Modeling Approaches to Brain-like Information Technology

Masumi Ishikawa, Fredrik Linåker, Masahiro Nagamatu, Takeshi Yamakawa, Syozo Yasui
Department of Brain Science and Engineering, Graduate School of Life Science and Systems Engineering,
Kyushu Institute of Technology, Kitakyushu, Fukuoka 808-0196, Japan
{ishikawa, linaker, nagamatu, yamakawa, yasui}@brain.kyutech.ac.jp

***Abstract*— As one of the research groups in the 21st Century COE Program on brain information technology, we present various modeling approaches towards its establishment.**

I. Introduction

To establish an emerging and multidisciplinary field of brain information technology, close collaboration across disciplines is of necessity. To this end we have developed various modeling approaches based, directly or indirectly, on mathematical, physiological, psychological and behavioral investigation into information processing of animals and humans.

Section 2 is a dynamical approach to solving satisfiability problems and constraint satisfaction problems. Section 3 proposes a principle of Convergence Suppression and Divergence Facilitation based on findings of physiology and its applications to ICA and analogical inference. Section 4 presents an application of SOR network to image enhancement, which is an extension of conventional self-organizing maps. Section 5 proposes a novel approach to localization of a mobile robot, i.e., location and orientation, in real time. Section 6 presents a method for improving the performance of reinforcement learning by directly using sensory information.

II. Solving SAT and CSP by Recurrent Neural Networks

A. Introduction

The SAT (SATisfiability problem of the propositional calculus) and the CSP (Constraint Satisfaction Problem) are problems to find a solution, i.e. a value assignment to all variables, which satisfies all given constrains. They can represent many problems in computer science and practical applications. However they are also well known NP-complete problems, and become intractable when problem size becomes large.

To solve these problems efficiently, local search methods are well used, however, they have a serious drawback of being trapped by local minima. We proposed a neural network called LPPH (Lagrange Programming neural network with Polarized High-order connection) for the SAT and the CSP. The LPPH finds the solution in the continuous state space. The dynamics of the LPPH has the following properties[1]: 1) in the state space it does not stop at any point which is not a solution, and 2) if it comes near to a solution, it converges to the solution. These properties do not exclude the case in which the dynamics moves around forever in the state space. However, experimental results show that the LPPH can find the solution more effectively than already proposed methods, without using any stochastic move or multiple restart. In this paper we explain the LPPH and its major results.

The advantages of the LPPH are: 1) it can find solutions efficiently, 2) it can flexibly and easily treat the additional requirements about solutions, and 3) it is suitable for massively parallel implementation in VLSI. In our COE program we study applications of the LPPH to decision making and motion planning of autonomous robots under several constraints.

B. Formulation of LPPH

The CSP is a combinatorial problem to find a solution which satisfies all given constraints. The CSP is defined by a triple (X, D, C), where $X = \{X_1, X_2, \cdots, X_n\}$ is a finite set of variables, $D = \{D_1, D_2, \cdots, D_n\}$ is a finite set of domains, and $C = C_1, C_2, \cdots, C_m$ is a finite set of constraints. If each D_i is restricted to be {true, false}, and each C_i is restricted to be a disjunction of literals (a literal is a variable or its negation), the CSP is reduced to be a SAT.

Let x_{ij} be a Boolean variable which represents variable X_i being assigned the jth value in D_i. x_{ij} is called a VVP (Variable-Value Pair). If x_{ij} is true ($x_{ij} = 1$), variable X_i is assigned the jth value in D_i. If x_{ij} is false ($x_{ij} = 0$), variable X_i is not assigned the jth value in D_i. Constraint C_r consists of a set of VVPs.

