Development of Spatial Cognition through Visuomotor Integration in Hierarchical Recurrent Neural Networks

Wataru Noguchi

Graduate School of Information Science and Technology

Hokkaido University

July, 2019

Acknowledgement

Foremost, I would like to express my gratitude to my advisers Prof. Masahito Yamamoto and Associate Prof. Hiroyuki Iizuka for their polite instruction and guidance, and for providing me the freedom and opportunities to work on a variety of problems. Their insightful comments and discussions with them presented me new ideas and perspectives, and encouraged me to complete research on this thesis.

I would also like to thank the committee of this thesis: Prof. Masahito Kurihara, Prof. Tetsuo Ono and Prof. Hidenori Kawamura for their insightful critiques and suggestions. Based on their comments, the description and organization of this thesis was refined for clearly showing contribution of this work.

I also thank all members of my research group, Autonomous Systems Engineering Laboratory. Working and discussing with them encouraged me.

Finally, I would like to thank to my family. I would not have continued my research without their supports.

This work was supported by JSPS KAKENHI Grant Number JP18J20404.

Contents

Chapter 1

Introduction

1.1 Spatial Cognition

Spatial Cognition in Animals

How can we acquire the concept of the space? The ability to recognize spatial position is an essential aspect of cognition for animals to live in the world. Animals can effectively explore the environment for foods and perform homing behavior by recognizing their spatial position in the environment. Without such navigation ability by the spatial recognition, animals cannot survive in the nature.

Almost all the existent animals have some kind of spatial recognition ability; however, the styles of the spatial recognition vary between species. *C. elegans* shows chemotactic or thermotactic behavior for locating foods or danger by using gradients of chemical density or temperature; it is a simple spatial recognition mechanism, which can be described as a stimulus-response model. Some kind of desert ants and bees have a path-integration ability which is an ability to tracking their position by internally integrating their moving speed and direction, and can return home along straight routes [MW88]. The path-integration requires memory and it is not just stimulus-response mechanism. Mammalian including we human, which has more sophisticated nervous systems, has more sophisticated spatial cognition abilities; they have internal representation of the spatial structure of external environment and use the representation for localizing themselves in the space.

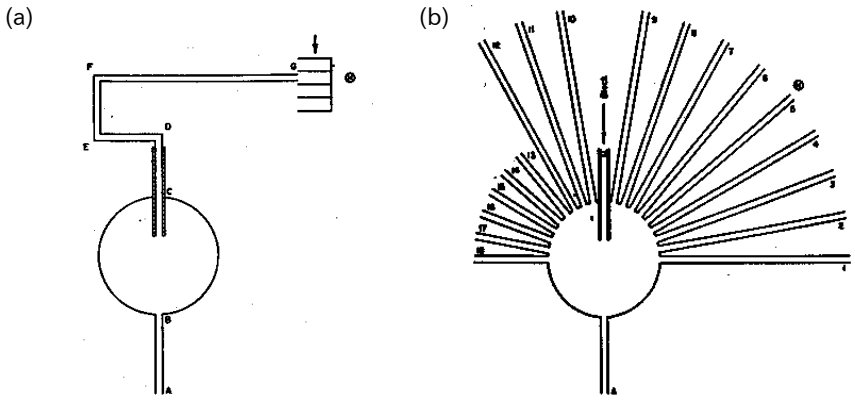

Figure 1.1: The Tolman's maze used in [Tol48]

The Cognitive Map

Tolman showed that rats have internal (or mental) representation like a map in their brain by performing an experiment of maze navigation in rats [Tol48]. In Tolman's experiment, rats explored a maze and learned the position of a food (Fig. 1.1 (a)). After learning the maze for reaching the food place, the structure of the maze was changed (Fig. 1.1 (b)). In the maze with novel structure, the rats go straight to the food place with newly appeared path in stead of taking the path learned during exploration of the previous maze. Because the selected path which straightly lead to the food place was novel for the rats, such shortcut behavior cannot be performed by just remembering previously performed actions. This experiment shows that the rats do not perform navigation by just reacting stimulus and there is some internal process like planning in their brain. Specifically, considering the fact that the rats solved the spatial navigation task, it is considered that the rats have an internal representation of a spatial structure of the external environment in their brain. Tolman called the internal representation of the space a cognitive map.

It is considered that there are two major abilities of the spatial cognition which was shown by the rats in the Tolman's experiment and we describe the details of these abilities as below. One ability is the recognition of the spatial structure of the space. The space can be generally considered as an Euclidean space which has three-dimensional structure and each object has its coordinate as an property in the three-dimensional space. By recognizing the three-dimensional structure of the space, the rats could recognize the spatial relationship between objects in their spatial coordinates. The rats also can recognize

how its spatial position changed by their self-motion, and they can recognize their spatial position even after passing through novel routes. The other ability is navigating from one place to another place by considering the spatial relationship between them. This navigation ability is active ability rather than passive ability in the sense that it requires planning how to move to the goal place in advance. If animals did not have planning ability, they cannot perform the spatial navigation even if they can recognize the spatial structure of the space. Above two abilities can be considered as parts of animals' general cognitive abilities. The recognition of the spatial structure is one of the abilities to recognize the abstract concept and the planning of the navigation using the spatial recognition is a part of general planning abilities. The planning using abstract concepts is necessary ability to highly adaptive behavior for living in the natural environment. Thus, studying the spatial cognition is one of the study for understanding general cognition.

As described above, studying the spatial cognition is necessary for understanding animals' remarkable abilities to live in the environment. Especially, because direct evidences to support the hypothesis that animals have an abstract map like representation in their brain have been found, many neuroscience study spatial cognition have been conducted. O'Keefe and Dostrovsky firstly found the hippocampal cells that fire when an animal locates at a specific place, which are called the place cells [OD71]. It is considered that place cells provide animals sense of spatial position and are the instance of the cognitive map in animals. Place cells have been firstly identified in rats. In rats brain, other type of spatial coding cells, e.g., head-direction cells which fire corresponding to animals' head direction (HD) [TMR90] and grid cells which fire at specific locations arranged in hexagonal grids on the space [FMW+04] have been found. HD cells and grid cells can track animals' head direction and spatial position from animals' movement signals even in dark environments, and it is considered that HD and grid cells comprise the path-integration ability [MBJ+06]. There are also brain cells that code information necessary for navigation behavior, e.g., distances from the goal and direction to the navigation goal [SFLU17]. These cells might provide necessary information for animals to perform navigation behavior as shown by the rats in Tolman's experiment.

Development of Spatial Cognition

The developmental process of spatial cognition have been studied in neuroscience. The spatial selective activities like place cells have been observed in rat at the first exploration of the external environment outside of its nest [WCBO10]. However, such spatial activities cannot work for spatial recognition because these are not associated with the external environment. In fact, young rats without enough spatial experiences failed to navigate goals without any cues, however, through their development, the rats became able to navigate to goals by using the internal spatial recognition without any visible cues [Sch85]. The spatial activities of the place cells became stable and robust along with the development of the spatial navigation ability. This observation suggest that rats develop the association between internal representation and external environment through experiences. Further, it was found that the place cell activities change depending on the environments of tasks [MRM15]. Thus, it is considered that the spatial recognition of position and its development are deeply combined with animals' experiences.

The development of the spatial cognition have been studied in neuroscience study as described above. These studies can investigate when the spatial cognition related abilities like spatial navigation or the spatial neural activities are developed, in what order these abilities or neural activities are developed, and what relationships exist between the spatial behavior and neuronal activities. However, only such reductionistic approach that tries to infer how the spatial cognition develops by observing the real animals is not sufficient. That is because the spatial cognition is generated by very complex system of the brain and the huge amount of observation is necessary for revealing such a complex system of the spatial cognition.

1.2 Computational Modeling of the Spatial Cognition

There is another approach for understanding the spatial cognition by reproducing the spatial cognition as models on computer simulation instead of by observing the behavior or neuronal activities of real animals. If a part of the spatial cognition was reproduced

in the simulation model, we can understand that the mechanism which was implemented on the simulation model contributes to the spatial cognition. There are some kind of computational model of the spatial cognition in term of the objective of the model.

1.2.1 Modeling the Spatial Cognitive Function

The spatial cognition is realized by specific kinds of neurons that activate depending on individual's states or activities related to space. Thus, modeling the such spatial activities of the neurons might provide useful insights about how the spatial cognition works.

The place cells and grid cells have special characteristic in their temporal activities such that these cells code the detailed place in their place or grid field by using phase in theta rhythm of the local potential fields, and some models have been proposed to explain how such characteristics are generated [BBO07]. These models aimed to model the spatial neuronal activities even in their spike characteristic. Sometimes, the grid cells or HD cells are considered as attractor networks for robustly maintaining the spatial position or direction and the models with attractor dynamics were proposed [MBJ$^+$06]. The network models by McNaughton et al. developed the grid-cell like activities by learning, however, the grid patterns were taught by another network as a tutor and the grid-like patterns was not associated with external environment.

Although above mentioned models that reproduce the activities of the spatial neurons could explain how the spatial activities are generated, these models did not put importance on explaining how the spatial activities are developed through experiences and were explicitly designed to produce spatial dependent activities. Thus, above models are not sufficient to understand the development of the spatial cognition.

1.2.2 Modeling the Development of the Spatial Cognition

Considering the fact that the spatial cognition is not innate ability and is developed through the experiences, developmental models are necessary for understanding the spatial cognition. There are also models for explaining the development of the spatial cognition. For explaining the development of the spatial cognition, the model should not be designed to develop the target spatial abilities. Concretely, for example, for investigating the

development of the place cells, the model should not be directly designed to generate place cells' activities and the place cells inputs or information equivalent to the place cells should not be provided to model.

Aota et al. proposed a neural network model that developed the place cell like activities by integrating visual information using self-organizing map and Hebbian learning over simplified visual experiences in very simplified environment [AMU99]. Stachenfeld et al. showed that a model of development of the grid cell-like representation by a model that learns predictive representation of the reward over the environment, which are called a successor representation [SBG17]. These studies indicate that the spatial representation can be developed by integrating information like vision or reward associated with the spatial position using temporal continuity.

Wyss et al. and Franzius et al. simulated the development of place or grid cells like neural activities from more realistic subjective visual experiences, i.e., visual image [WKV06, FSW07]. They used neural network models with hierarchical structure like visual cortex and the place cells or the grid cells are extracted as slowly changing feature of visual experiences. Their models could simulate the development the spatial representation in a similar way to real animals in the sense that these models use high-dimensional visual inputs which are closer to real animals than inputs of other model used. However, their models only passively received visual inputs and did not generated spatial movement by themselves, and these models cannot recognize the spatial relationships between places, namely, spatial structure, and cannot perform the spatial navigation.

Recently developed deep learning approach can realize the simulation where the recognizing environmental states and performing spatial navigation are jointly developed. Deep neural networks can acquire various abilities through the learning without any preconfigured functions and jointly develop the abilities of recognition of the environment and performing task-oriented behaviors in an end-to-end manner [LBH15]. Spatial navigation model by deep neural networks were also proposed [MBM$^+$16]. For the spatial navigation task, convolutional neural network for recognizing high-dimensional vision and recurrent neural network for memorize the past exploration are often used and these modules enable the model to effectively explore the environment and achieve goals. Most of such deep learning models were not constructed for the purpose of investigating how the spatial

cognition is developed. Although these models become able to solve the navigation task through only learning, they cannot perform the animal-like spatial navigation behavior, namely, short-cut behavior like rats in the Tolman's experiment. That is because these models did not develop the recognition of the spatial structure. On the other hand, there are studies that show that the recognition of spatial structure by using self-organized grid cell-like representation can be developed through the learning of the path-integration task on the deep neural network models [BBU+18, CW18]. Especially, Banino et al. constructed deep neural network model that perform spatial navigation using the developed grid-like representation including short-cut behavior. In the case of their models, the developed grid-like representation was used for recognizing the spatial structure of the environment. The spatial relationships between places were not explicitly provided by the inputs and the spatial navigation behavior was not taught by the experimenter but learned through only experiences. It means that their network develop the recognition of grid-like representation, the recognition of the spatial structure, and learned to use it for effectively navigating in the space through their experiences. However, the spatial position and direction as the place cell and HD cell were given to the model during the development. On the other hand, in the real environment, the spatial position or direction often cannot be directly identified from sensory observation like vision. Thus, their model cannot explain how animals could develop the recognition of the spatial structure from only their experiences like vision.

1.2.3 The Development of Spatial Cognition through Only Visuomotor Experiences

As described above, many computational model for explaining the development of the spatial cognition. However, there was no model that can develop the spatial cognition like rats in the Tolman's experiment from experiences similar to real animals, e.g., vision and motion. The model by Banino et al. can develop the grid cell-like representation through the learning of path-integration learning and the model could recognize the spatial position even in an un-experienced place; however, the model was trained on the place cell inputs, which provide spatial position explicitly. Because, in general, animals cannot

get sensory information that explicitly represent the spatial position, it is considered that the model by Banino et al. used information which is not available for animals originally. For understanding the development of the spatial cognition, a model that can develop the recognition of the spatial position and spatial structure through experiences is required. The problem is how to develop the recognition of the spatial position through only experiences which animals can sense and without explicit information of the spatial position.

Emergence of high-level cognition through experiences

One of reasons that many computational model of the development of the spatial cognition are explicitly designed for developing the spatial cognition by providing direct learning method of the spatial structure or some prior knowledge about the space like spatial position is that the spatial cognition seems to be too sophisticated to develop through only experiences. Actually, the sensory inputs themselves are not sufficient for developing the sophisticated cognitive abilities, however, the learning of the experiences could cause the emergence of the high-level cognition even if the learning mechanism was not designed for the development of the cognition. On the studies of optical illusion, which is one of the cognitive phenomena, a kind of optical illusion was emerged through learning of visual experiences using a deep neural network model [WKS+18]. In their study, a deep predictive coding network was trained on huge amount of video to just predict next frame of the video and the network perceived a illusionary motion in illusionary visual images. This is one of examples that cognitive ability or phenomena that cannot be predicted from experiences themselves could be emerged through only experiences, and it along with the idea that sophisticated spatial cognition shown by animals could be emerged through only experiences.

Spatial cognition through visuomotor experiences

As described above, we considered that the spatial cognition could be developed though only the leaning of experiences. It should be noted that the spatial cognition depends on form of the experiences. For example, although the models by Wyss et al. and Franzius et al. could develop the representation of place or direction considered only visual

experiences, these models cannot recognize the spatial structure, and consequently, cannot recognize spatial position in un-experienced place. That is because it is impossible to apply spatial position to a novel vision at novel (un-experienced) place. It means that the spatial recognition ability that works in novel place was not realized by only visual experiences. On the other hand, the model by Banino et al. [BBU+18] used motor sensory inputs for tracking spatial position and direction and can recognize spatial position even in novel place: their model developed the spatial cognition based on motor sensory experiences. In real rats, it was reported that self-motion is required for stable activity of the place cells [TKL+05]. Then, it is considered that the spatial recognition requires experiences of motion in addition to observation of external environments by vision. Banino et al. provide explicit representation of place to their models. In this study, on the other hand, we considered that, by experiences of visuomotor integration, the spatial cognition like recognition of spatial structure can be developed without providing explicit representation of place.

1.3 Research Objective

In this study, we simulate the development of the spatial cognition through the learning of only experiences like rats. Especially, we consider about the spatial cognition developed through visuomotor integrated experiences similar to rats. The visuomotor experiences themselves do not explicitly show the spatial position or direction, and spatial relationship between places. How the spatial cognition could be developed such visuomotor experiences is investigated. Reproduction of spatial representation in animals' brain like place and grid cells are not our target. Instead, we considered following two abilities of spatial cognition. One ability is the recognition of the spatial structure. The concept of the cognitive map indicates not only recognition of place but also recognition of the spatial relationships between places. By recognizing the spatial structure, it is possible to recognize the spatial position even in firstly visited place. The other ability is the spatial navigation considering the spatial relationship between places. Rats in Tolman's experiment performed short-cut behavior by voluntary selecting novel path to reach food place. It is the ability to voluntary use the recognition of spatial structure for performing navigation. We consider

these abilities are the core of spatial cognition for explaining the sophisticated spatial behavior performed by rats in Tolman's experiment.

The models that develop the spatial cognition are constructed based on deep neural networks. Especially, recurrent neural network (RNN) with hierarchical structure, which we call hierarchical recurrent neural networks (HRNN) structure are used for representing high-level concept of spatial structure. We tried to simulate the development of the spatial cognition by training the hierarchical RNN on the visuomotor experiences. To simulate the development of the spatial cognition in the similar condition to rats, the hierarchical RNN does not receive explicit representation of place like place cells and just receive vision and motion.

The learning of the hierarchical RNN is conducted in a visuomotor integrated way. Different from previous models that used only visual experiences [WKV06, FSW07], our model should consider self-motion. Our model is trained to recognized to relationships between vision and motion: how vision changes corresponding to motion and vice versa. Concretely, our model is trained to predict visuomotor sequences from only visual inputs or from only motion input in addition to from visual and motion inputs. We consider that such visuomotor integration is necessary for developing the recognition of spatial structure.

For developing the spatial navigation ability, the hierarchical RNN is trained to perform spatial navigation. In previous models, the recognition of spatial structure and spatial navigation ability were separately developed or only the recognition of spatial structure was developed. On the other hand, we considered that the spatial navigation not only uses the recognition of spatial structure but also contributes to the development of the recognition of the spatial structure. Then, the developments of the recognition of the spatial structure and the spatial navigation ability were conducted simultaneously in our simulation.

1.4 Outline of the Thesis

The rest of this thesis is organized as follows. In chapter 2, the development of the recognition of the spatial structure is simulated. Firstly, the hierarchical recurrent neural

networks (HRNN) with two levels of RNN is described. Then, the HRNN is trained to predict the visuomotor experiences of simulated mobile agent or real environment. Especially, the visuomotor integration learning on the scheme of prediction was introduced. It will be shown that the representation of the spatial structure is developed in the internal states of the HRNN through the visuomotor integrated learning.

In chapter 3, the development of the spatial navigation considering the spatial relationship is simulated. The navigational HRNN (NHRNN) model for performing the spatial navigation is introduced. The NHRNN is trained to navigate in an open space environment without any obstacles and in a maze like environment with obstacles that changes its structure by replacing obstacles. It will be shown that, through the training of the spatial navigation, the representation of the spatial structure and the spatial navigation ability are developed. Specifically, in the maze like environment, the NHRNN become able to perform short-cut behavior by considering the spatial position of the navigational goal.

In chapter 4, we summarize the thesis, and discuss about what can be explained from the results of our simulations and future perspective of simulation study on development of the spatial cognition.

Chapter 2

Development of the Recognition of the Spatial Structure in Hierarchical Recurrent Neural Networks

2.1 Introduction

Spatial position is an abstract concept that is not explicitly represented in sensory information animals can obtain. However, actually, animals can only obtain subjective sensory information like vision or self-motion. Then, how can recognition of the spatial position and even spatial structure be developed through such subjective experiences? Many studies have modeled the firing patterns of the brain cells related to the spatial recognition like place cells for both theoretical [CKBO13, MKM08, MBJ+06] and practical [JCG15, MWP04] objectives. However, these studies have focused on how positional information is stored and modulated in structured neural network models and how place and grid cells work in our brains. For these purposes, the correct spatial coordinates of current positions are provided while the model learns the spatial structures. Therefore, these models cannot be used to examine how recognition of the spatial structure emerges.

Without using any spatial coordinates of the current positions, Philipona et al. suggested that the neural activities of sensory inputs and proprioception can be used to deduce the dimensions of the spatial perception [PON03]. Terekhov and O'Regan used simple simulations, which did not assume a priori knowledge that space existed, to show

that spatial recognition can be obtained from sensorimotor dependencies [TO16]. The recognition of the spatial structure is stored as pairs of consistent sensorimotor associations. Wyss et al. proposed a neural model with a loss function design and visual cortex-like structure with which a mobile robot perceived successive visual images and learned the smooth and somewhat independent changes in the activities of higher-level neurons [WKV06]. As a result, the model created an internal representation that changed according to the different areas of the learned environment. In this chapter, in accordance with the idea that spatial recognition can be derived from sensorimotor associations without knowledge of the spatial coordinates of the current environment, we investigated how the understanding of spatial structures, such as their spatial coordinates in the environment, was self-organized by sequences of proprioceptive and visual inputs. In contrast to the Wyss model, we used only predictive learning of sensory inputs and motions and did not assume a loss function which evaluates how neurons are activated. Moreover, we investigated how the acquired understanding of the spatial structure was generalized to unknown situations differ from the previous studies.

Tani and Nolfi studied how symbols in the external world and sensorimotor experiences are self-organized in hierarchical neural networks [TN99]. Their model, which consisted of recurrent neural networks (RNNs), was trained to predict the future sensory inputs of a moving robot. Its design was based on the idea that an internal model is required to predict sensorimotor experiences in the external world. Through the learning of the prediction, the internal model is embedded in the dynamics of the RNN, and the mobile robot can predict sensory inputs that correspond to low-level environmental structures, such as local corners or branches, and high-level structures, such as a room consisting of a set of low-level corners or branches. Although the model can extract and integrate sequential patterns of the external environment, it can only memorize the patterns of sequential changes and cannot recognize spatial spread or topology. Yamashita and Tani extended the model by introducing different time scales in the lower and higher levels of the network [YT08]. The model was implemented on a humanoid robot, which was able to smoothly interact with the dynamic environment. However, spatial recognition was not their research focus because the initial position of the robot was always fixed and the model only memorized the sequential patterns in the teaching actions.

In this chapter, we constructed a hierarchical recurrent neural network (HRNN) model that can develop the spatial representation in its internal states through only prediction learning of subjective visuomotor experiences. The HRNN learned two-dimensional spatial structure rather than sequential structure of visuomotor experiences by receiving visuomotor sequences along with two dimensional movement in simulation or real environment. The HRNN was constructed as a deep neural network [LBH15] and can deal with high-dimensional visual input directly; thus, we can simulate the development of recognition of the spatial structure in more similar way to animals like rats than the previous models.

This chapter is organized as follows. In section 2.2, the structure of the HRNN model and how the HRNN learns visuomotor experiences were described. In section 2.3, the HRNN was trained on visuomotor experiences of a simulated mobile agent and we show that the spatial representation like the cognitive map was self-organized through the prediction learning. In section 2.4, the HRNN was trained on visuomotor experiences collected by using a human subject in a real environment and we show that the HRNN even developed the representation of the spatial structure of the real environment. In section 2.5, the HRNN was trained on visuomotor experiences with various complexity of the agent behavior and the effect of behavioral complexity on the development of the spatial representation was investigated. In section 2.6, a different model for simultaneously developing the recognition of place and head-direction based on prediction learning was proposed and it was shown that the developed recognition of place and head-direction could be shared between different environments. In section 2.7, we summarized this chapter and discussed about the contribution of the demonstrated results in the experiments to the development of the recognition of the spatial structure.

2.2 Hierarchical Recurrent Neural Networks

In this section, the details of our proposing HRNN model was described. Previous hierarchical network models [TN99, YT08] were constructed to investigate the self-organization of internal models through predictive learning. Our HRNN model also implemented a hierarchical structure to investigate cognitive maps created from low-level visuomotor ex-

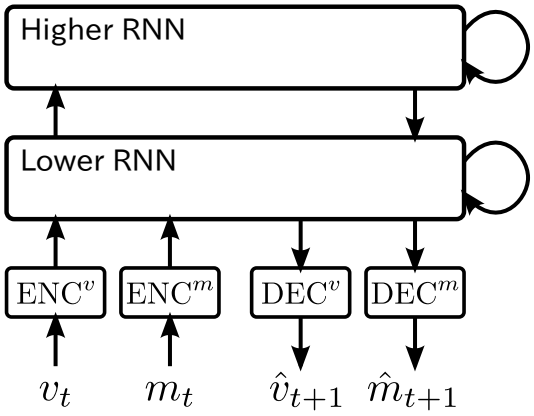

Figure 2.1: The schematic of the hierarchical recurrent neural networks (HRNNs)

periences. Because the cognitive map is a model of the external world, learning should be performed so that the model can predict future input sequences of motion. For such predictions, precise internal models of one's own world and the external world are important. One example of a prediction-based internal model is the forward model [KFS87], which predicts physical body dynamics in environments with estimated parameters. This approach is based on physical dynamics. Another example uses multimodal integration to predict future input sequences based on past sequences that were used as internal models. The advantage of this approach is that it does not directly depend on physical dynamics or advance knowledge of physical properties, as in animals. Thus, we used the multimodal integration approach in this study. In order to simulate the process underlying the creation of cognitive maps, the HRNN ran in the time series of vision and motion and predicted vision $\hat{v}_{t+1}$ and motion $\hat{m}_{t+1}$ while receiving v_t and m_t. The HRNN received only subjective vision and motion, and objective positional information, such as spatial coordinates, was not provided.

2.2.1 Structure of Hierarchical Recurrent Neural Networks

The HRNN mainly consisted of two layers of RNNs: lower-level and higher-level RNNs. Additionally, the HRNN had separate encoding and decoding layers for vision and motion. A schematic of the HRNN is shown in Figure 2.1. The details of the model's structure are described below. When the functions of the lower and higher layers are expressed

as RNN^{lower} and RNN^{higher}, respectively, the equations for the these layers in one-step processing are the following:

$$\boldsymbol{h}_t^{lower} = \text{RNN}^{lower}(\boldsymbol{f}_t^v, \boldsymbol{f}_t^m, \boldsymbol{h}_{t-1}^{higher}, \boldsymbol{h}_{t-1}^{lower}), \tag{2.1}$$

$$\boldsymbol{h}_t^{higher} = \text{RNN}^{higher}(\boldsymbol{h}_{t-1}^{lower}, \boldsymbol{h}_{t-1}^{higher}), \tag{2.2}$$

where $\boldsymbol{h}_t^{lower}$ and $\boldsymbol{h}_t^{higher}$ are the internal states of the lower and higher layers, respectively, and $\boldsymbol{f}_t^v$ and $\boldsymbol{f}_t^m$ are the features of the visual $\boldsymbol{v}_t$ and motion $\boldsymbol{m}_t$ inputs, respectively. The features, $\boldsymbol{f}_t^v$ and $\boldsymbol{f}_t^m$ are calculated as follows:

$$\boldsymbol{f}_t^v = \text{ENC}^v(\boldsymbol{v}_t), \tag{2.3}$$

$$\boldsymbol{f}_t^m = \text{ENC}^m(\boldsymbol{m}_t), \tag{2.4}$$

where ENC^v and ENC^m are non-linear transformation functions by neural networks as visual and motion inputs encoders, respectively. After RNN processing, the visual output $\hat{\boldsymbol{v}}_{t+1}$ and motion output $\hat{\boldsymbol{m}}_{t+1}$ are generated as follows:

$$\hat{\boldsymbol{v}}_{t+1} = \text{DEC}^v(\boldsymbol{h}_t^{lower}), \tag{2.5}$$

$$\hat{\boldsymbol{m}}_{t+1} = \text{DEC}^m(\boldsymbol{h}_t^{lower}), \tag{2.6}$$

where DEC^v and DEC^m are non-linear transformation functions by neural networks as visual and motion inputs predictors, respectively. Summarizing equations (2.1)-(2.6), the overall function of the HRNN is expressed as follows:

$$\hat{\boldsymbol{v}}_{t+1},\ \hat{\boldsymbol{m}}_{t+1} = \text{HRNN}(\boldsymbol{v}_t, \boldsymbol{m}_t, \boldsymbol{h}_{t-1}^{lower}, \boldsymbol{h}_{t-1}^{higher}). \tag{2.7}$$

As expressed in the above equations, the lower-level RNN interact with features of the visual and motion sequences, which dynamically change with every time step. Thus, the lower-level RNN involve dynamic sensory inputs, which indicates that the lower-level RNN focus on short-term dependency rather than long-term dependency. In contrast, the higher-level RNN receive inputs on the internal states of the lower-level RNN. Because the internal states of the lower-level RNN contain the short-term features of sensory inputs, the higher-level RNN can extract more abstract features from these inputs. By focusing on the visual and motion sequences related to spatial movement, the short-term features consist of future visual inputs and movement outputs, while the long-term features consist of input on the location of the agent. We expect that these long-term features, which make up the cognitive map, are formed in higher-level RNNs through the following training.

