Building a Cognitive Map Using an SOM^2

Kazuhiro Tokunaga, Tetsuo Furukawa

Abstract:

In this paper, we propose a new method for building an environmental map in a self-organizing manner using visual information from a mobile robot. This method is based on a Higher Rank of Self-Organizing Map (SOM^2), in which Kohonen's SOM is extended to create a map of data distributions (set of manifolds). It is expected that the "SOM" is capable of creating an environmental map in a self-organizing manner from visual information, since the set of visual information obtained from each position in the environment forms a manifold at every position. We also show the effectiveness of the proposed method.

Keywords: *self-Organizing map, map building, place cells, head direction cells, autonomous robot.*

1. Introduction

1.1. Aim of this study

The ability to build an environmental map based on sensor information is necessary for an autonomous robot to perform self-localization, identification of direction and self-navigation. In animals, a map building capability is very important to accomplish crucial behavior such as predation, nest homing, path planning, and so on. With regards to research on map building in animals, O'Keefe and Dostrovsky identified "place cells", which respond preferentially to specific spatial locations in the hippocampus of a rat [1]. The place cells encode the observed sensory information as the animal explores its environment. Moreover, O'Keefe and Nadel propounded the theory that animals build a "cognitive map" within the brain, based on research of the place cells [2]. It is thus evident that animals build a cognitive map, which plays a role in path planning using landmarks (i.e., particular information of local environments) coded by the place cells. Furthermore, Taube et al. identified "head direction cells", which respond preferentially according to the direction of the head [3]. The head direction cells are seen to be involved in the map building, since it is important to know one's own direction before moving to a destination. Therefore, a robot is expected to be able to perform navigation automatically using a cognitive map model that incorporates the mechanism of place cells and head direction cells in its implementation. Moreover, it may be possible to identify a mechanism from the model akin to the map building of animals.

With regards to a technical model for map building, we propose using a Higher Rank Self-Organizing Map (SOM^2) in this study. The SOM^2 proposed by Furukawa [4] is generally an extended model of Kohonen's SOM [5]. The SOM^2 has a SOM-type modular network structure but with nesting SOMs (Fig. 1(a)). It is the task of each SOM module in the SOM^2 to identify a manifold, which approximates a data vector set, thereby enabling the entire SOM^2 to find the formation of the fiber bundle in a self-organizing manner (Fig. 1b)). It has been suggested in [4] that by using this feature, the SOM^2 can, with unsupervised learning, build a map, which is able to estimate the

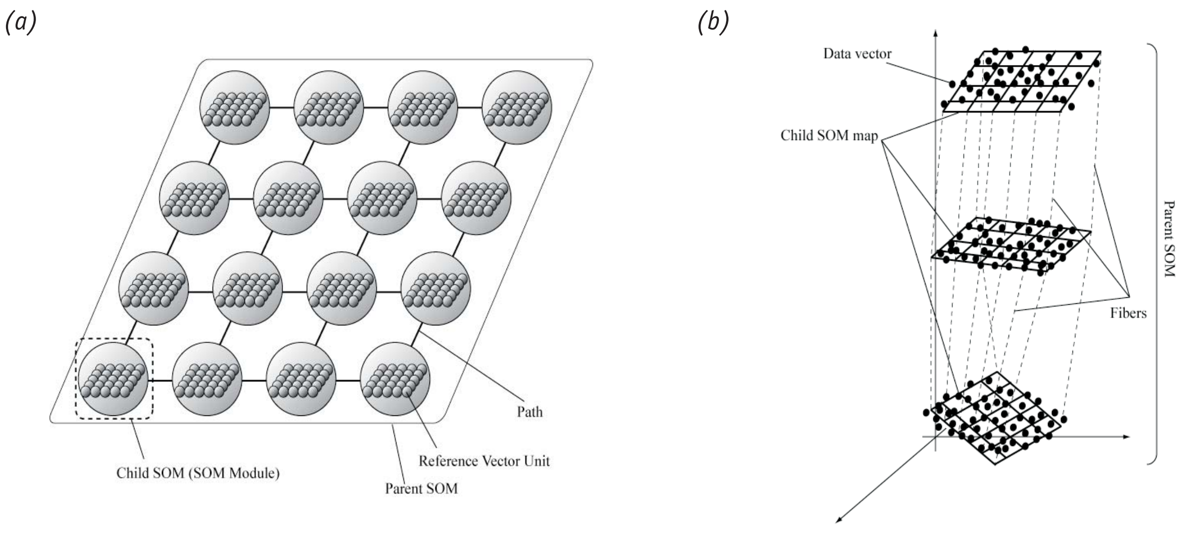


Fig. 1. Architecture of SOM^2. (a) the SOM^2 is a nest structure of SOMs. The position of each child SOM is kept with connecting between neighborhood child SOMs by path. (b) Each child SOM approximates each episode with a graph map (i.e., a manifold) through training of the episodes. The connecting the correspondence point of each map represents the fiber.

location and azimuth direction independently, based only on sets of the image data vectors observed from omni-directional vision sensors. It has also been suggested that the SOM^2 can, with unsupervised learning, build a cognitive map with the features given below, using only visual information; each module of the SOM^2 represents a place cell which codes the specific location, while each reference vector unit in the SOM module represents a head direction cell. However, a detailed verification has not yet been done. Hence, this study aims to confirm through various computer simulations, whether or not the position and the azimuth direction can be estimated from a map acquired by unsupervised learning of the SOM^2.