Let VVP x_{ij} represent the degree of certainty that the variable X_i is assigned the jth value of D_i. x_{ij} has the continuous value between 0 and 1. The dynamics of the LPPH is defined as follows:

$$\begin{aligned} \frac{dx_{ij}}{dt} &= x_{ij}(1 - x_{ij}) \sum_{r=1}^{m} w_r s_{rij}(\boldsymbol{x}) \\ \frac{dw_r}{dt} &= -\alpha w_r + h_r(\boldsymbol{x}), \end{aligned}$$

where $s_{rij}(\boldsymbol{x})$ represents a force put on x_{ij} for satisfying constraint C_r. w_r is the weight of constraint C_r. $h_r(\boldsymbol{x})$ represents the degree of un-satisfaction of constraint C_r. α is called the attenuation coefficient. In the LPPH, each variable changes its value so as to satisfy all constraints, and weight w_r increases, if constraint C_r is not satisfied. The LPPH searches a solution of the CSP by numerically solving the above dynamics.

C. Major results of LPPH

1) For the SAT we proposed a dynamics which can solve the problem efficiently.

2) It is known that attenuation coefficient strongly affects the efficiency of solving the problem, and it is also known that the best value of α strongly depends the problem, and it is hard to decide the best value in advance. To resolve this difficulty we proposed an extension of the LPPH[2] which has two kinds of weights, one is called short term memory and the other is called long term memory.
3) We extended the dynamics of the LPPH[3] to solve several variations of the SAT, such as, the SAT with an objective function, the SAT with a preliminary solution, and the MAX-SAT.
4) We proposed an efficient dynamics[4] for the CSP which include not only binary constraints but also the following constraints: ALT(n,S) (where S is a finite set of VVPs. The ALT constraint requires that at least n VVPs in S be true), ALF(n, S) (at least n VVPs in S be false), AMT(n, S) (at most n VVPs in S be true), and AMF(n, S) (at most n VVPs in S be false).
5) A method of parallel execution of the LPPH[5] was proposed. Experimental results show that high speedup radio is obtained. Furthermore this method is easy to implement by hardware.

III. Pruning, Selective Binding and Internal Models with Applications to ICA and Analogical Reasoning

The synaptic density of the kitten visual cortex increases rapidly following the birth, and then decreases as the visual experience proceeds. This decrease does not, however, occur in the visually deprived kittens. There are a number of similar findings in the developing brain. These observations indicate that a pruning mechanism is at work during early stages of visual learning. And such pruning is thought to manifest one of the fundamental strategies underlying the brain mechanisms.

Pruning is an important strategy in artificial neural networks as well. There, one wishes to find the minimum (necessary and sufficient) size/complexity for the neural network, for reasons relevant to (1) economy, (2) understanding / insight, (3) generalization ability. Generally, however, the minimum structure for the given task is not a priori known. A solution is pruning whereby a sufficiently large and complex structure is prepared initially and unnecessary connections are eliminated during the training. As such, pruning can be viewed as a connectionist's reflection of the Principle of Redundancy Reduction. Pruning algorithms such as Weight Decay and Optimum Brain Damage are widely known.

These algorithms have been applied mostly for single-output neural nets, in which case elimination of a hidden-output connection would imply removal of a hidden unit. This is not necessarily the case if the network has two or more output units. When the neural network is a multi-output one, one would have to specify what kind of pruning is desired. This consideration led to the pruning algorithm called Convergence Suppression and Divergence Facilitation (CSDF) whose objective is not only to minimize the number of active hidden units but also to make each surviving hidden unit be utilized jointly by as many as output units possible [6]. In addition

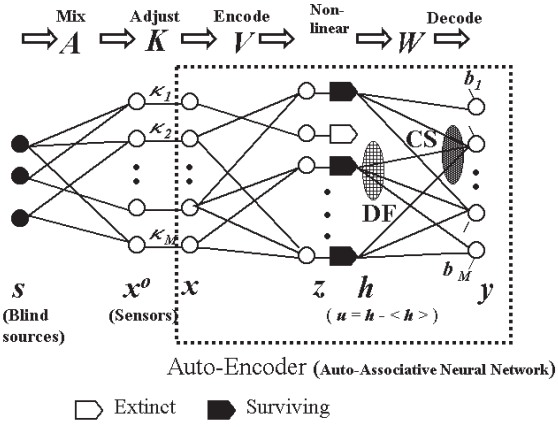

Fig. 1. Architecture for ICA

to the keywords (1)-(3) pointed out as merits of pruning, the CSDF algorithm is thought to give some more, i.e., (4) optimum modularization, (5) selective binding/association, and (6) induction of internal models, and (7) abstraction ability.