2.2.2 Visuomotor Prediction Learning

The HRNN was trained to predict future vision and motion through visual and motion sequences. Furthermore, in order to realize multimodal integration of vision and motion in the predictive learning scheme, the HRNN was also trained when either vision or motion information was not provided.

Crossmodal prediction

We recognize how the view changes as a result of our motion and, conversely, recognize how we have moved by the changes in the view. For example, we can imagine the visual flow when we walk with our eyes closed, and we can recognize how a camera moved from the visual flow seen on the monitor.

In addition, the HRNN was trained to predict visual sequences from motion sequences and motion sequences from visual sequences. In this crossmodal prediction task, the HRNN received visual and motion sequence inputs until a certain time step when the vision was shut down. The HRNN was trained to predict both visual and motion sequences from only motion. Predicting motion from vision is performed similarly. The HRNN fills the missing modality by feeding back the predicted output.

To summarize the training procedure, the HRNN learned through three tasks: prediction from both vision and motion (PVM), prediction from only motion (PoM), and prediction from only vision (PoV). The different tasks had different inputs. The PVM task involved both vision and motion, while the PoV and PoM tasks involved either vision or motion only; the missing input was compensated for by its prediction at the previous time step. The visual and motion inputs at each time step are defined as follows:

$$\boldsymbol{m}_t \leftarrow \begin{cases} \hat{\boldsymbol{m}}_t & \text{in case PoV} \\ \boldsymbol{m}_t & \text{otherwise} \end{cases}, \tag{2.8}$$

$$\boldsymbol{v}_t \leftarrow \begin{cases} \hat{\boldsymbol{v}}_t & \text{in case PoM} \\ \boldsymbol{v}_t & \text{otherwise} \end{cases}. \tag{2.9}$$

The $\hat{\boldsymbol{v}}_t$ or $\hat{\boldsymbol{m}}_t$ predicted in the previous time step was used as the input for the missing modality. Schematics of the computational flow in the three tasks are shown in Figure 2.2. The three training tasks were conducted for the same neural network to achieve these objectives at the same time.

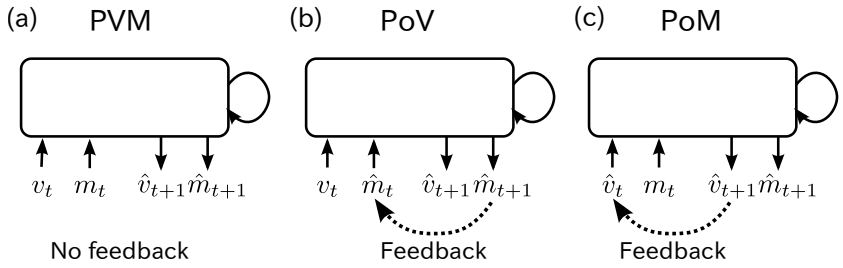

Figure 2.2: The three tasks for the training of the HRNN. (a) The PVM task: Both visual and motion information is provided. (b) PoV and (c) PoM tasks: Either vision or motion is not provided, and the prediction is substituted for the missing modality.

The learning processes were designed with the PoM and PoV crossmodal predictions in order to determine whether the agent was able to form a strong crossmodal association between vision and motion instead of simulating the developmental processes in the brain.

Learning objective

The learning of the HRNN was conducted with the objective of minimizing error between the prediction and target inputs for both vision and motion at each time step. Thus, the error function E, which should be minimized, is defined as follows:

$$E = \sum_{t=1}^{T} \left[E^v(\hat{\boldsymbol{v}}_{t+1}, \boldsymbol{v}_{t+1}) + E^m(\hat{\boldsymbol{m}}_{t+1}, \boldsymbol{m}_{t+1}) \right] \tag{2.10}$$

where T is the length of each visuomotor sequence, and E^v and E^m are the partial error functions for vision and motion, respectively. Through training to minimize the error E, the weights of the connections in the network were optimized for predicting future visual and motion inputs. E was shared between the PVM, PoV, and PoM tasks and the HRNN was trained to minimize the errors in all of the tasks.

The loss function L that the HRNN had to minimize is sum of the errors for three tasks. When the error functions for the PVM, PoV, and PoM tasks are denoted by E^{PVM}, E^{PoV}, and E^{PoM}, respectively, L is formulated as follows:

$$L = E^{PVM} + E^{PoV} + E^{PoM}. \tag{2.11}$$

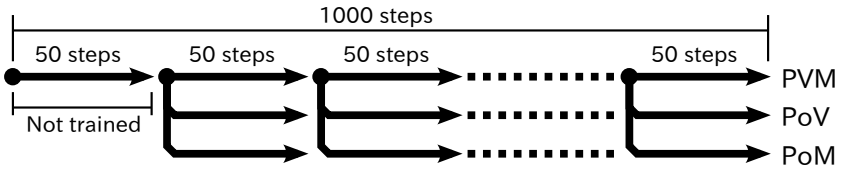

Figure 2.3: The training process for the three tasks, which was done in a single sequence.

Training method

The training of the HRNN was supervised in order to minimize the prediction errors for vision and motion. The backpropagation through time (BPTT) algorithm [RHW86, WZ95] was used to train the HRNN. By using BPTT, the gradient of L through the segment $\nabla_{\boldsymbol{\theta}} L$ is calculated, and the parameters of the HRNN are updated as follows:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \epsilon \nabla_{\boldsymbol{\theta}} L, \tag{2.12}$$

where ϵ is the learning rate. The BPTT training was performed within single small segment as described bellow.

Training schedule

The training for the three tasks (PVM, PoV, and PoM) was conducted as follows. First, the whole sequence was divided into small segments. The visual images and motions were presented to the agent in the first segment. The prediction outputs were not yet evaluated. In the next segment, the prediction outputs for the presentation of the visual images and motions were then evaluated (PVM). Thereafter, the prediction outputs for the presentation of either the visual images or the motions were evaluated (PoV and PoM). In the next PVM, the PoV and PoM conditions started at the end of the previous PVM. Therefore, the final internal states of the previous PVM condition were also the initial states of the three conditions. Training was performed in every segment until the end of the whole sequence. The training process is illustrated in Figure 2.3.

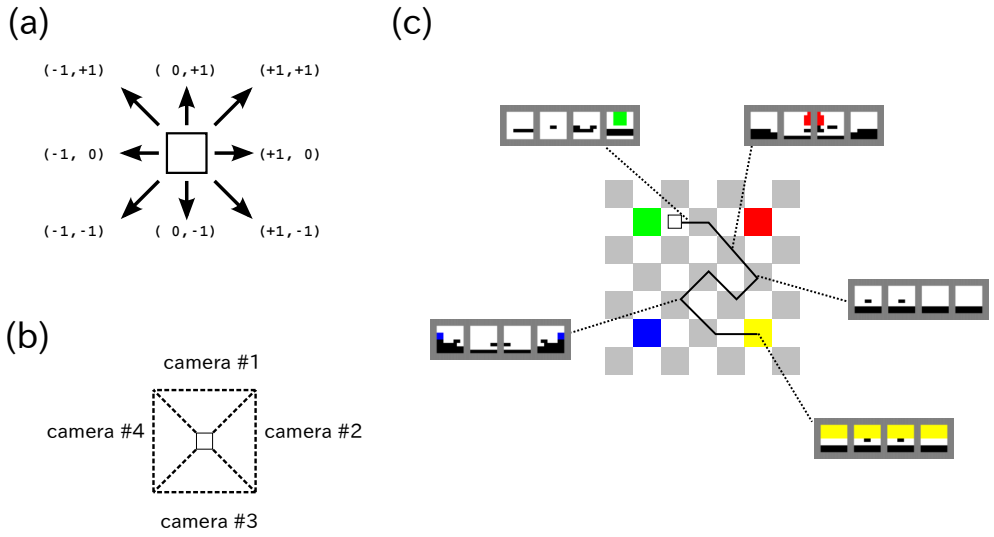

Figure 2.4: The agent that can move a unit distance in eight directions at one time step (a). The agent is equipped with four cameras for an omnidirectional view (b). A sample of motion and visual images obtained by the cameras (c). The floor of the arena had a black-white checkered pattern (the black area is shown in grey for visibility).

2.3 Development of the Recognition of the Spatial Structure through Visuomotor Integration

In this section, the HRNN model was trained on visuomotor sequences by a mobile robot that moved around a flat arena in a simulated environment, and we show that the HRNN can develop the spatial recognition through the prediction learning of subjective visuomotor experiences. Especially, the contribution of the visuomotor integration by the cross-model prediction learning to the development of the spatial recognition was investigated.

2.3.1 Simulation and Training

Simulation environment

In order to collect the visual and motion sequences learned by the HRNN, a mobile robot moved around the simulation environment. The mobile robot was modeled as an agent that can move around a two-dimensional flat arena. The agent was equipped

with omni-wheels and an omnidirectional camera, and it could travel in any direction with an omnidirectional view. The displacement of the agent was determined by two outputs, with one determining north-south displacement and the other determining east-west displacement.

Thus, the motion value at each time step was two-dimensional. The range of displacement values at each time step was $[-1, 1]$. The omnidirectional camera was implemented with four cameras that covered the entire view around the agent. Each camera targeted a different direction: north, south, east, or west. The agent always sensed the omnidirectional visual images that were captured by the four cameras attached to the robot. The size of each visual image was 8×8 pixels, and each pixel in the image had three channels (RGB). Thus, the dimension of the vision input become $4 \times 8 \times 8 \times 3 = 768$.

In the environment where the agent moved, there are four colored landmarks (Fig. 2.4). These landmarks floated like a balloon and were arranged such that they formed a square. The cameras always captured these landmarks above the horizontal line. The four landmarks were placed at $(10, 10)$, $(-10, 10)$, $(10, -10)$, and $(-10, -10)$, with the center of the arena at the origin $(0, 0)$. The distance between the centers of the neighboring landmarks was 20 units. Thus, it took at least 20 time steps to reach one landmark from another. The agent was expected to create a cognitive map of this environment.

Restricted area

In Tolman's experiment [Tol48], a rat learned the spatial environment of a maze in order to obtain a food reward. Even when the structure of the maze was changed, the learned route was no longer available, and the food was placed at the same location, the rat reached the food by passing through a previously unknown shortcut. These results indicated that the rat not only remembered the learned route before the structure of the maze was changed but also recognized the spatial relationships between different places in the maze and the location of the food. In other words, the rat recognized that the unknown shortcut would lead to the known location based on the cognitive map developed in the brain. Therefore, the cognitive map was evaluated by testing whether the HRNN recognized unknown trajectories to a known location.

In order to implement the unknown trajectory, we set a restricted area between red

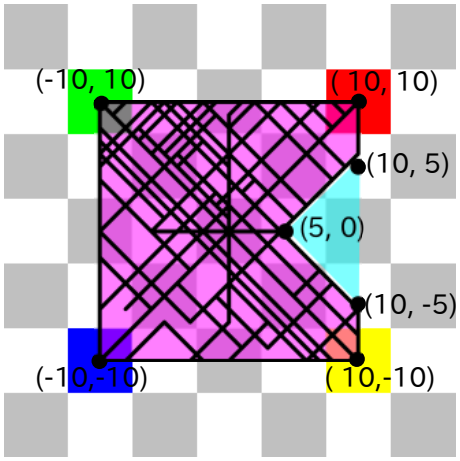

Figure 2.5: An example of the trajectory of the agent's motion in 1,000 steps. The agent moves around the area bounded by the landmarks, which is colored in purple. The light blue area is the restricted area where the agent is not allowed. Because of the restricted area, the agent must make a detour to go to the red landmark from the yellow one and vice versa, and the agent cannot know that the red and yellow landmarks are placed like the green and blue ones.

and yellow landmarks, as shown in Fig. 2.5. The agent was not allowed to go into the restricted area, which was defined by the interior of a triangle with vertices at $(5, 0)$, $(10, 5)$, and $(10, -5)$. Although we did not have a wall or partition marking the restricted area, the agent was controlled so it did not enter the restricted area while moving to collect the training data. Thus, the agent did not learn the motion required to trace the shortest path between the two landmarks. However, if the HRNN had the ability for spatial recognition through the use of an acquired cognitive map like rats, it should be able to predict correct visual images even when the agent moved on the unknown shortest path.

Because the outputs of the HRNN were not coordinates of robot positions but visual and motion predictions, we could not directly evaluate the spatial recognition ability of the HRNN through its position estimation ability. However, predicted vision, which was considered the recognition of locations in the HRNN, can be used for evaluating the spatial recognition of the HRNN. Because future vision strongly correlated with current vision, evaluating the cognitive map with predicted vision should be conducted when visual

information is not provided (PoM task of crossmodal prediction). If the HRNN predicted the correct colors corresponding to the landmark locations, even when the agent reached the landmarks by passing through the restricted area in the PoM task, this indicated that the HRNN acquired spatial understanding with the cognitive map, as described above.

Training Data

In order to collect visuomotor sequences for the simulation environment, the movement of the agent was controlled with predetermined rules. The direction of movement of the agent was determined by choosing one of the destinations at the center of the landmarks and center of the arena. The agent moved a single unit in each direction if it could decrease the distance to the destination on each axis, i.e., north-south or east-west. The destination was randomly reset with a 10% probability at every step. The starting point of the agent was randomly initialized in the square enclosed by the centers of the landmarks. Consequently, the agent moved within the square and did not go outside the square due to the control rules. If the agent had to cross the restricted area to reach the destination, the destination was reset at the boundary of the restricted area. An example of the visual images along a sample trajectory is shown in Fig. 2.4 (c).

The agent moved 1,000 time steps as one sequence from the starting point, which was randomly determined for every sequence. We collected 100 sequences for training of the HRNN. To test spatial recognition ability, we also collected sequences without restricting the area. These collected sequences were divided into segments of 50 time steps, so that each sequence comprised 20 segments of small sequences.

Training and Results

As described above, the HRNN learned the visuomotor sequences that were collected by the robot, and its acquired spatial recognition ability was tested on sequences with a restricted area.

Both lower and higher RNNs consists of GRUs [CGCB14] with 256 units. The vision and motion encoders ENC^v and ENC^m is a fully-connected layer with 128 hidden units. The vision and motion predictors DEC^v and DEC^m consists of two fully-connected layers with 128 hidden units for both vision and motion, and output units with the same

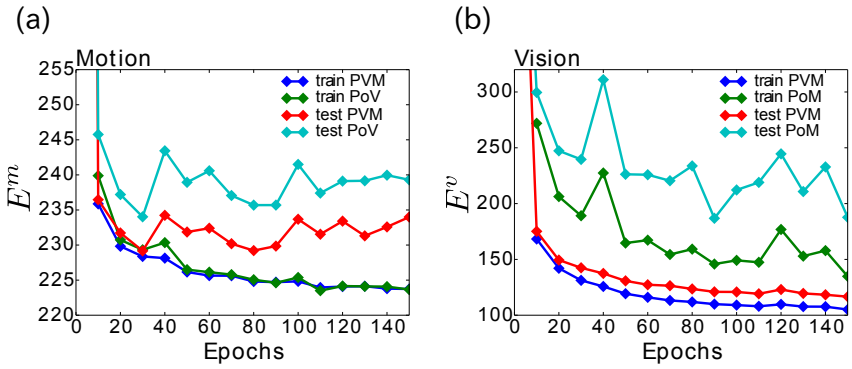

Figure 2.6: Prediction errors during motion (a) and vision (b) training. The errors for the PVM task are shown for both motion and vision. For motion, the errors for the PoV task are shown, and, for vision, the errors for the PoM task are shown. The errors for the test sequences are calculated from 100 sequences collected without the restricted area.

dimensionality as the vision and motion inputs, respectively. All hidden units in the encoders and predictors had ELU activation [CUH16] and the output units of vision and motion predictors had logistic-sigmoid and tanh non-linearlity, respectively. The vision error E^v was calculated as binary cross entropy and motion error E^m was calculated as mean squared error. To prevent our model from overfitting to the training sequences, an L1-norm of the model's parameters was added to the training loss with coefficients of 10^{-3}. The learning rate ϵ was adapted with the Adam method with default control parameters [KB15]. Training was conducted with the minibatch learning method with a minibatch size of 10. In a single epoch, the number of training was 100 (sequences) $\times$ 19 (segments) $\times$ 3 (conditions) = 5,700. This epoch was repeated.

Figure 2.6 shows the prediction errors of the visual and motion sequences for the training and test datasets. The errors of the training data for the visual and motion sequences successfully decreased during training. To see the effects of overfitting, the errors in the test data are also shown in the graphs. First, we focused on errors in the PVM task. For motion, the test data errors seemed to increase around 80 epochs. In contrast, the visual errors for the test data decreased with the training data errors. This occurred because the teaching signals of the motion sequences had only two dimensions with values that were discretized into only three values, i.e., -1, 0, or 1. Learning motion

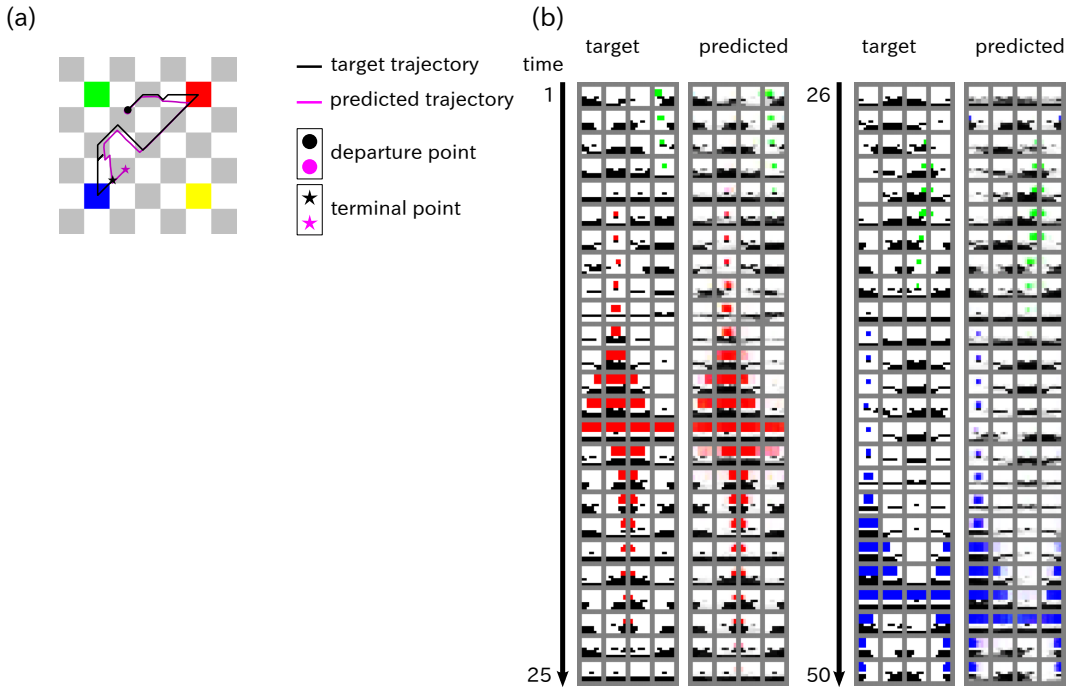

Figure 2.7: An example of predicted motion sequences (a) and vision sequences (b). The motion sequences are shown as the trajectory of the agent position. The target sequences are also shown. The visual and motion sequences for the target sequences correspond to each other. The target visual sequences are visual images from the corresponding position in the target motion sequence. The predicted sequences of motion and vision are sequences that are predicted in the PoV and PoM tasks, respectively.

sequences was relatively easier than learning visual sequences that have 768 dimensions. Second, we focused on the errors for the crossmodal prediction tasks (PoV and PoM). For motion (PoV), the test data errors also seemed to increase. For vision (PoM) in which the test sequence errors were bigger than those for the training sequences, the test errors decreased at a similar rate as the training errors. Therefore, to predict vision the model did not seem to overfit for the training sequences.

In a subsequent analysis, we used the model obtained with 90 epochs of training when the visual errors were minimal before motion overfitting progressed.

Crossmodal prediction abilities

In order to confirm that the HRNN recognized relationships between vision and motion, we visualized the visual and motion sequences that were predicted from only vision or motion sequences of the test data, respectively. This was similar to asking a subject what he/she is going to see if he/she moves along a given motion pattern from the current position or what paths he/she takes when the visual flow is provided. Figure 2.7 (a) shows the trajectory for the motion sequences that were predicted from vision alone. The target and predicted trajectories differed somewhat because the positional differences accumulated as the time steps proceeded. However, the moving directions at each step were almost correct.

Figure 2.7 (b) shows the visual sequences that were predicted when the motion sequences were provided. The predicted visual images correctly reproduced the colored landmarks at the correct times. Because the visual sequences were predicted without any external visual inputs, the HRNN recognized the relationships between the colored landmarks or floor patterns and how the motion sequences drove the agent in the environment. The HRNN also acquired an internal representation of the current position because the proper landmark colors were reproduced. If the agent did not know where it was, the reproduced colors would be wrong. These results showed that the HRNN successfully learned the correlation between the visual and motion sequences with the crossmodal prediction tasks.

2.3.2 Self-organized spatial representation

Internal states analysis

In order to analyze how the HRNN embedded the external structure into its own internal state, we visualized the states of the recurrent layers of the HRNN during the prediction. The dimensionality of the states was reduced to two dimensions with a principal component analysis (PCA). Figure 2.8 shows the visualized states when the agent moved between landmarks or the centers of the arena and not between the red and yellow landmarks, which was the unknown trajectory. The states of the lower-level and higher-level RNNs are shown, and the colors of the lines corresponding to the color of

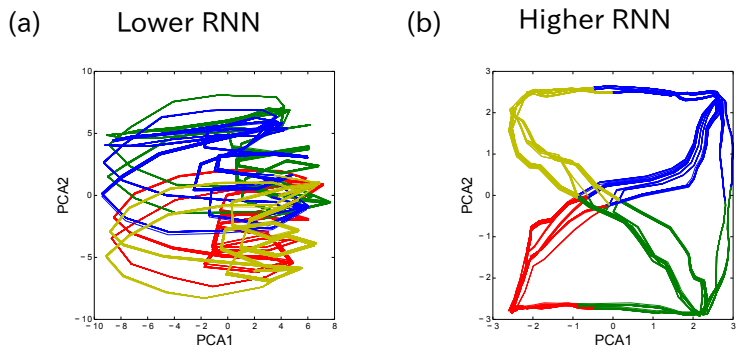

Figure 2.8: Internal states of the model when predicting visuomotor sequences. The states are mapped onto the two-dimensional space based on the results of the PCA analysis. The colors of the lines correspond to the nearest landmark from the true position (unpredicted position) of the agent. The agent does not enter the restricted area. (a) The states of the lower-level RNN. (b) The states of the higher-level RNN.

the nearest landmark from the current position of the agent. For the lower-level RNNs, even though the states were organized by color, lines with different colors overlapped in the two-dimensional space, which indicated that the states had higher-dimensional structures. In contrast, the states of the higher-level RNNs crossed only at the boundaries of the colors, and the shapes of the trajectories had the same topology as the trajectories of the agent's movements in the arena. These results showed that the HRNN recognized landmarks not by memorizing sequential experiences but with the relationships among various landmarks.

Generalization ability to restricted areas

The HRNN recognized the topological layouts of the landmarks. However, it was unclear if the HRNN created the cognitive map because the agent knew where it was, even in unknown areas. Here, we investigated whether the acquired internal model was a cognitive map by testing the generalization ability of the HRNN towards unknown motion paths in the PoM task. The predicted visual image when the trained model received the motion sequences when passing through unknown paths that have never been traversed during training is shown in Fig. 2.9. The red landmarks were correctly predicted along the unknown and shortest trajectory from the yellow landmark. This indicated that the

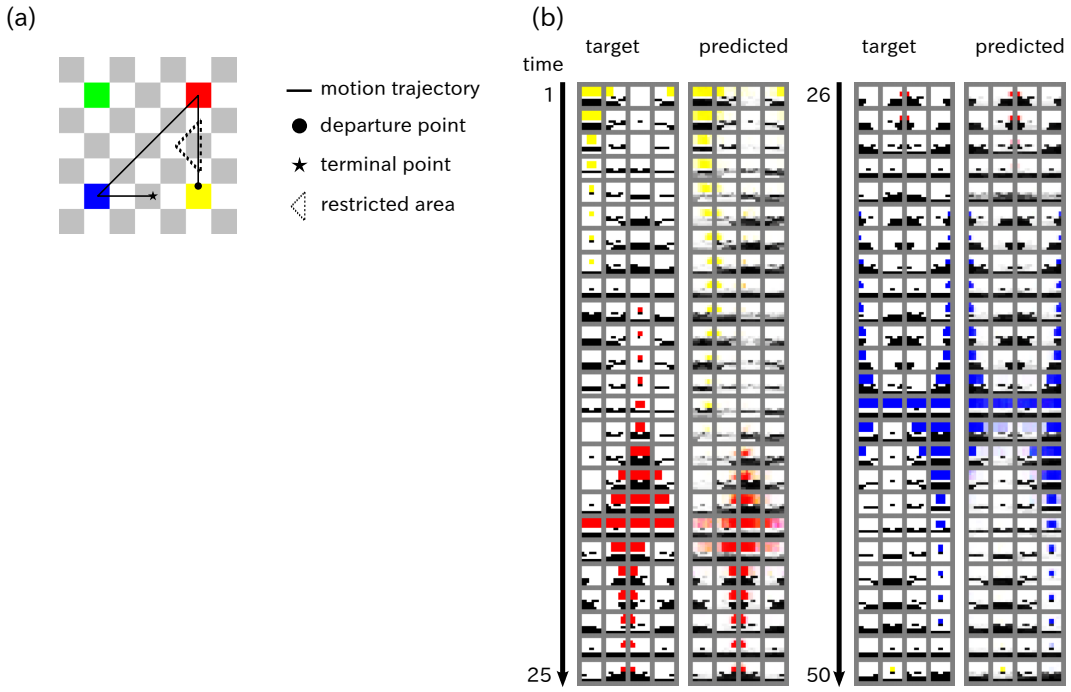

Figure 2.9: Examples of motion sequences passing through the restricted area (a) and vision sequences for the restricted area (b).

HRNN recognized the space between the red and yellow landmarks through which it could pass. In other words, the HRNN created a map with local visuomotor experiences even for unknown areas, and the map was called a cognitive map. Furthermore, the internal states of higher-level RNNs when the agent moved in the restricted area are shown in Fig. 2.10. The state for the restricted area, which was the shortest path between the red and yellow landmarks, traced a similar trajectory for states in the experienced areas between the blue and green landmarks, which had the same spatial relationship as that between the red and yellow landmarks. Figure 2.11 compares the trajectories for both the simulation space and internal state space between the shortest path crossing the restricted area and the detour path through the center of the arena. The HRNN clearly distinguished the shortest path from the detour path. These analyses showed that the HRNN extracted the spatial structure of the learned environment from visuomotor sensory inputs only and interpolated the unknown area with the acquired spatial recognition ability.

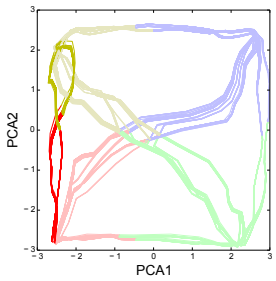

Figure 2.10: The internal states of the model when the agent passes through the restricted area. The internal states for the unrestricted area is shown in a light color.

