

## Research Article

# Parameterless-Growing-SOM and Its Application to a Voice Instruction Learning System

**Takashi Kuremoto, Takahito Komoto, Kunikazu Kobayashi, and Masanao Obayashi**

*Graduate School of Science and Engineering, Yamaguchi University, Tokiwadai 2-16-1, Ube, Yamaguchi 755-8611, Japan*

Correspondence should be addressed to Takashi Kuremoto, wu@yamaguchi-u.ac.jp

Received 5 January 2010; Revised 22 April 2010; Accepted 21 June 2010

Academic Editor: Ivo Bukovsky

Copyright © 2010 Takashi Kuremoto et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

An improved self-organizing map (SOM), parameterless-growing-SOM (PL-G-SOM), is proposed in this paper. To overcome problems existed in traditional SOM (Kohonen, 1982), kinds of structure-growing-SOMs or parameter-adjusting-SOMs have been invented and usually separately. Here, we combine the idea of growing SOMs (Bauer and Villmann, 1997; Dittenbach et al. 2000) and a parameterless SOM (Berglund and Sitte, 2006) together to be a novel SOM named PL-G-SOM to realize additional learning, optimal neighborhood preservation, and automatic tuning of parameters. The improved SOM is applied to construct a voice instruction learning system for partner robots adopting a simple reinforcement learning algorithm. User's instructions of voices are classified by the PL-G-SOM at first, then robots choose an expected action according to a stochastic policy. The policy is adjusted by the reward/punishment given by the user of the robot. A feeling map is also designed to express learning degrees of voice instructions. Learning and additional learning experiments used instructions in multiple languages including Japanese, English, Chinese, and Malaysian confirmed the effectiveness of our proposed system.

## 1. Introduction

Kohonen's self-organizing map (SOM) is a kind of a neural network which maps a high-dimensional input onto a regular low-dimensional grid orderly by unsupervised learning schemes [1–4]. Because of its simple algorithm and powerful performance, SOM has been developed and applied widely to the fields of pattern recognition, signal processing, intelligent control, and so on [5–15]. In a website of SOM library [6], more than 7,000 papers concern with this technique are collected.

Generally, SOM algorithm maps an  $n$ -dimension feature data in an input space  $\mathbf{x}(x_1, x_2, \dots, x_n)$  to a unit  $i$  in a low-dimensional output space with connections  $\mathbf{m}_i(m_1, m_2, \dots, m_n)$  by a simple rule using Euclidean distance, winner-takes-all,

$$c = \arg \min_i (\|\mathbf{x} - \mathbf{m}_i\|), \quad (1)$$

that is, a high-dimensional input is corresponded to a most suitable unit  $i$  with position  $c$ , best-match-unit (BMU) on

the output map. For all inputs and initial connections with random values, a competitive learning rule enhances that the input data with similar features keep closely on the visualized topological output map

$$\Delta \mathbf{m}_i = \alpha h_{ci} (\mathbf{x} - \mathbf{m}_i), \quad (2)$$

where  $\alpha$  is learning rate and  $h_{ci}$  is a neighborhood function

$$\mathbf{h}_{ci} = \exp\left(-\frac{\|c_i - c\|^2}{2\sigma^2}\right). \quad (3)$$

Here,  $c_i$ ,  $c$  denote the positions of an arbitrary unit on the output map and BMU, respectively,  $i = 1, 2, \dots, k \leq N \times M$ ,  $\sigma$  is a constant. Obviously,  $\mathbf{h}_{ci}(\mathbf{x}) \geq 0$ ,  $\mathbf{h}_{ci}(0) = 1$ , and  $\mathbf{h}_{ci}(\infty) = 0$ .

In fact, the size of the output space  $N \times M$  in the original SOM is fixed in advance, and parameters such as learning rate  $\alpha$  and the scale of neighborhood  $\sigma$  are often determined empirically. These constraints result in 2 kinds of problems

in technical applications [6–14]:

- (1) the fixed size of output map prevents additional learning when new feature data are presented and BMUs are difficult to be found on the trained output map;
- (2) annealing schemes for tuning the learning rate and the neighborhood size are necessary to improve the operation rate of output map; however, it usually increases computational load to realize the annealing.

Variations of SOM with growing structures are proposed to solve the first problem [7–10]. The basic idea of these kinds of SOM is to set the output feature map with a small size initially, for example, 2 units, then insert rows/columns into the map in training, where/when a most visited BMU exists [7, 10] or the deviation of the distance between the units on input layer and output map [8, 9]. We proposed another kind of method to solve the lack of units by using a memory layer to store matured units on the feature map during training process and release the matured units to be initialized, that is, the units come to available to be reused [12, 13]. When a feature data set is input to the learning system, the process searches corresponding BMU on memory layer at first, feature map which is produced by SOM just become to an intermediate map, so we called it transient SOM (T-SOM).

To solve the second problem, there have been also various approaches such as reducing learning rate ( $\alpha$  in (2)) and neighborhood size ( $\sigma$  in (3)) linearly, that is, multiplying attenuation coefficients, calculating the neighborhood size in the input space, or using Kalman filters to find BMU on the output space [6]. Berglund and Sitte proposed a low-cost parameterless SOM algorithm (PLSOM) recently which uses the fitting error between the input and the map only to decide the annealing schemes [11].

In this paper, we combine the idea of growing SOM algorithm and the method of PLSOM to construct a novel SOM names parameterless-growing-SOM (PL-G-SOM) to tackle both problems of SOM described above. This new PL-G-SOM increases its structure adapting to the input data, and anneals parameters to realize sensitive clustering on the output space automatically. We also adopt PL-G-SOM into a voice instruction learning system where it serves as an automatic classifier of input features as well as T-SOM has been applied to a hand image instruction learning system [12, 13] and a voice instruction learning system [14].