1.2. Related works

Map building is an important theme in studies involving autonomous robots. Consequently, in recent years, various methods for building an environmental map have been proposed. A classical map building method, dead reckoning, can estimate the location and pose (or direction) of a robot by calculating the displacement from an internal sensor, such as a rotary encoder, acceleration sensor, and so on. The proposal method builds the map from only the arrangement of the memory of sensor information, while the dead reckoning build the map by using odometry information. On the other hand, several methods for map building using external sensors such as a laser range finder, vision sensor, and so on, have also been proposed. The most popular method, known as SLAM [6], is often used in map building using external sensors [7], [8], [9]. SLAM can perform self-localization and estimate the structure of the environment around the robot simultaneously, making it a technologically excellent method for map building. Nevertheless, SLAM requires a highly accurate observation model and locomotion model, a priori, since it is necessary to understand the correct structure of the environment [10]. The observation and locomotion models provide the physical measurements of the environment and the physical location of the robot using external sensors, respectively. The observation and locomotion models provide the physical measurements of the environment and the physical location of the robot using external sensors, respectively. The SLAM builds the map based on measurements provided from these models. However, it is difficult to develop the models, which flexibly build the map, since the conditions of the environment, and the sensors are nonstationary. Besides, the correspondence between SLAM and map building in an animal's brain has not yet been identified. In contrast, a method has been proposed, called "topological map building", that builds the map abstracted by a graph [11]. Nodes and edges in the graph represent specific locations (areas) and pathways between areas, respectively. Typically, sensor information for landmarks is memorized to nodes, while pathways between landmarks are stored as edges. Then self-localization and path planning can be performed by matching the sensor information to the map. Since each node memorizes the information that represents a local area, it is not necessary to comprehend the correct structure of the environment. In addition, the method requires no physics models beforehand. Moreover, it is very interesting because this method resembles the cognitive map based on place cells in the hippocampus of an animal's brain. However, this method has two important issues: how are the allocation of nodes and the connection of paths decided in a self-organizing manner. As a solution to these issues, applying a self-organizing neural network (SONN), such as the Self-Organizing Map (SOM), Topology Representing Network (TRN), and so on, has been suggested [12]. Reference vectors of the SONN are the nodes that memorize the information of specific areas. Moreover, paths that connect reference vectors represent the pathway between nodes. The SONN can perform the allocation of nodes and the connection of paths in a self-organizing manner with unsupervised learning. Tanaka *et al.* proposed an implementation model incorporating the place cell in the hippocampus using a TRN [13]. This method does not, however, build a cognitive map in the same way as an animal, because GPS information is included as training input. In addition, K. Chokshi *et al.* proposed a method for self-localization using the categorization of vision information by an SOM [14]. Their method is also an implementation model of the place cell. Nevertheless, since these methods do not include the functionality of the head direction cell, a robot cannot identify its own direction.

In contrast, our proposal method builds the map that can independently estimate the position and direction from only the visual information with the unsupervised learning. Each module of SOM^2 is a node, which memorizes the visual information that represents the local environment in a self-organizing manner. In addition, the memorized visual information is ordered corresponding to direction with the unsupervised learning. The features are similar to ones of the place cells and head direction cells in the hippocampus. Thus, the method incorporates not only the functionality of the place cells, but also that of the head direction cells. Moreover, the topology of the map acquired with SOM^2 and the topology of the environment's geography are nearly equal. Therefore, the self-navigation of the robot can be very easily performed by using the map.

2. Map building using an SOM^2

In this study, we aim to show that the SOM^2 can create a self-organizing map in which the position and orientation of a robot can be estimated using only vision sensor information. First, we explain how to acquire the map using an SOM^2.

When a mobile robot equipped with a vision sensor gets a birds-eye view of the surrounding area at place A in the environment as shown in Fig. 2a), the episode of vision sensor information is distributed as a manifold in a multi-dimensional vector space (sensor space). If the mobile robot observes vision sensor information by rotating 360 degrees at place A, then the episode of vision sensor information is distributed as a one-dimensional toroidal manifold (Fig. 2b)). In addition, if the mobile robot moves from place A to place B, the episode of sensor information at place B forms a manifold near place A (Fig. 2b)). Thus, the episodes observed at consecutive places in the environment form continuous manifolds in sensor space (Fig. 2b)). Moreover, the correspondence point between the manifolds corresponds to the shooting angle of the vision sensor (that is, the azimuth direction of the environment). Therefore, it is expected that the position and azimuth direction can be estimated using a map created based on the distance and correspondence point between manifolds.