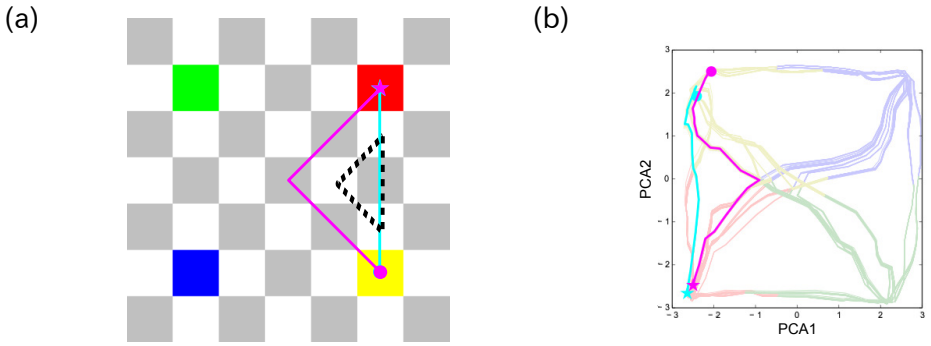

Figure 2.11: (a) The motion trajectories that pass to the restricted area (light blue) and that do not pass to the area (pink). (b) The internal states formed during corresponding motion trajectories.

2.3.3 Analysises on the development of the cognitive map

Formation of the cognitive map

We analyzed the cognitive map in the HRNN in a different way. To more directly clarify the correspondence between the cognitive map in the HRNN and the external environment, we painted the two-dimensional space with the colors of the visual images that the agent predicted. The color was painted at the position in two-dimensional space that corresponded to the current position in the environment. The space was painted by the predicted visual images when the agent moved on the test motion sequences. [1] First,

[1]To paint the space, the pixel values in the visual images of the sequences are summed for each RGB channel. The summed values are then accumulated at the point that corresponds to the current position of the agent. After the values are summed over all motion sequences, the accumulated values

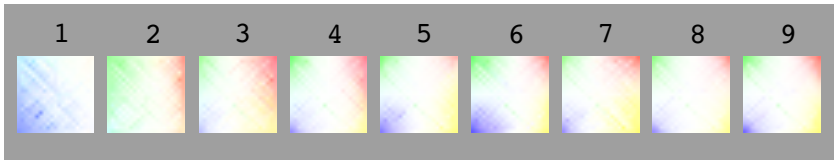

Figure 2.12: The space painted by the color of the predicted visual images and its transition during training. The number above each painted space indicates the training epochs.

Table 2.1: The prediction performance of the trained models after passing through restricted areas of different sizes ($n = 0, 1...9$). The percentages show the matching accuracy of the predicted colors after passing through the restricted area. For details, see the main text.

n	0	1	2	3	4	5	6	7	8	9
Accuracy	100%	100%	100%	100%	100%	100%	100%	96%	100%	94%

the agent moved around for 100 steps while receiving visual and motion inputs in order to recognize the current position. Then, the agent was provided only the test motion sequence without visual information for 900 steps; the color that occupied the predicted image the most at every step was identified. The colors were painted in the current position. Ten different test motion sequences were used. Figure 2.12 shows how the painted space changed in training epochs. The painted space was blurry in early epochs and gradually became consistent with the true arrangement in the arena as training progressed. These results showed that the HRNN created the cognitive map by repeatedly learning sequences.

Effect of the size of the restricted area

To investigate how much the size of the restricted area affected the predictions, we trained the model with restricted areas of 10 different sizes and tested the predictions. When the restricted area was defined by three vertices, i.e., $(10 - n, 0)$, $(10, n)$, and $(10, -n)$, 10 different areas were created by changing n from 0 to 9. When n equaled 0, no areas were restricted. We performed five different learning simulations starting with random initial configurations for each n and obtained five different trained models. To evaluate

are normalized for visualizing over each pixel.

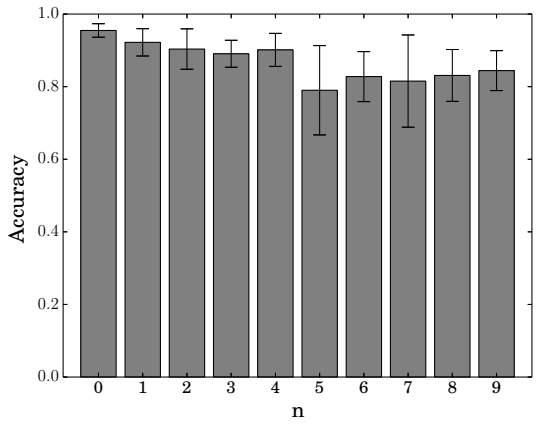

Figure 2.13: The prediction performance of trained models on the path through restricted areas with different sizes ($n = 0, 1...9$). The matching accuracy for predicted and real vision when the agent is on the path between the red and yellow landmarks, which passes through the restricted area, is shown.

the performance of the trained models, the colors of the predicted visions and real visions were compared while the agent passed between the red and yellow landmarks through the restricted area. The experimental processes were as follows. First, the trained agent was controlled to arrive at the yellow or red landmark within 280 steps. During this, the visual and motion sequences were given to the agent. Then, the only straight motion sequence to the opposite yellow or red landmark across the restricted area was given. Vision was not provided but predicted by the model. Finally, the predicted vision was compared with real vision from the environment. The color of the vision was determined with *hue-saturation-value* (HSV) color space [2]. Performance was calculated over 100 paths from yellow to red and from red to yellow, and a total of 200 trials were conducted in the evaluations of each trained model. Table 2.1 shows the matching accuracies of the

[2]The visual colors are determined with HSV color space. First, visual images are converted from the RGB space to the HSV color space with the *hue*, *saturation*, and *value* channels. Second, the converted visual images are labeled with colors determined by the mean value over the image. If the *saturation* value is zero, the color of the images is determined to be white. Otherwise, the colors are determined by the *hue* value. We assumed that there are six colors besides white for labeling (red, yellow, green, cyan, blue, and magenta), and the *hue* value of the HSV space was divided into six regions that correspond to these six colors.

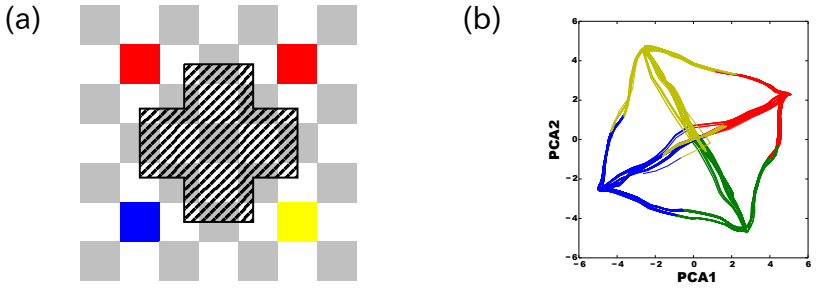

Figure 2.14: (a) The ambiguous environment that contained two landmarks with the same color (two red landmarks). The agent could not sense any landmarks in the striped area due to the limitation of the maximum distance in which the camera could capture the landmarks. (b) The internal states of the higher-level layer of the model that was trained for the ambiguous environment are shown in (a). When the agent is near the top-left red landmarks, the internal states are colored green.

predicted colors only when the agent arrived at the opposite landmark. Figure 2.13 shows the matching accuracies for the path between red and yellow. These results showed that the cognitive map was robustly acquired against the size of the restricted area.

Learning in an ambiguous environment

Although the exact spatial coordinates were not given to the agent directly, the spatial positions corresponded uniquely to the different visual patterns. Thus, the HRNN might have utilized a one-to-one mapping between them, and the spatial relationships between the positions might not have been learned. However, this was not the case in the HRNN. In order to show that the HRNN acquired the spatial relationships, the HRNN was trained for an ambiguous environment in which the agent could not learn one-to-one mapping, as shown in Fig. 2.14 (a). In the HRNN, the maximum distance in which the camera was able to capture the landmarks was limited to 4 units. The area in which the agent could not capture any landmarks is illustrated in the figure. Furthermore, two landmarks had the same red color. The HRNN was trained for the visuomotor sequences that were collected in such an ambiguous environment in which the other configurations were the same as those in the previous experiment. As a result, the acquired internal states were self-organized in the PCA space in a way that was similar to that described in the above

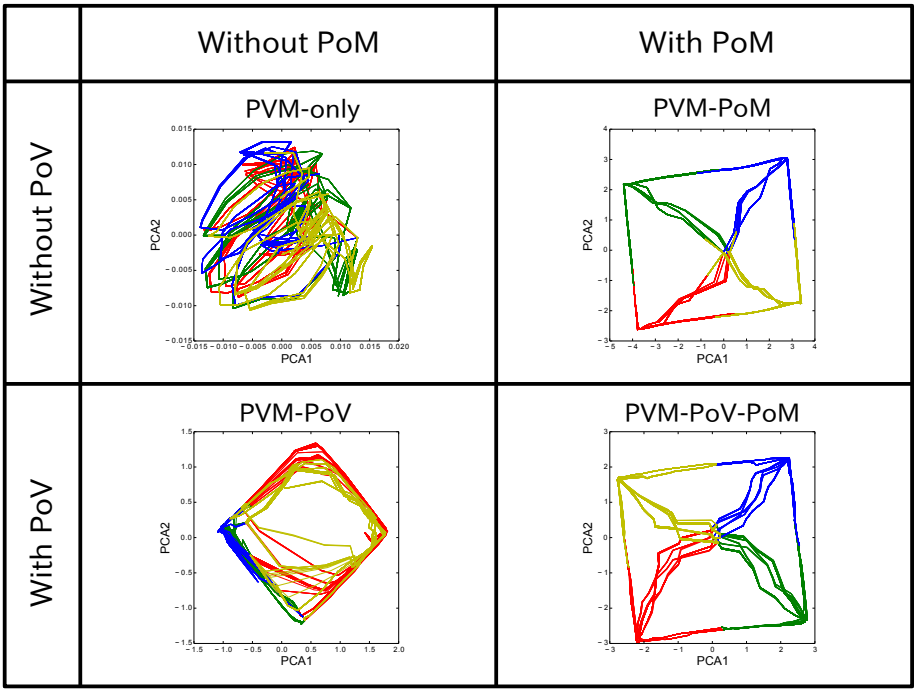

Figure 2.15: Comparison of the internal states of the higher-level RNNs between different learning conditions. The color is painted in the same manner as described in Fig. 2.8.

results (Fig. 2.14 (b)). Therefore, the HRNN obtained the spatial recognition not by learning the one-to-one mapping between visual patterns and the spatial positions but by associating the visual and motion sequences.

Analyzing the impact of the learning of crossmodal predictions

The above experiments showed that the HRNN acquired spatial recognition, even for the unknown area. In order to analyze how the learning of crossmodal predictions affected the acquired internal states, the internal states of higher-level RNNs were compared among four different models with different training conditions: PVM-only task (PVM-only condition), PVM and PoV tasks (PVM-PoV condition), PVM and PoM tasks (PVM-PoM condition), and all tasks (PVM-PoV-PoM condition). In this case, the HRNN was trained on the visuomotor sequences with the restricted area removed in order to focus on the effects of learning the PoV and PoM tasks. We used the model that was obtained with 300 epochs of training for every training condition.

Figure 2.15 compares the internal states of the higher-level RNNs. The internal states

Table 2.2: Accuracy of the internal state classification for higher-level RNNs by k-means clustering for models with different training conditions.

Training condition	Max	Min	Average
PVM-PoV-PoM	95.6%	83.7%	90.7%
PVM-PoM	97.1%	89.4%	92.4%
PVM-PoV	84.7%	48.0%	59.4%
PVM-only	76.2%	35.9%	46.5%

under the PVM-PoV-PoM condition formed a topology that corresponded to the learned environment. In the PVM-only condition, however, the trajectory of the internal states was scrambled and disorganized. This might have occurred because future inputs tended to be similar to current inputs, and recognition of the topology of the environment was not difficult in the PVM task. For models trained under the PVM-PoM condition, the states of the model were organized into a topology like that in the PVM-PoV-PoM condition. However, for the model trained under the PVM-PoV condition, the states of the trained model formed a square, which did not seem to be the cognitive map. Because local visual sequences can provide information related to corresponding motion, global spatial recognition is assumed to not be required for the PoV task, and learning in the PoV task did not make the HRNN organize the cognitive map. Conversely, because local movements contained no information about the global position, memorizing the global position in its own states, which leads to the self-organization of the cognitive map, was important.

Quantitative evaluation of the cognitive map

To quantitatively evaluate cluster formation in the internal states, we performed four different learning simulations starting from random initial configurations for each condition and constructed five trained models, including the trained model in previous sections. The internal states that were colored in the same way as Fig. 2.8 were investigated. To evaluate cluster formation, the k-means clustering method was applied as follows. First, internal states with the same color were assigned to the same color group. Next, regardless of the colors, we created four clusters of the internal states with k-means clustering ($k = 4$). Next, we found the best matches between the color groups and the clusters such that the states of the same clusters were in the same color group. The accuracy

Table 2.3: Evaluation of the consistency between vision and motion generated in mental simulations. The sequences with 150 steps that were obtained from the early 200 steps (except the first 50 steps) are used for each of the 10 test sequences. A total of 1,500 steps are evaluated for each model.

Training condition		1st trial	2nd trial	3rd trial	4th trial	5th trial	Average
PVM-PoV-PoM	Steps with any color	1311	1103	1309	1232	1221	1235.2
	Matched steps	571	188	1051	852	321	596.6
	Accuracy	44%	17%	80%	69%	26%	47%
PVM-PoM	Steps with any color	1243	1388	1318	1143	1085	1235.4
	Matched steps	228	454	504	123	54	272.6
	Accuracy	18%	33%	38%	11%	5%	21%
PVM-PoV	Steps with any color	742	648	248	854	557	609.8
	Matched steps	14	67	56	26	40	40.6
	Accuracy	2%	10%	23%	3%	7%	9%
PVM-only	Steps with any color	154	198	275	165	331	224.6
	Matched steps	42	85	116	22	98	72.6
	Accuracy	27%	43%	42%	13%	30%	31%

of assigning cluster internal states to the same color group was used as an index of the self-organization of the internal states.

Table 2.2 shows the results of the k-means clustering classification. The models trained by the PoM task (PVM-PoM and PVM-PoV-PoM conditions) showed consistently high accuracy. In contrast, the models trained under the PVM-only and PVM-PoV conditions showed lower accuracy. These results confirmed that the PoM task was a major factor in the self-organization of the cognitive map.

Mental simulation experiment

In the previous sections, the analyses were performed by providing the trained model with external inputs from the environment. We imagined visual images that corresponded to imagined motion by using the cognitive map as mental simulation. The ability for mental simulation with the cognitive map is evaluated here.

For the ability for mental simulation, the accuracy between the imagined visual images and the real visual images was evaluated when the agent moved in the environment with imagined motion sequences. To obtain the imagined visual and motion sequences, the outputs $\boldsymbol{v}_{t+1}$ and $\boldsymbol{m}_{t+1}$ of the model were fed back into the inputs. The network model became an autonomous dynamic system, and we obtained visual and motion sequences without providing external inputs. Real visual images were also obtained by moving the

agent according to the imagined motion sequences. To compared the real and imagined visual images, the color of the visual images were determined as discussed above. To avoid the evaluations becoming too high because of blank (no landmark) visual images, we only evaluated when the color landmarks appeared, which indicated that colors other than white were detected, in real visual images.

In the experiment, the agent first moved around for 100 time steps to recognize the current position in the same way as in the previous sections. Next, the agent imagined the visual and motion sequences for 200 time steps by feeding back the predictions as inputs. Subsequently, the agent moved according to the imagined motion sequences to obtain real visual images. The visual images for the first 50 time steps were ignored because there were no clear differences. The visual images for the remaining 150 time steps were used for the analysis.

Table 2.3 shows the results of the evaluation, which was conducted with the same trained models used in previous sections. The models trained under the PVM-PoV-PoM condition showed the highest average accuracy. Although the models trained under the PVM-only condition showed the second highest accuracy, the number of real visual images that captured any colored landmarks was much less for models trained under the PVM-only condition. This occurred because the generated motion sequences often became monotonous and the agent tended to go outside of the training area. Thus, the models trained under the PVM-only condition were thought to be less able to consistently generate vision and motion between them over the long term. It is notable that models trained under the PVM-PoM condition showed decreased accuracy than the models trained under the PVM-only condition, while these models organized the cognitive map in their internal state. Such one way crossmodal training seemed to cause biases and inconsistencies in the acquired internal recognition for vision and motion, and cross-modal integration was not developed. Alternatively, the models trained under the PVM-PoM condition could not use the acquired cognitive map for voluntary movements. The models trained under the PVM-PoV-PoM condition were thought to use the cognitive map to not get lost in the arena. However, in some trials, even these models that were trained under the PVM-PoV-PoM condition were less accurate, and the ability to use the cognitive map was considered unstable.

We can always sense visual and motion inputs, and the creation of crossmodal predictions is not explicitly required. Thus, training for the PVM-only task seemed sufficient for generating consistent visual and motion sequences. However, this was not the case in our results. The results showed that explicit crossmodal predictions in which the balance between vision and motion was maintained were required for consistent sequence generation.

2.3.4 Discussion

The HRNN acquired an internal model of the spatial structure of the external environment with two-dimensional topology. Because the objective of the training was to minimize the prediction error, a cognitive map was useful for reducing error. Moreover, the cognitive map was not formed in both layers of the RNN but rather in the higher-level RNN (Fig. 2.8), which indicated that the hierarchical structure contributed to the self-organization of the cognitive map. The crossmodal prediction task became difficult because of the missing information. In such situations, the HRNN reduced the prediction errors by remembering the current location in the higher-level RNN, which stably recognized the location because the higher-level RNN was not directly connected to unstable and dynamic external inputs. Thus, the division of the roles between the two levels was considered self-organized. Therefore, the hierarchical structure in the HRNN worked well for extracting the environmental structure. In previous studies of hierarchical structure [TN99, YT08], models were trained to only predict with visual and motion inputs (like the PVM task in the current study). Thus, while an internal model of sequential experiences was acquired, an internal model of the structure of the external environment was not. As described in the experiments comparing training conditions, self-organization of the cognitive map required the learning of crossmodal predictions. Moreover, predictions using sequential dependency were difficult because there were no sequential rules for the agent to determine the movement direction at the data collection phase in this study. Thus, the creation of a map of the external environment is very needed to reduce prediction errors. Thus, this learning allowed the HRNN to organize the cognitive map.

Our results showed that the internal states were organized and formed clusters that corresponded with the spatial positions when the agent learned the prediction of the

visuomotor sensory inputs under the PoM task (Fig. 2.15). The acquired model successfully recognized the current position, even when an unknown trajectory was traced. However, the ability of mental simulation was not stably generated, even for the best model that learned under all tasks (PVM, PoV, and PoM) (Tab. 2.3). In other words, our current model sometimes failed to create the ability for long-term imagination of the associations between vision and motion. In animals, neural activities that are specific to planning in the hippocampus exist, and neural replay is performed for memory consolidation [DKW09, OBS$^+$15]. It is crucial for animals to acquire the ability for long-term imagination. The reason our current model could not achieve this stably was that our predictive learning scheme aimed to minimize the prediction errors between time points t and $t+1$. The model always received external current inputs and predicted motion and vision at the next time step. Long-term imagination requires the ability for predictions for a longer time without external inputs. However, why do animals need to use long-term imagination instead of performing one-step predictions? One possible answer is that animals intend to achieve something in the future. To get to a certain destination, they need to make a plan and imagine how they will move and what they will see. Paine and Tani (2005) incorporated this higher-level intentional flow in the hierarchical structure in a robot navigation task. The implementation of intentional flow produces long-term imagination and stabilizes the ability for mental simulation. In fact, the PoM task can be considered a condition that is similar to intention in the sense that the agent is asked to produce the visual images that the agent is going to see with determined motion sequences and minimize the prediction errors even though the motion sequences were not produced by themselves. Conversely, the PoV task can also be considered an intentional condition in the sense that the agent must imagine future motion to trace the path following the determined visual sequences. Because the PoM task contributed to the formation of the clusters of internal states, intentions that simultaneously realize the long-term imagination of vision and motion were necessary for forming and stabilizing the cognitive maps in a computational model and in animals. In Paine's study, the robot did not create a cognitive map because the robot started from a fixed home position and only remembered the sequential flow to achieve the goals. However, by implementing intention, the HRNN acquired the ability to generate voluntary movements by using the self-organized cognitive

map.

The mental simulation in the HRNN was not stable compared to that described in Tani (1996). During a mental simulation with continuous sensory inputs, the simulation (prediction) error accumulated at each step, and an expanding discrepancy between the simulated and true sensory inputs was unavoidable. However, Yamashita and Tani (2008) showed that mental simulation with continuous sensory inputs was successful to some extent with a hierarchical RNN. In their model, the continuous sensory inputs were abstracted into higher-level sequential segments. The abstracted segments then acted as the discrete branching events described in Tani's study, and the mental simulation was achieved. However, the mental simulation was unstable even though the HRNN had a similar hierarchical structure as that in Yamashita's model. One possible explanation was that the HRNN abstracted the continuous inputs as spatial positions rather than as sequential events as described in the above discussion. A question that arises is how the discretization of experienced sensory flows and static spatial coordinates can be realized in a single neural network dynamics. This could be an interesting problem for the construction of navigation behavior from the use of the cognitive map.

2.4 Development of the Recognition of the Spatial Structure through Human Visuomotor Experiences

In this section, we investigate whether the HRNN also can develop the spatial recognition through the visuomotor experiences in real environment.

2.4.1 Learning in Real Environment

Visuomotor Experiences in Real Environment

We collected the visuomotor experiences by a human subject; the subject walked around in a room with a helmet equipped with a head-mount camera and accelerometers (Fig. 2.16). The head-mount camera captured first person view visual images and the accelerometers,

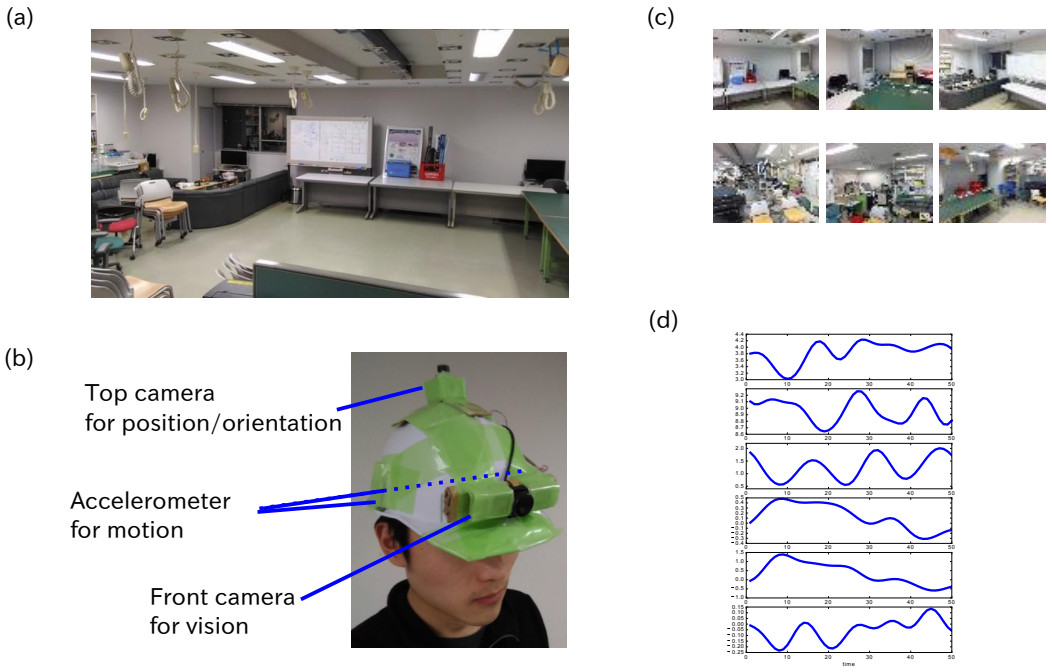

Figure 2.16: (a) Experimental environment (a room). (b) The helmet for collecting visuomotor sequences and the spatial position and orientation of the subject. (c) Examples of the collected vision. (d) Examples of the collected motion sequences.

which were attached on the sides of the head of the subject, could measure translational and rotational accelerations. The accelerometers were attached near both temples so that they located close to otolith organ and semicircular canal, which are organs that provide sense of translational and rotational acceleration to human. The accelerometers could measure three dimensional accelerations for both translation and rotation. For determining position and orientation of the subject in the room, another camera was attached on top of the helmet. The top camera capture AR markers on the ceiling and the position and orientation of the subject can be calculated based on reference positions associated with the AR markers. The calculated positions or orientations of the subject were not used during the training of the HRNN.

The visual images by the head-mount camera were used as the visual sequences and the accelerations by the accelerometers were used as the motion sequences. The visual images, accelerations and the positions and orientations of the subject were captured with 10 fps. The captured visual images were resized to 48×64 pixels and the dimension

Table 2.4: The structures of ENC^v and DEC^v

ENC^v							
layer	type	size	channel	kernel size	stride	padding	activation
1	input ($\boldsymbol{v}_t$)	64x 48	3	-	-	-	-
2	conv	32 x 24	8	3 x 3	2 x 2	1 x 1	ReLU
3	conv	16 x 12	16	3 x 3	2 x 2	1 x 1	ReLU
4	conv	8 x 6	32	3 x 3	2 x 2	1 x 1	ReLU
5	conv	4 x 3	64	3 x 3	2 x 2	1 x 1	ReLU
6	fully connected	1 x 1	64	-	-	-	ReLU

DEC^v							
layer	type	size	channel	kernel size	stride	padding	activation
1	input ($\boldsymbol{h}_t^{lower}$)	1 x 1	256	-	-	-	-
2	fully connected	1 x 1	64	-	-	-	ReLU
3	fully connected	4 x 3	64	-	-	-	ReLU
4	conv	4 x 3	128	3 x 3	1 x 1	1 x 1	ReLU
5	upsample	8 x 6	128	-	-	-	-
6	conv	8 x 6	64	3 x 3	1 x 1	1 x 1	ReLU
7	upsample	16 x 12	64	-	-	-	-
8	conv	16 x 12	32	3 x 3	1 x 1	1 x 1	ReLU
9	upsample	32 x 24	32	-	-	-	-
10	conv	32 x 24	16	3 x 3	1 x 1	1 x 1	ReLU
11	upsample	64 x 48	16	-	-	-	-
12	conv	64 x 48	3	3 x 3	1 x 1	1 x 1	Sigmoid

of the motion which comprised of the accelerations was 12. The visuomotor sequences were captured during $2,500$ seconds while the subject freely walked around in the room; it means $2,5000$ frames of visuomotor sequences were collected. Then, the collected sequences were splitted into 50 sequences in which each sequence comprise of 500 steps of vision and motion. All of these visuomotor sequences were used for the training of the HRNN.