CSDF is used in a new ICA method based on an auto-encoder performing sensor-signal identity mapping[7][8][9] (Fig.1). An internal model of the external signal-mixing situation emerges due to the CSDF pruning, and the hidden units that survive the CSDF pruning reconstruct the source signals. This ICA method which requires no pre-processing such as whitening is characterized by its high adaptability and robustness, as is demonstrated by trouble cases such as sudden increase of the source signals, sudden failure of sensors and so on.

As another example, CSDF is applied in a neural network for analogical learning/inference [9][10] (Fig.2). Internal abstraction models together with abstraction/de-abstraction bindings are generated as a result of the CSDF pruning coupled with the backpropagation training. The internal abstraction model acts as an attractor for new relevant dataset, a process corresponding to analogical memory retrieval.

IV. Intuitive Image Enhancement using SOR

A. Introduction

SOR network is an extension of a SOM, and can extract desired I/O relationship of a target system using I/O vector pairs and evaluations instead of teacher signals[11]. In the SOR network, even intuition of an user can be adopted as evaluations. The effectiveness of the SOR network is verified by applying it to image enhancement based on user intuition.

B. Intuitive Image Enhancement Using SOR Network

1) SOR Network: The SOR network consists of an input layer, an output layer and a competitive layer which include n, m and N units, respectively as shown in Fig.3. The unit j in the competitive layer is connected to the units in the input and output layers with reference vector. The network can be established by learning in order to approximate the desired I/O relationship of a target system.

The SOR network has two modes, learning mode and execution mode. In learning mode, the random I/O vector pairs are applied, as the learning vectors, to the input and

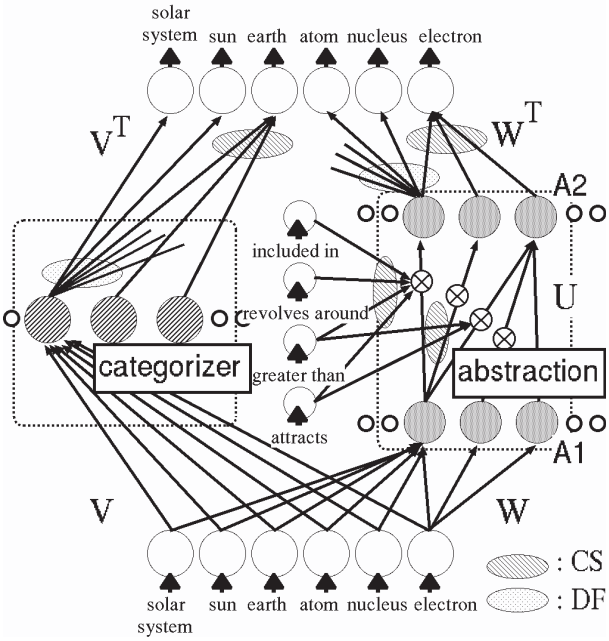

Fig. 2. Architecture for analogical learning

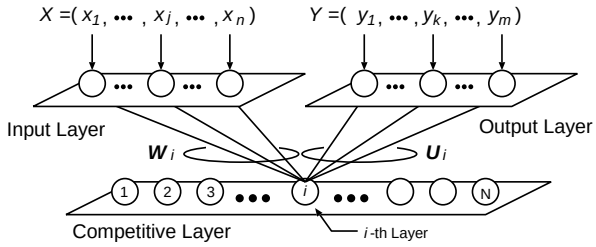

Fig. 3. Structure of the SOR network.

the output layers together with the evaluations corresponding to the I/O vector pairs. The evaluation may be assigned by the network designer, given by the intuition of the user or obtained by examining the system under test. The positive evaluation causes the self-organization of attraction to the learning vector and the negative one does that of repulsion from the learning vector. The reference vectors are arranged in area where desired I/O vector pairs exist by learning. After the learning, the output of the network represents the weighted average of the weight reference vector corresponding to the similarity between the reference vector and the input vector.