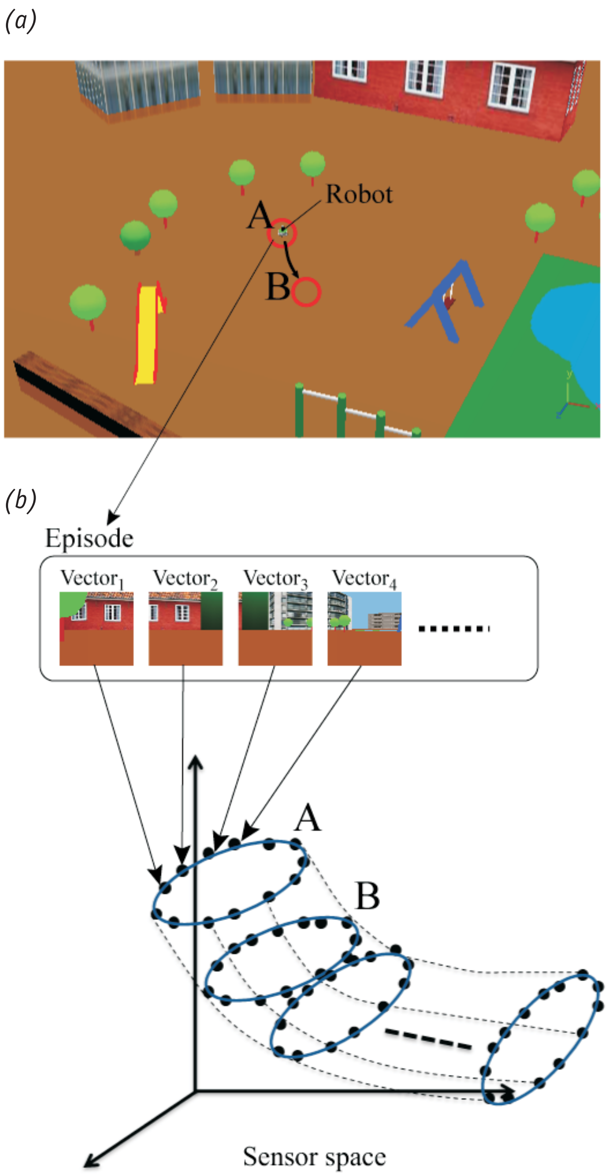


Fig. 2. The episode of vision sensor information is distributed as a manifold in a sensor space. If the mobile robot moves from place A to place B, the episodes of sensor information at place B from a manifold near place A. Moreover, the correspondence point between the manifolds corresponds to the shooting angle of the vision sensor.

For this method, we employ the SOM^2 proposed by Furukawa. The SOM^2 is an extension of the SOM in which each reference vector unit in the conventional SOM is replaced by an SOM module. In other words, the SOM^2 is a nest structure of SOMs (Fig. 1a)). In this paper, the SOM module (child level) is called the "child SOM", while the whole SOM (parent level) is called the "parent SOM". In the SOM^2, sets (episodes) of vector data are given to the SOM^2 as training data. The vector data for each episode are distributed on each of the subspaces in vector space (Fig. 1b)). Each child SOM approximates each episode with a graph map (i.e., a manifold) through training of the episodes. Here, training of each child SOM is performed in such a way that the correspondence points on the map in each child SOM are uniform. Thus, connecting the correspondence point of each map represents the fiber. Besides, the parent SOM orders the maps formed by child SOMs. Therefore, an SOM^2 can create a map of manifolds.

In building an environmental map using an SOM^2, a Ring SOM (RSOM), which approximates the distribution of data vectors by a one-dimensional toroidal manifold, is em-ployed for each child SOM (Fig. 3). Hereafter, this SOM^2 is referred to as the RSOM×SOM. The episode sets of vision sensor information observed at various places are given to the SOM^2 as training episodes. After training, each child SOM forms a manifold of each place's episodes. Here, the correspondence points in the map of each child SOM are constant, that is, the environmental azimuth direction is constant. Each reference vector unit of the RSOM represents a head direction cell. Moreover, the parent SOM creates a map with the topology (i.e., topology at the positions of the manifolds in sensor space) of the positions in the environment preserved. As a result, the map of the parent SOM itself represents a geometrical map of the environment. Each module of parent SOM represents the place cells. In addition, the azimuth directions of the environment in each RSOM are ordered in a self-organizing manner.

Restrictions on the method, however, include that the working environment is open without obstacles, in which robot cannot pass through and robot's view is interrupted, and that similar visual information does not exist in the environment. It is certainly possible to apply the method in a non-limited environment by enhancing the SOM^2. (This is addressed in subsection 5.1.) Nonetheless, the aim of this study is to verify that an SOM^2 can create a self-organizing map in which the position and orientation of a robot can be estimated using only vision sensor information. The robot's working environment for this study is set as follows.

(A) The working environment is open without obstacles. Moreover, the robot can see faraway buildings and mountains, etc. as shown in Figs. 4 and 5.

(B) The robot has an omni-directional camera as vision sensor.

(C) Only visual information is assumed to be observed by the robot sensors.

In (A), under normal circumstances, it is preferable that the robot can build the map while looking around with a single directional camera. Building the preferable map from partial information is difficult without enhancing the algorithm for the SOM^2. Thus, in this study, the episodes are acquired from an omni-directional camera.

3. Algorithm for the SOM^2 (RSOM×SOM)

In this section, the algorithm for the RSOM×SOM is explained. The RSOM×SOM is an SOM^2 in which each child SOM is replaced by a RSOM. The difference between an RSOM and SOM is the definition of the distance measure between reference vectors on the map, since the reference vector is allocated on a one-dimensional toroid in

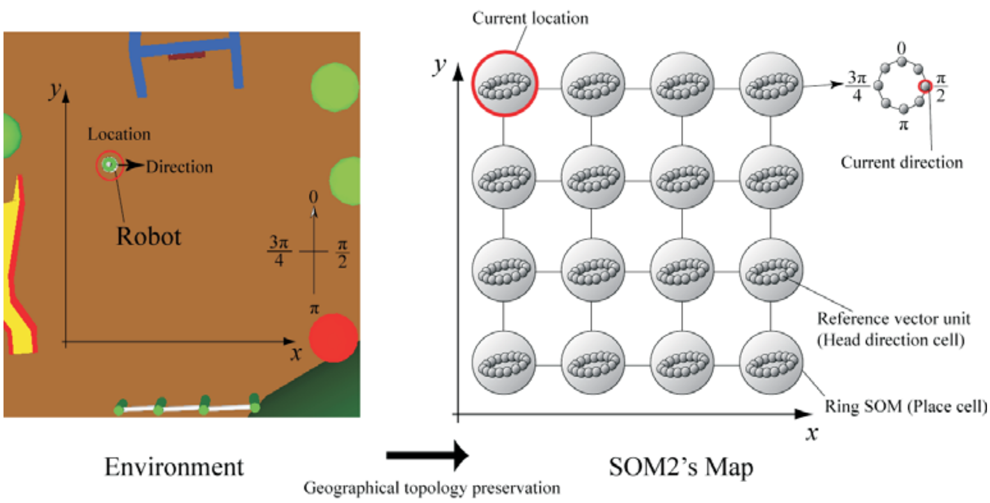


Fig. 3. Architecture of RSOM×SOM that is engineering model of the place cells and the head direction cells. The map of the parent SOM itself represents a geometrical map of the environment. Each reference vector unit of RSOM represents a head direction cell. Moreover, Each RSOM represents the place cells.