Network Settings

We use the Continuous-Time RNN (CTRNN) that has a time constant parameter τ, which determines the time scale [Bee95, YT08]. The lower and higher RNNs were the CTRNN layers with 256 and 128 neurons, respectively. The time constant τ of the lower and higher RNNs was 2 and 25, respectively. The motion encoder ENC^m is a fully-connected layer with 64 hidden units with ReLU activation and DEC^m consists of two fully-connected layers with 64 hidden units with ReLU activation and 12 output units for motion with tanh activation, respectively. The visual encoder ENC^v and DEC^v consists of convolutional neural networks. The structures of DEC^v and DEC^v are shown in Tab. 2.4. The vision error E^v and motion error E^m were calculated as mean squared error. The L1-norm regularization term with respect to RNNs' parameters was added to the loss with coefficients of 0.05. The length of each visuomotor segment for the actual training

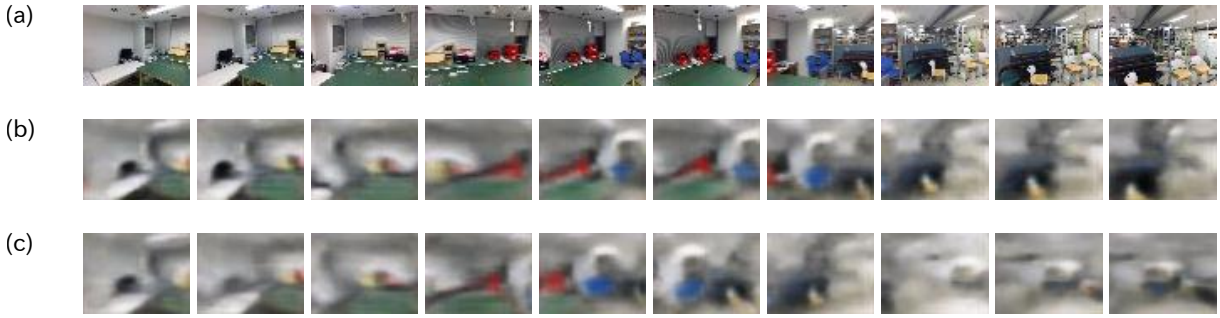

Figure 2.17: A example of input visual sequences (a) and corresponding predicted visual sequences from both vision and motion inputs (b) and from only motion input (c).

is 50. The HRNN was trained over the training visuomotor sequences 200 times.

Training Results

After the training, the HRNN became able to predict vision and motion as in Fig. 2.17 and 2.18. These figures show the predicted vision and motion in the PVM setting, the predicted vision in the PoM setting, and the predicted motion in the PoV setting. Although the predicted visual images were blurry, they capture the characteristic of landmarks in the room. In the case of the PoM setting, although the predicted visual images in the latter half of the sequence look very different from the ground truth images, that in first half were correctly predicted; it means that the HRNN could generate prediction corresponding to motion. The motion predictions in the case of the PoV setting were not so close to the ground truth of motion but roughly close to the ground truth. This is because small variation of the motion did not affect the visual sequences and the visual sequences did not contains sufficient information for predicting details of the motion sequences.

2.4.2 Spatial Representation Developed in the Real Environment

We visualize the internal states of the trained HRNNs for investigating the obtained internal recognition in the trained HRNNs. The internal states of the RNN laysers were visualized by PCA (Fig. 2.19). In the case of the lower RNN, each point of the internal states are colored according to the orientation (direction) of the subject; a specific color

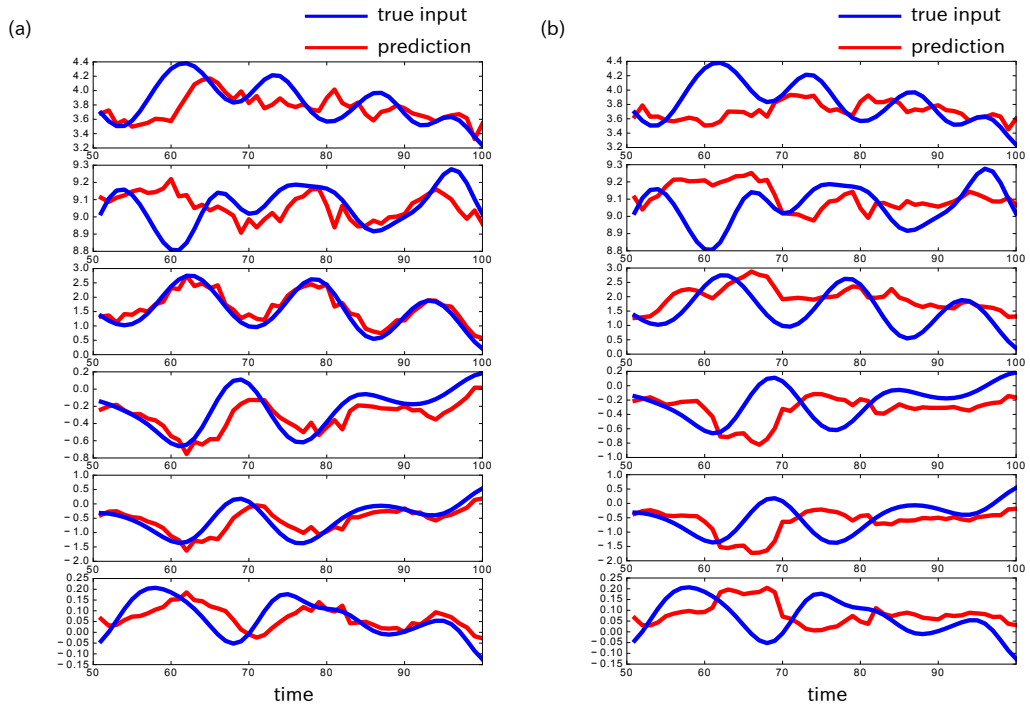

Figure 2.18: A example of input motion sequences (blue line) and corresponding predicted motion sequences (red lines); the predicted motion sequences from both vision and motion inputs (a) and from only visual input (c).

is assigned to each direction based the HSV color space, which is a torus color space. The internal states of the lower RNN roughly were arranged along with the colors and formed circle-like shape; it corresponded to the periodic characteristic of direction. In the case of the higher RNN, each point of the internal states are colored according to the position of the subject. For coloring the internal states of the higher RNN, values were assigned to each position. Red, blue, green, and yellow corresponded to the four corners of the area where the subject walked, and linearly interpolated colors were assigned to other positions. The internal states of the higher RNN were organized by colors and it is considered that the higher RNN represented the spatial position of the subject in it internal states.

In this experiment in real environment, the direction of the subject changed during the subject walked different from the previous experiments where the agent did not change its direction. As shown in the results of the internal states analysis, the HRNN develop the

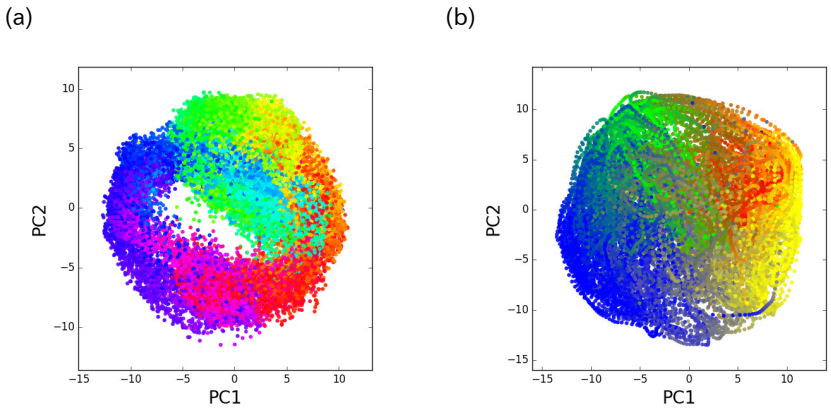

Figure 2.19: (a) The internal states of the lower RNN colored according to the spatial orientation. (b) The internal states of the higher RNN colored according to the spatial position.

Table 2.5: Results of regression analysis of the internal states.

Input	Error distance [m]: avg. (std.)	
	train	test
Raw visual images	0.59 (0.32)	0.92 (0.51)
Internal states of Lower RNN	0.43 (0.25)	0.44 (0.25)
Internal states of Higher RNN	0.36 (0.24)	0.36 (0.24)

representation of both direction and position in different RNN layers. That is because the lower and higher RNNs had different time scales; the lower RNN had fast time scale and the higher RNN had slow time scale. As shown in the previous studies [WKV06, FSW07], spatial position has slower dynamics than direction. Thus, the lower RNN could recognize the direction with fast time scale, and on the other hand, the higher RNN could recognize the spatial position with the slow time scale.

Quantitative Evaluation of the Developed Spatial Representation

We quantitatively evaluated the internal representation of spatial position developed in the internal states by linear regression analysis. Ridge regression models were constructed to predict the spatial position of the subject from the internal state of the RNNs, and the prediction errors were used for evaluation of the spatial representation. If the internal states were well organized so that they change corresponding to the spatial position of the subject, the regression model could accurately predict the spatial position. In this evalu-

ation, all the two consecutive data points such that the distance between their recorded positions was larger than 0.5 meter were excluded as noisy outliers. After excluding the outliers, the number of the data points became $24,739$ and the regression models were trained on 90 % of the data points and the other 10 % was used for test data. Three regression models were constructed for predicting the spatial position from the internal states of the lower RNN, that of the higher RNN, and raw visual images. The results of the regression analysis are shown in Tab. 2.5. The average error distances between ground truth position and predicted position by the regression models are shown. The regression model using raw images overfitted on the training data. On the other hand, the regression models using the internal states well generalized to the test data, and the regression model using the internal states of the higher RNN more accurately predicted the position than that using the internal states of the lower RNN. This results show that the higher level RNN with slower time scale in the hierarchical structure of RNNs could effectively develop the representation of the spatial position through the visuomotor experiences in the real environment.

2.4.3 Discussion

In this section, it was shown that the HRNN is scalable to realistic situation by training it on the visuomotor experiences collected in the real environment. The result showed that the HRNN also can develop the representation of direction in addition to the spatial position. The representations of the spatial position and direction were developed in the higher and lower RNNs, respectively. It is considered that the differences in the time scales of the RNNs contributed the development of the spatial position and direction; the spatial position was represented in the higher RNN with slow time scale and the direction was represented in the lower RNN with fast time scale. This result is analogous to the results in [WKV06, FSW07].

In this case, the visual input was more high-dimensional and complex than in the previous section, and the motion input was not just displacements but accelerations. Although different time scales in RNNs and CNN were introduced for dealing with realistic visuomotor experiences, no specific function was assumed in these modules and the spatial representations of position and direction were developed through only prediction learning.

This result indicates that the spatial recognition is not innate ability and can be developed through visuomotor experiences even in real environments.

2.5 Effect of Behavioral Complexity on the Development of the Recognition of the Spatial Structure

Animals can develop spatial recognition through only subjective visuomotor sequences, and the subjective visuomotor sequences depend on the animals' behavior. Conversely, the behavior of animals changes depending on their recognition. Thus, behavior and recognition develop through interaction with each other. In the case of spatial recognition, it was observed that the spatial behavior of a rat changed along with the development of spatial representation in its brain [WCBO10]. However, because the behavior and spatial recognition changes simultaneously, it is unclear how behavior affects the development of spatial recognition.

In this section, we simulate the development of spatial recognition using controlled behaviors. We focus on the relation between the complexity of spatial behavior and the development of spatial recognition. How the developed spatial recognition depends on the complexity of the behaviors is investigated. The complexity of the behaviors is interpreted as the randomness of the spatial movement pattern. The HRNN model was trained on visuomotor sequences with movements of different values of randomness. It is expected that the developed recognition is different for different randomness of the movement, and the effect of the movement pattern on the developed spatial recognition is investigated.

2.5.1 Simulation and Training

A mobile robot was made to move around in the simulation environment. It was modeled as an agent that can move around in a two-dimensional flat arena. The agent can sense visual images through an attached camera on its head and proprioceptive self-motion. The movement pattern is controlled by the randomness parameter η. Visuomotor sequences for different randomness η are prepared for the simulation of the development of spatial recognition.

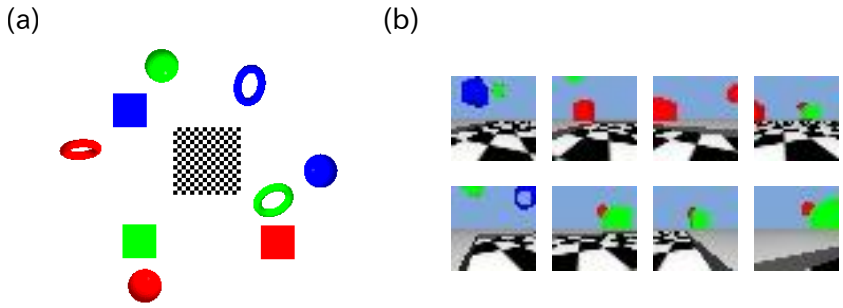

Figure 2.20: (a)Overview of the simulated environment. (b) Examples of the agent's vision. (c)Hierarchical recurrent neural network (HRNN).

Simulation environment

The simulation environment is shown in Figs. 2.20 (a) and (b). There are several floating objects that constitute a landscape for the agent's visual experiences. The agent moved within the arena that is indicated by the floor having a checkered pattern. The arena wherein the agent could move around is enclosed by an invisible fence. The fence is low and does not obstruct the agent's view. The size of the arena is 20×20 units of distance.

Movement pattern of the simulated agent

The agent moved by unit distance in one simulation step. The moving direction was the same as the agent's heading direction. The head direction changed with every time step. The new head direction was obtained by adding a random value $\epsilon \sim \mathcal{N}(0, \eta^2)$ to the value of the current head direction. Thus, the value of η (the standard deviation of ϵ) determined the randomness of the exploration by the agent in the arena. The unit of η is degree. If the agent hits the fence as a result of movement, the agent rebounded at the fence and the head direction was changed at the beginning of the next step (new head direction was perturbed by ϵ). The examples of movement pattern for different values of η are shown in Fig. 2.21.

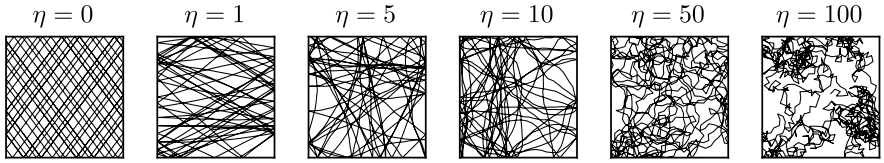

Figure 2.21: Examples of the agent's movement pattern for various values of η during 1,000 steps.

Table 2.6: The structures of ENC^v and DEC^v

ENC^v

layer	type	size	channel	kernel size	stride	padding	activation
1	input ($\boldsymbol{v}_t$)	32 x 32	3	-	-	-	-
2	conv	16 x 16	8	3 x 3	2 x 2	1 x 1	ReLU
3	conv	8 x 8	16	3 x 3	2 x 2	1 x 1	ReLU
4	conv	4 x 4	32	3 x 3	2 x 2	1 x 1	ReLU
5	fully connected	1 x 1	64	-	-	-	ReLU

DEC^v

layer	type	size	channel	kernel size	stride	padding	activation
1	input ($\boldsymbol{h}_t^{lower}$)	1 x 1	256	-	-	-	-
2	fully connected	1 x 1	64	-	-	-	ReLU
3	fully connected	4 x 4	64	-	-	-	ReLU
4	conv	4 x 4	32	3 x 3	1 x 1	1 x 1	ReLU
5	upsample	8 x 8	32	-	-	-	-
6	conv	8 x 8	16	3 x 3	1 x 1	1 x 1	ReLU
7	upsample	16 x 16	16	-	-	-	-
8	conv	16 x 16	16	3 x 3	1 x 1	1 x 1	ReLU
9	upsample	32 x 32	16	-	-	-	-
10	conv	32 x 32	3	3 x 3	1 x 1	1 x 1	Sigmoid

Visuomotor sequences

The agent's motion $\boldsymbol{m}_t$ is represented as two-dimensional vectors calculated using ϵ as follows:

$$\boldsymbol{m}_t = (\cos(\epsilon), \sin(\epsilon)). \tag{2.13}$$

When the agent collided with the fence, the motion $\boldsymbol{m}_t$ was determined so that the moving direction was reflected at the fence.

The size of the visual image captured by agent's camera $\boldsymbol{v}_t$ was 32×32, and each pixel of the image had three channels (RGB). The agent receives the motion and vision resulting from the movement in one simulation step, and only the subjective motion and vision are the inputs from the environment to the agent.

Training

The HRNN was trained to predict the agent visuomotor sequences. Different HRNNs were trained with the sequences produced for various η. We describe an HRNN trained the with the motion sequences for $\eta = \alpha$ as per HRNN-$\alpha\eta$ below (e.g., HRNN-10η is the HRNN trained with the motion sequences of $\eta = 10$).

The next motion was determined with random fluctuations so that what the HRNN can predict is $\boldsymbol{m}_{t+1} = (0, 1)$, which is the expected value of motion calculated from ϵ in all conditions. Thus, the motion prediction would not contribute results.

Training Settings

For collecting the sequences, the agent moved in the arena as follows. First, the agent was placed at a random position in the arena with a random head direction. The agent then moved 500 steps following the movement pattern defined by η, and the motion and visual inputs were stored as training sequences. One hundred sequences with different initial positions and directions were prepared for the training. The random initial condition of the agent is required for exploring the entire arena when η is very small and the agent moves monotonically and periodically.

Network Settings

The lower and higher RNNs were the CTRNN layers with 256 and 128 neurons, respectively. The time constant τ of the lower and higher RNNs was 2 and 25, respectively. The motion encoder ENC^m is a fully-connnected layer with 64 hidden units with ReLU activation and DEC^m consists of two fully-connected layers with 64 hidden units with ReLU activation and 2 output units for motion with tanh activation, respectively. The visual encoder ENC^v and DEC^v consists of convolutional neural networks. The structures of DEC^v and DEC^v are shown in Tab. 2.6. The vision error E^v and motion error E^m were calculated as mean squared error. In order to prevent the HRNN from overfitting to the training sequences, an L1-norm of the HRNN's parameters was added to the objective of minimization with coefficients of 10^{-3}. The length of each visuomotor segment for the actual training is 50 (a single training sequence is divided into 10 segments). The Adam

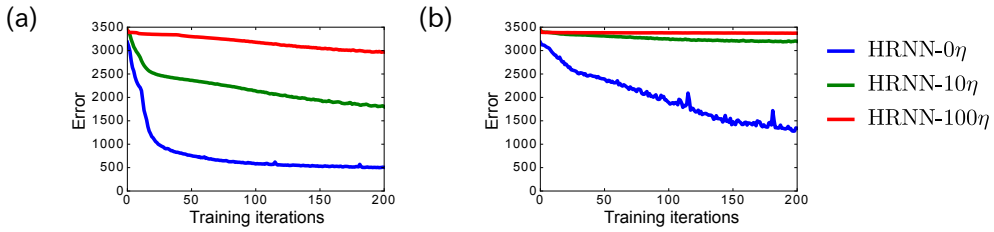

Figure 2.22: Error in vision predicted from vision and motion (a) and that predicted from only motion (b) during the training. Errors are shown for $\eta = 0, 10,$ and 100 (HRNN-0η, HRNN-10η, and HRNN-100η).

algorithm [KB15] was used for updating parameters.

We prepared visuomotor sequences with $\eta = 0, 10,$ and 100. Three different HRNNs for different values of η were trained 200 times over training sequences. The obtained abilities of the trained HRNNs are shown below.

2.5.2 Effect of Behavior on Prediction Ability

Figure 2.22 shows the errors of vision during training. The prediction errors for vision from both the previous vision and motion inputs and from only the motion inputs are shown for each HRNN trained with a different values of η. The larger η is, the slower the rate of decrease in the error. Moreover, the error of vision predicted from only motion using HRNN-100η remained almost the same. Figure 2.23 shows examples of the visual images predicted by the trained HRNNs. The movement when η is 0 was used to obtain the results for all the HRNNs. It was shown that the trained HRNNs—except for HRNN-100η—can predict visual images as a result of training. In the case of HRNN-100η, the predicted vision does not clearly contain any colored landmark. This is because, in cases wherein η is large, the movement pattern is almost random and the HRNN could not predict visual sequences at all. The visual images predicted using only motion are shown in Fig. 2.23 (bottom), and it shows how well the trained HRNN constructed the internal model of the external environment. The vision for HRNN-0η with colored landmarks using only motion was predicted almost correctly although the floor pattern was not predicted. HRNN-10η was also able to predict the colored landmarks using only motion although the predicted vision is blurry. HRNN-100η could not predict vision as in the above results.

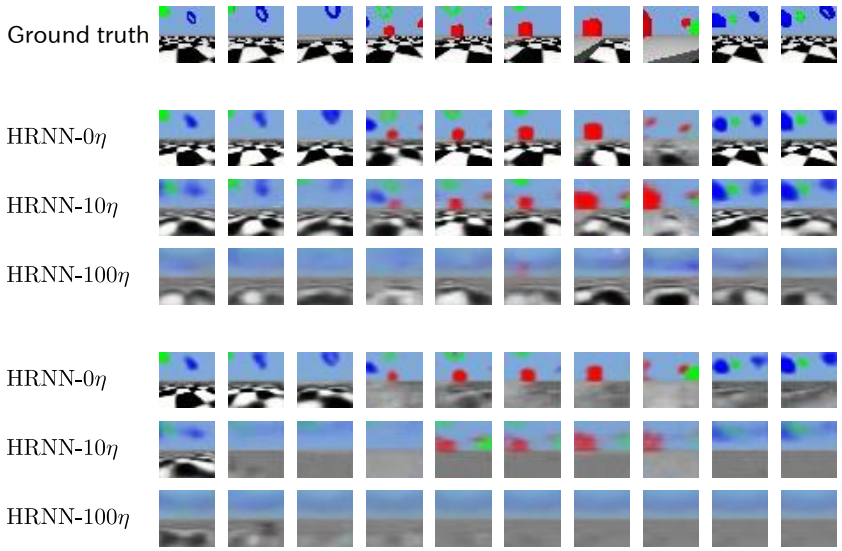

Figure 2.23: Examples of visual images predicted by trained HRNNs, along with a movement with $\eta = 0$. Top: the true visual images. Middle: predicted visual images. Bottom: visual images predicted using only motion. The results for different HRNNs with $\eta = 0, 10,$ and 100 (HRNN-0η, HRNN-10η, and HRNN-100η) are shown.

These results indicate that the HRNN was able to derive the internal model of the environment that was associated with external visual sequences if the randomness of the agent's movement (η) was not too high during training.

2.5.3 Effect of Behavior on Spatial Recognition

We visualize the internal states of the trained HRNNs for investigating the obtained internal recognition in the trained HRNNs. In order to visualize the internal states of the RNN layers, the dimensionality of the states was reduced to two dimensions with a principal component analysis. Figure 2.24 shows the visualized internal states of the slow RNN. The internal states for the HRNNs trained with various values of η are shown in the figure. The internal states were colored according to the current agent's position. For coloring the internal states, RGB values were assigned to each position. Red, blue, green, and yellow corresponded to the four corners of the arena, and linearly interpolated colors were assigned to other positions. In the case of HRNN-10η, the internal states were organized by color, i.e., spatial position, and it is considered that the HRNN recognized

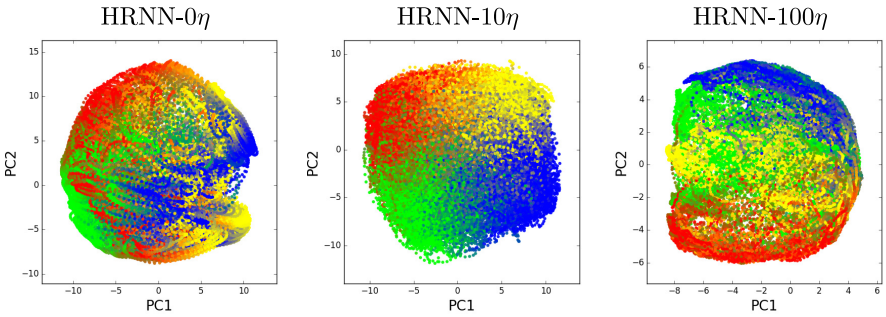

Figure 2.24: Internal states of slow RNN while predicting visuomotor sequences. The states of the various HRNNs trained with $\eta = 0, 10,$ and 100 (HRNN-0η, HRNN-10η, and HRNN-100η). Each point of the states is colored corresponding to the agent's current position as described in the main text.

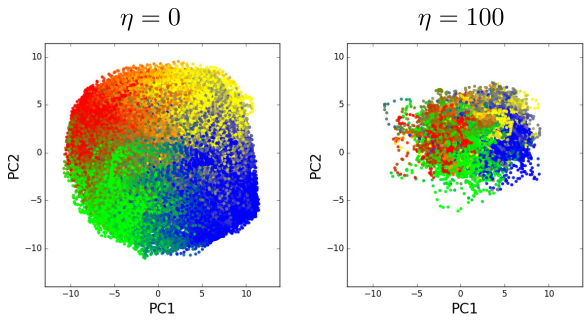

Figure 2.25: Internal states of slow RNN of HRNN-10η while predicting visuomotor sequences of unexperienced movement patterns with $\eta = 0$ and 100.

the spatial structure of the environment. In the cases of HRNN-0η and HRNN-100η, the internal states were somewhat organized by color, but different colors overlapped each other. These internal states are not considered as the internal model of space because these states were not arranged corresponding to the topological layout of the environment wherein the agent moved. It should be noted that the HRNN-0η did not obtain the internal model of the spatial structure while the HRNN-0η could predict the visual sequences from motion only. This may be because the structure obtained for HRNN-0η is not spatial but sequential. These results show that development of spatial recognition requires appropriate randomness of movement.

Figure 2.25 shows the internal states of HRNN-10η when the HRNN-10η received

visuomotor sequences produced with other values of η. In the case of the sequence with $\eta = 100$, the internal states were not organized by the agent's position as in Fig. 2.24. This is because the movement with $\eta = 100$ is almost random and the HRNN could not use sequential memory to recognize the agent's position. On the other hand, in the case of the sequence with $\eta = 0$, the internal states are organized by the agent's position although the internal model of the spatial structure could not be developed in the HRNN trained on the movement for $\eta = 0$ (HRNN-0η). This means that, once the spatial recognition was developed in an appropriate movement patterns, it could be used for other movement pattern, except for movement with too much randomness.

Evaluation of obtained internal model

In order to evaluate the spatial recognition of the trained HRNNs quantitatively, we constructed regression models for predicting the spatial position of the agent from the internal states. A similar method is used in real animal experiments to evaluate the place cell neurons [WM93]. If the HRNN obtained the internal model of the spatial structure in its internal states, the regression model can predict the actual position accurately from the states. The regression model outputs the prediction of the position by considering the internal states at each time step. We used the first and second principal component (PC) of the internal states in this evaluation for investigating how well the HRNN extracted the spatial structure as a low-dimensional representation. We used a linear regression model in this evaluation. Thus, the internal states at various positions were required to be arranged corresponding to the spatial arrangement of the positions for the accurate prediction of the positions. In this evaluation, to conduct a deeper investigate into how the spatial recognition depends on randomness η, a larger number of training visuomotor sequences were used with different values of η. We prepared visuomotor sequences with $\eta = 0, 0.1, 1, 5, 10, 50,$ and 100. We trained five different HRNN-$\alpha\eta$ for each $\eta = \alpha$ with different random initial configurations. To obtain the internal states used in this regression, the trained HRNNs received test visuomotor sequences. Visuomotor sequences with $\eta = 0$ (without any randomness of movement) were used for all the HRNN-ηs to eliminate errors caused by the randomness of the movement. The internal states after 50 steps in each sequence were used to optimize and evaluate the regression models.

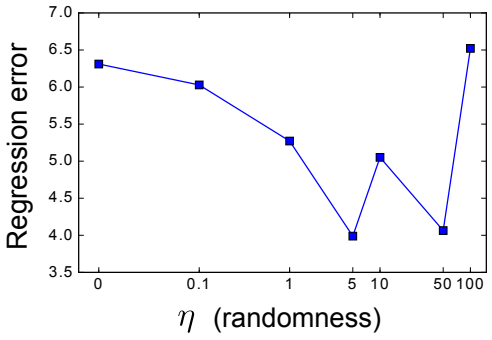

Figure 2.26: Errors in prediction of the agent's position using the internal states of slow RNN with linear regression models. Regression models are prepared for each HRNN trained with a different values of η. The graph is plotted using a log scale for η except near $\eta = 0$ where a linear scale is used.

