An Autonomous Mobile Robot with Functions of Action Learning, Memorizing, Recall and Identifying the Environment Using Gaussian Mixture Model

Masanao Obayashi[1], Taiki Yamane[1], Takashi Kuremoto[1], Shingo Mabu[1], and Kunikazu Kobayashi[2]

[1]Yamaguchi University, 2-16-1 Tokiwadai, Ube, Yamaguchi 755-8611, Japan
{m.obayas, s050vk, wu, mabu}@yamaguchi-u.ac.jp
[2]Aichi Prefectural University, Nagakute, Aichi, 480-1198, Japan
kobayashi@ist.aichi-pu.ac.jp

Abstract. In this paper, behavior scheme of autonomous mobile robots to achieve the objectives of them in environments are proposed, having function of identifying the current environment in which they are placed and making use of learning, memorizing and recalling behaviors of corresponding to each of plural different environments. Specifically, each robot has the function of identifying the environment using some behavioral statistical data for each environment, and if the robot has already experienced the environment, it behaves by making use of own experienced data stored in the database, otherwise it performs a new behavior learning and adds the learning results into the database.

Keywords: intelligent robot, chaotic neural network, reinforcement learning, identification of the environment, Gaussian mixture model

1 Introduction

In recent years, researches on intelligent robots and their practical application have been attracting attention. For example, meal transport robot, workplace patrol robot, document delivery robot, etc. and their applications have expanded dramatically.

However, such robots are not called so intelligent, because that almost all of them are given the path to the place of destination and only move along the same path repeatedly. Furthermore, in the field of discrimination of the environment for agent action learning, we can hardly see papers about such a subject. As previous researches on identification method of the environment, there are a few ones that are our papers published in the past, for example, the method [1] that check in order the environments in the database whether the series of actions in the environment information in the database are appropriate for the current environment or not. The other is the method [2] identifying the environment using the feedback SOM with high efficiency and high precision. In this paper, behavior scheme of autonomous mobile robots to achieve the objectives of them in environments are proposed, having functions of identifying the current environment in which they are placed and making use of learn-

ing and memorizing behaviors of corresponding to each of plural different environments. Specifically, each robot has the function of identifying the environment using some behavioral statistical data for each environment, and if the robot has already experienced the environment, it behaves by making use of own experienced data memorized in the database constructed by chaotic neural networks (CNNs) and Gaussian mixture models, otherwise it performs a new action learning and adds them into the database. In this paper, as the database to perform the memorization and recall of pairs of action and perceptual information corresponding to the action, we also adopt chaotic neural networks (CNNs) [3] which is a coupled system of chaotic neuron that models the refractory and analog input and output characteristics, as an action learning method, we adopt reinforcement learning, a kind of representative machine learning.

2 Whole structure of the system

The whole structure of the system including the proposed agent and environment are shown in Figure 1. The structure of the agent is inside surrounded by a broken line, it consists of three functions: identification of the environment, action learning and database with pairs of state of the environment and action corresponding to it. At first, the agent receives state information from the environment in which the agent is placed, it identifies the environment in the database as same as the facing environment, in other word, the agent investigates the database whether there is the environment that matches the facing environment in it, or not. If the information about the facing environment has already memorized in the environment, it behaves by making use of own experienced data memorizing in the database, otherwise it performs a new action learning and adds the results of action learning into the database.