The rest of this paper is organized as follows. Section 2 presents the details of PL-G-SOM. Section 3 shows a voice instruction learning system using PL-G-SOM. In Section 4, instruction learning experiments with 4 languages were reported to confirm the ability of learning and additional learning of the proposed system. Section 5 is the conclusion.

## 2. A New SOM: PL-G-SOM

*2.1. Growing of Output Map.* To construct a growing SOM which is more sensitive to larger categories of input data comparing with the SOM with fixed size in advance, different

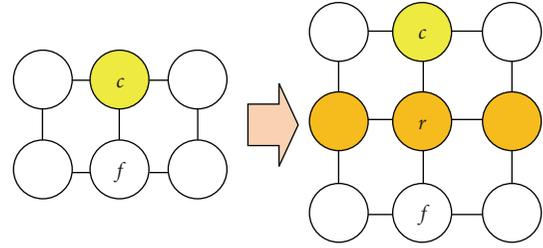


FIGURE 1: Insert a row/column into the feature map. Unit  $c$  is a BMU, and  $f$  is the farthest unit among the neighbors of  $c, r$  the inserted row/column.

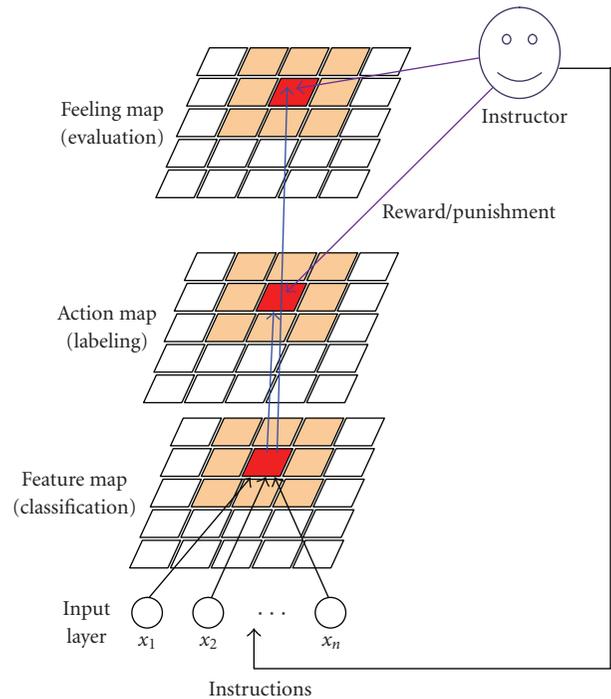


FIGURE 2: The structure of a voice instruction learning system using PL-G-SOM. It is similar to the system using T-SOM in [12–14], however, instead of memory layer of BMU in T-SOM, each map grows with training. Annealing schemes of their neighborhood size and learning rates are given by PL-G-SOM.

criteria have been proposed. Fritzke chose to insert a new row/column adjacent to a most often visited BMU in his Growing Grid [7]. The reason for this criterion of map enlargement is that the earlier map may be considered as a coarse one and likelihood BMUs need to raise their resolution to deal with the change of input. Meanwhile, Bauer and Villmann suggested adding units in the direction or even new dimension of the largest error between input data and the output map in their GSOM [8, 9]. However, the process of enlarging the output map either in Growing Grid or GSOM is similar and it is shown in Figure 1. In fact, when a new row/column needs to be inserted to the neighbor of a BMU  $c$ , for example, in the middle of  $c$  and  $f$ , the weights

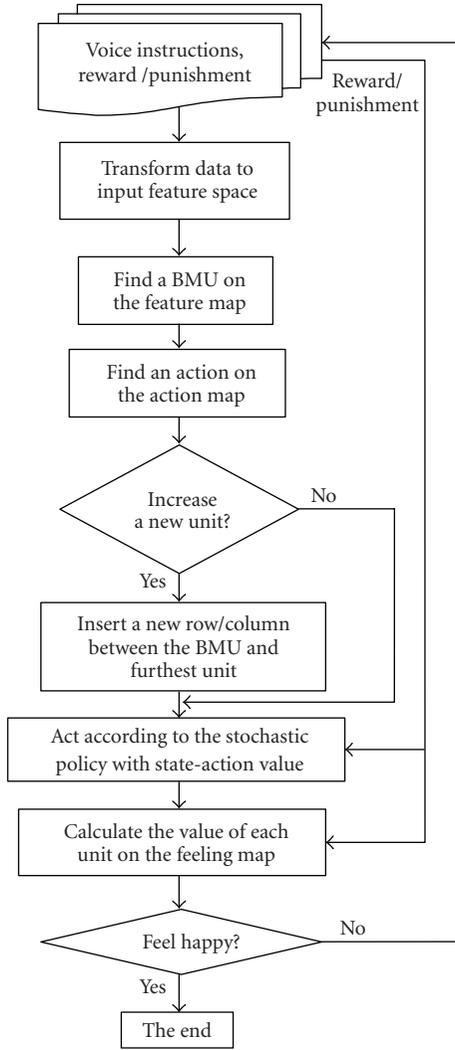


FIGURE 3: Flow chart of the proposed voice instruction learning system processing.

of connections between input and new nodes take average values of  $c$  and  $f$ :

$$\mathbf{m}_r = 0.5 (\mathbf{m}_c + \mathbf{m}_f), \quad (4)$$

and so do them of  $r$ 's neighbors

$$\mathbf{m}_{r\pm l} = 0.5 (\mathbf{m}_{c\pm l} + \mathbf{m}_{f\pm l}), \quad (5)$$

where  $l = 1, 2, \dots, N$  or  $M$ . Unit  $f$  is chosen which has a largest Euclidean distance from the BMU  $c$  among the neighbors of  $c$ , and after this process, the map size changes to  $N \times (M + 1)$ , or  $(N + 1) \times M$ .