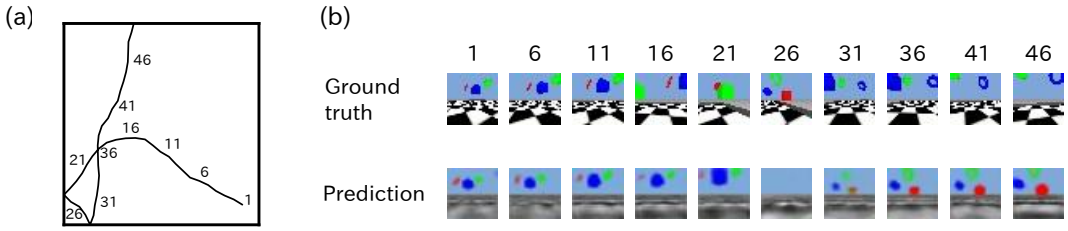

Figure 2.27: (a) Trajectory of movement for $\eta = 10$. (b) Visual sequence predicted by HRNN-0η using only motion according to the movement shown in (a). The numbers in the figures indicate the time steps of movement.

Figure 2.26 shows the evaluation results (errors of the actual positions from the internal states) of the regression models for different values of η. The regression errors for a small value of η ($\eta = 0$) and large value of η ($\eta = 100$) are larger than that for an intermediate value of η. This result is consistent with the visualization results of the internal states, i.e., the states are organized by spatial position for an intermediate value of η. This result also quantitatively shows that the internal model obtained for a very small value of η does not represent the spatial structure.

We further investigate the difference between the internal models obtained through movements with small randomness ($\eta = 0$) and intermediate randomness ($\eta = 10$). Figure 2.27 (a) shows a trajectory of the agent for $\eta = 10$, and Fig. 2.27 (b) shows the vision

predicted using only the motion for HRNN-0η according to the trajectory. The numbers in the figure indicate the time steps from the beginning of prediction. From steps 11 to 21, the agent turned; however, the predicted vision was not changed corresponding to the turning but changed as if the agent had proceeded in a straight line. Although the predicted vision changed at approximately 26 steps, the predicted vision subsequently also changed as if the agent were moving along a straight line. This indicates that the HRNN trained with a small-η movement only recognized that the agent is rebounded at the boundary and did not consider that the agent can turn at any position. In other words, if the HRNN was trained on the movement with a very small randomness, the HRNN could not develop the recognition of the spatial adjacency of the positions because of the limited exploration in the environment.

In above experiment, at least one of the external inputs (vision and motion) were always available to the HRNNs. In order to investigate the richness of the internal recognition in trained HRNNs, we let the trained HRNNs generate a visuomotor sequence as a mental simulation, wherein both the vision and motion of the external environment are not available. In the mental simulation, the outputs $\hat{\boldsymbol{v}}_{t+1}$ and $\hat{\boldsymbol{m}}_{t+1}$ of the HRNN were fed back into the inputs. The HRNN became an autonomous dynamical system and changed its internal states without providing external inputs. Figure 2.28 shows the internal states for the mental simulation. The internal states are mapped into the PC spaces which are the same as those used in Fig. 2.24 for each HRNN. The results for HRNN-0η and HRNN-10η are shown. For both the HRNN-0η and HRNN-10η, the internal states are shown for two different trials of mental simulation. In the case of HRNN-0η (small η), the internal states for different trials fall into different small subspaces in the mapped PC space. It appears that, once the state falls into the small subspace, it does not escape from there and becomes a limit cycle. This is because the movement for a very small randomness produces a periodic visuomotor sequence that depends on the initial position and angle of the agent, and the transition between different periodic sequences does not occur. Thus, the internal model obtained for the trained HRNN represents the sequential pattern of the trained sequences. In contrast, in the case of HRNN-10η (intermediate η), the internal state became wider than that of HRNN-0η although it did not cover the overall space. The internal states did not fall into a small subspace as in the case of HRNN-0η.

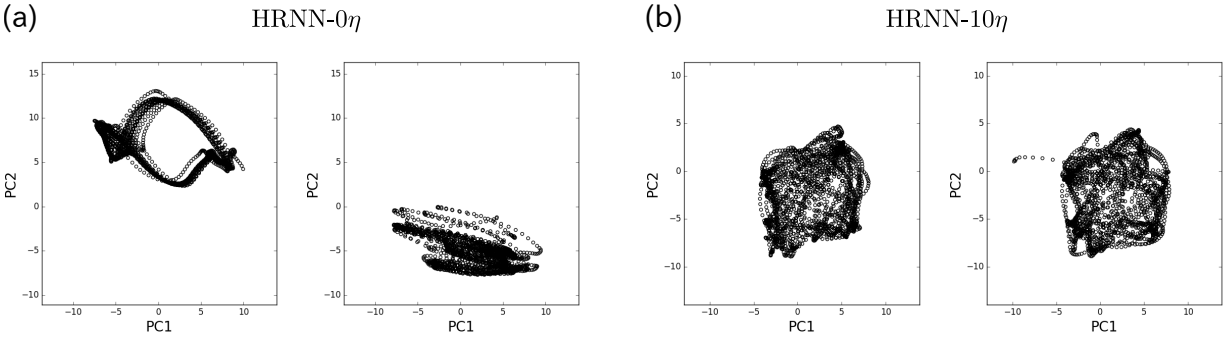

Figure 2.28: Internal states of slow RNN during mental simulation with 2,000 steps by (a) HRNN-0η and (b) HRNN-10η. The states of two trials of mental simulation are shown for both HRNN-0η and HRNN-10η.

This indicates that the wider region of the internal state's space was connected. This is because the visuomotor sequences were not deterministic, and the HRNN was required to consider the possible but not completely predictable future sequences. This means that the HRNN encloses the instability as possible actions and can imagine the positions in response to the actions. Thus, it constitutes the spatial recognition as a cognitive map.

2.5.4 Discussion

In previous studies, the movement pattern for the spatial translation is hand-tuned such that the spatial recognition is well developed [WKV06, FSW07]. In this section, we investigated how the movement pattern could affect the development of spatial recognition. We showed that spatial recognition is not developed if η, namely, the randomness of the movement (variation of turning), is too low or too high, and is developed if η has an appropriate intermediate value.

In the case of a small value of η, the HRNN cannot recognize the spatial relationships between the positions in the arena. In this case, the HRNN developed an internal model of the visuomotor sequences with respect to sequential events rather than spatial structure at such low randomness. This is because the HRNN could easily predict the input by remembering the visuomotor sequence, and there was no need to recognize the spatial structure. In the case of a large η, the HRNN could not predict the visual sequence from only the motion, and the spatial recognition was not developed. This is because of the

difficulty of prediction with high randomness of movement. In the case of high randomness of movement, there are many possible visual sensations in the next step, and the HRNN should take into consideration these possibilities. In the case of an intermediate value of η, the movement is along a straight line to some extent, and it is not difficult to estimate the vision of the next step based on the past sequence. The movement obtained for an intermediate value of randomness could be considered to be an intentional movement to achieve a goal. We considered that such intentional movement is required for the development of spatial recognition. In fact, the complexity of behavior varies during the developmental processes of human infants [TTK99]. To reduce the uncertainty of the environment or to achieve goals by performing intentional movements, it is necessary to reduce the complexity of the behaviors for new born infants; however, to obtain representations such as a cognitive map, the same complexity of behaviors as that in our model would be required. In our model, the developmental process is not implemented but it might be possible to explain the changing complexity of the behaviors of human infants in such a context using our model. The randomness of motion was required not only during development but also after cognitive skills have been developed. It is also shown that the behavioral variability or noise exists for adapting to changes of motor control system even in well skilled adult songbirds [TB07]. For investigating the effect of variability of motion shown in human infants and songbirds, the developmental process should be introduced.

2.6 Development of the Shared Spatial Representation in Different Environments

Natural world has different visual appearances between different environments. Even in such variety of the external sensory inputs from the environment, we can recognize the world with the same sense of spatial metric. Such sense of shared spatial metric between different environments is considered to be provided by the sense of self-motion. The sense of motion is always consistent between different environments and animals can have the consistent spatial metrics by using self-motion. In fact, even in a dark environment, the grid cells and head-direction cells change their activity according to the rats' spatial movements [MBJ$^+$06]; that means rats can keep track of its spatial

position and orientation even without external sensory cues. By using self-motion, these spatial cells can have shared representations between different environments with different visual appearances. Such shared representation allows the rats to use the same metrics in different environments and is necessary for navigation for their lives.

In this section, we propose a model that can develop the shared spatial representations between different environments through visuomotor integration. In addition to investigate how the spatial representation is shared between different environments, we simulated the development of the spatial representations of place and direction as differentiated representations. For realizing development of such shared representation of place and direction, we constructed another kind of hierarchical recurrent neural network model. The proposed neural network model has three recurrent layers that have different properties on available motion inputs or a way of connection each other. The proposed model was equipped on a simulated mobile agent and trained on visuomotor experiences of the agent in two different environments with different visual appearances. In the following, we showed that the representations of place, direction and visual appearances of the environments were developed in different layers of the proposed network because of difference of available motion inputs.

2.6.1 Recurrent Neural Network for Developing Shared Spatial Recognition

A schematic view of the network is shown in Fig. 2.29. The network receives visuomotor inputs and predicts the next visual images as same as the HRNN. The network mainly consists of three recurrent layers and the convolutional neural networks (CNN) [LBBH98] for recognizing and generating visual images.

The network is constructed so that it has more reliability on motion inputs than visual inputs. All the three RNNs can use the visual inputs, however, visual inputs are masked with a probability of 99% while the motion inputs are always provided. Therefore, for correctly predicting visual sequences, the network should update its internal recognition of agent position and orientation according to agent's motion.

The three RNNs are called the rotational, translational, and visual RNNs according

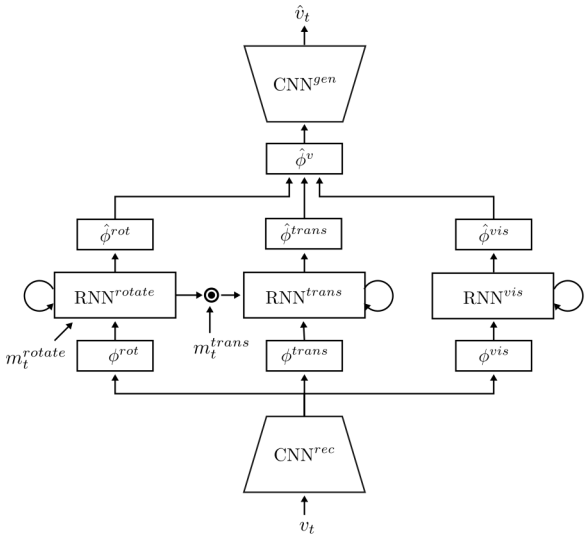

Figure 2.29: The proposed network model.

to their available inputs. The rotational and translational RNNs receive the agent's rotational and translaional velocities, respectively. The visual RNN receives only visual inputs and does not receives self-motion related inputs. Especially, the translational RNN receives the internal states of the rotational RNN in addition to the translational velocity. Concretely, the product of the rotational RNN's states and the translational velocity is fed into the translational RNN. The use of translational velocity is based on the previously proposed model of path-integration in rats [MBJ+06]. Visual prediction is realized by using the outputs of all the RNN layers and we expected that the RNN layers develop different internal representation for predicting vision under their different properties. It is expected that the rotational RNN develops the directional representation because of their input of the rotational velocity. If the rotational RNN's states have directional representation, the input for the translational RNN (the product of the rotational RNN's states and translational velocity) can represent which direction the agent moves, and the translational RNN can recognize the change of the agent's position. As the visual RNN does not receive self-motion inputs, the visual RNN can have the representation only related to visual appearance of the environments.

In each simulation step, the network receives visual and motion inputs and predicts the visual image at the next time step. Firstly, the CNN module receives visual input

and transforms it into a feature vector $\boldsymbol{f}_t^v$ as follows:

$$\boldsymbol{f}_t^v = \mathrm{CNN}^{rec}(\boldsymbol{v}_t), \tag{2.14}$$

where CNN^{rec} is visual recognizer that consists of convolutional layers. Then, the feature $\boldsymbol{f}_t^v$ are transformed again into feature vectors $\boldsymbol{f}_t^{rot}$, $\boldsymbol{f}_t^{trans}$, and $\boldsymbol{f}_t^{vis}$ for the three RNNs as follows:

$$\begin{gathered}\boldsymbol{f}_t^{rot} = \phi^{rot}(\boldsymbol{f}_t^v), \quad \boldsymbol{f}_t^{trans} = \phi^{trans}(\boldsymbol{f}_t^v), \\ \boldsymbol{f}_t^{vis} = \phi^{vis}(\boldsymbol{f}_t^v),\end{gathered} \tag{2.15}$$

where ϕ^{rot}, ϕ^{trans}, and ϕ^{vis} represent the functions of fully connected layers. Then, the feature vectors are passed to three RNN layers and the RNNs update their internal states as follows:

$$\boldsymbol{h}_t^{rot} = \mathrm{RNN}^{rot}(m_t^{rot}, \mathrm{M}(\boldsymbol{f}_t^{rot})), \tag{2.16}$$

$$\boldsymbol{h}_t^{trans} = \mathrm{RNN}^{trans}(\boldsymbol{h}_{t-1}^{rot} \odot m_t^{trans}, \mathrm{M}(\boldsymbol{f}_t^{trans})), \tag{2.17}$$

$$\boldsymbol{h}_t^{vis} = \mathrm{RNN}^{vis}(\mathrm{M}(\boldsymbol{f}_t^{vis})), \tag{2.18}$$

where $\boldsymbol{h}_t^{rot}$, $\boldsymbol{h}_t^{trans}$, and $\boldsymbol{h}_t^{vis}$ are the internal states of the RNNs, and RNN^{rot}, RNN^{trans}, RNN^{vis} are the functions of the RNNs, and M is masking function that replaces the input vector by zero with probability of 99%. As described above, the rotational RNN receives rotational velocity m_t^{rot} and the translational RNN receives the product of the rotational RNN's state $\boldsymbol{h}_{t-1}^{rot}$, which is in the previous time step $t-1$, and the translational velocity m_t^{trans}, in addition to the visual feature. The outputs of the RNNs are then transformed into another feature vector $\hat{\boldsymbol{f}}_t^v$ as follows:

$$\hat{\boldsymbol{f}}_t^v = \hat{\phi}^v(\hat{\boldsymbol{f}}_t^{rot}, \hat{\boldsymbol{f}}_t^{trans}, \hat{\boldsymbol{f}}_t^{vis}), \tag{2.19}$$

where,

$$\begin{gathered}\hat{\boldsymbol{f}}_t^{rot} = \hat{\phi}^{rot}(\boldsymbol{h}_t^{rot}), \quad \hat{\boldsymbol{f}}_t^{trans} = \hat{\phi}^{trans}(\boldsymbol{h}_t^{trans}), \\ \hat{\boldsymbol{f}}_t^{vis} = \hat{\phi}^{vis}(\boldsymbol{h}_t^{vis}),\end{gathered} \tag{2.20}$$

and $\hat{\phi}^{rot}$, $\hat{\phi}^{trans}$, $\hat{\phi}^{vis}$, and $\hat{\phi}^v$ are the function of fully connected layers. Finally, the visual prediction $\hat{\boldsymbol{v}}_{t+1}$ is generated from $\hat{\boldsymbol{f}}_t$ by using another convolutional network CNN^{gen}, which consists of the transposed convolutional layers [ZKTF10], as follows:

$$\hat{\boldsymbol{v}}_{t+1} = \mathrm{CNN}^{gen}(\hat{\boldsymbol{f}}_t). \tag{2.21}$$

Table 2.7: Structure of CNN^{rec} and CNN^{gen}

CNN^{rec}

layer	type	size	channel	kernel	stride	padding	activation
1	input ($\boldsymbol{v}_t$)	32 × 32	3	-	-	-	-
2	conv.	16 × 16	16	3 × 3	2 × 2	1 × 1	ReLU
3	conv.	8 × 8	32	3 × 3	2 × 2	1 × 1	ReLU
4	conv.	4 × 4	64	3 × 3	2 × 2	1 × 1	ReLU
5	fully connected	1 × 1	128	-	-	-	ReLU
6	fully connected	1 × 1	64	-	-	-	ReLU

CNN^{gen}

layer	type	size	channel	kernel	stride	padding	activation
1	input ($\hat{\boldsymbol{f}}_t^v$)	1 × 1	64	-	-	-	-
2	fully connected	1 × 1	128	-	-	-	ReLU
3	fully connected	4 × 4	64	-	-	-	ReLU
4	transposed conv.	8 × 8	32	4 × 4	2 × 2	1 × 1	ReLU
5	transposed conv.	16 × 16	16	4 × 4	2 × 2	1 × 1	ReLU
6	transposed conv.	32 × 32	3	4 × 4	2 × 2	1 × 1	tanh

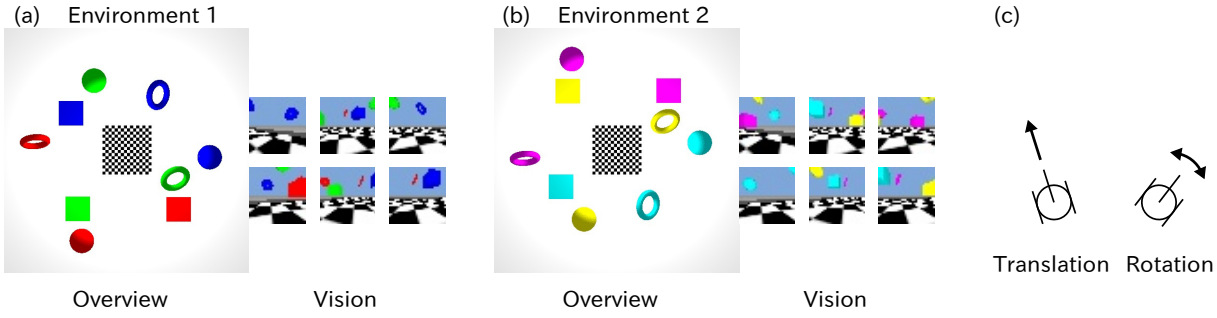

Figure 2.30: The simulated environments (a, b) and agent (c). Two different environments with different visual appearances are simulated. The agent's possible movements are moving forward (translation) and turning left or right (rotation).

The LSTM (long short-term memory) network [HS97] was used for all RNNs, which have 64 hidden units. All of the fully connected layer (ϕ^{rot}, ϕ^{trans}, ϕ^{vis}, $\hat{\phi}^{rot}$, $\hat{\phi}^{trans}$, $\hat{\phi}^{vis}$, and $\hat{\phi}^{v}$) has 64 neurons and ReLU function is used for their activation function. The structure of CNN^{rec} and CNN^{gen} are shown in Tab. 2.7.

2.6.2 Simulation and Training

To simulate the development of place and head-direction cells, a mobile agent is modeled to move around in two different simulated environments. The agent is equipped with a neural network model and learns to predict visuomotor sequences. The details of simulation are described below.

Simulated agent and environment

The simulated agent and environment are shown in Fig. 2.30. There are two environments with different visual appearances (referred as environment 1 and environment 2). The agent can move around in the two-dimensional flat arena while receiving visuomotor sensory inputs, i.e., visual images captured by a head-mounted camera and the agent's velocity as self-motion. There are several floating colored objects that constitute a landscape for agent's visual experiences. The arena wherein the agent can move around is enclosed by a fence. The fence is low and does not obstruct the agent's view. The size of the arena, which is same between the two environments, is 20×20 units of distance.

The agent can move around by choosing one of three pre-defined discretized actions, i.e., moving forward and turning left or right, at each step. The speeds of forward and turning are one unit distance and 15 degree per step, respectively. The agent's motion is autonomously controlled to move toward a destination, which has been randomly selected within the arena. To reach the destination, the agent firstly turns toward the destination and then moves forward. When the agent reaches the destination, a new destination is selected. Additionally, to avoid monotonous movement patterns, the destination is changed with probability of 0.02 each step regardless of whether or not the agent reaches the destination.

The agent's motion is represented by rotational velocity m_t^{rot} and translational m_t^{trans}, each of which has scalar value. The size of the visual image $\boldsymbol{v}_t$ captured by agent's camera is 32×32, and each pixel of the image has three channels (RGB). The range of value of each element in the visual images is $[-1, 1]$.

Training data For the training, the agent moved in the arena and visuomotor sequences were collected. We collected 100 visuomotor sequences for the environment 1 and 2; each sequence comprised of the agent's visual and motor sensory inputs in 1,000 simulation steps. Totally, 200 visuomotor sequences were collected for the training. The initial position and orientation of the agent were randomly determined at the beginning of each sequence.

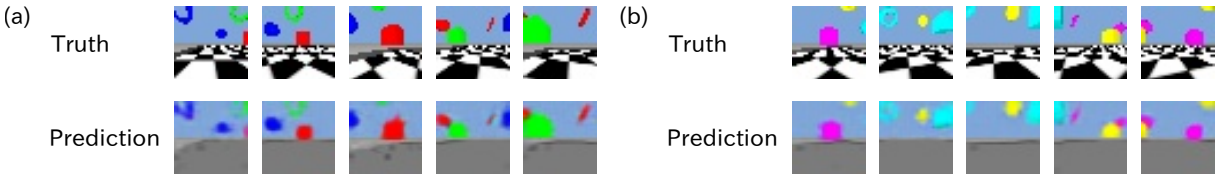

Figure 2.31: The predicted visual sequences by the trained model for both environments: the environment 1 (a) and 2 (b). The ground truth of visual sequences are also shown. The visual images for each 5 time steps in successive sequences are shown. The random mask for visual inputs was applied as same as during training.

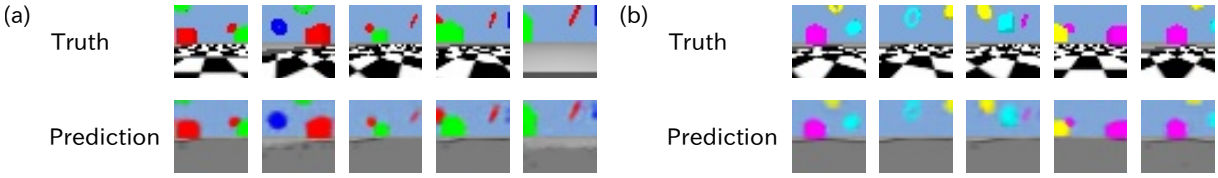

Figure 2.32: The predicted visual sequences without visual inputs in the environment 1 (a) and 2 (b). The ground truth of visual sequences are also shown. The visual images at time steps of 960, 970, 980, 990, 1000 are shown.

Prediction learning The proposed network is trained to predict the visual sensory inputs of the agent. The objective of the training is to minimize the visual prediction error. The vision error E^v was calculated as mean squared error. The cross-modal prediction learning in the training of the HRNN was not used for this model. Instead, the visual input masking realized the visuomotor integration for developing the spatial representation.

The parameters of the network were optimized by using the Adam algorithm [KB15]. For a single update of the parameters, 100 steps of the sequence were used and BPTT was performed for the 100 steps; consequently, 10 updates was performed in a single training sequences. The mini-batch size was 10. The network was trained 300 times over the training sequences.

Training Results

In Fig. 2.31, the samples of predicted visual sequences by the trained network are shown with ground truth sequences. As shown in the figure, the trained network could correctly

predict the visual landmarks according to the agent's movement. Considering that the visual inputs to the RNN layers were masked with 99% probability, these results indicate that the trained network could update its internal recognition by using the inputs of rotational and translational velocities.

To evaluate how long the trained network can keep producing the agent's visual sequence, we let the trained network to predict sequences where the visual inputs were not provided except for the first step of the sequence. The predicted visual sequences without visual inputs are shown in Fig 2.32. The predicted and true visual images in the latter of the sequence (from 950 to 1000 steps) are shown in the figure. Even though the visual inputs were not provided during the prediction, the trained network could correctly predict visual sequences for long time steps. These results indicate that the trained network developed an internal representation that was updated by using the agent's motion for predict visual sequences.

2.6.3 Development of Spatial Representations of Place and Direction Shared in Different Environments

To investigate what kinds of internal recognition was developed in the trained network, we visualize the internal states of the trained network. For the visualization, the dimensionality of the internal states was reduced to two dimensions by principal component analysis (PCA). PCA was applied to the internal states for both environments. Figure 2.33 shows the visualized internal states of RNNs. In the figure, while the internal states are shown separately by two different environments, the same areas of the PC space are shown for both of two environments. The visualized states of the rotational and translational RNN are colored according to the agent's head-direction and spatial position, respectively. In the case of the rotational RNN, the internal states formed a circle and were arranged according to the agent's head-direction. Further, the states were mapped on the same region for both environments. The arrangements of the color (the head-direction) were different between environments; this is because the network developed the internal recognition based on not absolute coordinates of the simulation but subjective visuomotor inputs. The representation in the rotational RNN reflected the agent's head direction and not vi-

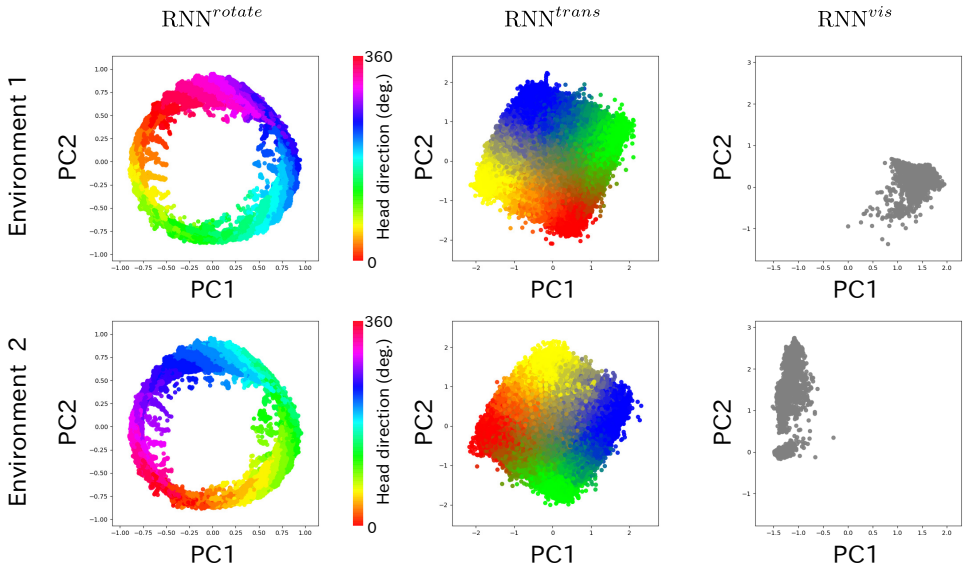

Figure 2.33: The results of principal component analysis. The internal states of the rotational (left), translational (middle) and visual (right) RNNs are shown for two environments; top for environment 1 and bottom for environment 2. The rotational RNN's states are colored according to the agent's head direction. For coloring the translational RNN's states, RGB values were assigned to each position in the arena; red, blue, green and yellow corresponded to the four corners of the arena, and linearly interpolated colors were assigned to other position.

sual appearance, and can be considered the head-direction cell-like representation. In the case of the translational RNN, the internal states were organized by color, i.e., the spatial position. The states for different environments were mapped on the same region and the arrangements of the color were different between environments as same as the rotational RNN. The representation in the translational RNN can be considered as the place cell-like spatial representation. Different from the rotational and translational RNN, the internal states of the visual RNN formed two separated clusters according to the difference of the environments. Because the visual RNN represented the difference of environments, the network could correctly predict visual sequences where the position and direction were represented by the same internal states between different environments.