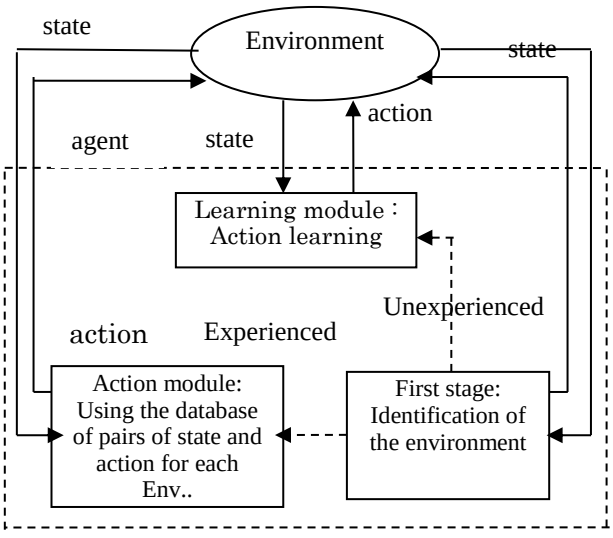

Fig. 1. Whole structure of the proposed autonomous agent

3 Action learning

In this paper, as an action learning approach, Q learning method is used, which is one of typical reinforcement learning method. For more information, see reference [4]. After end of action learning, pairs of action and perceptual information corresponding to the action and Gaussian mixture model (GMM) about the environment are added as labeled information into the database which consisting of CNNs and GMMs.

4. State-action storage system

After end of action learning, pairs of useful state and action sequences by trial and error learning are stored in the state-action storage unit, that is, in the database. There are two kinds of memory; short term memory and long term memory. In this paper, the short-term memory corresponds to the transient memory when an action learning, while long-term memory does to the memory gotten at the end of action learning, which is transferred to the state-action storage unit called database. CNNs are adopted as the long-term memory. We describe CNN below shortly.

4.1 Chaotic neural network (CNN)

CNN is configured by interconnecting chaotic neuron models, represented by the following formula [3].

$$x_i(t+1) = f(y_i(t+1) + z_i(t+1)) \tag{1}$$

$$y_i(t+1) = k_r \cdot y(t) - \alpha \cdot x_i(t) + a_i \tag{2}$$

$$z_i(t+1) = k_f \cdot z_i(t) + \sum_{j=1}^{N} w_{ij} x_j(t) \tag{3}$$

$$w_{ij} = \frac{1}{P} \sum_{p=1}^{P} (2x_i^p - 1) \cdot (2x_j^p - 1) \tag{4}$$

Here, $x_i(t), y_i(t), z_i(t)$ are output value of ith neuron at the time t, internal state on the refractory and internal state of the interaction, respectively. α is a constant parameter, w_{ij} is the connection weight from jth neuron to ith neuron, k_r is the time decay constant for the refractory, k_f relates to the interaction time decay constant, a_i compound term of external input and the threshold of the input, x_i^p means ith component of pth storage pattern, P means the total number of storage patterns, N shows the total number of neurons.

4.2 Memorizing and recall of an environmental data using mutual associative type CNN

While our mutual associative CNN (MACNN) has same structure of auto-associative CNN (AACNN) and MACNN operates same as AACNNs, it looks working as mutual associative memory model. Figure 2 shows the structure of the MACNN, consisting

of three parts; input, hidden and output part. In the hidden part of neurons, random data are stored in order to make correlations between input data and output data weaken. When pairs of both environment and learned action corresponding to the environment gotten by the action learning are memorized into MACNN, the perceptual patterns are memorized into the input part of neurons and action patterns are memorized into the output part of the neurons using Eq. (4). When making use of the MACNN, the current environment perceptual pattern is set as the output (x in Sec 4.1) of the input part of neurons, fixing their values, let MACNN operate, after the MACNN converges, a corresponding action pattern is recalled on the output part of neurons. Please refer to chapter 11 in the reference [4] for detail explanation

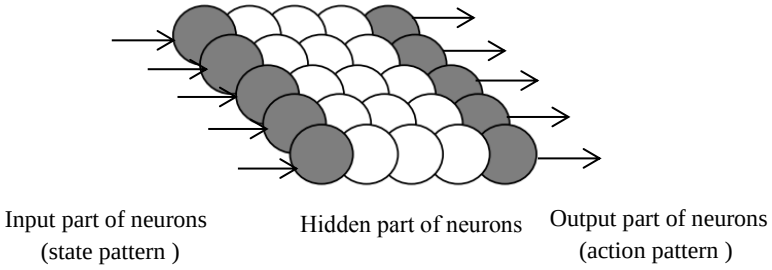

Fig. 2. Structure of mutual associative type CNN (MACNN)