2) Image Enhancement and Difficulties: In order to enhance a contrast of an image, intensities of all pixels are transformed using a mapping curve[12]. This process is called image enhancement. Generally the mapping is decided from the histogram of the original image, thus the relationship between the histogram and mapping curve affects the performance of enhancement. There are many researches for image enhancement in which an enhanced image having strong contrast can be obtained. However, images obtained by these methods are sometimes unnatural and have too strong contrast, thus the images are not suitable for representing the user intuition.

3) Intuitive Image Enhancement Using SOR Network: In this paper, image enhancement method, in which the desired intensity mapping curve for the histogram of an original image is generated by the SOR network, is described. The histogram of an original image is represented as the 256-dimensional vector $\boldsymbol{x} = (x_0, \cdots, x_i, \cdots, x_{255})$, where x_i is the number of pixels whose intensity is i. The intensity mapping curve is represented as the 256-dimensional vector $\boldsymbol{y} = (y_0, \cdots, y_k, \cdots, y_{255})$, where y_k is the output intensity for the input intensity k. $\boldsymbol{x}$ and $\boldsymbol{y}$ are employed as the input vector and the output vector of the SOR network, respectively.

The evaluation of the relationship between $\boldsymbol{x}$ and $\boldsymbol{y}$ is given by the user watching the image obtained by the intensity mapping curve $\boldsymbol{y}$. The learning of SOR network is achieved using these learning vectors and their evaluations. After the learning, the SOR network exhibits the desired relationship between intensity histogram and intensity mapping curve based on the user intuition. The intensity histogram of the image which should be enhanced is applied to the SOR network, and the desired intensity mapping curve is generated by execution mode of SOR network.

4) Experimental Results: For obtaining the learning vectors 20 images are prepared. Each image is enhanced by randomly generated 20 mapping curves. The user evaluates the 400 enhanced images with his/her intuition. Learning of the SOR network is achieved using these learning vectors and their evaluations. The 5 testing images are enhanced by the linear transformation, the histogram equalization and the proposed method. Fig.4 shows the original and enhanced images of testing image. The user who evaluated the learning images preferred the images enhanced by the proposed method to other images.

C. Conclusions

The SOR network can construct desired I/O relationship based on intuitive or objective evaluations. The effectiveness of the intuitive image enhance method using SOR network is verified.

V. Vision Based Real Time Robot Localization

A. Introduction

Localization using vision is still an open problem. Cameras have a virtually unlimited range, can cover large field-of-views at high update rates, and are available at low cost and with low power consumption. A difficult part of using camera images for robot localization is, however, map representation. Constructing and using a full 3D map in real-time involves immense processing and storage resources.

Instead, localization based on local appearances has been investigated. This involves storing a set of images at different locations, along with information about where the images were taken. These memorized images can be compared against the current camera images, and the exact location can thereby be estimated.

Often, the robot will not be precisely on any of the memorized locations so the image-matching mechanism thereby needs to be robust to small movements and rotations. In summary, an appearance-based approach requires that *a*) a large set of images can be stored in the robot memory, *b*) that there is a fast and robust mechanism for matching against these images.