We use the same growing process here however, a new criterion to choose the BMU is proposed by concerning with a reinforcement learning algorithm when SOM is adopted into a human-machine interaction learning system. The detail will be given in Section 3.

**2.2. Annealing of Parameters.** To decide the learning rate and the size of neighborhood function, we adopt the method of PLSOM proposed by Berglund and Sitte [11]. Either the learning rate  $\alpha = \varepsilon(t)$  or the neighborhood size  $\sigma(t)$  is calculated by the distance between input and the BMU:

$$\begin{aligned} \varepsilon(\mathbf{t}) &= \frac{\|\mathbf{x}(\mathbf{t}) - \mathbf{m}_c(\mathbf{t})\|^2}{r(\mathbf{t})}, \\ r(t) &= \max(\|\mathbf{x}(t) - \mathbf{m}_c(t)\|^2, r(t-1)), \\ r(0) &= \|\mathbf{x}(0) - \mathbf{m}_c(0)\|^2, \end{aligned} \quad (6)$$

$$\begin{aligned} \mathbf{h}_{ci}^\varepsilon(\mathbf{t}) &= \exp\left(-\frac{\|\mathbf{c}_i - \mathbf{c}\|^2}{(\sigma(\varepsilon(\mathbf{t})))^2}\right), \\ \sigma(\varepsilon(\mathbf{t})) &= \sigma_{\max} \cdot \varepsilon(\mathbf{t}), \quad \sigma(\varepsilon(\mathbf{t})) \geq \sigma_{\min}, \end{aligned}$$

where  $\sigma_{\max}$ ,  $\sigma_{\min}$  are positive parameters, for example, the value may be the size of the map and 1.0, respectively.

The competitive learning rule of the connections between input and output units that is, (2), can be changed to an online learning algorithm

$$\begin{aligned} m_i(\mathbf{t} + 1) &= m_i(\mathbf{t}) + \Delta m_i(\mathbf{t}) \\ &= m_i(\mathbf{t}) + \varepsilon(\mathbf{t}) \mathbf{h}_{ci}^\varepsilon(\mathbf{t}) (\mathbf{x}(\mathbf{t}) - m_i(\mathbf{t})). \end{aligned} \quad (7)$$

### 3. A Voice Instruction Learning System Using PL-G-SOM

A voice instruction learning system is supposed as an internal model of an autonomous robot which performs kinds of available actions when external signal in voices is presented at first and learns to output requested actions using the reward or punishment from the instructor. So the system supports the robot to keep learning and additional learning abilities. For example, a robot with the voice instruction learning system is able to “understand” human’s instructions in different languages, or a pet robot like “AIBO” [16] comes easily to used to change a new owner.

**3.1. The Structure.** To realize the human-machine interaction, an internal model of autonomous robot is constructed as shown in Figure 2. The structure is similar to a learning system using Transient-SOM (T-SOM) which is proposed in our previous work [12–14]. In [12, 13], a hand image instruction learning system which has 5 layers including Input Layer, Feature Map, Action Map, Feeling Map, and Memory Layer is composed with SOM algorithm and reinforcement learning rules. Instructions to the robot are presented by kinds of shapes of human’s hand, and robots categorize them, that is, image signals in an 80-dimensional space with SOM, and the instructions are labeled with a series of autonomous actions according to a stochastic policy. Instructor observes the action of the robot and provides reward/punishment of the action to robot, so the action policy of the robot is able to be modified to cooperate with the instructions of hand images. For online learning and additional learning, T-SOM adopted a memory layer which