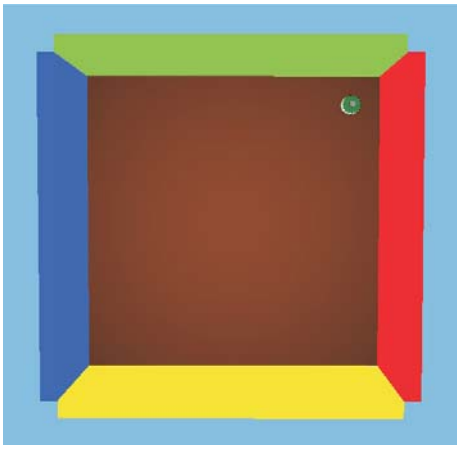


Fig. 4. Type 1 environment.

Fig. 5. Type 2 environment.

the RSOM, but on a lattice in the SOM. Otherwise they are the same.

First, we define certain variables. Suppose there are I training episodes where each episode is composed of J data vectors. The i-th episode is defined as $D_i = \{x_{i1}, x_{i2},..., x_{ij},..., x_{iJ}\}$, where x_{ij} is the vector data. Furthermore, the parent SOM is composed of K RSOM modules. Each RSOM module (i.e., child SOM) has L reference vectors. Now, the set of reference vectors in the k-th RSOM module is defined as $w^k = \{x^{k1}, x^{k2},..., x^{kl},..., x^{kL}\}$.

In the training of a RSOM×SOM, the following three processes are repeated: (1) evaluative process, (2) cooperative process, and (3) adaptive process. These processes are explained below.

(1) Evaluative process

First, error e_{ij}^{kl} between each data vector and each reference vector in all child SOMs is calculated as follows:

$$e_{ij}^{kl} = \left\| w^{kl} - x_{ij} \right\|^2 . \tag{1}$$

Here, the index of the best matching unit (BMU) l_{ij}^{k*} is defined as

$$l_{ij}^{k*} = \arg\min_l e_{ij}^{kl} . \tag{2}$$

Next, error E_i^k in each child SOM module for each episode is calculated as

$$E_i^k = \frac{1}{J}\sum_{j=1}^{J} e_{ij}^{k*} \tag{3}$$

where e_{ij}^{k*} is the error of BMU l_{ij}^{k*} for x_{ij}. Thus, error E_i^k in each child SOM module is the mean of e_{ij}^{k*} for all data vectors in one episode. Moreover, the best matching module (BMM) l_i^* for the i-th episode is defined as

$$k_i^* = \arg\min_k E_i^k . \tag{4}$$

(2) Cooperative process

In the cooperative process, the learning rates α_i^k and β_{ij}^l are calculated to decide the update values of all reference vectors. α_i^k is defined as follows:

$$\alpha_i^k = \frac{\phi_i^k}{\sum_{i'} \phi_{i'}^k} \tag{5}$$

$$\phi_i^k = \exp\left(-\frac{d\left(k, k_i^*\right)^2}{2\sigma_P^2(t)} \right) \tag{6}$$

$$\sigma_P(t) = \sigma_{Pmin} + (\sigma_{Pmax} - \sigma_{Pmin}) \cdot \exp\left(-\frac{t}{\tau_P}\right), \tag{7}$$

where α_i^k is the learning rate at the parent level, ϕ is a neighborhood function, and $d(k, k_i^*)$ represents the distance between the k-th module and the BMM on the map. In addition, $\sigma_P(t)$ represents a neighborhood radius, which decreases exponentially with learning step t. In this study, $\sigma_P(t)$ is defined as Eq. (7). σ_{Pmax} and σ_{Pmin} are the maximum and minimum radii of the neighborhood function, respectively. Moreover, τ_P is a time constant for decreasing the speed of the neighborhood radius. Next, β_{ij}^l is defined as follows:

$$\beta_{ij}^l = \frac{\phi_{ij}^l}{\sum_{j'} \phi_{ij'}^l} \tag{8}$$

$$\phi_{ij}^l = \exp\left(-\frac{d(l, l_{ij}^{**})^2}{2\sigma_C^2(t)}\right) \tag{9}$$

$$\sigma_C(t) = \sigma_{Cmin} + (\sigma_{Cmax} - \sigma_{Cmin}) \cdot \exp\left(-\frac{t}{\tau_C}\right). \tag{10}$$

β_{ij}^l is the learning rate at the child level. Note that the distance between the l-th reference vector and the BMU of the BMM on the map is calculated as Eq. (9). This encourages the preservation of homogeneity in the map of each child SOM. σ_{Cmax}, σ_{Cmin}, and τ_C are the maximum radius, minimum radius, and time constant at the child level.

(3) Adaptive process

All reference vectors are updated by

$$w^{kl} = \sum_{i=1}^{I} \sum_{j=1}^{J} \alpha_i^k \beta_{ij}^l x_{ij}. \tag{11}$$

In training an RSOM×SOM, the above three processes are repeated.

