing different sensory information.

In this work, we use a simulated robot with a field of view (FOV) of about 320° simulating a rat's FOV [32]. Although small FOVs down to 60° produce the same place and HD cells [31], we need to test whether this still works on a real robot considering the complexities of the real world such as changing lighting conditions and noisy sensory information. Besides, we need to implement our approach on a real robot platform and qualitatively evaluate the navigation performance, for example, to test the accuracy of localization and orientation representation in the GWR network.

V. Conclusion

We have presented a neural network architecture that navigates a simulated robot to a desired position based on modeled place and HD cells. The architecture includes modeling hippocampal cells from visual inputs, map learning through self-organization, neural planning and action control. The positional and directional information is represented by the ensemble activity of place and HD cells respectively, which are modeled via unsupervised learning from visual image directly. A behavior-imitation mechanism is used to learn



(a) (b) (c)

Fig. 7: Robot navigates to a desired goal. The robot's current real position is represented by the small green node and its representation in the map is represented by the dark blue neuron. The desired goal is shown in red and the neuron brightness indicates the reward signal. The robot reaches the goal by continously moving to the neighboring neuron that has the highest signal activity. The bump of the red circle surrounding the robot indicates the robot's next desired direction. The red solid line represents the robot's moving trajectory and the color of passed neurons is changed into cyan.

actions supporting state transitions from a previous phase where an experimenter drives a robot to complete the same transitions successfully. The proposed neural system enables a robot to reach the desired position smoothly and effectively by using its own camera only, without the need for external sensors.

References

[1] S. Thrun and J. J. Leonard, "Simultaneous localization and mapping," In Springer handbook of robotics, Springer Berlin Heidelberg, pp. 871-889, 2008.

[2] G. Silveira, E. Malis, and P. Rives, "An efficient direct approach to visual SLAM," IEEE Transactions on Robotics, 24(5)," pp. 969-979, 2008.

[3] C. Eden, "Cognitive mapping," European Journal of Operational Research, 36(1), pp. 1-13, 1988.

[4] E. C. Tolman and C. Edward, "Cognitive maps in rats and men," Psychological Review, 55(4), pp. 189-208, 1948.

[5] S. Vasudevan, S. Gchter, V. Nguyen, and R. Siegwart, "Cognitive maps for mobile robots–an object based approach," Robotics and Autonomous Systems, 55(5), pp. 359-371, 2007.

[6] J. O'Keefe and L. Nadel, "The hippocampus as a cognitive map," Oxford: Clarendon Press, 1978.

[7] J. S. Taube, R. U. Muller and J. B. Ranck, "Head-direction cells recorded from the postsubiculum in freely moving rats. I. Description and quantitative analysis," Journal of Neuroscience, 10(2), pp. 420-435, 1990.

[8] J. O'Keefe and N. Burgess, "Geometric determinants of the place fields of hippocampal neurons," Nature, 381(6581), pp. 425-428, 1996.

[9] E. I. Moser, E. Kropff, and M. B. Moser, "Place cells, grid cells, and the brain's spatial representation system," Annual Review of Neuroscience, 31, pp. 69-89, 2008.

[10] T. Hartley, N. Burgess, C. Lever, F. Cacucci, and J. O'Keefe, "Modeling place fields in terms of the cortical inputs to the hippocampus," Hippocampus, 10(4), pp. 369-379, 2000.

[11] N. Cuperlier, M. Quoy and P. Gaussier, "Neurobiologically inspired mobile robot navigation and planning," Frontiers in Neurorobotics, 2007.

[12] L. Wiskott and T. J. Sejnowski, "Slow feature analysis: Unsupervised learning of invariances," Neural Computation, 14(4), pp.715-770, 2002.

[13] F. Schönfeld and L. Wiskott, "Modeling place field activity with hierarchical slow feature analysis," Frontiers in Computational Neuroscience, 2015.

[14] D. Sheynikhovich, R. Chavarriaga, T. Strsslin, and Gerstner, "Spatial representation and navigation in a bio-inspired robot," In Biomimetic Neural Learning for Intelligent Robots, Springer, Berlin, Heidelberg, pp. 245-264, 2005.

[15] A. Arleo and W. Gerstner, "Spatial cognition and neuro-mimetic navigation: a model of hippocampal place cell activity," Biological Cybernetics, 83(3), pp. 287-299, 2000.

[16] C. Weber and J. Triesch, "From exploration to planning," In International Conference on Artificial Neural Networks (ICANN 2008), pp.740-749, 2008.

[17] M. Witkowski, "An action-selection calculus," Adaptive Behavior, 15(1), pp.73-97, 2007.

[18] C. Giovannangeli and P. Gaussier, "Autonomous vision-based navigation: Goal-oriented action planning by transient states prediction, cognitive map building, and sensory-motor learning," In IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2008, pp. 676-683, 2008.

[19] W. Yan, C. Weber, and S. Wermter, "A neural approach for robot navigation based on cognitive map learning," In International Joint Conference on Neural Networks, IJCNN 2012, pp. 1-8, 2012.

[20] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, F. F. Lei. and A. Farhadi, "Target-driven visual navigation in indoor scenes using deep reinforcement learning," In IEEE International Conference on Robotics and Automation (ICRA 2017), pp. 3357-3364, 2017.

[21] X. Zhou, C. Weber, and S. Wermter, "Robot Localization and Orientation Detection Based on Place Cells and Head-Direction Cells," In International Conference on Artificial Neural Networks (ICANN 2017), pp. 137-145, 2017.

[22] S. Marsland, J. Shapiro, and U. Nehmzow, "A self-organising network that grows when required," Neural networks, 15(8), pp.1041-1058, 2002

[23] C. Weber and S. Wermter, "A self-organizing map of sigma-pi units," Neurocomputing, 70(13), pp.2552-2560, 2007.

[24] S. I. Amari, "Dynamics of pattern formation in lateral-inhibition type neural fields," Biological Cybernetics, 27(2), pp. 77-87,1977.

[25] W. Erlhagen and E. Bicho, "The dynamic neural field approach to cognitive robotics," Journal of Neural Engineering, 3(3), pp. 36-54, 2006.

[26] M. Toussaint, "A sensorimotor map: Modulating lateral interactions for anticipation and planning," Neural Computation, 18(5), pp. 1132-1155, 2006.

[27] N. Cuperlier, M. Quoy, P. Laroque and P. Gaussier, "Transition cells and neural fields for navigation and planning," Mechanisms, Symbols, and Models Underlying Cognition, pp.147-152, 2005.

[28] F. Schönfeld and L. Wiskott, "RatLab: an easy to use tool for place code simulations," Frontiers in Computational Neuroscience, 7. 2013.

[29] I. Borg and P. J. Groenen, "Modern multidimensional scaling: Theory and applications," Springer Science & Business Media, 2005.

[30] W. Yan, C. Weber. and S. Wermter, "Learning indoor robot navigation using visual and sensorimotor map information," Frontiers in Neurorobotics, 7, 2013.

[31] M. Franzius, H. Sprekeler and L. Wiskott, "Slowness and sparseness lead to place, head-direction, and spatial-view cells," PLoS Computational Biology, 3(8), pp. 1605-1622, 2007.

[32] A. Hughes, "A schematic eye for the rat," Vision Research, 19(5), pp. 569-588, 1979.