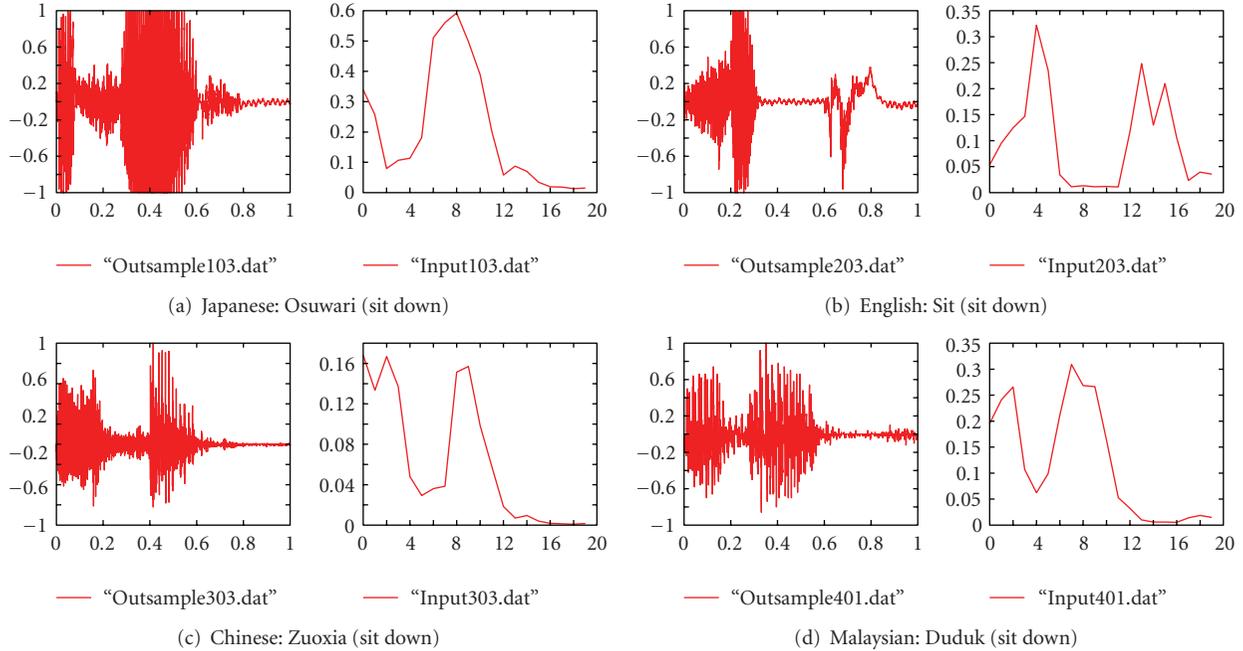


FIGURE 4: An instruction (*sit down*) features input to robot in different languages. Left: sound waves; right: normalized features in 20-dimensional space.

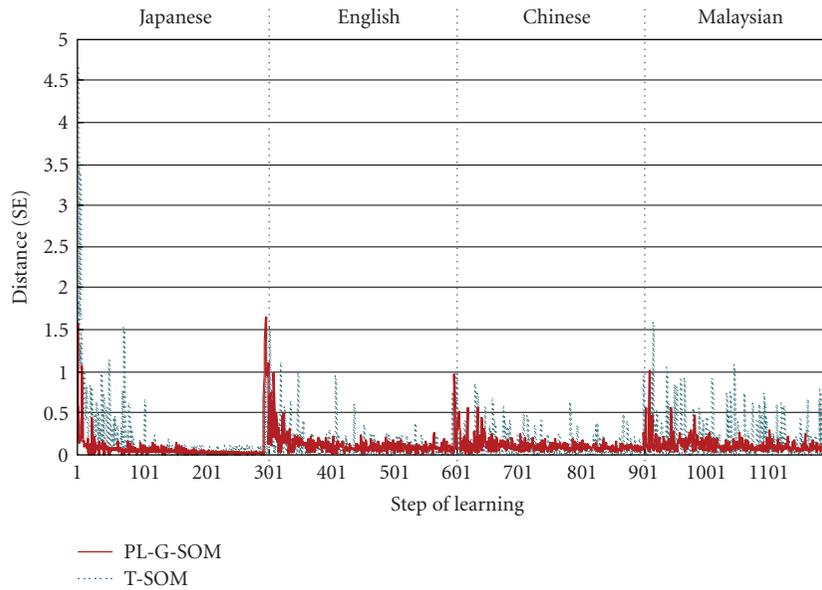


FIGURE 5: Comparison of the Euclidean distance (SE: squared error) between input and BMUs in learning (the first 300 iterations)/additional learning process between T-SOM [12–14] (broken lines) and PL-T-SOM (solid lines).

stores “matured” BMUs, and input features are matched with units on the memory layer before executing SOM on feature map. We also adopted annealing plan to decide the size of neighborhood and learning rate into T-SOM, and a voice instruction learning system using the improved T-SOM named PL-T-SOM was developed in [14]. However, a problem that exists in T-SOM is that its memory layer stores only the value of matured units without the topology of the feature map. Even if the memory layer could remember the

topology of the feature map trained online, the new topology would not be able to be established on it. For this reason, we propose a new voice instruction learning system using PL-G-SOM given in Section 2 instead of T-SOM.

In Figure 2, Feature Map is the basic growing SOM and the size of Action Map and Feeling Map growing with the Feature Map too. In fact, instructions given by voice data are transformed into feature vectors of input space (layer) at first, the PL-G-SOM algorithm is then executed

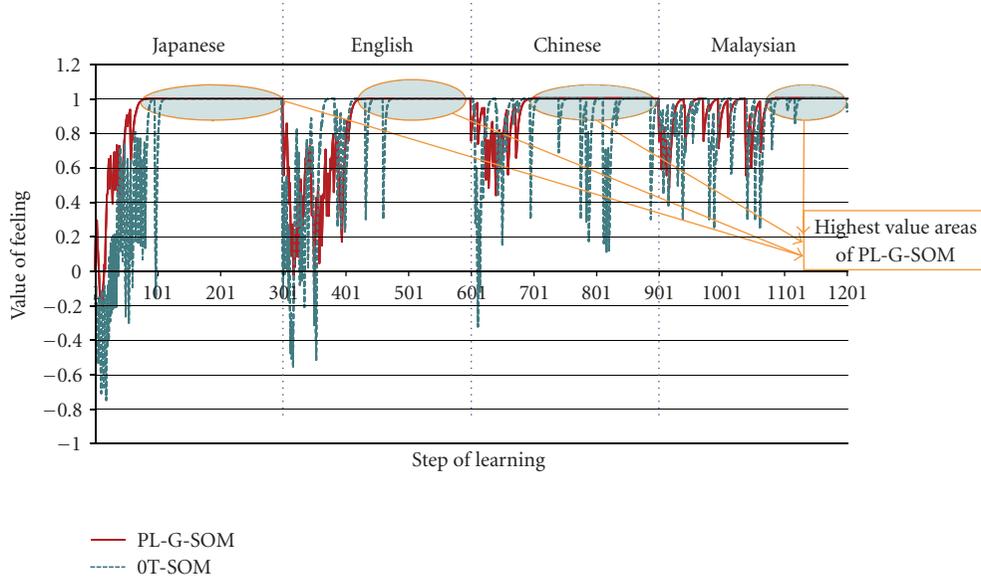


FIGURE 6: Feeling values rose to the maximum happiness 1.0 according to training iterations. PL-G-SOM proposed here (solid lines) showed faster and longer convergence than T-SOM in [12–14] (dash lines).