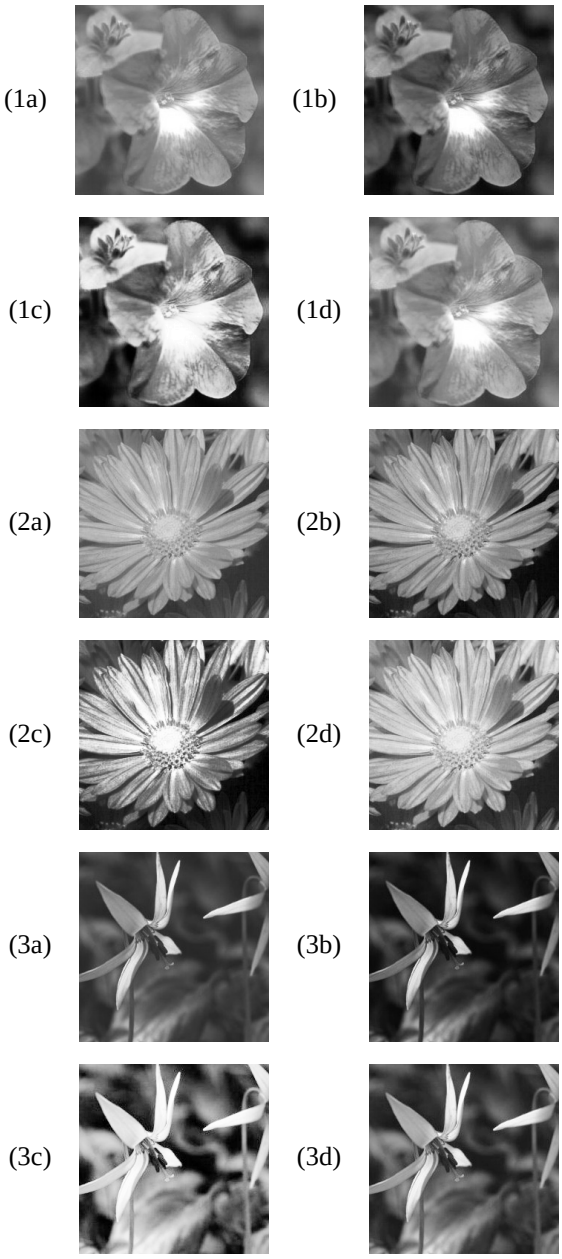

Fig. 4. Testing images and their enhanced images. (a) the original image. (b) image enhanced by linear transformation. (c) image enhanced by histogram equalization. (d) image enhanced by the proposed method.

B. Proposed approach

We have created an image processing algorithm which extracts low-dimensional representations of images in real-time. Unlike PCA, which is the predominantly used reduction technique, our technique is not based on assumptions about the input distribution, and therefore more directly suited for life-long learning. Eigenspace-based techniques namely need training sets correctly depicting all future inputs, or an adaptive on-line updating scheme, putting real-time operation in question as all stored image data need to be updated if the eigenvectors are modified.

Our technique provides compact representations for the images, as well as a solution to the alignment problem. Shifts have no effect on our representations as they are built as group averages over the translation group. A global feature-vector is extracted where each element g corresponds to a shift in the image $\boldsymbol{X}$ (indices into which are modulo N_0 and N_1):

$$A[f](\boldsymbol{X}) = \frac{1}{N_0 N_1} \int_{t_0=0}^{N_0} \int_{t_1=0}^{N_1} f(g(t_0,t_1)\boldsymbol{X}) dt_1 dt_0. \quad (1)$$

where $(g\boldsymbol{X})(\boldsymbol{n}) = \boldsymbol{X}(\boldsymbol{n}')$, with $\boldsymbol{n}' = \boldsymbol{n} + \boldsymbol{t}$.

If an omnidirectional camera is used, a rotation invariant representation of each location can be constructed. Instead of introducing rotational invariance at the local pattern level using the 2D Euclidean motion group, thereby discarding all distinction between horizontal and vertical features, it is in our system rather introduced on the global level. This is accomplished by switching from a Cartesian $\boldsymbol{n} = (n_0, n_1)$ to a polar $(\tilde{n}_0, \tilde{n}_1)$ image coordinate system, adapted to the omniview sensor:

$$\tilde{n}_0 = \gamma + \phi(n_1/\beta) cos(2\pi n_0/\alpha) \quad (2)$$

$$\tilde{n}_1 = \delta + \phi(n_1/\beta) sin(2\pi n_0/\alpha) \quad (3)$$

where ϕ describes the (cata-)dioptric profile, (γ, δ) is the location of the actual center in the omniview image, and α and β are horizontal and vertical fidelities, respectively. Due to the low dimensionality and rotation invariance of the resulting representations, the system can process 5,000 localization hypotheses l simultaneously in real-time, on a linked standard PC. The system was implemented on a real robotic platform, with Sequential Monte Carlo localization:

$$Bel(l_t) = \alpha P(s_t|l_t) \int P(l_t|l_{t-1}, a_{t-1}) Bel(l_{t-1}) dl_{t-1}, \quad (4)$$

where s are sensor readings and a odometry readings.