Next, we confirmed if the visual RNN represented the difference of environments by swapping the visual RNN's states between two environments. First, the network receives

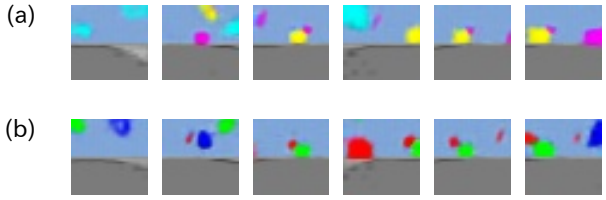

Figure 2.34: The predicted visual sequences in the environment 1 where the internal states of the visual RNN were replaced by that in the environment 2 (a). The predicted images when the internal states were not replaced are also shown (b).

the visuomotor sequences in the environment 2 and the sequence of the visual RNN's internal states was recorded. Then, the network receives the visuomtor sequences in the environment 1 and predict visual sequences where the sequence of the visual RNN's states were replaced by the that were recorded in the environment 2. The sequence of the rotational and translational RNN's states were kept original as in the environment 1. Figure 2.34 shows the predicted visual images using the original states and the recorded states. Although the internal states of the rotational and translational RNN were kept as is in the environment 1, the predicted visual images clearly showed visual features of the environment 2. This result shows that the difference of environments was recognized by the visual RNN. In other words, the rotational and translational RNN did not represent the difference of environments, at least it did not have effects on the visual prediction. Note that the predicted visual images by the original and recorded states in Fig. 2.34 showed the similar visual flow (the objects moved the same way in the images). This is because the same states of the rotational and translation RNN were used. This also indicates that the representations of direction and position in the rotational and translational RNN were shared between two environments.

2.6.4 Discussion

Our proposed network model could successfully develop the prediction ability in a visuomotor integrated way as a result of training. By using motion inputs, i.e., the rotational and translational velocities, our network could develop the internal representation that changed according to the agent's motion. In fact, our network could predict the agent's

visual sequences long time using only motion inputs (Fig. 2.32). The internal states analysis showed that our network developed the directional and positional representations in the rotational and translational RNNs, respectively (Fig. 2.33 left and middle). The developed directional and positional representations were efficient for keeping track of the agent's spatial position and orientation. In such spatial representation, certain positions or directions were uniquely mapped onto the internal states of the network. As a result, our network had not to memorize the association between patterns of visual and motion sequences; our network predicts vision by using the spatial map like representation rather than by memorizing the orders of visual images that have been seen during the training, which were determined by the agent's motion patterns. Moreover, the spatial representations were shared between different environments. By representing the difference of environments in the visual RNN (Fig. 2.33 right), our network could correctly predict visual sequences in two different environments with the same the internal spatial representation. It means that our network used the same spatial metrics in different environments. Such shared spatial representation could not be realized by the models like [WKV06, FSW07] that use only visual sensory inputs because, in different environments with different visual appearance, it is not possible to extract the shared spatial concepts from only visual inputs. Our results showed that the spatial representation shared between different environments with different visual appearances could be developed by prediction learning with visuomotor integration.

Banino et al. also proposed the RNN model that can develop the grid-like spatial representation, which is similar to the grid cells found in rats' brain [BBU+18], that works as spatial metrics independent from visual appearance. However, objective inputs of position and orientation are provided externally as teaching signals and the teaching signals are always same between different environments. In that sense, the development of shared spatial representation in their model is considered as a trivial result. In contrast to their model, our proposed network was trained on only subjective visuomotor inputs and the directional, positional and visual representations were self-organized in different RNN layers. We considered that our results showed that subjective visuomotor experiences are sufficient to develop the spatial representation that works in different environments.

2.7 Conclusion

In this chapter, we conducted experiments for investigating how the spatial representation can be developed only through the learning of visuomotor experiences where the spatial position or direction, and spatial relationship between places were not explicitly presented.

In section 2.2, we proposed the HRNN model for simulating the development of the spatial representation. The HRNN had no prior knowledge about the spatial structure; no specific functions were implemented in advance to the experiences. The training of the HRNN was designed as the visuomotor prediction learning as in a visuomotor integrated way. This prediction learning is unsupervised learning and did not provide any explicit information of the spatial position and spatial relationships between places. In later sections, the development of the spatial recognition was simulated using the HRNN in various conditions.

In section 2.3, the HRNN was trained on the visuomotor experiences of the simulated mobile agent and the development of the recognition of the spatial structure was investigated. As a result, the HRNN developed the representation of the spatial structure in the higher RNN and it could recognize the agent spatial position even in the un-experienced area. By comparing the developed representation between different conditions of the prediction learning, it was shown that the long-term prediction learning without external visual inputs (PoM task) was necessary for developing the spatial representation.

In section 2.4, the development of the spatial representation in the HRNN from the visuomotor experiences collected in the real environment was simulated. The difference of the time scale in RNNs was introduced by CTRNN and the representations of the spatial position and direction were developed as slow and fast features in the visuomotor sequences. This experiment demonstrated that the scalability of the HRNN model that it can develop the spatial recognition from the realistic visuomotor experiences.

In section 2.5, for investigating how the spatial recognition is affected by the behavior, the HRNN was trained in various visuomotor sequences with various complexity of the behavior. As a result, it was shown that the representation of spatial structure was not developed when the behavior was too deterministic or random and the moderate randomness or complexity of visuomotor experiences was necessary for developing the

spatial representation.

In section 2.6, the development of the spatial representation that had the same spatial metric between different environments was simulated in another proposed RNN model. The model had RNN modules that separately receive the translational and rotational velocities, respectively, and developed the representations of spatial position and direction through only visual prediction learning. Further, by dealing with visual characteristic of different environments with another RNN module, the developed spatial representations were shared between different environments. The model also was not designed to have any specific function for recognizing spatial structure and such spatial representation shared between different environments was developed through only prediction learning.

As summarized above, the experiments in this chapter showed that the spatial recognition can be developed only through the learning of visuomotor prediction learning. Especially, the HRNN developed the spatial representation of the environment on various kinds of visuomotor experiences in various environments although the structure of the RNN was generally same in these experiments. As discussed in section 2.5, for predicting visuomotor sequences along with spatial movements with adequate complexity, it is more effective to recognize the spatial structure than to memorize the sequential structure of the visuomotor sequences to predict the sequences. Therefore, no matter what the environment is, the HRNN could developed the spatial representation of the environments through visuomotor experiences in the environment. That is because the HRNN did not assume any specific spatial structure of the environment. These results suggest that the spatial recognition of animals is not a predefined ability but a developed ability as a result of a generalization of their experiences.

Chapter 3

Development of the Spatial Navigation in Hierarchical Recurrent Neural Networks

3.1 Introduction

The spatial representation like place or grid cells contribute to spatial navigation [MGRO82, BBMB15]. However, the spatial representation itself is not sufficient to perform spatial navigation and it should be properly integrated with spatial navigation behavior. Spatial navigation requires planning of routes to reach the navigational goal and animals should actively use the spatial representation to find paths which lead to the navigational goal. Actually, frontal lobe which generally involves planning become active when human is solving spatial navigation [EPJS17]. However, how the recognition of the spatial structure is integrated into navigation ability are not understood well.

In the research of mobile robotics, simultaneous localization and mapping (SLAM) is studied for realizing a robot that can recognize its spatial position. Using SLAM algorithm, a robot can construct a spatial map through exploration of its environment. However, SLAM algorithm was designed by humans and is only suitable for constructing a map of the environment and localizing it in the map. Consequently, the navigation algorithm was developed separately from SLAM algorithm, and the integration of localization and navigation was also designed by humans. There exist studies that propose models

that integrate the development of spatial representation and navigation or some other cognitive functions (e.g., language) [THTI17, EAE+15]. However, the algorithms for constructing such spatial representation or other functions are still designed by humans such that roles of the modules in the models or sensory information are defined in advance. Although some SLAM algorithms can be a model for explaining how the mechanisms of spatial recognition work in the brain [ZS17, TYT17], how the spatial recognition ability can be developed is not considered. Therefore, a model that can develop the spatial recognition and navigation from no predefined functions needs to be developed.

Recently, advanced deep learning approaches are able to simulate development of the ability to recognize high-dimensional data or perform complex tasks from no predefined functions [LBH15]. Spatial navigation ability is also modeled using deep neural network (DNN) [MBM+16, CSP+18, PSN+17, PML+, SNS18]. These navigation models with DNN were trained to reach a specific target indicated by certain features (e.g., visual images, language sentence, or relative position of target). The training of these models was conducted in an end-to-end manner wherein the models have no knowledge about their environments in advance. The trained models could recognize complex high-dimensional inputs like vision and effectively navigate to the targets even in unknown situations. However, it was not shown that the models developed spatial representation. In fact, the navigation behaviors were achieved not by considering the spatial position of the target and navigating agent (e.g., by comparing current and target visual images [PML+], by searching around the target object [CSP+18], or by moving toward the target given by a relative position [PSN+17]); consequently, shortcut behavior was not realized here. In other words, although these DNN models could develop complex navigation ability from no predefined functions, the navigation using spatial recognition such as a cognitive map was not realized.

In this chapter, we extended the HRNN model proposed in the previous chapter and proposed the navigational HRNN (NHRNN) model for realizing the development of navigation ability based on a self-organized spatial representation; another RNN for controlling spatial navigation was added. As same as the previous chapter, only subjective visuomotor inputs are available for the NHRNN and the NHRNN had no pre-defined functions, for investigating the self-organization of the spatial representation and the spatial navigation

ability. The previous HRNN model could develop the spatial representation by prediction learning of visuomotor experiences. The NHRNN model learned and performed spatial navigation by generating visuomotor sequences as same as prediction learning. Thus, it is considered that the NHRNN can learn the recognition of the spatial structure and spatial navigation as an unified process, i.e., generation of vision and motion. Consequently, it is expected that the NHRNN could develop the spatial navigation integrated with the self-organized spatial representation.

This chapter is organized as follows. In section 3.2, the structure of the NHRNN model and how the NHRNN learns spatial navigation were described. In section 3.3, the NHRNN was trained in a simple environment where there is no obstacles and we show that the abilities of bottom-up spatial recognition and top-down navigation control were simultaneously developed through the training of the spatial navigation in such simple environment. In section 3.4, the NHRNN was trained in a maze like environment which had some obstacles and changed its structure by changing arrangement of the obstacles. Through the training of this maze environment, the spatial navigation ability based on the spatial representation was developed for effectively perform navigation in various structures of the maze. In section 3.5, we summarized this chapter and discussed about the contribution of the demonstrated results in the experiments to the development of the spatial navigation ability.

3.2 Navigational Hierarchical Recurrent Neural Networks

Spatial navigation requires bottom-up recognition of the external sensory inputs and top-down intentional control of behaviors. The HRNN in the previous chapter realized the bottom-up recognition of spatial position and somehow top-down process in the sense that the HRNN could recognize position even in an unexperienced area; however, the intentional top-down process was not considered in the HRNN. In this section, we extended the HRNN by adding another RNN layer for controlling spatial navigation behavior as a top-down process.

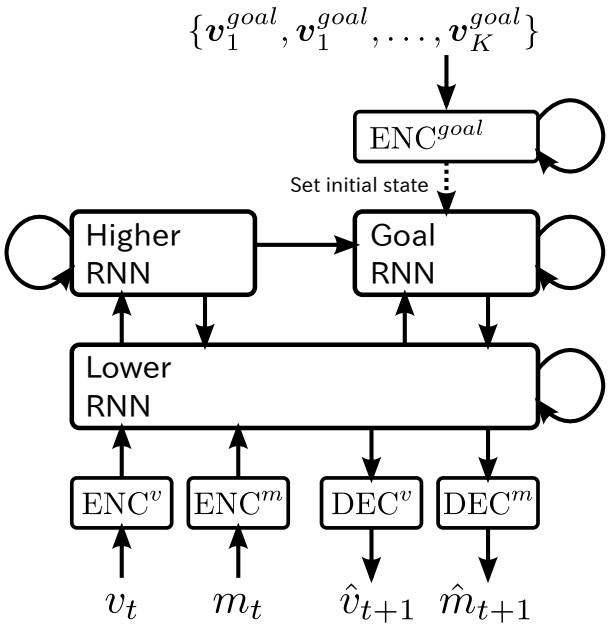

Figure 3.1: The schematic of the navigational hierarchical recurrent neural network (NHRNN)

3.2.1 Structure of Navigational Hierarchical Recurrent Neural Networks

A schematic view of the extended model, which we call navigational HRNN (NHRNN), is shown in Fig. 3.1. The NHRNN is constructed to generate navigation behavior by receiving the The NHRNN has three RNN layers, namely, the lower and higher RNNs as same as the HRNN and the goal RNN. The goal RNN controls the generation flow of the lower RNN by setting the initial states of the internal states, as described later.

When the functions of the goal RNN and RNN^{goal}, the equations of these RNN layers of NHRNN in one-step processing is expressed as follows:

$$\boldsymbol{h}_t^{lower} = \text{RNN}^{lower}(\boldsymbol{f}_t^v, \boldsymbol{f}_t^m, \boldsymbol{h}_{t-1}^{higher}, \boldsymbol{h}_{t-1}^{goal}), \tag{3.1}$$

$$\boldsymbol{h}_t^{higher} = \text{RNN}^{higher}(\boldsymbol{h}_{t-1}^{lower}, \boldsymbol{h}_{t-1}^{higher}), \tag{3.2}$$

$$\boldsymbol{h}_t^{goal} = \text{RNN}^{goal}(\boldsymbol{h}_{t-1}^{lower}, \boldsymbol{h}_{t-1}^{higher}, \boldsymbol{h}_{t-1}^{goal}), \tag{3.3}$$

where $\boldsymbol{h}_t^{goal}$ is the internal states of the goal RNN. The visual output $\hat{\boldsymbol{v}}_{t+1}$ and motion output $\hat{\boldsymbol{m}}_{t+1}$ are generated as same as in the HRNN by using $\boldsymbol{h}_t^{lower}$.

Optimizing the initial states of a goal RNN

For performing navigation, our model had to generate the predicted motion sequences required to visit the destinations. As described later, the destinations are indicated by the subjective visual images and the multiple destinations might be given for a single navigation, in the following experiments. Control of the navigation to the destinations is realized by the goal RNN. Because the goal RNN does not directly connect raw-level input or output, it can deal with long-term dynamics, such as motion generation as a top-down process. Generating goal-directed behaviors has been shown to be done by optimizing the initial states of the RNN with a genetic algorithm or backpropagation [PT05, NNT08]. Especially, they used an RNN with slow time scales for controlling goal-directed behaviors. In their previous studies, the values of the initial states themselves were directly optimized. Instead, in this study, the initial states were generated from the visual images of destinations of navigation task. Inspired by encoder-decoder network [CVMG+14], the images of destinations are encoded into the intentional states by the goal encoder module. Receiving the images of destinations, the goal encoder can encode the information of the destinations into its output, and the output can be used as the initial states for goal-directed motion.

The encoder RNN received the images of the destinations as sequential inputs and transformed them into a vector as the initial states of the goal RNN $\boldsymbol{h}_0^{goal}$.

$$\boldsymbol{h}_0^{goal} = \mathrm{ENC}^{goal}(\boldsymbol{v}_1^{goal}, \boldsymbol{v}_2^{goal}, ..., \boldsymbol{v}_K^{goal}) \tag{3.4}$$

where ENC^{goal} is the goal encoder and $\boldsymbol{v}_k^{goal}$ is the images of the k-th destination for the navigation where K is the number of the destinations. Although RNN is suitable for the goal encoder in the case that the number of the destinations K is variable, both feedforward and recurrent neural networks can be used as the goal encoder. The goal encoder is optimized to minimize the errors between the predictions and teaching sequences in the same manner as the other modules.

3.2.2 Learning of Spatial Navigation

The NHRNN is trained to generate visuomotor sequences as same as in the training of the HRNN. When no goal for navigation are given the NHRNN is just trained to predict

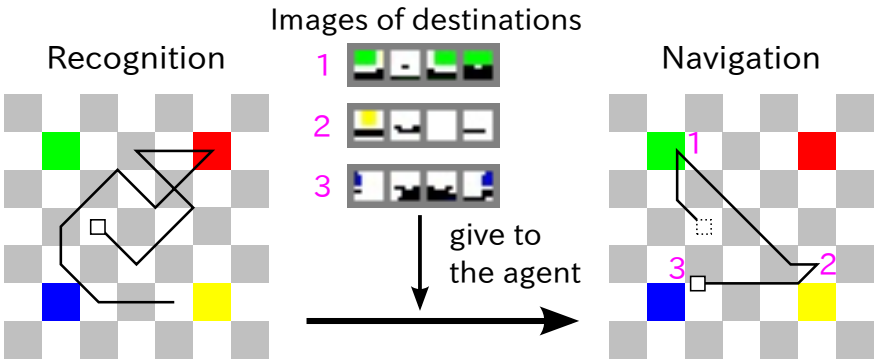

Figure 3.2: The task for the robot. In the recognition phase, the robot freely moves in the environment. In the navigation phase, the robot should follow the shortest path for achieving given destinations.

future visuomotor inputs. In the prediction learning without navigational goals, it is expected that the NHRNN develops the spatial representation as same as the HRNN. When some goals are given the NHRNN is trained to generate visuomotor sequences along with visuomotor sequences performed by expert's navigation behavior; the training of the navigation is conducted in a imitation manner. For performing navigation according to the given images, the NHRNN generates the initial states of the goal RNN from the destination images by the goal encoder. The goal encoder is trained only when navigation goals are given. The objective of the learning was minimizing the errors between the predicted (or generated) and teaching sequences in both the recognition task and goal-directed task. For multimodal integration, the three way training of PVM, PoV, and PoM used in the training of the HRNN is also used in the training of the NHRNN.

3.3 Spatial Navigation in Simple Environments

In this section, we trained the NHRNN model to perform spatial navigation in a simple simulated environment.

3.3.1 Navigation Task and Training

Navigation Task

In order to investigate how voluntary spatial movements are realized through the integration of rich visual information and motion, we designed a task in which the mobile robot performed navigation behaviors in a simulated environment. In this simulation, a mobile robot similar to the experiment in section 2.3 is simulated.

The task for the agent was navigating the destinations shown as the visual images of the agent. The navigation task consisted of two phases, namely the recognition phase and the navigation phases (Fig. 3.2). In the recognition phase, the agent freely moved within the arena and can recognize it spatial location. In the navigation phase, the agent moved toward the destination points indicated by the images captured at the destination points. The agent was abruptly given the visual images for the destination points, and the agent had to navigate to the places where the visual input was the same as the images of destinations given. There could be multiple destination points. If multiple visual images were given, the agent was supposed to visit them in the same order as the given images. The visual images of the destinations were given once: when the phase changed to the navigation phase. The two phases were successively performed as described below. The visual images of the destination points were given at a certain time in the recognition phase. Once the agent received the images of the destination points, the agent was expected to perform navigation behavior so that it visits the given destination points. To accomplish the navigation task, the agent had to recognize its current position while moving around during the recognition phase and consider how to get to the destinations from the position that the agent recognized.

Training

Training Settings

The training data is constructed as follows. First, the robot moves around for 300 time steps while the destination points randomly changes with 10% at each time step. The motion sequences and the visual images during this are used for the training data for the recognition phase. Next, soon after 300 time steps, the target visual images (the number

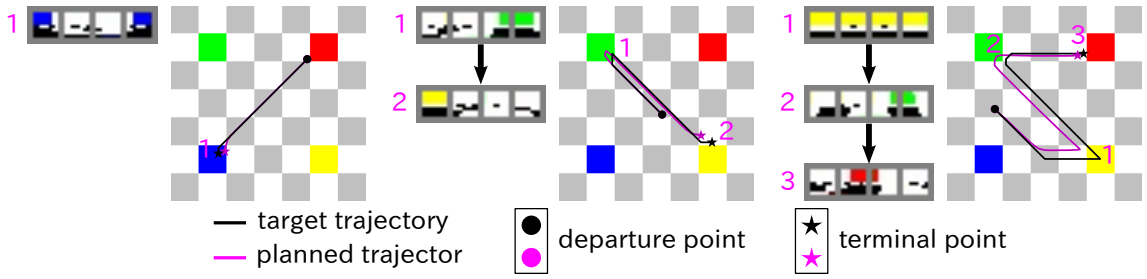

Figure 3.3: Examples of generated navigation behavior by trained model. One sample is shown for each condition where the number of destinations is from 1 to 3.

of targets is randomly decided from 1 to 3) are given and set the initial states of the goal RNN. The robot moves to visit all the target destinations in a designated order for 50 time steps. The motion sequence is given by the designed controller. When the robot arrives at all targets before 50 time steps, the robot stays there. The motion sequences and visual images during this navigation are used for the training data for the navigation phase. 100 sequences for the recognition phase are prepared for training. 10 different navigation sequences are created after a single recognition phase sequence, which means there are $1,000 (= 100 \times 10)$ sequences for the navigation sequences for training. To evaluate the learning results, we created 10 sequences for the recognition phase and $100 (= 10 \times 10)$ sequences for the navigation phase.

Network Settings

The lower, higher, and goal RNNs are GRU with 128 units. The goal encoder consists of GRU with 128 hidden units and a fully connected layer for goal image encoder. The images of destinations are preprocessed by the goal image encoder before receiving by the GRU. The vision and motion encoders ENC^v and ENC^m is a fully-connected layer with 64 hidden units. The vision and motion predictors DEC^v and DEC^m consists of two fully-connected layers with 64 hidden units for both vision and motion, and output units with the same dimensionality as the vision and motion inputs, respectively. The activation function for hidden units in all the fully-connected layers was ELU, and that for the output units of vision and motion predictors were logistic-sigmoid and tanh non-linearlity, respectively. The vision error E^v was calculated as binary cross entropy and

Table 3.1: The error distances for destinations of each visiting order. The errors in learned conditions where the number of destination is not greater than 3 and unknown conditions where the number of destinations is more than 3 are shown.

Number of destinations	Visiting order				
	1st	2nd	3rd	4th	5th
1	1.61				
2	1.82	2.12			
3	1.43	2.72	1.86		
4	1.78	2.41	16.29	2.43	
5	1.69	2.46	10.24	17.87	2.61

motion error E^m was calculated as mean squared error. The L1-norm regularization was applied to the all RNNs with coefficient of 10^{-3}

Training Results

Our model is trained 200 times over the training sequences and the abilities of obtained model are evaluated in the following. Fig. 3.3 shows the trajectories of target and planned (or generated) motion sequences in the navigation phase for test sequences. Although there are some differences between target and planned trajectories, the planned trajectories pass close to the destinations for all destinations. Thus, it is considered that the our model acquires the abilities to control motion corresponding to given visual images of destinations.

For the quantitative evaluation of motion planning abilities of obtained model, how close the robot approaches to the given destinations is evaluated. For each destination, the distance from the destination point to the closest point in the trajectory of the planned motion sequences is calculated as errors. The errors are separately calculated to the destination orders. Although the number of the target destinations during training is not greater than 3, we can test how it behaves when the destinations more than 3 are given. Table 3.1 shows the calculated error distances from the destinations. When the number of destination is not greater than 3, the average error distances are less than the distances the robot can moves in two steps ($= 2\sqrt{2}$). This results shows that our model successfully obtains the motion planning abilities. The errors for the second destination becomes larger than the third destination. It seems strange considering the fact that the

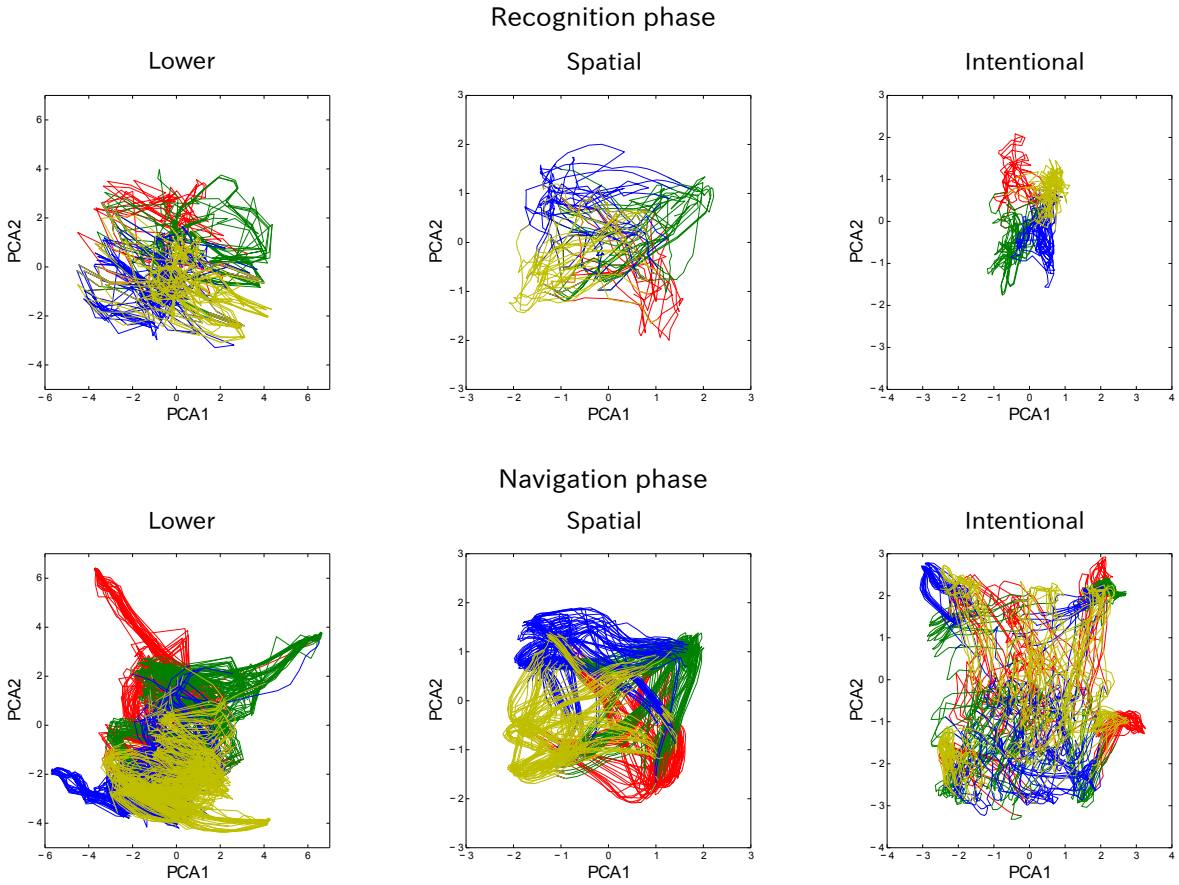

Figure 3.4: The internal states of the trained model in the recognition and navigation phase. The colors of the lines correspond to that of the nearest object from current position. The states are mapped onto the two dimensional space based on the result of PCA analysis.

second destination appears earlier. This is probably because visiting the last destination is easier than the others in a sense that the agent needs to visit the destinations while remembering the next destinations for the first and second, but it is not the case for the third. Moreover, because the error is accumulated during planning for the first destination, the error distance for the second destination is bigger than that for first destination. In the case that the number of destinations is four or five, the error distances for third and fourth is very large. This result shows that our model can only recognize the first, second and final destinations and cannot recognize more than 3 destinations.

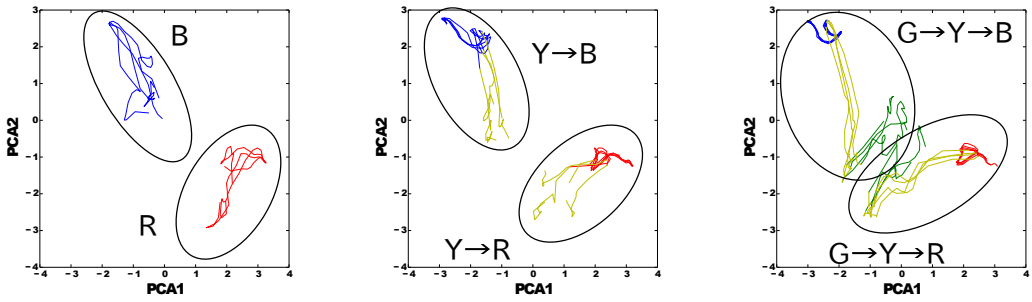

Figure 3.5: The internal states of the goal RNN. In this case, the colors of the lines correspond to that of the nearest object from current destination point. Two different trajectories of the states are compared in single plot for each condition where the number of destinations is from 1 to 3. It is clear that the trajectories are different even for the same destination except for terminal destination if the visiting order is different.