4. Simulation

This section presents the verification results for two types of simulations. The purpose of the simulations is to confirm whether the RSOM can build an environmental map to estimate the position and head direction using only vision sensor information.

4.1. Framework for simulations

Using the "Webots" robotics simulation software developed by Cyberbotics Ltd., we created two types of working environments for the simulations. Type 1 is an environment in which four walls are painted red, blue, green, and yellow (illustrated in Fig. 4). Type 2 is a park-like working environment shown in Fig. 5. The area in which the robot is able to move is a meter long by a meter wide (Figs. 4 and 5). Furthermore, the robot has an omni-directional vision sensor. Fig. 6 is an example of a panoramic image taken from the omni-directional vision sensor. The size of the panoramic image is 512 x 64 pixels. In addition, the colors of the panoramic image are converted to 64 colors (in other words, red, green and blue are converted to 4 colors).

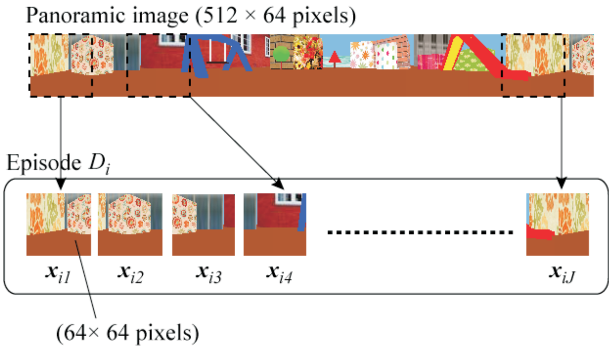


Fig. 6. Upper image is example of panoramic image observed from vision sensor. The episode given to RSOM×SOM is created from the panoramic image.

Next, we explain the episodes given to the RSOM×SOM. An episode $D_i = \{x_{ij}\}$ is created from the observed panoramic image. Data vector x_{ij} is a color histogram vector extracted from an image (size 64x64) clipped from the panoramic image. For all data vectors, the color histograms are extracted evenly from the entire panorama image.

Next, the simulation flow is explained. First, the robot moves around randomly in the environment, while simultaneously, observing the panoramic images at various positions in the working environment. Next, the set of episodes extracted from the panoramic images are given to the RSOM×SOM as training episodes. Here, note that X-Y coordinate is not included in training episodes. The training of the RSOM×SOM is performed offline. After the training process, two places (A) and (B) are verified to confirm whether the RSOM has built an environmental map that can estimate the position (brief X-Y coordinate) and head direction using only vision sensor information from the two types of environments.

(A) Confirmation of estimating the position

The BMM on the RSOM×SOM is monitored when the robot moves to an arbitrary place in the working environments. If the RSOM×SOM can build a map in which the topology of the geography is preserved, then the topology of the robot's places is almost the same as that of the BMM's positions.

(B) Confirmation of estimating the head direction

First, after the robot is put in an arbitrary place, the episode observed from this place is given to the RSOM×SOM. The robot is turned to face north and then, a BMM corresponding to the episode is decided. In addition, a BMU in the BMM is decided after the color histogram vector of the front image (64x64 pixels) is given to the BMM. If the robot is rotated on the spot, the BMU will change continuously on the map of the RSOM. Thus, the map of the RSOM preserves the topology of the direction. In addition, it is expected consistency be maintained in the reference vectors of every module.

4.2. Simulation results in Type 1 environment

First, before the training of the RSOMxSOM, the robot moved randomly in the environment, and simultaneously, observed the set of panoramic images. In Fig. 7 the trajectory of the robot is depicted by "-" , while "●" denotes the positions at which the panoramic images were observed. Note that the direction of the robot was not fixed. Panoramic images were observed from 200 positions; in other words, there were 200 training episodes. In addition, there were 32 data vectors per episode.

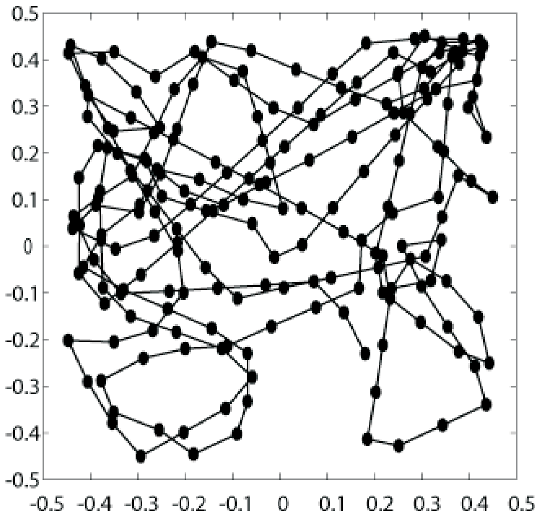


Fig. 7. The trajectory of the robot at observation of visual information in type 1 environment. The trajectory of the robot is depicted by "-", while "●" denotes the positions at which the panoramic images were observed.

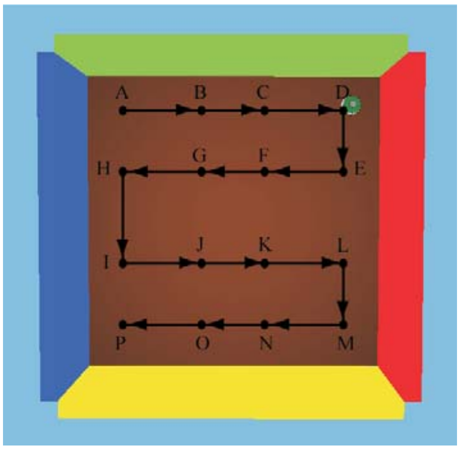


Fig. 8. the trajectory of the robot in "(A) Confirmation of estimating the position" in type 1 environment.