As a small-size robot and robot environment was used, the actual location could be tracked with an overhead camera, allowing an exact performance evaluation. In the 90x50 centimeter arenas, memorized by 112 uniformly collected images, the circular 5-centimeter robot has an average localization error of less than 4 centimeters. Details of this polar higher-order local autocorrelation (PHLAC) and applications can be found in [13].

C. Conclusions

The PHLAC representations allow a large set of locations to be stored as an appearance-based map. The robot can match incoming camera images against these stored representations in real-time, and maintain multiple simultaneous localization hypotheses to resolve perceptual aliasing situations. If repositioned randomly, the robot will quickly recover the correct location and orientation by automatically integrating odometry and sensory information across time in the Bayesian framework.

VI. Effective Reinforcement Learning Using Sensory Information

A. Introduction

It is well known that reinforcement learning necessitates a large number of episodes for learning. To decrease computational cost various methods have been proposed.

As far as we know, sensory information is used only for measuring the environment or for association of actions and sensory information [14][15]. To improve the performance of reinforcement learning, more direct use of sensory information in reinforcement learning is required.

This paper proposes to introduce sensory signals into reinforcement learning in a direct way to increase its learning speed and the probability of reaching a goal, and to decrease the probability of collision[16].

B. Q-Learning

Q-learning estimates a value as a function of a state and an action. We adopt Q-learning here, because in a mobile robot a value to be estimated heavily depends not only on a state, i.e., location and orientation, but also on an action taken by a mobile robot. A value function, $Q(s,a)$, in Q-learning is updated as,

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left[r + \gamma \max_{a'} Q(s',a') - Q(s,a)\right] \quad (5)$$

where $Q(s,a)$ is a value function of a state-action pair, s is a state, a is an action, s' is the next state, α is a step size, r is a reward, and γ is a discount rate.

C. Improvement of reinforcement learning

A naive idea of introducing sensory information into reinforcement learning is to directly use sensory information as reward. For example, if an obstacle is observed in a particular direction by an ultrasonic sensor, negative reward is given in Q-learning. However, as can be seen in the update equation in Eq.(refeq:Qval), it reduces a value function only at the current location of a mobile robot. When the observed obstacle is located far away, however, the reduction of a value function at the current location is inappropriate.

Instead of reducing a value function only at the current location of a mobile robot, the reduction of value functions in the neighborhood of an obstacle seems reasonable. This is exactly an idea which we propose in this paper. Consider a line segment connecting the current location of a mobile robot and the obstacle detected by a sensor. We propose to assign the maximum reduction of a value function at the location of the obstacle and no reduction at the current location of a mobile robot. These two extrema are linearly interpolated on the line segment. The reduction of value functions in a region instead of that only at the current location of a mobile robot in conventional reinforcement learning realizes faster learning of a value function.

However, this reduction causes the following difficulties in a mobile robot. Firstly, a mobile robot tends to make a big detour, because a region near an obstacle has small values, hence is hard to approach. This big detour makes the length of a path of each episode larger, hence should be avoided. Secondly, the reduction sometimes prohibits a mobile robot to pass through a narrow corridor due to the decrease of a value function in the corridor. This should also be avoided.

To overcome these difficulties, we, therefore, restrict the range of reduction of a value function; only the region where the distance to an obstacle is less than a given threshold is subject to reduction of a value function. This threshold is expected to further improve the performance of reinforcement learning. The selection of the threshold is essential, but difficult to determine. We propose to determine it experimentally by selecting the threshold with the best performance.

D. Computer Experiments

We carry out computer experiments to evaluate the effectiveness of the proposed method in terms of the learning speed, the number of goals reached, the number of collisions, and the length of a path in each episode.

We use 2 kinds of environment. Due to space limitation, however, we show the results for only the complex environment C in Figure 5. The area of the environment is $4m \times 4m$ and is composed of 20×20 grids. The mobile robot has 3 actions, i.e., forward moving, turning right by 10 degrees, and turning left by 10 degrees. The distance to an obstacle is measured by ultrasonic sensors. The state of the mobile robot is defined by its location (one of 20×20 grids) and orientation (one of 8 sectors). Table I presents parameters used in reinforcement learning.