5. Identification of the environment

When being stored the features of plural different environments into the database, environmental state patterns of first several random steps from the start position are gotten many times for each environment, and statistical information for these patterns is memorized by Gaussian mixture model (GMM). On the other side, when identifying the facing environment, its GMM is gotten in the same manner. Comparing the GMM in the database and the GMM for the facing environment, we judge whether the agent has already experienced the facing environment or not. Details are described below.

5.1 Identification method of the environment

In this section, we describe the method to identify the environment. We propose the identification method using GMM. GMM is represented by mean μ and variance σ^2. GMM is used as a distribution function for identification of the environment. GMM is built up using the amount of observation data $s\ (\in R^1)$ resulting from (T × X) times actions where X means the number of trials, one trial is T step random actions from start position. In 5.1.1, building up algorithm of GMM is explained.

5.1.1 Mean and variance configuration algorithm

The algorithm for determining the mean and variance of constructing a GMM is shown below.

Step1 : (Initialize) Using the first observation state data $s\ (\in R^1)$, mean and variance of the first Gaussian distribution $f_1(s)$ are set as follows,

$$\mu_m = s, \sigma_m^2 = 1.0\ (m = 1)$$

Here, m is number of Gaussian distribution $f_m(s)$ that makes up the MMG.

Step2 : Modify the mean and variance using the next data s as follows,

(1) when $-3\sigma_m < \min|\ \mu_m - s\ | < 3\sigma_m$,

$$\mu_m = (1-\alpha) * \mu_m + \alpha * s\ ,\ \sigma_m^2 = (1-\alpha) * \sigma_m{}^2 + \alpha * (x-\mu_m)^2, \tag{5}$$

(2) otherwise, generate next number of Gaussian distribution that makes up the GMM as follows,

$$m = m+1\ ,\ \mu_m = s\quad ,\ \sigma_m = 1.0\ .$$

Step3 : Back to Step2 if it is within the specified number of times, otherwise , GMM, $f(s)$, obtained by using the mean-variance obtained by the algorithm is represented by Equation (6).

$$f(s) = \sum_{m=1}^{M} \frac{w_m}{K} * f_m(s) \tag{6}$$

$$f_m(s) = \frac{1}{\sqrt{2\pi\sigma_m{}^2}} \exp(-\frac{(s-\mu_m)^2}{2\sigma_m{}^2}) \tag{7}$$

Here, K : the number of data, w_m : the number of data used during mth Gaussian distribution creation, M is the number of Gaussian distributions.

5.1.2 Identification algorithm of the environment

At first, the feature vector for each environment in the database to identify the environment are defined as U-dimensional vector consisting of U-dimensional probability values calculated using GMM (Eq. (6)) by observation resulting from T times random actions from start position in the current environment. Here U is the number of different kinds of the states within the T observation states. We identify the environment by the degree of similarity between each of all the feature vectors in the database and the current probability vector. The algorithm is as follows,

Step 1: Using the observation state vector, calculate all the feature vectors corresponding to each environment presented by GMM in the database and the current probability vector for the observation state vector.

Step 2: Calculate the degrees of similarity $\cos\theta$ between the feature vector $\vec{a}$ and the current probability vector $\vec{b}$, here θ is angle between the feature vector and the current probability vector.

$$\cos\theta = \frac{\vec{a}\cdot\vec{b}}{|\vec{a}|\cdot|\vec{b}|} = \frac{\sum_{i=1}^{U}(a_i \cdot b_i)}{\sqrt{\sum_{i=1}^{U}(a_i)^2}\sqrt{\sum_{i=1}^{U}(b_i)^2}} \tag{8}$$

Here, $\vec{a} = (a_1, \cdots, a_U)$, $\vec{b} = (b_1, \cdots, b_U)$, $l = \arg\max_j \theta_j$, $j \in L$. Here , L is the number of the storage environments in the database, that is, they have already learned and stored.