on the Feature Map, and the rules of growing given by (4) and (5) (Figure 1) are also applied to increase Action Map and Feeling Map. Action Map is composed by those units which correspond to the units on Feature Map; that is, each unit on Action Map represents each kind of features of input data. The units on Action Map are labeled by a reinforcement learning algorithm given by Section 3.2 to limit each feature to adaptive actions of the robot. Feeling Map has the same distribution of units as which of Action Map. The action number that comes from Action Map is furnished with a feeling value which expresses the degree of the action mastered by the robot. The details of Feeling Map are described in Section 3.3.

**3.2. Reinforcement Learning Algorithm.** The value of units on Action Map is given by a value function of state and action, that is, (8), where  $Q(s_t, a_t(i))$  has the value of selected action  $a_t(i)$  when the robot is in the state  $s_t$ , and  $Q(s_0, a_0(i)) =$  random numbers initially:

$$Q_{t+1}(s_{t+1}, a_{t+1}(i)) = Q_t(s_t, a_t(i)) \pm r, \quad (8)$$

where  $\pm r$  is the empirical value of reward (+)/punishment (−) given by the instructor, for example, a positive constant when the robot acted correctly according to its policy function and a negative constant oppositely.

Now suppose that there are  $N \times M$  units that exist on Action Map; that is,  $N \times M$  states exist in the environment of Markov decision process (MDP), each unit has  $K$  actions to be selected available, then a reinforcement learning (RL) algorithm [17] can be used to label the classes of the states which are units on Action Map yielded by the Feature Map. According to (8), a Q-value table can be established as shown in Table 1.

For each state  $s_t$  that is, presented voice instruction, robot intends to select a valuable action  $a_t(i)$  according

TABLE 1: Q-value table. Each unit of Action Map has a  $Q_t(s_t, a_t(i))$  value corresponding to an action.

$s_t$	$a_t(1)$	$a_t(2)$	$a_t(i)$	$a_t(K)$
1	6	2	−8	0
2	10	1	0	1
...				
$N \times M$	0	2	7	2

to a stochastic action policy  $\pi$  given by Gibbs distribution (Boltzmann distribution) as shown in

$$\pi_t(a_t(i) | s_t) = \frac{e^{Q_t(s_t, a_t(i))/T}}{\sum_{j \in A} e^{Q_t(s_t, a_t(j))/T}}. \quad (9)$$

Here,  $T$  is a positive parameter named temperature [17], higher  $T$  causes an active exploration of actions (each action is selected under a similar probability), and lower  $T$  gives a greedy selection of the action with higher  $Q$  value oppositely.

We propose to use  $Q(s_t, a_t(i))$  as a criterion of growing the size of Feature Map, Action Map, and Feeling Map. In fact, when the robot chooses an action with high  $Q(s_t, a_t(i))$  but instructor judges that it is wrong, then a new row/column is inserted nearby the  $s_t$ , that is, the BMU  $c$ . The growing process is described in Section 2.1.

**3.3. Feeling Map.** To express the degree of how a voice instruction is learned by robot, a Feeling Map which has the same number of units with Action Map is designed (Figure 2). The distance from input pattern to BMU of Feature Map and the reward from instructor are used to calculate feeling values which is normalized in  $[-1.0, 1.0]$  where high positive value means happiness and 0.0 is the

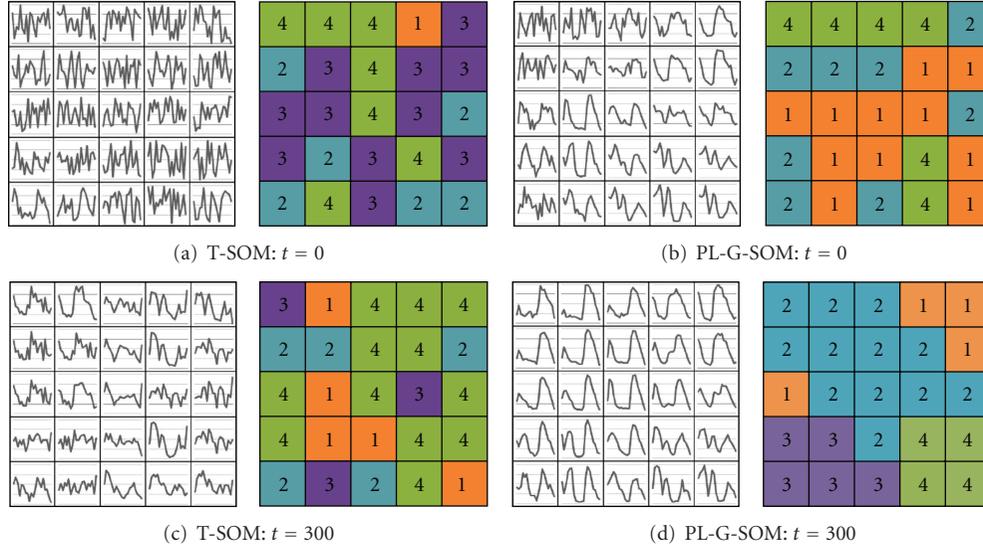


FIGURE 7: Change of each map during learning process. Left maps show Feature Maps and Right maps show Action Maps in (a)–(d). Comparing the learning result showed by (c) and (d), it is easy to find that except of the action “1”, PL-G-SOM showed to be more effective in gathering the similar input on its Action Map as neighbored output comparing with T-SOM.