3.3.2 Internal Representation for Bottom-up Spatial Recognition and Top-down Navigation Control

In order to analyze how our model embeds the spatial recognition and navigational intention onto its own internal states, we visualize the internal states of three recurrent layers in our obtained model. For this analysis, we create additional sequences of motion and visual images obtained through trajectories that go through the landmarks as destinations in all possible orders (e.g. Blue, Red, Green, Yellow, Blue-Red, Blue-Green, Blue-Yellow,...). The number of destinations is not greater than 3. In the additional sequences, the navigation phase starts from selected four points near the four landmarks.

For the visualization, the number of dimensions for the internal states of the recurrent neural network is reduced into two dimensions using principal component analysis (PCA). Fig. 3.4 shows the visualized states for the lower, higher and goal RNNs in both recognition and navigation phases. The colors of the lines corresponds to the color of the nearest landmark from the current position of the robot at each step. The internal states of the higher RNN seem to self-organized in the way that the internal states forms clusters with the same arrangement of landmarks in the environment, and this arrangement is shared between recognition and navigation phases. Therefore, it is considered that the higher layer contains the locational recognition, which can be regarded as the cognitive map,

which self-organizes through the training where the vision and motion are given. On the other hand, in the lower and goal RNNs, the different colors overlap with each other over larger area than in the higher layer, and the forms of these states quite differ between recognition and navigation phase. In case of the lower layer, it is considered that the difference of the internal states is caused by the difference of agent's motion between two phases, which are determined by whether the agent moves toward the given destinations. In case of the goal RNN, as the states of the goal RNN are set by encoded intention, it is natural that the states differ between two phases. After all, the lower and goal RNNs do not recognize the spatial position of agent.

In order to analyze how the the goal RNN controls motion planning, we focus on the internal states of the goal RNN in the navigation phase. Fig. 3.5 shows the states during motion planning for two different terminal destinations with the same relay points for each condition where the number of destination is one, two and three. It shows that the trajectories of states are different between two different terminal destinations even if the relay points are the same. This is because the internal states need to remember the difference of the terminal destinations. It is also noted that there are slight variations in the trajectories of states for the same destinations and the variations maybe correspond to the starting point of the navigation phase. These results show that our model controls motion by gradually changing the internal states in order to achieve the recognized destinations.

3.3.3 Discussion

Our proposed model obtains spatial recognition, namely the cognitive map, and navigation control ability in higher-level layers from only vision and motion experiences. The formed internal states of the higher RNN are not different between recognition and navigation phases in contrast to the the goal RNN. This is because the obtained cognitive map is useful for both prediction and planning. Thus, it is considered that our model plans navigation behavior from given destinations by considering the change of the agent's position using the cognitive map.

The results of internal state analysis show that our model recognizes the destinations from visual images and controls navigational robot behavior by the goal RNN. However,

in the recognition phase, the goal RNN does not work and it is not biologically plausible, considering that animals always move autonomously even when no goal is given. This is because the intention is forcibly embedded into the goal RNN, and this mechanism should be improved.

3.4 Spatial Navigation Behavior based on Developed Spatial Representation

In this section, we simulated the robot navigation in an environment where the starting position was not fixed and the obstacles were placed randomly. In such navigation task, the robot should perform two different levels of navigational behaviors: obstacle avoidance and moving toward the given target. In this case, each RNN layer in the NHRNN has a different timescale for realizing the different levels of behaviors. Although the RNN layers have different timescales, no function is implemented in the layers and the behaviors should be obtained through learning of the task. The NHRNN was trained for controlling the robot's navigation behavior. It was seen that different levels of behaviors are self-organized in the hierarchical structure of RNNs. Especially; the spatial representation is self-organized in the RNN layers with a slow timescale. The model performed the navigation behavior using spatial representation. Details of the experiment are described in the following sections.

3.4.1 Navigation Task and Training

In this section, a mobile robot performs navigation behaviors in a simulated environment where obstacles are randomly placed and the trained path is not necessarily available. The navigation goal is indicated by the visual images as same as the previous section. The robot is required to obtain spatial representation for the navigation from subjective visuomotor experiences. The robot equips the NHRNN as a controller, and we investigate how the hierarchical structure controls goal-directed behaviors while avoiding obstacles.

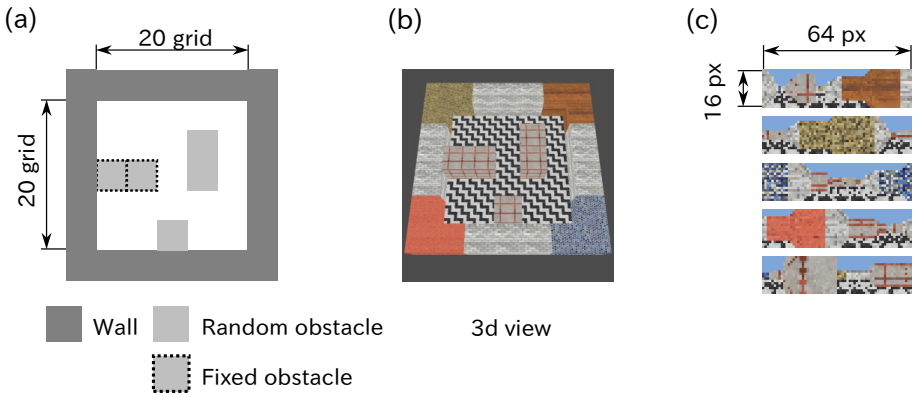

Figure 3.6: (a) Overview of the environment. (b) 3D overview of the environment. (c) Sample visual images captured by robot's camera.

Robot and Environment

The simulation environment is a square arena surrounded by textured walls. In this environment, position is represented by grid coordinates. The size of the arena is 20 × 20 grid cells. The walls have characteristic textures at their corners, which can work as landmarks. The floor is also textured. In addition to the walls, there are five objects in the arena that act as obstacles. These obstacles have the same texture but different from the wall one. Two of them are fixed obstacles whose positions do not change, whereas others are random obstacles whose positions change randomly during each trial of the navigation. The simulated environment is organized as shown in Fig. 3.6. The fixed obstacles are placed between the upper-left and lower-left corners, and the straight paths between them are unavailable.

The mobile robot is modeled as an agent that can move around the environment. The agent, who is equipped with an omnidirectional camera, can obtain an omnidirectional view while moving around the arena. The agent can move to eight adjacent cells in one time step. The omnidirectional camera is used as a capturing system that consists of four cameras covering the entire view around the agent. Each camera faces a different direction (north, south, east, or west). The size of the captured visual image for each camera is 16 × 16, and each pixel of the image has three channels (RGB). Four captured images are used as a single image by combining them horizontally, and the size of the visual image input is 64×16 with the three colored channels. While capturing the images,

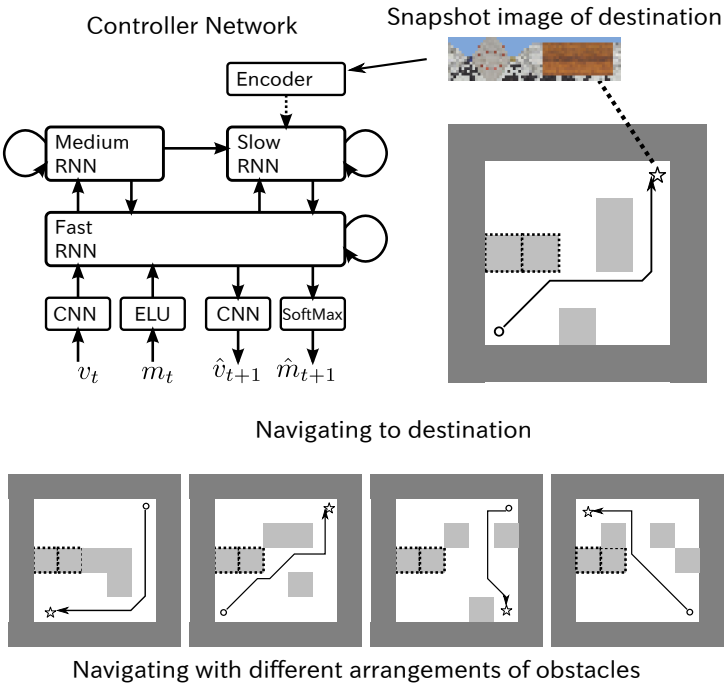

Figure 3.7: Navigation task. Top: Controller network receives snapshot images and navigates the robot in the environment using predictive motion output. Bottom: In the navigation task, the placement of obstacles, starting position, and destination are changed during each trial of navigation.

the positions of the cameras are perturbed by Gaussian noise with a mean of zero and standard deviation of 0.1, where the size of single grid cell is defined as 1.

Navigation Task

The navigation task for the agent is to reach the destinations indicated by the visual images captured at the destination points same as the previous section. However, in this experiments, the positions of three obstacles are changed at the beginning of every navigation trial as described above, and the agent has to avoid the obstacles while reaching the destinations. Therefore, the agent has to consider how to reach the destinations and what action should be taken to avoid getting stuck in the obstacles. An overview of the navigation task is presented in Fig. 3.7.

The NHRNN was trained as a controller in the navigation task. The internal states of all RNNs were initialized by zero values at the beginning of the recognition phase.

Training Data

The training data for learning was constructed in the similar way to previous. In this case, the length of the recognition phase was 100 time steps. The number of destinations was fixed to one and the length of navigation phase was not predefined; the navigation phase last until the agent achieved the destinations during this data collection. The agent's motion was controlled to trace the path found by the A* search algorithm. Three hundred sequences were prepared for the recognition phase. Ten different navigation sequences were created from the same starting position after a single sequence of the recognition phase; this indicated that there were 3,000 (300×10) sequences for the navigation sequences.

Network Settings

The lower, higher, and goal RNNs are CTRNNs with 128, 64 and 32 neurons, respectively. The time constant of CTRNNs for the lower, higher, and goal RNNs was 2, 10, and 20, respectively. In this section, with respect to the time constant of CTRNNs, we call the lower, higher, and goal RNNs fast, medium, and slow RNNs, respectively. The goal encoder consists consists of the visual encoder which is the same as the visual encoder for input visual images and a fully-connected layer with 32 hidden units; the goal visual image was firstly transformed by the visual encoder, and then the transformed vector was encoded into the initial states of the goal RNN by the fully-connected layer.

The motion encoder ENC^m is a fully-connected layer with 64 hidden units and the motion predictor DEC^m consists of two fully-connected layers with 64 hidden units and eight output units for motion, respectively. For the activation function of the output units in DEC^m, softmax function was used for outputting probabilities of each discrete action. The structures of the visual encoder DEC^v and predictor DEC^v which consists of CNNs are shown in Tab. 3.2. The vision error E^v was calculated as mean squared error and motion error E^m was calculated as cross entropy. To prevent our model from overfitting to the training sequences, an L2-norm of the model's parameters was added to minimize the objectives with coefficients of 10^{-3}.

Our model was trained 10 times in all the training sequences, and the abilities of the obtained model were evaluated.

Table 3.2: Structure of ENC^v and DEC^v

ENC^v

layer	type	size	channel	kernel	stride	padding	activation
1	input ($\boldsymbol{v}_t$)	64 × 16	3	-	-	-	-
2	conv.	32 × 9	8	4 × 2	2 × 2	1 × 1	ELU
3	conv.	16 × 5	16	4 × 2	2 × 2	1 × 1	ELU
4	conv.	8 × 3	32	4 × 2	2 × 2	1 × 1	ELU
5	fully connected	1 × 1	64	-	-	-	ELU

DEC^v

layer	type	size	channel	kernel	stride	padding	activation
1	input ($\boldsymbol{h}_t^{lower}$)	1 × 1	128	-	-	-	-
2	fully connected	1 × 1	64	-	-	-	ELU
3	fully connected	8 × 3	32	-	-	-	ELU
4	transposed conv.	16 × 6	16	4 × 2	2 × 2	1 × 0	ELU
5	transposed conv.	32 × 10	8	4 × 2	2 × 2	1 × 1	ELU
6	transposed conv.	64 × 16	3	4 × 2	2 × 2	1 × 2	sigmoid

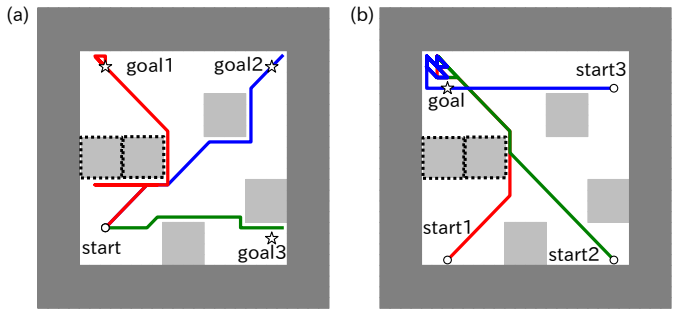

Figure 3.8: (a) Examples of navigation behaviors from one starting position to different destinations performed by trained model. (b) Examples of navigation behaviors from different starting positions to a single destination performed by trained model.

Training Results

To evaluate the navigation ability, the outputs of the trained model were used as the actual motion generation to move the agent. Figure 3.8 (a) shows sample trajectories of the agent's position during navigation. The agent navigated three different destinations from the same starting position by recognizing the image of the destinations. The agent successfully reached the destinations while avoiding the obstacles. This means that our model can recognize the existence of obstacles and select motions appropriately to reach the given destinations. Figure 3.8 (b) shows other sample trajectories. It shows the case of different starting points and the same goal. Our model also successfully navigated the specific goal from different starting points.

To investigate how the destination was represented in the internal states of our model, we collected the internal states while the agent navigated toward different destinations and analyzed them. All obstacles except the two fixed ones were removed. First, we divided the grid arena into 5×5 cells to classify the internal states corresponding to the destinations. The area with an obstacle cannot be a destinations; therefore, the total number of categorized destinations totaled to 23. In this experiment, the agent started the navigation task from five different initial positions (four corners and center of the arena). The agent is required to navigate to the destinations designated by the snapshot images from the initial positions. Totally, we obtained $115(23 \times 5)$ navigation behaviors and internal state changes for analysis. The internal states were visualized in two-dimensional (2D) space using principal component analysis (PCA). The first and second components were used for visualization. Figure 3.9 shows the visualized internal states of the slow RNN colored by the categorized destinations. The states were numbered with respect to the position in the arena (corresponding positions are shown in the right figure). The internal states of the fast and medium RNN did not seem to be organized by destination position (not shown). In contrast to our previous results [NIY17], the current spatial position of the agent in this experiment was also not self-organized in the medium RNN. This is because the obstacles were placed randomly and it was difficult to recognize the agent's current position. The initial states of the slow RNN seemed to align corresponding to the topological layout of positions in the arena. Moreover, the internal states kept a distance from those for other destinations in the PCA space throughout the navigation. This supports the explanation that our model realizes the navigation by remembering the positions of destinations rather than the sequence of actions.

3.4.2 Shortcut Behavior

Next, we investigated how the agent behaves when a shortcut path appears by removing the fixed obstacles which always exist during training. Our model is trained to navigate straight to the destination if there are not obstacles on the way because of the A* algorithm. If our model well generalizes these experiences during training and recognizes the space of the arena as spatial representation (not remembering the action sequences to the destination), the agents would be able to follow the newly appearing shortcut path.

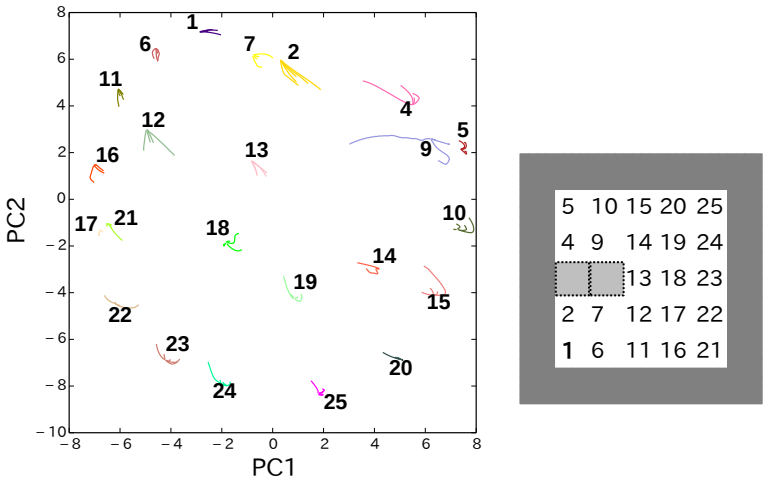

Figure 3.9: Visualization of PCA results with respect to the position of destination. Numbers shown together with the internal states of the slow RNN indicate position of destinations. Numbers in the left figure correspond to the numbers shown in the map on the right side.

In this experiment, the two fixed obstacles were removed; this means that the shortcut path between the upper-left and lower-left corners became available. All obstacles other than the two fixed obstacles were not placed. The starting and destination points were set to the upper-left and lower-left corners, respectively. Figure 3.10 shows the trajectories of the agent when the obstacles were removed. For comparison, the trajectories in the case when the obstacles were present are also shown. If the obstacles were present, agent reached the destination by taking a detour. On the contrary, in the case when the obstacles were removed, the agent passed through the area where the fixed obstacles were placed.

To quantitatively evaluate the shortcut behavior, we performed the obstacle-removing experiments between the upper-left and lower-left corners 100 times for each direction and calculated the reaching rate and path length. We set the starting and destination positions from the area of size 4×4 at the upper-left and lower-left corners. We assumed that the agent reaches the destination when it enters the area including the destination location. Path length was calculated as accumulated Euclidean distance on the way to the destination area after the agent starts navigation. The reaching rate and path length were calculated for each path from the upper-left to lower-left and vice-versa. For

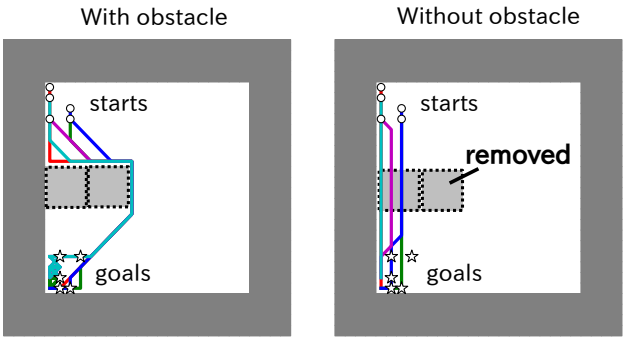

Figure 3.10: Navigation behaviors in the cases when fixed obstacles are present and removed. Five different behaviors are illustrated.

Table 3.3: Reaching rate and path length

		Path Length		Reaching
		avg.	s.d.	Rate
lower-left->upper-left	with obstacles	28.6	11.4	94%
	without obstacles	18.2	2.7	100%
upper-left->lower-left	with obstacles	22.3	2.6	100%
	without obstacles	15.6	1.3	100%

comparison, these values were also calculated in the case when the obstacles were present. Table 3.3 shows the calculated results. The reaching rate was 100 %, which shows that the agent can successfully reach the destination even in an unknown situation where the fixed obstacles are removed. The path length when the obstacles were removed was much shorter than when they were present. This shortcut behavior indicates that our model realizes the obstacle avoidance and goal-directed behavior by not remembering the sequence of actions but by always selecting proper actions considering the environmental state and position of the desired destination.

We conducted the internal states analysis again for investigating how our model recognizes the environmental states and selects actions for reaching the destination. We collected the internal states in the abovementioned experiment. Here, the agent navigated from each of the 4 $\times$ 4 cells at the lower-left corner to a destination point at the upper-left corner. As a result, we obtained 16 trajectories of internal states of each layer of the RNNs. The collected internal states were mapped to a PCA space by a mapping function constructed for the abovementioned previous PCA. We focused on the internal

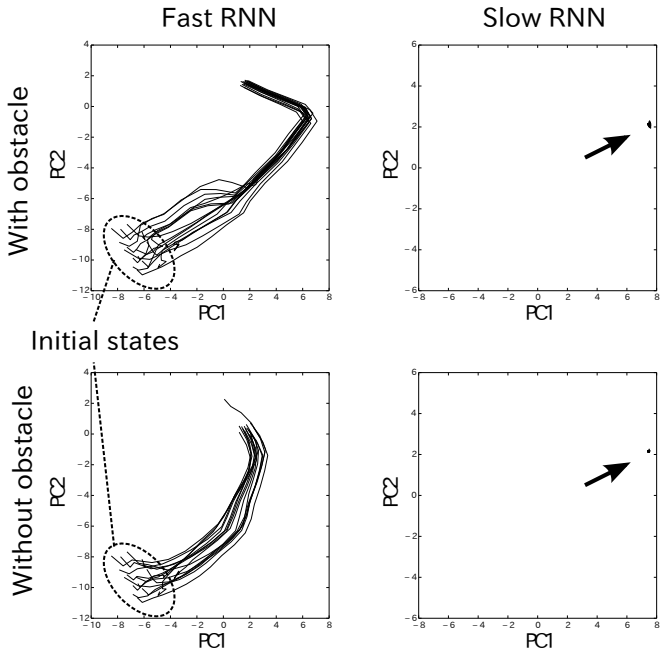

Figure 3.11: Visualization of PCA results. Internal states of the fast and slow RNNs are illustrated for cases when the obstacles are present and removed.

states of the fast and slow RNNs. Figure 3.11 shows the results of the internal states analysis. Comparison of the internal states of the fast RNN revealed that they changed differently in the second half of its trajectory. Especially, there was a clear difference in the first component of the fast RNN. The values of the first component changed more in the case with the fixed obstacles than in the case without them. It can be said that our model recognizes whether the obstacles are present, and the fast RNN changes in response to the wall avoidance behaviors. If the obstacles are not present, the fast RNN does not need to change the internal states and can maintain them while reaching the destination. On the contrary, there was no difference in the internal states of the slow RNN for different conditions. The internal states were maintained as is same regardless of the existence of the walls. This means that the slow RNN encodes the intentions to the destinations as top-down regulation to the behaviors.

Closed-loop Mental Simulation

By feeding back the prediction as intentions for the top-down regulations, our model can autonomously generate visuomotor sequences like mental simulation. In contrast to the

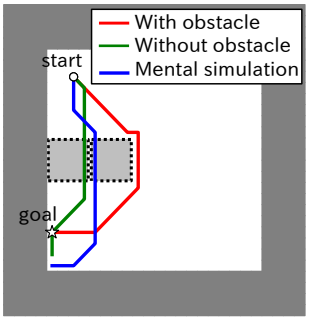

Figure 3.12: Example of mentally simulated navigation behaviors.

above result of interactive motion generation, the agent behaves in the internal mental world rather than the actual external environment. This experiment can clarify how the internal world is constructed inside the agent. After the snapshot image was presented, external visual inputs were shut, and the agent generated the visuomotor sequences in closed-loop manner. The starting point of the navigation was the upper-left corner and its destination was at the lower-left. Figure 3.12 shows an example of the simulated behaviors. For comparison, the generated behaviors in interaction with external environment are shown for both cases: with and without obstacles. The mentally simulated trajectory passed over the obstacles and is close to the trajectories in the case without obstacles. This result shows that our model assumes that there is no obstacle in the internal world. In other words, our model knows that, if the obstacles are removed, the agent can pass through the area where the obstacles were previously placed, even though our model has never entered that area during the training. This result is consistent with the results of the internal state analysis that the slow RNN controls the behaviors by position-based representation. Moreover, it is considered that our model obtains the spatial structure of the external world such as the cognitive map by generalizing the experiences.

3.4.3 Discussion

The HRNN developed different levels of navigation functions: obstacle avoidance by fast RNN and controlling goal-directed behavior, in hierarchical structure by only learning the navigation task. Especially, the slow dynamics of neurons control the fast dynamics in a

top-down manner. Previously, Ito et al. studied a network with hierarchical structure that controls a humanoid robot's behaviors by neurons of slow dynamics called parametric bias [INHT06]. When the robot's behavior is guided by a human assistant, the network changes the internal states of slow dynamics and the robot starts performing the guided behavior. In contrast to our model, this is more like a bottom-up formation of the intentions. Our model tries to find a way to achieve its goal regardless of environmental situations (i.e. disappearing obstacles). This could be a top-down regulation of behaviors from the slow dynamics as intentions. Our experimental results showed that our model successfully changes its behaviors in response to the different initial positions and placements of the obstacles. This means that our model successfully recognizes the visual sensory inputs by the CNN and uses the recognition for selecting actions in addition to the top-down controlling signal from the slow RNN.

The different functions were self-organized in the different timescales of the RNNs. Previously, Paine and Tani simulated the self-organization of the hierarchy of timescales in RNN [PT05]. The RNN was used as a robot controller for solving a navigation task in a simple maze. The weights of the RNN were optimized using genetic algorithm, and the wall avoidance and top-down control of goal-directed motion sequences were self-organized. Their experiment showed that hierarchical structure with multiple timescales is useful to realize the integration of low and high-level abilities to complete the task. However, the navigation task started from a fixed home position and the structure of the maze was fixed throughout the experiment; therefore, the task was completed only by remembering the sequences of actions, and cognitive map-based navigation behavior was not developed. Hwang et al. also used the model with multiple timescales for realizing goal-oriented behaviors (grasping a specific object) from goal-oriented states of slow RNN [HJKT16]. In their experiment, the network had to generate different behaviors to complete a task, which was shown as a video of human gesture, with same internal states of slow dynamics in response to various situations. This task is similar to our simulation where the positions of obstacles and goal change. The trained model in Hwang's study showed robustness against the gestures performed by non-trained human indicators. In contrast to Hwang's study, in which the proposed model was proved robust against relatively small perturbation, our model showed generalization ability against a

novel situation where fixed obstacles, which always exist during training, were removed. Such generalization ability against the novel situation is owing to the self-organized spatial representation. As indicated by the experimental results, our model developed spatial representation (Fig. 3.9) and performed the shortcut behavior by using that representation (Fig. 3.10). In the navigation task, there are numerous possible behaviors that our model should generate because our model has to navigate to any destinations from any starting points. Therefore, it is almost impossible to remember the action sequences for reaching the destinations. Instead, by developing the spatial representation of the destination, our model could reach the destination by only considering which direction the destination is, rather than by remembering the action sequences. Additionally, although it is possible to change the internal states of the slow RNN, the model spontaneously learned to keep the initial states encoded from the snapshot image. Such static representation in a slow RNN corresponds to the fact that the position of the destination does not change during navigation. We considered that the acquired spatial recognition has similar concepts to the cognitive map of rats in Tolman's experiment.

3.5 Conclusion

In this chapter, how the spatial navigation using the self-organized spatial representation can be developed through only visuomotor experiences was investigated.

In section 3.2, the NHRNN model for performing the spatial navigation based on the HRNN. The NHRNN model had three RNN modules: the lower and higher RNN as same as the HRNN and additional goal RNN for controlling the spatial navigation. The NHRNN could generate navigation behavior by setting the initial internal states of the goal RNN based on the visual image of the navigation goal. The navigation was conducted by generating visuomotor sequences along with navigational behaviors. The visual image of the goal is subjective vision of the navigation agent and the inputs that contained explicit information of the spatial position were not given. In later sections, the NHRNN was trained to perform spatial navigation and the development of the spatial navigation through visuomotor experiences was investigated.

In section 3.3, the NHRNN was trained on a simple environment that is an open space

where there was not obstacles. After the training of navigation in the environment, the representation of the agent's spatial position was developed in the higher RNN as same as in the previous chapter and the representation of the navigation goal was developed in the goal RNN. In this experiment, the goal representation in the goal RNN did not show the spatial position of the navigation goals and the spatial navigation considering the spatial position of the navigation goal was not performed.

In section 3.4, the NHRNN was trained on a maze like environment where the structure of the maze could change by replacing the obstacles. As a result of the training, the NHRNN developed the spatial navigation ability using the spatial representation of the goal, which was developed in the goal (slow) RNN. Further, the NHRNN performed the shortcut behavior like rats in the Tolman's experiment performed.