(i) When $\theta_l \le \theta_{th}$, the agent judges that the storage environment l is most similar to the current environment, and pairs of stored state and action corresponding to the environment l is determined to be available, moves to action module. (θ_{th} is the threshold of the similarity of the environment.)

(ii) When $\theta_l > \theta_{th}$, the agent judges that there is no storage environments to be available and moves to action learning module.

6. Computer simulation

In the maze problem, the performance comparison of the proposed method and conventional method [2] through the following two simulations is carried out. The proposed method identifies the mazes using Gaussian mixture models, and also the conventional method identifies using feedback SOMs. As data for construction of identification systems and identification of unknown mazes for each maze, when first 5 actions (T = 5 in Section 5.1) from the start position is called one trial, we get the 150 perceptional patterns through 30 trials (X = 30) .

6.1 Preparation

Put the following assumptions at GMM construction of the environment.

(i) GMM is constructed by the data during several actions from the start position in the maze.

(ii) The agent acts T steps randomly without use of the storage information.

(iii) The agent stays in the same position when colliding with the obstacle.

(iv) It is possible for the agent to take only one of next three actions: forward, left and right without backward per a step.

In this simulation, it is assumed that the agent is able to perceive an aisle or an obstacle toward the five directions up to one square away around it as shown in Figure 3.

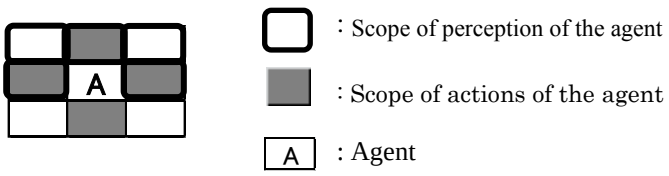

Fig. 3. Scope of perception and action of the agent

6.2 Observation state description

The feature extraction algorithm of the environment is as follows,

Step1 : Put the agent at the start position.

Step2 : Repeat Step3 to Step4 T (number of actions) times.

Step3 : Move one cell to random direction and get the observation information according to the following equation,

$$evi\ (\text{direction}) = \begin{cases} 1 & : \text{adjacent is obstacle,} \\ 0 & : \text{adjacent is aisle,} \end{cases} \tag{9}$$

$evi\ (\text{direction})$: observation information toward the observation direction.

Step4: Get a feature value, a perceptual pattern information named "state", for each step according to the following equation,

$$\begin{aligned} \text{state} = {} & evi\ (\text{right}) + 2 \cdot evi\ (\text{front right}) + 4 \cdot evi\ (\text{front right}) \\ & + 8 \cdot evi\ (\text{left}) + 16 \cdot evi\ (\text{front left}). \end{aligned} \tag{10}$$

Here, state $(0 \le state \le 31)$ represents one of the situations $(= 2^5)$ around the agent (See Fig. 3)..

6.3 Simulation 1

We consider multiple maze problems. After end of action learning for each maze, pairs of perceptual pattern and action along optimal path from the start to the goal were stored into the database created by the CNN group and one of the stored mazes is given to the robot to verify whether the maze is correctly identified or not. The robot has perception-action capabilities as shown in Figure 3, assuming that the robot always faces the upward direction and is able to move only one cell in an action. The predefined number of mazes are generated with the following procedure.

(i) The size of the maze is randomly determined
(ii) A start and goal positions in the maze are randomly decided.
(iii) Walls in the maze are randomly placed

Here, white cell is passage, black cell is wall, S is a start position, G indicates a goal. Among the mazes created by above procedure, mazes that can't reach the goal (e.g., bottom right of Figure 6), or has shorter path less than T steps are excluded. As a result, the discrimination rate in the conventional method and the proposed method obtained in the simulation are shown in Table 1. It shows the average discrimination rate of 10 trials (the number of available mazes per trial is 25).