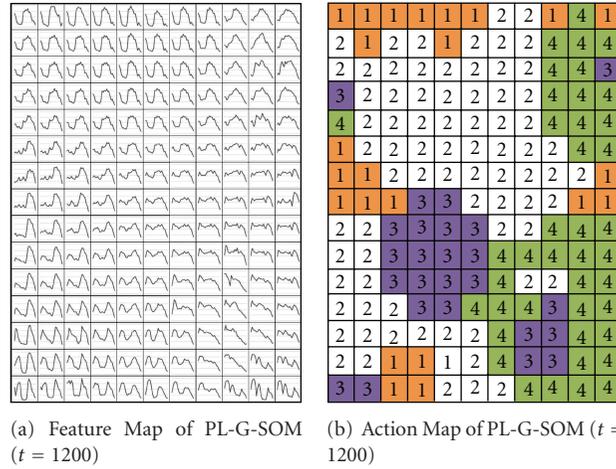


FIGURE 8: Results of feature classification and instruction learning/additional learning using PL-G-SOM. The sizes of maps grew to  $11 \times 15$  (165 units) when they began with  $5 \times 5$  (25 units) in the experiment.

initial value of each unit; negative values express sadness. The learning algorithm which was also used in [12–14] is given by

$$F_{t+1}(i) = F_t(i) \pm aC - bD_i, \quad (10)$$

where  $F(i)$  notes the feeling value of unit  $i$  on the Feeling Map (zero initially),  $C$  notes the continue times of reward or punishment,  $D_i$  is the Euclidean distance (squared error) between the unit on Feature Map corresponding to  $i$ , and the input data,  $a, b$  are constants and  $0 < a < 1$ ,  $0 < b \ll 1$ .

## 4. Experiments

**4.1. Descriptions.** Learning and additional learning experiments were performed using the system with PL-G-SOM

proposed in Section 3 and the system with T-SOM in [12–14].

Four kinds of voice instructions were used in experiments: *sit down*, *lie down*, *stand up*, and *walk*. Instructions in Japanese were used to training the system. Additional learning using voice instructions given by other languages was executed after training using the Japanese. Three kinds of languages: English, Chinese, and Malaysian were used to confirm additional learning ability of the system. The voices were recorded in a normal room by 3 males who pronounced each instruction 3 times. So, there were 3 samples of one instruction used for each kind of languages while 4 actions with 48 samples.

Sound waves were preprocessed by normalization and noise elimination, and windowed by 20 intervals to yield

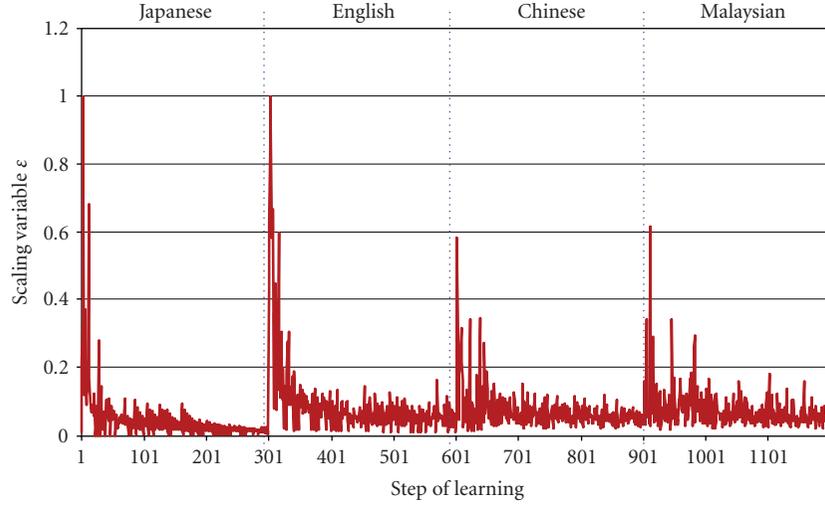


FIGURE 9: The change of scaling variable  $\varepsilon$  of PL-G-SOM in the learning/additional learning process.

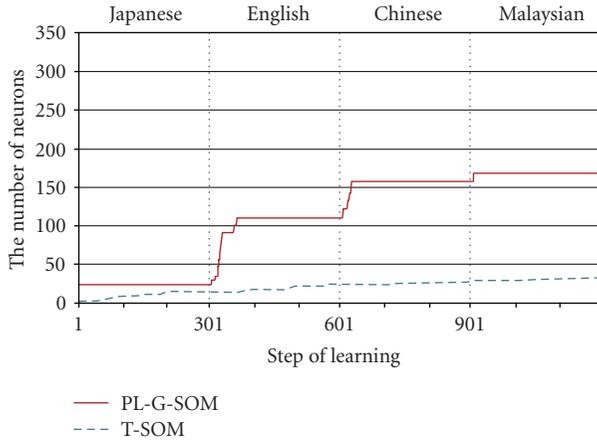


FIGURE 10: Comparison of recognition error (SE: squared error between input and BMU) in learning process between T-SOM and PL-T-SOM.

20 feature vectors of input space. Figure 4 shows an example of instruction “*sit down*” pronounced in Japanese (“*Osuwari*”), English (“*Sit*”), Chinese (“*Zuoxia*”), and Malaysian (“*Duduk*”). Parameters used in the experiments are shown in Table 2.