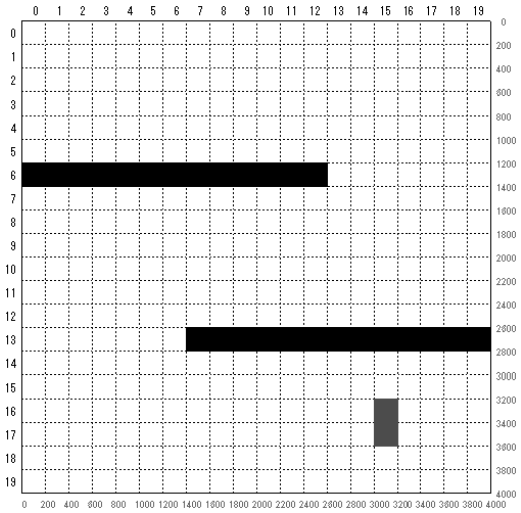

Fig. 5. The complex environment C. Black rectangles are the obstacles and the grey rectangle is the goal.

TABLE I
PARAMETERS IN REINFORCEMENT LEARNING

reward for action	-7.50	α	0.40
reward for goal	20.00	γ	0.99
reward for collision	-250.00	ϵ	0.01

Figures 6 and 7 illustrate the number of goals reached and the number of collisions in environment C. Figure 6 illustrates that a large value of the threshold improves the speed of reinforcement learning until 1000 episodes. It also shows that the number of goals reached for the threshold of $800mm$ levels off after 1500 episodes. This is because the mobile robot has difficulty in passing through a narrow corridor for the threshold of $800mm$ after sufficient learning.

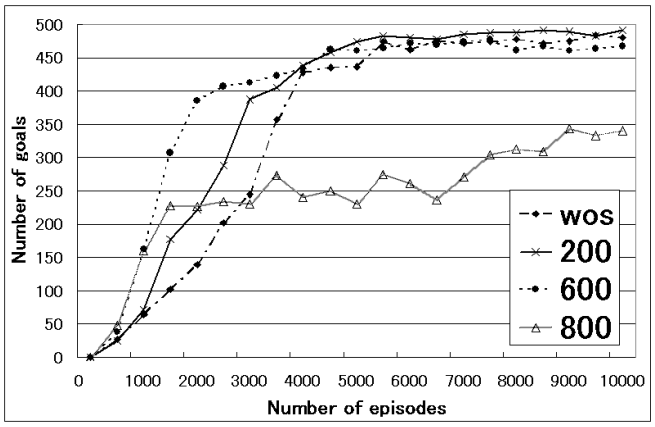

Fig. 6. The number of goals reached over episodes in the complex environment C. **wos** stands for the case without sensors. **200**, **600** and **800** stand for the cases with sensors with thresholds of $200mm$, $600mm$ and $800mm$, respectively.

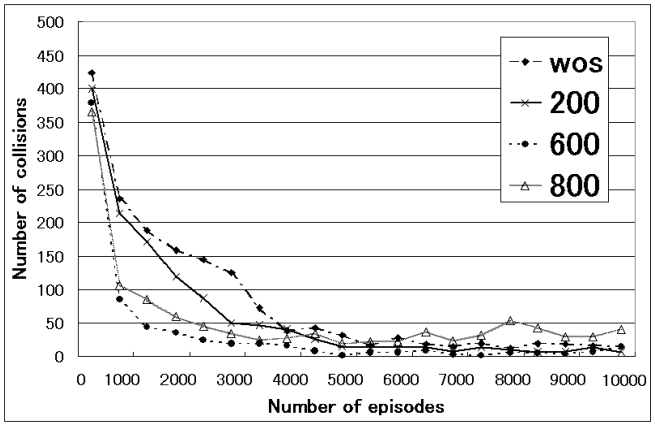

Fig. 7. The number of collisions over episodes in the complex environment C. **wos** stands for the case without sensors. **200**, **600** and **800** stand for the cases with sensors with thresholds of $200mm$, $600mm$ and $800mm$, respectively.