As described above, the NHRNN developed the spatial representation through visuomotor experiences during the training of spatial navigation. Especially, the spatial representation developed in the NHRNN in the maze like environment is considered to be different from the spatial representation developed in the HRNN in the previous chapter; it was the representation of the spatial position of the navigation goal. Such spatial representation of the goal can be developed only in the case of the spatial navigation; wowever, we consider that, in this maze environment, spatial navigation learning is necessary to develop the spatial representation, even for the agent's spatial position not for the goal's spatial position. As discussed in the previous section, long-term prediction learning is necessary for developing the spatial representation. However, it is difficult to completely predict vision in such various maze environments as the whole structure of the maze is unknown to the NHRNN, and visual prediction learning could not work and not encourage the development of the recognition of the spatial structure, in this case. Instead, in this maze case, the navigation learning encouraged the development of the spatial representation. That is because the spatial navigation also requires long-term prediction ability not in term of visual prediction but in term of change of the spatial position. The spatial position was not affected change of the maze structure and the prediction by the spatial navigation worked. In addition, as discussed in section 3.4, the change of the maze structure encourage the development of the recognition of the spatial structure instead of memorization of the action sequences in the spatial navigation. The result show that not

only bottom-up visuomotor inputs but also top-down intention to effectively navigate to goals have affect the development of the spatial representation.

Chapter 4

Conclusion

4.1 Summary

In this thesis, to investigate how the spatial cognition can be developed through experiences, the development of the spatial cognition was simulated using artificial neural network models. Especially, the development of the spatial cognition in similar condition to rats was considered; only visuomotor experiences which contains no explicit information of the spatial position or direction and spatial relationships between places were given to the models, and teaching of these spatial knowledge was never provided. We summarized the thesis below.

In chapter 1, we described the spatial cognition in animals and studies on the spatial cognition. The simulation studies on the development of the spatial cognition and difference between the previous and our approaches were described. Especially, we described the necessity of the simulation through only experiences and the importance of the visuomotor integration for studying the development of the spatial cognition. Then, the objective of our study was stated.

In chapter 2, the HRNN model was introduced and the development of the representation of the spatial structure was simulated through visuomotor prediction learning using the HRNN. The HRNN had hierarchical structure of RNNs without any predefined functions and was trained to just predict visuomotor sequences. In section 2.3, we firstly trained the HRNN model on visuomotor experiences of the mobile agent in a simple simulated environment and showed that the recognition of the spatial structure was developed

through the visuomotor prediction learning. Especially, the visuomotor integration learning for predicting visuomotor sequences from only motion sequences was required for the development of the recognition of the spatial structure. In section 2.4, we also simulated the development of the recognition of the spatial structure on visuomotor experiences in real environment. In section 2.5, it was shown that adequate randomness of spatial movement was necessary for developing the cognitive map like spatial representation by comparing the developed internal representations on different movement randomness. In section 2.6, the development of the spatial representation shared between different environment with different visual characteristics was simulated using a different model. Then, these results were summarized in section 2.7.

In chapter 3, the NHRNN model was proposed by extending the HRNN and the development of the spatial navigation ability was simulated. The NHRNN was trained to navigate toward navigational goals indicated by visual images in a form of generating visuomotor sequences as same as the HRNN. In section 3.3, the NHRNN was trained to perform navigation of the simulated agent in an open space environment without obstacles, and as a result, the agent's spatial position and the navigation goals were represented in the higher-level RNNs separately. In this case, the representation of the navigational goals had no spatial structure. In section 3.4, the NHRNN was trained on the maze like environment where the obstacles existed and whose structure was changed by replacing the obstacles. In this case, the representation of spatial position of the navigational goals was developed. This results indicate that the recognition of the spatial structure is also developed for effectively performing spatial navigation.

As summarized above, the development of the spatial cognition through only visuomotor integrated experiences was simulated. Especially, the recognition of the spatial structure that can enable the model to recognize spatial position even in novel place by considering the spatial relationship and the spatial navigation ability by considering the spatial position of the navigational goal were developed. These spatial cognitive abilities are similar to that performed by rats in Tolman's experiment [Tol48]. Our results were different from previous studies for simulating the development of the spatial cognition in the sense that, in our simulation, no explicit information about the space such as spatial position and spatial relationships between places was given to the model and the model

just learned visuomotor sequences as same as real rats. In the following, we further discuss about the differences of our simulation from previous studies, the contribution of our results, and future direction of the simulation approach for studying the development of the spatial cognition.

4.2 Discussion

In this section, we discuss about contributions of our simulation results to study on the development of the spatial cognition.

4.2.1 Hierarchical Structure and Visuomotor Integration

In this study, the development of the spatial cognition, namely, the recognition of spatial structure and the spatial navigation ability, was simulated through visuomotor prediction learning on the proposed hierarchical recurrent neural networks. The hierarchical structure of RNN enabled the model to contain representations with multiple time scales and the spatial representation of position was self-organized in the higher level RNN. The spatial representation in the proposed models was developed not just by mapping visuomotor sensory inputs, rather, it was generated by the models internally. This internally generated spatial representation could generate prediction about visuomotor inputs in a top-down manner; and, the top-down generation process realized the spatial recognition in an unknown area and the shortcut behavior. In addition to hierarchical structure, visuomotor integration learning contributed the development of the top-down structure. The visual inputs were different depending on the spatial position, on the other hand, the self-motion did not depend on the spatial position; and the models could acquire the recognition such that the spatial position always changes corresponding to self-motion everywhere in the environment. The recognition of the relationship between spatial position and self-motion was consistent even in the unknown area and the HRNN could generated correct visual images after passing the unknown area by top-down spatial recognition of the higher RNN as shown in section 2.3. In section 3.4, the NHRNN developed the spatial navigation ability with short-cut behavior by considering the spatial position of the navigation. The developed spatial navigation behavior is also top-down generation pro-

cess in the sense that the NHRNN generate motion so that the agent go straight toward navigational goal unless it faces obstacles. The development of this top-down navigation behavior is also because of consistency of self-motion. Because the self-motion consistently change the agent's spatial position, it is more effective to select motion by considering the spatial position using internal spatial representation than by considering bottom-up information about environment, i.e., vision. As described above, the hierarchical structure and visuomotor integration could develop top-down generation of visual prediction or navigational behavior by using the spatial representation that consistently works even in a novel spatial situation.

4.2.2 Not Only One Spatial Coding

Our simulation indicates that the spatial representation that realize spatial understanding was similar to real animals. Actually, although the developed spatial representation in our model was not same as the spatial representation in real animals, namely, place and grid cells, the model showed spatial recognition abilities like understanding spatial position in novel place and short-cut behavior. The developed representation of the spatial structure in our simulations were place cell-like representation in the sense that the internal states had one-to-one correspondence with the places. However, a single neuron did not correspond a specific place and the spatial position were coded by overall hidden neurons of the RNN. Additionally, the representation had grid cell-like characteristic in the sense that the activation of the neurons changes according to input motion; the developed spatial representation could perform path-integration like ability. In the developed representation, the spatial position was changed linearly corresponding to value of each hidden neuron in the RNN as we could observe the two-dimensional structure of the internal representation by PCA. Such representation could directly represent the spatial structure of Euclidean space. The results of our simulation show that well studied place and gird cells are only instances of possible spatial representation.

4.2.3 Spatial Representation Developed for Spatial Navigation

In our simulations, the recognition of spatial structure was developed through visuomotor prediction learning and spatial navigation task, and there were two form of the spatial position represented in the RNN's internal states: the spatial position of the agent and the navigational goal. The representation of the agent's position was developed for predicting future observation that changed according to the agent's movement. As the agent's movement contained randomness, to predict observation correctly, the spatial representation of the agent's spatial position was required. On the other hand, the representation of the navigational goal was developed for performing navigation in the maze like environment. In the maze that changed its structure, it is required to navigate by considering the spatial structure and the spatial position of the navigational goal. These simulation results showed that the representation of spatial positions of the agent and the navigation goal was developed for different objectives even though both of them were the representation of spatial position. The model by Banino et al. [BBU+18] used the grid cell representation that developed as the representation of the agent spatial position as the navigational goal's representation. Especially, the model was designed by experimenter to use the same spatial representation between the agent and the navigational goal. By modeling such reusing of the agent's spatial representation for spatial navigation, how the spatial navigation ability was developed cannot be fully understood. Even if the grid cell representation is useful for spatial navigation, it is not obvious that the grid cell which is developed for representing the agent's spatial position is also used for representing the navigational goal's spatial position. In our simulations, the representation of the spatial position was not explicitly given to the model and it was developed through the learning of spatial navigation independently of the agent's spatial position. It indicates that real animals also develop their internal spatial representation of the navigational goals through their experiences for effectively navigating in complex natural environment apart from their own spatial position.

4.2.4 Future perspectives

Although we simulated the development of the spatial cognition and showed some useful insights about the spatial cognition as described above, there are problems current simulations and remained questions. We describe about our consideration of these problems and questions for providing future perspectives.

Development of the grid cells

The developed spatial representation is considered to be enough to say it realizes the spatial cognition as described above discussion, however, it is considered that developed spatial code had problems. One problem is that the space is coded by the activation value of the neurons and its magnitude. The activation function used in hidden nodes of the RNNs in our simulations has lower and upper bounds. Because of this limitation of range of activation, the values of the hidden neurons could not become larger from a certain distance in a travelling of a space. It means that the internal representation of the spatial structure had limited size. However, the real space had infinite size conceptually. Thus, the developed spatial recognition in the HRNN could not be applicable to space that is larger than the represented space in the internal model of the HRNN. Although it is considered that the spatial code of real animals as grid cells also has limitation in recognizable distance, it is more flexible or applicable to larger space. The spatial code by the grid cells is generated by combining multiple grid cells with multiple grid scales. In the spatial code by the grid cells, a major factor is not the magnitude of the neural activation but the length of the spatial period. Thus, the limitation of the neural activation's magnitude does not regulate the capacity of the recognition of spatial distance. In fact, Banino et al. showed that the spatial position can be recognizable even in larger environment than ever-experienced environment by the spatial code using multiple grid scales [BBU$^+$18]. Another problem is the robustness against noise. In the case of the spatial code by grid cells, it is considered that small errors by noise can be corrected as the grid cells are forming attractor network [MBJ$^+$06]. On the other hand, the spatial code by magnitude of neurons as developed in our simulation could easily accumulate errors and the internal recognition of the spatial position is easily drifted by noise. Therefore, as described above, the developed spatial representation was different

from real animals and inefficient for accurately recognizing spatial position or distance. That maybe because the model was trained in not large and noiseless environment. How the grid cells can be developed through only visuomotor experiences should be considered.

Remapping of place cells

Because the place cells, the spatial representation in real animals, exist in hippocampus which deeply involves memory, the spatial cognition is considered to be deeply associated with memory mechanism. One example phenomena in spatial cognition that involves memory is the remapping of the place cells; the hippocampul place cells are re-assigned for places when rats entering into different environment from another environment. Specifically, patterns of place cell population assigned for environments are orthogonal representations each other [MRM15]. With huge number of combination of assignment patterns from population of hippocampul cells and their orthogonal representation, it is considered that rats effectively and robustly memorize a huge amount of environments. In our simulation in section 2.6, although the development of spatial recognition between different environment with different visual characteristic was considered, the development of such remapping phenomena was not simulated as only two environments were used and there are no need to efficiently memorize different environments. If the development of the remapping phenomena was simulated, we would get better understandings about the relationships between memory and spatial cognition.

Simulation including the behavioral development

In chapter 3, we simulated the development of the spatial representation through the learning of spatial navigation. However, the behaviors that the NHRNN learned were fixed during training although real animal's behaviors change throughout their development [WCBO10]. In section 2.4, we showed that the HRNN differently developed spatial representation depending on the agent's behaviors; however, the difference of behaviors was externally introduced by the experimenter and not generated by the HRNN itself. That is because the HRNN has no mechanism to generate behaviors intentionally and cannot change the agent's behavior voluntary. On the other hand, the NHRNN has mechanism to generate behaviors to perform spatial navigation based on given navigation

goals. In current study, as the NHRNN was trained to perform navigation in a imitation manner. However, for example, by training the NHRNN in a reinforcement learning, it is possible to change behaviors during training. The simulation where the agent change its behavior depending on the internal spatial representation is necessary for investigating how the developments of spatial representation and spatial navigation behaviors interact each other.

Hierarchical cognitive maps

We have cognitive maps with different scales. Some has small scale for a single room and some has large scale for a entire country. In large scale cognitive map, some parts that exist in real environments are omitted, e.g., small buildings are not described in a map of the entire country. It means that the large scale cognitive map is an abstract representation of the spatial environment. Then, how such large scale abstract cognitive is developed? We considered that the development of the abstract cognitive map is conceptually same as that of small detailed cognitive map that was simulated in our simulation; the abstract map can be developed for predicting future observation or performing spatial navigation. The reason for abstraction of the space might be the limitation of the size of the space that can be represented as the cognitive map and the inefficiency of creating the detailed large scale map. Then, the resolution of the map is decreased by omitting small and minor landmarks. For navigating in large scale environment like a city, using two scale of the cognitive map is an effective way. Firstly, relay points that should be visit to reach the final destination are planned by using the large scale abstract cognitive map. Then, by using the small scale cognitive maps, the navigation between the relay points are performed. By performing such hierarchical navigation, the detailed large scale cognitive map is not necessary. For simulating the development of hierarchical representation of the cognitive maps, it is considered that the simulation in larger and more complex environment than in our current simulations and the additional hierarchical structure for multiple cognitive maps with different scales are required.

Studying disability of spatial cognition

Current study focused on how the spatial cognition can be developed; however, how the disability on spatial cognition can be caused should be considered. Real animals sometimes failed to properly develop the spatial cognition. Held and Hein showed that kitten with lack of experiences of active movement on visuomotor experiences during developmental age cannot properly develop spatial cognition integrated with self-motion and the kitten was not able to walk around as well as a kitten had experiences with active movement [HH63]. In our experiments, it was observed that the HRNN failed to develop the spatial recognition when it did not learned the long-term visual prediction task (PoM task) in section 2.3 and when the agent behavior did not have adequate complexity in section 2.5. It means that the developmental failure on the spatial cognition due to lack of necessary learning or experiences also can be simulated as well as successful development. Such developmental model that can simulate both success and failure of development depending on how it learns is certainly necessary for investigating the developmental failure and we consider that our simulation demonstrated that our models can be used for investigating what cause disabilities on spatial cognition during development.

Extendability of model to general knowledge development

We finally reconsider the potential of our models beyond the developmental model of the spatial cognition. In this study, although we used only vision and motion for sensory inputs for the models, the models could applied to other modalities such as auditory or tactile sensors because the models had no assumption about the input modalities; how to recognize the sensory inputs was learned by the models themselves in an end-to-end manner. Thus, conceptually, the development of the spatial cognition from any kinds of sensorimotor experiences can be simulated in our proposed methods. Further, even abstract concepts like meaning of words also can be used as inputs to our models, and it is considered that our models also can be extendable as a model of the development of the internal representation of the general knowledge. It was hypothesized human can store knowledge about objects or concepts as if they placed on two-dimensional space in their brain similar to the physical space; such internal representation of knowledge is also called as the cognitive map and it is considered that knowledge become flexibly

manipulable by being placed on the cognitive map [EC14, COB16]. Such cognitive map about general knowledge is considered to be similar to the cognitive map about the spatial structure; it is speculated that the difference between them is only what is associated with the map. However, as general knowledge is an abstract concept that should be generated or extracted from sensory inputs, the problem of how the cognitive map of general knowledge can be developed is not exactly the same as that of the cognitive map of the spatial structure. Thus, it is considered that the current model cannot realized the development of the knowledge cognitive map; however, if it was realized, it could contribute to understand how human or animals' remarkable abilities of thought using abstract concept work in their brain.

Bibliography

[AMU99] Yoshito Aota, Yoshihiro Miyake, and Seiji Ukai. Neural network modeling of hippocampal place cells in rats. *The transactions of the Institute of Electronics, Information and Communication Engineers. D-II*, 82(12):2355–2366, dec 1999.

[BBMB15] Daniel Bush, Caswell Barry, Daniel Manson, and Neil Burgess. Using grid cells for navigation. *Neuron*, 87(3):507–520, 2015.

[BBO07] Neil Burgess, Caswell Barry, and John O'keefe. An oscillatory interference model of grid cell firing. *Hippocampus*, 17(9):801–812, 2007.

[BBU+18] Andrea Banino, Caswell Barry, Benigno Uria, Charles Blundell, Timothy Lillicrap, Piotr Mirowski, Alexander Pritzel, Martin J Chadwick, Thomas Degris, Joseph Modayil, et al. Vector-based navigation using grid-like representations in artificial agents. *Nature*, 557(7705):429, 2018.

[Bee95] Randall D Beer. On the dynamics of small continuous-time recurrent neural networks. *Adaptive Behavior*, 3(4):469–509, 1995.

[CGCB14] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning*, 2014.

[CKBO13] Guifen Chen, John A King, Neil Burgess, and John O'Keefe. How vision and movement combine in the hippocampal place code. *Proceedings of the National Academy of Sciences*, 110(1):378–383, 2013.

[COB16] Alexandra O Constantinescu, Jill X O'Reilly, and Timothy EJ Behrens. Organizing conceptual knowledge in humans with a gridlike code. *Science*, 352(6292):1464–1468, 2016.

[CSP+18] Devendra Singh Chaplot, Kanthashree Mysore Sathyendra, Rama Kumar Pasumarthi, Dheeraj Rajagopal, and Ruslan Salakhutdinov. Gated-attention architectures for task-oriented language grounding. In *AAAI Conference on Artificial Intelligence (AAAI-18)*, 2018.

[CUH16] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). In *International Conference on Learning Representations (ICLR 2016)*, 2016.

[CVMG+14] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, 2014.

[CW18] Christopher J. Cueva and Xue-Xin Wei. Emergence of grid-like representations by training recurrent neural networks to perform spatial localization. In *6th International Conference on Learning Representations (ICLR2018)*, 2018.

[DKW09] Thomas J Davidson, Fabian Kloosterman, and Matthew A Wilson. Hippocampal replay of extended experience. *Neuron*, 63(4):497–507, 2009.

[EAE+15] Susan L Epstein, Anoop Aroor, Matthew Evanusa, Elizabeth I Sklar, and Simon Parsons. Learning spatial models for navigation. In *International Workshop on Spatial Information Theory*, pages 403–425. Springer, 2015.

[EC14] Howard Eichenbaum and Neal J Cohen. Can we reconcile the declarative memory and spatial navigation views on hippocampal function? *Neuron*, 83(4):764–770, 2014.

[EPJS17] Russell A Epstein, Eva Zita Patai, Joshua B Julian, and Hugo J Spiers. The cognitive map in humans: spatial navigation and beyond. *Nature neuroscience*, 20(11):1504, 2017.

[FMW+04] Marianne Fyhn, Sturla Molden, Menno P Witter, Edvard I Moser, and May-Britt Moser. Spatial representation in the entorhinal cortex. *Science*, 305(5688):1258–1264, 2004.

[FSW07] Mathias Franzius, Henning Sprekeler, and Laurenz Wiskott. Slowness and sparseness lead to place, head-direction, and spatial-view cells. *PLoS computational biology*, 3(8):e166, 2007.

[HH63] Richard Held and Alan Hein. Movement-produced stimulation in the development of visually guided behavior. *Journal of comparative and physiological psychology*, 56(5):872, 1963.

[HJKT16] Jungsik Hwang, Minju Jung, Jinhyung Kim, and Jun Tani. A deep learning approach for seamless integration of cognitive skills for humanoid robots. In *Development and Learning and Epigenetic Robotics (ICDL-EpiRob), 2016 Joint IEEE International Conference on*, pages 59–65. IEEE, 2016.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[INHT06] Masato Ito, Kuniaki Noda, Yukiko Hoshino, and Jun Tani. Dynamic and interactive generation of object handling behaviors by a small humanoid robot using a dynamic neural network model. *Neural Networks*, 19(3):323–337, 2006.

[JCG15] Adrien Jauffret, Nicolas Cuperlier, and Philippe Gaussier. From grid cells and visual place cells to multimodal place cell: a new robotic architecture. *Frontiers in neurorobotics*, 9, 2015.

[KB15] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *The International Conference on Learning Representations (ICLR 2015)*, 2015.

[KFS87] Mitsuo Kawato, Kazunori Furukawa, and R Suzuki. A hierarchical neural-network model for control and learning of voluntary movement. *Biological cybernetics*, 57(3):169–185, 1987.

[LBBH98] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[LBH15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[MBJ+06] Bruce L McNaughton, Francesco P Battaglia, Ole Jensen, Edvard I Moser, and May-Britt Moser. Path integration and the neural basis of the'cognitive map'. *Nature Reviews Neuroscience*, 7(8):663, 2006.

[MBM+16] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*, pages 1928–1937, 2016.

[MGRO82] RGM Morris, Paul Garrud, JNP al Rawlins, and John O'Keefe. Place navigation impaired in rats with hippocampal lesions. *Nature*, 297(5868):681, 1982.

[MKM08] Edvard I Moser, Emilio Kropff, and May-Britt Moser. Place cells, grid cells, and the brain's spatial representation system. *Neuroscience*, 31(1):69, 2008.

[MRM15] May-Britt Moser, David C Rowland, and Edvard I Moser. Place cells, grid cells, and memory. *Cold Spring Harbor perspectives in biology*, 7(2):a021808, 2015.

[MW88] Martin Müller and Rüdiger Wehner. Path integration in desert ants, cataglyphis fortis. *Proceedings of the National Academy of Sciences*, 85(14):5287–5290, 1988.

[MWP04] Michael J Milford, Gordon F Wyeth, and David Prasser. Ratslam: a hippocampal model for simultaneous localization and mapping. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 1, pages 403–408. IEEE, 2004.

[NIY17] Wataru Noguchi, Hiroyuki Iizuka, and Masahito Yamamoto. Hierarchical recurrent neural network model for goal-directed motion planning using self-organized cognitive map. In *Proceedings of the Twenty-Second International Symposium on Artificial Life and Robotics 2017 (AROB 22nd 2017)*, pages 73–78, 2017.

[NNT08] Ryunosuke Nishimoto, Jun Namikawa, and Jun Tani. Learning multiple goal-directed actions through self-organization of a dynamic neural network model: A humanoid robot experiment. *Adaptive Behavior*, 16(2-3):166–181, 2008.

[OBS+15] H Freyja Olafsdottir, Caswell Barry, Aman B Saleem, Demis Hassabis, and Hugo J Spiers. Hippocampal place cells construct reward related sequences through unexplored space. *Elife*, 4:e06063, 2015.

[OD71] John O'Keefe and Jonathan Dostrovsky. The hippocampus as a spatial map. preliminary evidence from unit activity in the freely-moving rat. *Brain research*, 34(1):171–175, 1971.

[PML+] Deepak Pathak, Parsa Mahmoudieh, Guanghao Luo, Pulkit Agrawal, Dian Chen, Yide Shentu, Evan Shelhamer, Jitendra Malik, Alexei A Efros, and Trevor Darrell. Zero-shot visual imitation.

[PON03] David Philipona, J Kevin O'Regan, and J-P Nadal. Is there something out there? inferring space from sensorimotor dependencies. *Neural computation*, 15(9):2029–2049, 2003.

[PSN+17] Mark Pfeiffer, Michael Schaeuble, Juan Nieto, Roland Siegwart, and Cesar Cadena. From perception to decision: A data-driven approach to end-to-end

motion planning for autonomous ground robots. In *2017 ieee international conference on robotics and automation (icra)*, pages 1527–1533. IEEE, 2017.

[PT05] Rainer W Paine and Jun Tani. How hierarchical control self-organizes in artificial adaptive systems. *Adaptive Behavior*, 13(3):211–225, 2005.

[RHW86] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–, October 1986.

[SBG17] Kimberly L Stachenfeld, Matthew M Botvinick, and Samuel J Gershman. The hippocampus as a predictive map. *Nature neuroscience*, 20(11):1643, 2017.

[Sch85] Françoise Schenk. Development of place navigation in rats from weaning to puberty. *Behavioral and neural biology*, 43(1):69–85, 1985.

[SFLU17] Ayelet Sarel, Arseny Finkelstein, Liora Las, and Nachum Ulanovsky. Vectorial representation of spatial goals in the hippocampus of bats. *Science*, 355(6321):176–180, 2017.

[SNS18] Gabriel Sepulveda, Juan Carlos Niebles, and Alvaro Soto. A deep learning based behavioral approach to indoor autonomous navigation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[TB07] Evren C Tumer and Michael S Brainard. Performance variability enables adaptive plasticity of 'crystallized'adult birdsong. *Nature*, 450(7173):1240, 2007.

[THTI17] Akira Taniguchi, Yoshinobu Hagiwara, Tadahiro Taniguchi, and Tetsunari Inamura. Online spatial concept and lexical acquisition with simultaneous localization and mapping. In *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, pages 811–818. IEEE, 2017.

[TKL+05] Alejandro Terrazas, Michael Krause, Peter Lipa, Katalin M Gothard, Carol A Barnes, and Bruce L McNaughton. Self-motion and the hippocampal spatial metric. *Journal of Neuroscience*, 25(35):8085–8096, 2005.

[TMR90] Jeffrey S Taube, Robert U Muller, and James B Ranck. Head-direction cells recorded from the postsubiculum in freely moving rats. i. description and quantitative analysis. *The Journal of neuroscience*, 10(2):420–435, 1990.

[TN99] Jun Tani and Stefano Nolfi. Learning to perceive the world as articulated: an approach for hierarchical learning in sensory-motor systems. *Neural Networks*, 12(7):1131–1141, 1999.

[TO16] Alexander V Terekhov and J Kevin O'Regan. Space as an invention of active agents. *Frontiers in Robotics and AI*, 3:4, 2016.

[Tol48] Edward C Tolman. Cognitive maps in rats and men. *Psychological review*, 55(4):189, 1948.

[TTK99] Gentaro Taga, Rieko Takaya, and Yukuo Konishi. Analysis of general movements of infants towards understanding of developmental principle for motor control. In *Systems, Man, and Cybernetics, 1999. IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on*, volume 5, pages 678–683. IEEE, 1999.

[TYT17] Huajin Tang, Rui Yan, and Kay Chen Tan. Cognitive navigation by neuro-inspired localization, mapping and episodic memory. *IEEE Transactions on Cognitive and Developmental Systems*, 2017.

[WCBO10] Tom J Wills, Francesca Cacucci, Neil Burgess, and John O'keefe. Development of the hippocampal cognitive map in preweanling rats. *Science*, 328(5985):1573–1576, 2010.

[WKS+18] Eiji Watanabe, Akiyoshi Kitaoka, Kiwako Sakamoto, Masaki Yasugi, and Kenta Tanaka. Illusory motion reproduced by deep neural networks trained for prediction. *Frontiers in psychology*, 9:345, 2018.

[WKV06] Reto Wyss, Peter König, and Paul FM J Verschure. A model of the ventral visual system based on temporal stability and local memory. *PLoS Biol*, 4(5):e120, 2006.

[WM93] Matthew A Wilson and Bruce L McNaughton. Dynamics of the hippocampal ensemble code for space. *Science*, 261(5124):1055–1058, 1993.

[WZ95] Ronald J Williams and David Zipser. Gradient-based learning algorithms for recurrent networks and their computational complexity. *Back-propagation: Theory, architectures and applications*, pages 433–486, 1995.

[YT08] Yuichi Yamashita and Jun Tani. Emergence of functional hierarchy in a multiple timescale neural network model: a humanoid robot experiment. *PLoS Comput Biol*, 4(11):e1000220, 2008.

[ZKTF10] Matthew D Zeiler, Dilip Krishnan, Graham W Taylor, and Rob Fergus. Deconvolutional networks. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2528–2535. IEEE, 2010.

[ZS17] Taiping Zeng and Bailu Si. Cognitive mapping based on conjunctive representations of space and movement. *Frontiers in Neurorobotics*, 11:61, 2017.