**4.2. Results and Analyses.** Either T-SOM or PL-G-SOM realized 100% recognition rates for 4 actions in different languages after learning and additional learning. However, PL-G-SOM showed faster and better convergence than T-SOM when the Euclidean distance (SE: squared error) between input and BMUs (Figure 5). This means that the classification to the input pattern was executed more efficiently by PL-G-SOM. Furthermore, the feeling values which express instruction recognition rate showed more obviously that correct actions of robot corresponding to instructions in voices were acquired more quickly and stably (Figure 6). Figure 7 shows the internal states of Feature Map

TABLE 2: Parameters used in the experiments.

Description	Symbol	Quantity
Size of Feature Map of T-SOM	$N \times M$	$5 \times 5$
Size of initial PL-G-SOM	$N \times M$	$5 \times 5$
Iteration times for one instruction	$t$	$300 \times 4$
Temperature	$T$	1.0
Number of instructions (actions)	$a(i)$	4
Maximum/Minimum neighborhood in PL-G-SOM	$\sigma_{\max}, \sigma_{\min}$	$N \times M / 2, 0.7$
Reward for one action selected	$r$	10.0
Parameters of Feeling Map	$a, b$	0.2, 0.05
Number of samples	—	48
Sampling rate	—	8 KHz
Sample size	—	8 bit
Channel	—	monaural

(left) and Action Map (right) changing in training. The curves in each unit on Feature Map in Figure 7 express values of  $\mathbf{m}_i$  ( $m_1, m_2, \dots, m_n$ ),  $i = 1, 2, \dots, N \times M$ . Numbers with different colors on the Action Map express the different actions which were classified (labeled) by the reinforcement learning process described in Section 3.

Figures 7(a) and 7(b) show the initial states of T-SOM and PL-G-SOM, where random numbers were used. Figures 7(c) and 7(d) are the results of learning using Japanese instructions. Comparing with T-SOM, PL-G-SOM showed more effective on the topology formation of actions; that is, numbers of actions on the Action Map clustered more clearly. After additional learning, that is, using English, Chinese, and

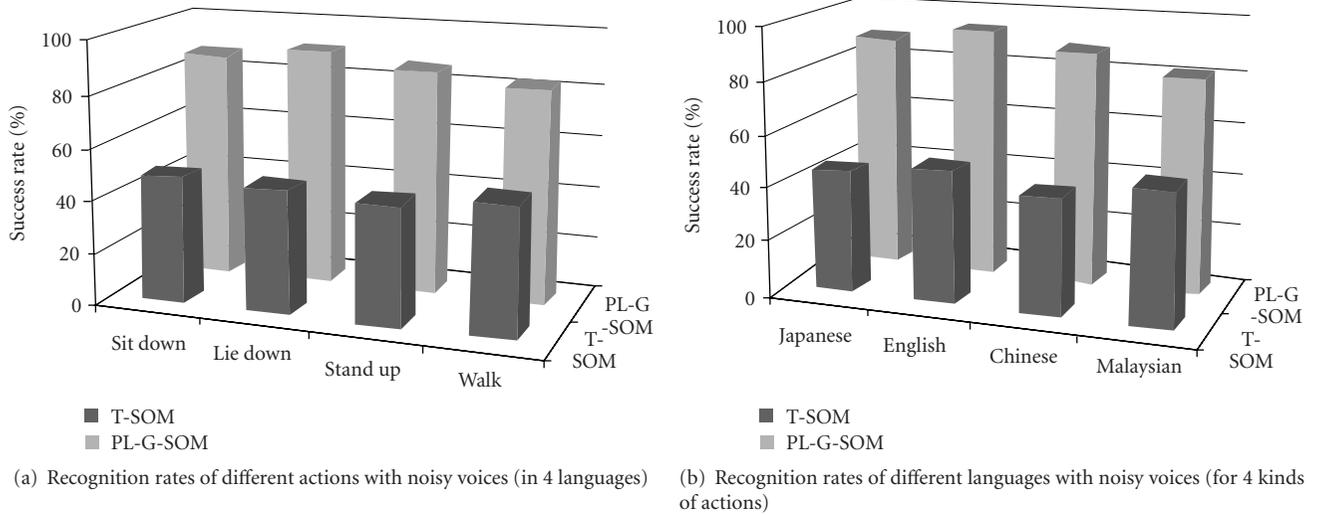


FIGURE 11: Comparison of recognition rates between T-SOM and PL-T-SOM using voice instructions with 10% noises.

TABLE 3: Recognition rates of different actions with noises.

Language	Method	Success rate (10% noises)	Success rate (20% noises)	Success rate (30% noises)
<i>Japanese</i>	T-SOM	45.9	40.9	35.0
	PL-G-SOM	88.5	57.8	46.6
<i>English</i>	T-SOM	50.8	45.3	27.0
	PL-G-SOM	93.2	57.6	39.9
<i>Chinese</i>	T-SOM	44.5	41.1	32.5
	PL-G-SOM	88.3	60.1	43.9
<i>Malaysian</i>	T-SOM	51.0	40.0	37.1
	PL-G-SOM	81.9	54.9	47.2
Average	T-SOM	48.0	41.6	33.1
	PL-G-SOM	86.7	57.6	44.4

Malaysian 300 times, respectively, the size of Feature Map and Action map of PL-G-SOM grew from 25 ( $5 \times 5$ ) units to 165 ( $11 \times 15$ ) (Figure 8).