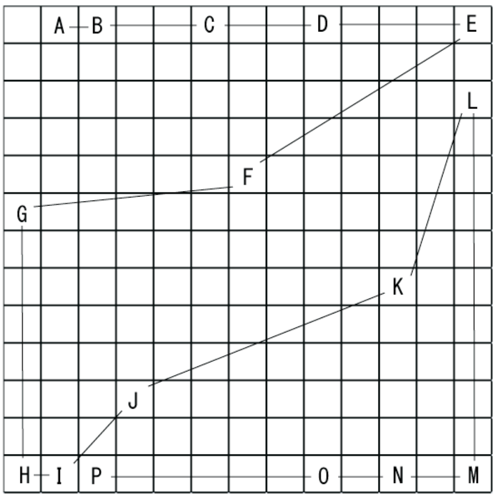
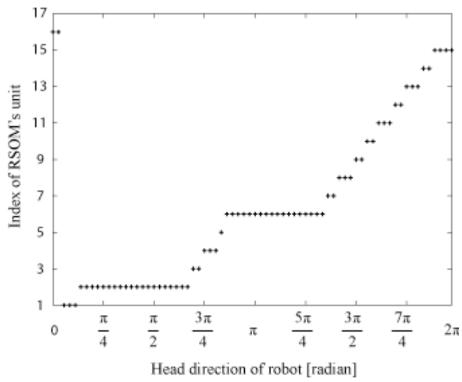


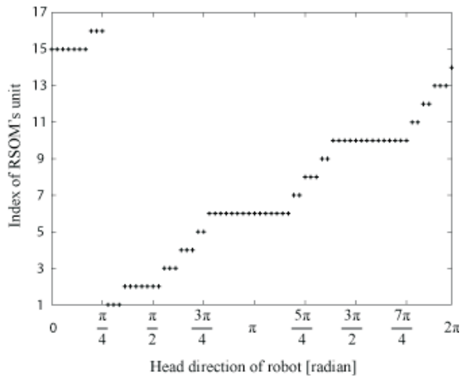
Fig. 9. The result of map building using RSOM×SOM in type 1 environment. The lattices denoted by letters of the alphabet are the BMMs at the positions shown in Fig. 8.

The results of "(A) Confirmation of estimating the position" are shown in Figs. 8 and 9. Fig. 8 shows the trajectory of the robot. In addition, Fig. 9 shows the RSOM×SOM's map in which each lattice corresponds to an RSOM module. The episodes observed at positions "A" to "T" in Fig. 8 were given to the RSOM×SOM as test data. Moreover, the lattices denoted by letters of the alphabet in Fig.9 are the BMMs at the positions shown in Fig. 8. Thus, it is possible for an RSOM×SOM to build a map that preserves the topology of the geography. The same result was obtained consistently despite training being repeated several times. The results of "(B) Confirmation of estimating the head direction" are shown in Figs. 10 a), (b), and (c). Each figure is a result at putting the robot on A, F, and J places in Figure 8, respectively. Having been placed at each position, the robot was rotated 360 degrees in intervals of 5 degrees. In Figs. 10a), (b), and (c), the relationship between the head direction of the robot and the BMU is shown. These results confirm that the head direction of the robot and the BMUs change continuously. Moreover, the head direction was able to be estimated by BMU easily.

(a)



(b)



(c)

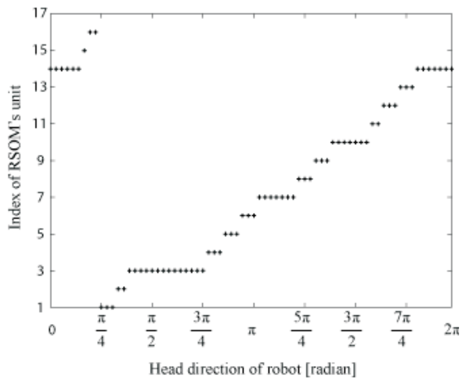


Fig. 10. The result of "(B) Confirmation of estimating the direction" in type 1 environment. (a), (b), and (c) are the relationships between head direction of robot and RSOM's unit at position A,F, and J in Fig. 8 respectively.

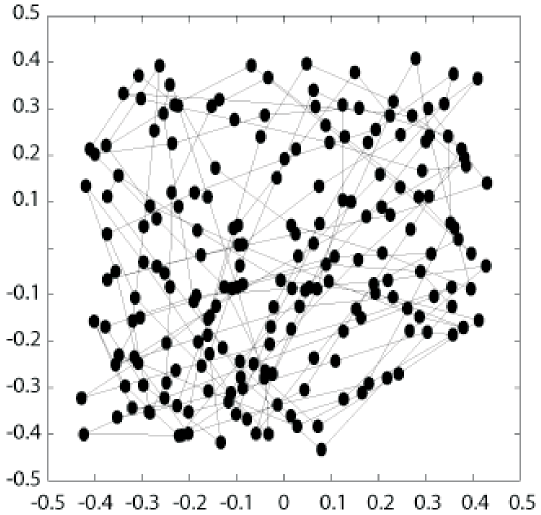


Fig. 11. The trajectory of the robot at observation of visual information in type 2 environment.