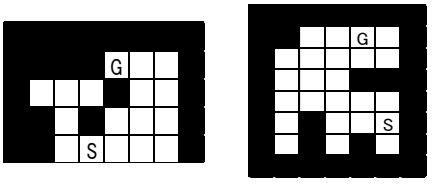

Fig. 4. Failure examples of identification using the conventional method

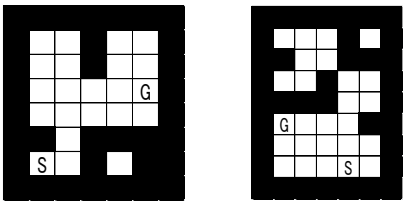

Fig. 5. Left : success example of identification, right : failure example of identification using the proposed method

Table 1. Results of simulation 1

	The conventional method	The proposed method
Discrimination rate	65.20 [percent]	71.40 [percent]

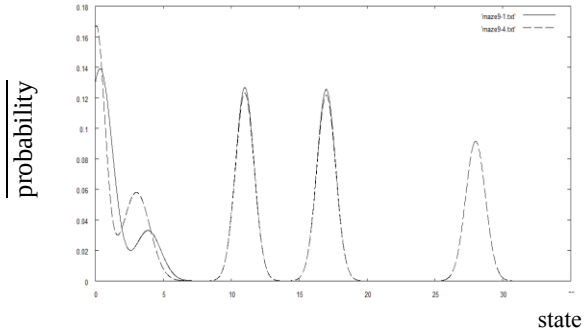

Fig. 6. Gaussian mixture probability density functions generated by left maze in Figure 5 (two times construction)

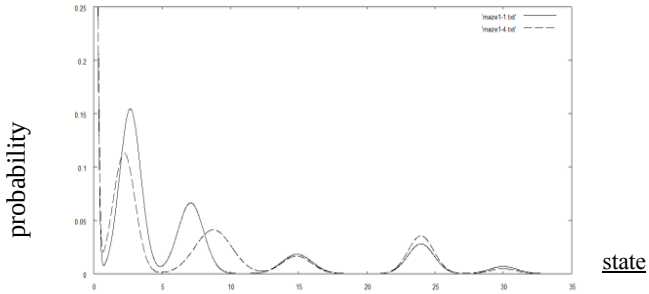

Fig. 7. Gaussian mixture probability density functions generated by right maze in Figure 6 (two times construction)

Figure 6 and 7 show Gaussian mixture probability density functions generated by left maze and right maze in Figure 5, respectively. These are constructed by the proposed method two times in the same manner. The former is success example and the latter is failure example. The former case looks less difference between two probability density functions than the latter one.

6.4 Simulation 2

Figure 8 shows 4 mazes to store into the database after end of action learning. White rhombus and black one mean a start position and goal position, respectively. Triangles means the optimal path at the end of learning and these results are memorized in the database. In this simulation, at first, for 4 mazes shown in Figure 8, optimal paths from the start to the goal through Q learning are learned, after end of learning, pairs of perceptual patterns and their optimal actions for them, and next, for each maze GMM is constructed by the proposed method. Both information are memorized into the database. Lastly, we try to solve a little large-size maze shown in Figure 9, while making use of these information of 4 mazes in the database of the agent with much less of additional action learning. As comparing the method with our proposed method, the method [2] as same as the proposed system without use of the discrimination method of the environment, namely "feedback SOM" is adopted. The left maze in Figure 9 shows the failure results of the conventional method because that the agent behaved along while using information about the incorrect maze (b) instead of correct maze (a) in Figure 8.

Figure 9 shows Simulation result (Left : the conventional method, right : the proposed method) for a little large-scale maze, making use of storage information about mazes in Figure 8. In Figure 9, the white triangles (T = 5) mean the action series of the agent when identifying the environment and black ones mean the action series by making use of the stored information. At the start position, the proposed method correctly identified the facing environment as maze (a) and it succeeds to get the goal. However, the conventional method failed to identify the facing the maze as maze (b) in shown Figure 8.