The scaling variable  $\varepsilon$  used in PL-G-SOM ((6)-(7)) changed with the training, and by Figure 9, one can confirm that  $\varepsilon$  decreased eventually during learning at first; however, when a new kind of language was input, the scaling variable  $\varepsilon$  suddenly changed to be larger and repeated its annealing scheme. Figure 10 shows the increase of the number of units on Memory Layer of T-SOM and the increase of the number of units on PL-G-SOM. Both units grew with additional learning and the number of units on Memory Layer of T-SOM stopped at 33, meanwhile 140 units were inserted into PL-G-SOM each layer. To confirm the robustness of the two learning system, we also tested noisy samples.

Table 3 shows the results of recognition rates of different actions with 10%, 20%, and 30% noises added to the 48 voice samples (i.e., N% of data in 20 dimensions were replaced by random numbers between  $[0.0, 1.0]$ ). The average rate of success actions using T-SOM and PL-G-SOM was 48.0% and 86.7, respectively, given by 10 times of executions. Table 4

shows the results of recognition rates of different languages with the respective noisy samples.

Figure 11 shows the comparison of recognition rates of T-SOM and PL-G-SOM when 10% noises existed in all 48 instruction samples.

The results using PL-G-SOM proposed here show advantages than those with conventional learning system in all cases. In fact, we also investigated the use of frequency features for recognition of different instructions, however, similar results were observed in the experiments.

## 5. Conclusion

PL-G-SOM, a novel self-organizing map, was proposed using a reinforcement learning algorithm and annealing schemes of parameters. Online learning and additional learning are available with PL-G-SOM, and it was adopted into a voice instruction learning system of autonomous robot instead of conventional T-SOM. Experiments results showed that the advantage of the new learning system is speed and noise robustness.

TABLE 4: Recognition rates of different languages with noises.

Instruction	Method	Success rate (10% noises)	Success rate (20% noises)	Success rate (30% noises)
<i>Sit down</i>	T-SOM	48.8	55.2	33.6
	PL-G-SOM	87.2	62.0	42.0
<i>Lay down</i>	T-SOM	47.2	34.0	36.4
	PL-G-SOM	90.8	67.2	56.0
<i>Stand up</i>	T-SOM	45.6	33.2	25.2
	PL-G-SOM	86.4	48.4	33.2
<i>Walk</i>	T-SOM	50.4	44.0	37.6
	PL-G-SOM	82.4	52.8	46.4
Average	T-SOM	48.0	41.6	33.1
	PL-G-SOM	86.7	57.6	44.4

## Acknowledgment

This paper was supported by Grant-in-Aid for Scientific Research (JSPS nos. 20500207, 20500277).

## References

- [1] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological Cybernetics*, vol. 43, no. 1, pp. 59–69, 1982.
- [2] T. Kohonen, "Analysis of a simple self-organizing process," *Biological Cybernetics*, vol. 44, no. 2, pp. 135–140, 1982.
- [3] T. Kohonen, "The self-organizing map," *Neurocomputing*, vol. 21, no. 1–3, pp. 1–6, 1998.
- [4] T. Kohonen, *Self-Organizing Maps*, Springer Series in Information Sciences, Springer, Berlin, Germany, 1995.
- [5] M. Cottrell, J. C. Fort, and G. Pagès, "Theoretical aspects of the SOM algorithm," *Neurocomputing*, vol. 21, no. 1–3, pp. 119–138, 1998.
- [6] Bibliography of SOM, <http://www.cis.hut.fi/nnrc/refs/>.
- [7] B. Fritzke, "Growing grid—a self-organizing network with constant neighborhood range and adaptation strength," *Neural Processing Letters*, vol. 2, no. 5, pp. 9–13, 1995.
- [8] H.-U. Bauer and T. Villmann, "Growing a hypercubical output space in a self-organizing feature map," *IEEE Transactions on Neural Networks*, vol. 8, no. 2, pp. 218–226, 1997.
- [9] T. Villmann and H.-U. Bauer, "Applications of the growing self-organizing map," *Neurocomputing*, vol. 21, no. 1–3, pp. 91–100, 1998.
- [10] M. Dittenbach, D. Merkl, and A. Rauber, "Growing hierarchical self-organizing map," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '00)*, vol. 6, pp. 15–19, July 2000.
- [11] E. Berglund and J. Sitte, "The parameterless self-organizing map algorithm," *IEEE Transactions on Neural Networks*, vol. 17, no. 2, pp. 305–316, 2006.
- [12] T. Kuremoto, T. Hano, K. Kobayashi, and M. Obayashi, "For partner robots: a hand instruction learning system using transient-SOM," in *Proceedings of the 2nd International Conference on Natural Computation and the 3rd International Conference on Fuzzy Systems and Knowledge Discovery (FSKD '06)*, pp. 403–414, 2006.
- [13] T. Hano, T. Kuremoto, K. Kobayashi, and M. Obayashi, "A hand image instruction learning system using transient-SOM," *Transaction on Society of Instrument and Control Engineering*, vol. 43, no. 11, pp. 1004–1006, 2007.
- [14] T. Kuremoto, T. Komoto, K. Kobayashi, and M. Obayashi, "A voice instruction learning system using PL-T-SOM," in *Proceedings of the 2nd International Congress on Image and Signal Processing (CISP '09)*, pp. 4294–4299, 2009.
- [15] V. Moschou, D. Ververidis, and C. Kotropoulos, "Assessment of self-organizing map variants for clustering with application to redistribution of emotional speech patterns," *Neurocomputing*, vol. 71, no. 1–3, pp. 147–156, 2007.
- [16] AIBO, <http://www.jp.aibo.com/>.
- [17] S. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, The MIT Press, London, UK, 1998.