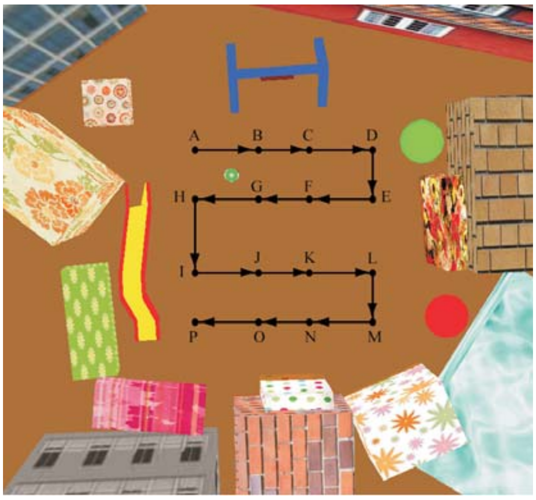


Fig.12. the trajectory of the robot in "(A) Confirmation of estimating the position" in type 1 environment.

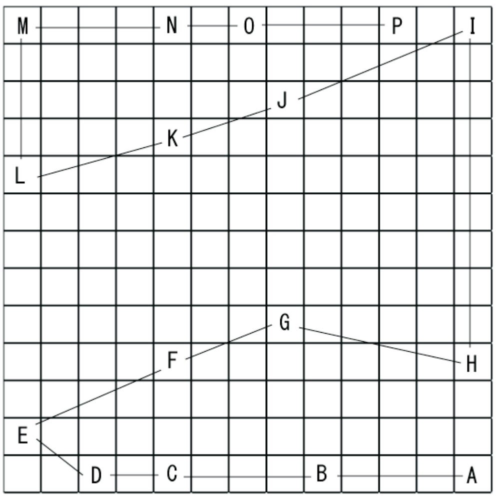


Fig.13. The result of map building using RSOM×SOM in type 2 environment. The lattices denoted by letters of the alphabet are the BMMs at the positions shown in Fig. 12.

4.3. Simulation results in Type 2 environment

Fig. 11 shows the trajectory of the robot in environment of Fig. 5. The panoramic images were observed at 200 positions; in other words, there were 200 training episodes. In addition, there were 32 data vectors per episode. The results of "(A) Confirmation of estimating the position" are shown in Figs. 12 and 13. These results suggest that an RSOM×SOM is able to build a map that preserves the topology of the geography even if the visual information varies. The results were consistent despite training being done several times. The results of "(B) Confirmation of estimating the head direction" are shown in Figs. 14. Each figure is a result at putting the robot on A, G, and K places in Figure 14, respectively. In the results, BMU corresponding to the head direction of the robot has not been continuously changed. These results suggest that estimation of the head direction was difficult because of the existence of a similar color histogram.

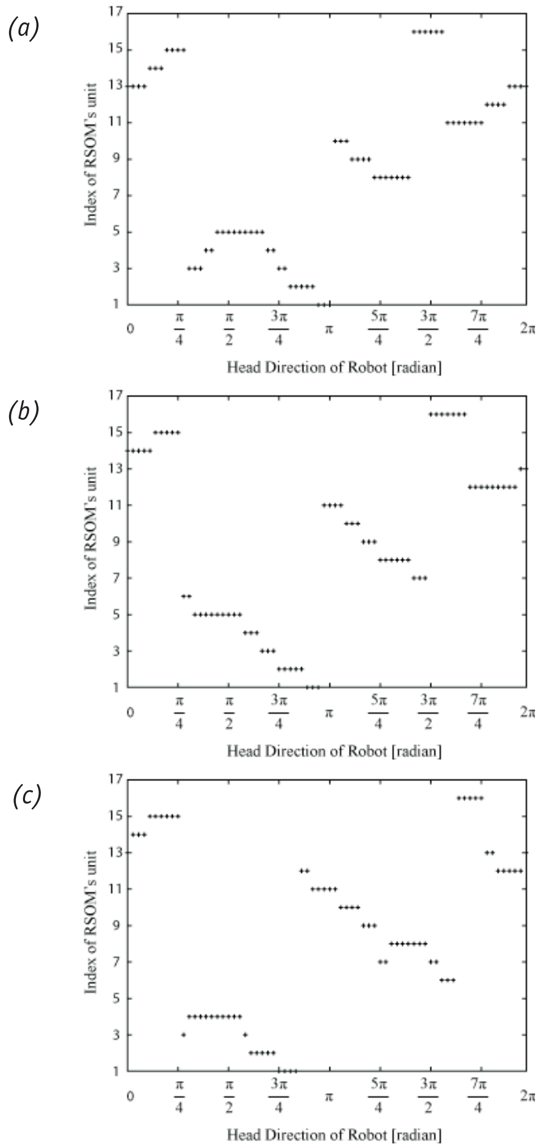


Fig.14. The result of "(B) Confirmation of estimating the direction" in type 2 environment. (a), (b), and (c) are the relationships between head direction of robot and RSOM's unit at position A,G, and K in Fig. 12, respectively.

5. Discussion

5.1. Map building in complex environments

We have described map building using an SOM^2 in a complex environment containing obstacles and similar vision information. In an adaptation of the SOM^2, child SOMs (i.e., place cells) are allocated to unreachable places, since the parent SOM is fixed in the lattice topology. A solution to this problem is to use a self-organizing neural network as the parent SOM, such as the NG and TRN, where the network topology is not fixed. It is shown in [4] that the parent and child of the SOM^2 can be designed using any SONN, besides the SOM. It is expected that the place cells are allocated only to the subspace in which input episodes are distributed, by replacing parent SOMs with NGs. Besides, when similar vision information exists, then it is considered that it is necessary to introduce the method of the map building including a time transition to the algorithm of SOM^2.