E. Conclusions

Direct use of sensory signals into reinforcement learning improves the performance of reinforcement learning. Computer experiments demonstrate that the number of goals reached increases more than 2 times faster than that by conventional Q-learning. The larger the value of the threshold is, the faster the learning speed is. However, when the value of the threshold is too large, the number of goals reached levels off.

VII. Conclusions and Discussions

We have presented various modeling approaches to brain-like information technology. We aim at finding exquisite information processing mechanisms of the brain through developing their models. We are hoping to develop brain-like LSI devices based on resulting models. We also plan to apply the models and LSI devices to groups of mobile robots under collaboration.

Acknowledgments

This research was supported by the 21st Century COE Program by the Ministry of Education, Culture, Sports, Science and Technology (MEXT), Japan.

References

[1] M. Nagamatu and T. Yanaru, On the Stability of Lagrange Programming Neural Networks for Satisfiability Problems of Propositional Calculus, Neurocomputing, pp.119-133, 1995.

[2] M. Nagamatu and T. Yanaru, Solving SAT by Lagrange Programming Neural Network with Long and Short Term Memories, Information Modelling and Knowledge Bases XI, IOS Press, pp.289-301, 2002.

[3] M. Nagamatu and T. Nakano and N. Hamada and T. Kido and T. Akahosi, Extensions of Lagrange Programming Neural Network for Satisfiability Problem and its Several Variations, Proceedings 9th International Conference on Neural Information Processing, pp.1781-1785, 2002.

[4] T. Nakano and M. Nagamatu, Solving CSP via Neural Network Which Can Update All Neurons Simultaneously, Proceedings of SCIS/ISIS2004, 2004.

[5] K. Zhang and M. Nagamatu, Parallel Execution of Neural Networks for Solving SAT, Proceedings of SCIS/ISIS2004, 2004.

[6] S. Convergence suppression and divergence facilitation: Minimum and joint use of hidden units by multiple outputs. Neural Networks, 10:2, pp.353-367, 1997.

[7] Yasui,S. A conventional auto-associative neural network separates blind sources without adding intentional algorithms other than pruning. In Neural Networks for Signal Processing XI, Eds. D.J. Miller et al.,(IEEE Press), pp.313-322, 2001.

[8] Yasui,S., Blind source separation by auto-encoder neural network with structure pruning. Proc. of Int. Invitational Workshop on Intelligent Interface Devices, pp.55-58, 2002.

[9] Yasui, S., Pruning, selective binding and emergence of internal models: Applications to ICA and analogical reasoning, In: Soft Computing for Information Processing and Analysis Enhancing the Power of the Information Technology, Eds. M. Nikravesh et al., (in press), 2004.

[10] Yasui, S., Abstraction based connectionist analogy processor. Int'l J. of Applied Mathematics and Computer Science, 10:4, pp.791-812, 2000.

[11] T. Yamakawa and K. Horio, "Self-Organizing Relationship (SOR) Network," IEICE Trans. on Fundamentals, Vol.E83-A, pp.1674-1678, 1999.

[12] A. Resenfeld and A.C. Kak, *Digital picture processing*, Academic Press, Inc., 1976.

[13] F. Linåker and M. Ishikawa, Rotation invariant features from omnidirectional camera images using a polar higher-order local autocorrelation feature extractor. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2004)*, 2004.

[14] S. Zilberstein, R. Washington, D. S. Bernstein, A. I. Mouaddib, Decision-Theoretic Control of Planetary Rovers, Plan-Based control of Robotic Agents, LNAI, No.2466, pp.270–289, 2002.

[15] E. Zalama, J. Gomez, M. Paul, J. Peran, Adaptive Behavior Navigation of a Mobile Robot, IEEE Trans. SMC, Part A: Systems and Humans, vol.32, pp160–169, 2002.

[16] Keiji Kamei, Masumi Ishikawa, More Effective Reinforcement Learning by Introducing Sensory Information, IJCNN, pp.3185-3188, Budapest, Hungary, July 25-29, 2004.