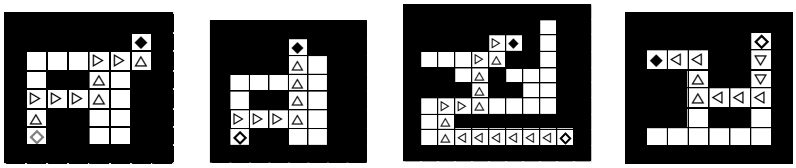

Fig. 8. The 4 optimal paths to be stored into the database after action learning

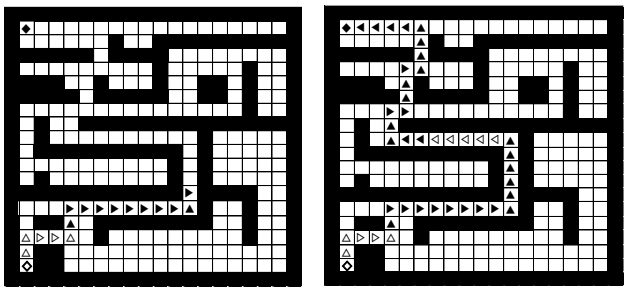

Fig. 9. Simulation result (Left : the conventional method, right : the proposed method) , making use of storage information about mazes in Figure 8

Figure 10 shows two Gaussian mixture probability density functions for maze (a) and (b). It says that they are very similar. Table 2 shows similarities between the maze (a) and (b) in Figure 8 and the maze of Figure 9 for the first T steps from the start position of the maze. The table shows the maze (a) is more similar than maze (b) for the maze (9) at the start point.

Table 2. Similarity between maze (a) , (b) in Figure 8 and the first T step action from the start position of maze of Figure 9.

Maze (a)	Maze (b)
0.988	0.856

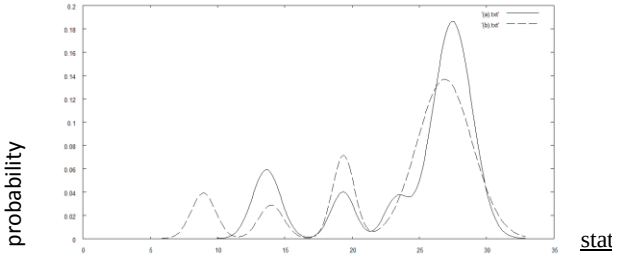

Fig. 10. Gaussian mixture probability density function obtained using maze (a) and (b) in Figure 8 (maze (a) : solid line, maze (b) : broken line)

7. Conclusions and Future Challenges

We proposed the construction method of a kind of the intelligent agent that has functions of action learning, memorizing, recall and identifying the environment using Gaussian mixture model. The robot with these functions could go to the goal efficiently, making use of the information of their experienced environment before.

Reference

[1] M. Obayashi, K. Narita, T. Kuremoto, K. Kobayashi, "A Reinforcement Learning System with Chaotic Neural Neworks-Based Adaptive Herarchical Memory Structure for Autonomous Robots, Proc. of International Conference on Control, Automation and Systems 2008 (ICCAS 2008), pp. 69-74, 2008

[2] M. Obayashi, K. Narita, Y. Okamoto, T. Kuremoto, K. Kobayashi and L. Feng, A Reinforcement Learning System Embedded Agent with Neural Network-Based Adaptive Hierarchical Memory Structure", In Advances in Reinforcement Learning, Chapter 11 , pp.189-208, IN-TECH , 2011

[3] M. Adachi, K. Aihara: "Associative Dynamics in a Chaotic Neural Network", *Neural Networks* ,Vol. 10, No. 1, pp.83-98, 1997

[4] R.S. Sutton, A.G. Barto, Reinforcement Learning : An Introduction, Cambridge, MA:MIT PRESS, 1988