5.2. Feature extraction from vision information

In this study, the color histogram was used as a simple feature extraction, since the study aims to verify the map building by SOM^2. It was shown that the map building from only color information was possible by the simulations. However, it is difficult in the real environment to distinguish the local environment from only color information. Therefore, the technique for recognizing the environment such as SIFT [15] is requested to be used as a feature extraction. 5.3. RSOMxSOM's responses for changing environment.

Fig. 15. Type 2 environment to which the human is added.

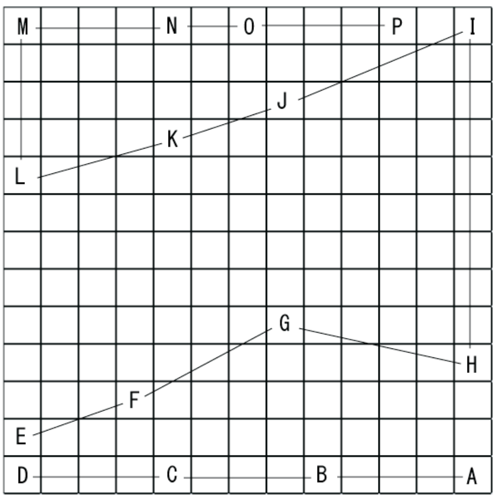


Fig.16. The result in type 2 environment to which the human is added.

Even if some of the environment change, it is possible to estimate the position and direction with map that was built by training of the SOM^2. We experimented as follows to verify the issue. First, the map was built in the type 2 environment. Next, a human's object was put on the environment (Fig.15). Namely, some of the environment was changed with the human's object. Final, (A) and (B) of simulation were verified. The results are shown to Fig.16. BMMs on the map changed as shown in Fig.16 when the robot was moved as shown in Fig.12 on the environment. There is little difference between result (Fig.13) of the environment where the human's object is not put and this result (Fig.16). Thus, it is suggested that a part of change not influence the position estimation.

6. Conclusion

In this paper, we confirmed that the SOM^2 could build a cognitive map that includes features of the place cells and head direction cells. It was shown that both the position and the azimuth direction could be estimated from the map acquired by unsupervised learning of the SOM^2. The SOM^2 model is not based on the neurological function of the hippocampus, but is modeled technologically in a topological way. A model that imitates the function of the cognitive map in animals more closely can be developed by creating an algorithm that introduces a time transition of information into the SOM^2.

AUTHORS

Kazuhiro Tokunaga*, Tetsuo Furukawa - Department of Brain Science and Engineering, Kyushu Institute of Technology, 2-4 Hibikino, Wakamatsu-ku, Kitakyushu 808-0196, Japan. E-mails: tokunaga@brain.kyutech.ac.jp, furukawa@brain.kyutech.ac.jp.
* Corresponding author

References

[1] O'keefe J., Dostrovsky J., "The hippocampus as a spatial map: preliminary evidence from unit activity in the freely moving rat ", *Brain. Res.*, vol. 34, 1971, pp. 171-175.

[2] O'keefe J. , Nadal L., *The Hippocampus as a Cognitive Map*, Oxford: Oxford. University Press, 1978.

[3] Taube J.S., Muller R.U., Ranck J.B. Jr., "Head-direction Cells Recorded from the Postsubiculum in Freely Moving Rats", *Journal of Neuroscience*, vol. 10, 1990, pp. 436-447.

[4] Furukawa T., "SOM of SOMs", *Neural Networks*, vol. 22, issue 4, 2009, pp. 463-478.

[5] Kohone T., *Self-Organizing Maps*, Springer-Verlag: New York, 2001.

[6] Gamini Dissanayake M.W., Newman P., Clark S., Durrant-whyte H.F., Csorba M., "A solution to the simultaneous localization and map building (SLAM) problem", *IEEE Transactions on Robotics and Automation*, vol. 17, 2001, pp. 229-241.

[7] Durrant-Whyte H.F., Bailey T., "Simultaneous Localisation and Mapping (SLAM): Part I The Essential Algorithms", *Robotics and Automation Magazine*, vol. 13, 2006, pp. 99–110.

[8] Sim R., Elinas P., Griffin M., “Vision-based SLAM using the rao-blackwellised particle filter”.” In: *IJCAI Workshop on Reasoning with Uncertainty in Robotics*, 2005.
[9] Se S., Lowe D.G., Little J.J., “Vision-based global localization and mapping for mobile robots,” IEEE Transactions on Robotics, vol. 21, Issue 3, 2005, pp. 364-375.
[10] Yairi T., “Map Building without Localization by Dimensionality Reduction Techniques”. In: *The 24th International Conference on Machine Learning (ICML-2007)*, 2007.
[11] Trullier O., Wiener S.I., Berthoz A., Meyer J., Berthelot P.M., “Biologically-based Artificial Navigation Systems: Review and prospects”, *Progress in Neurobiology*, vol. 51, 1997, pp. 483-544.
[12] Martinetz T., Schulten K., “Topology Representing Networks”, *Neural Networks*, vol. 7, issue 3, 1994, pp. 507-522.
[13] Takahashi T., Tanaka T., Nishida K., Kurita T., “Self-organization of place cells and reward-based navigation for a mobile robot”. In: *Proceedings of ICONIP’01*, 2001, pp. 1164-1169.
[14] Chokshi K., Wermter S., Panchev C., Burn K., “Image Invariant Robot Navigation Based on Self Organizing Neural Place Codes”, *Lecture notes in computer science*, vol. 3575, 2005, pp. 88-106.
[15] Loww D.G., “Distinctive Image Features from Scale-Invariant Keypoints”, *International Journal of Computer Vision*, vol. 60, no. 3, 2004, pp. 91-110.