International Journal of Networked and Distributed Computing, Vol. 2, No. 3 (August 2014), 166-174

One-D-R-A-G-SOM and its Application to a Hand Shape Instruction Learning System

Takashi Kuremoto*, Takuhiro Otani, Shingo Mabu, Masanao Obayashi
Graduate School of Science and Engineering, Yamaguchi University
Ube, Yamaguchi, 755-8611, Japan
E-mail: {wu, mabu, m.obayas}@yamaguchi-u.ac.jp

Kunikazu Kobayashi
School of Information Science and Technology, Aichi Prefectural University
Nagakute, Aichi, 480-1198, Japan
E-mail: kobayashi@ist.aichi-pu.ac.jp

Corresponding author. E-mail: wu@yamaguchi-u.ac.jp
Tel: +81-836-85-9520

Abstract

In this paper, a novel self-organizing map (SOM) named "One-D-R-A-G-SOM" is proposed. It is a kind of one dimensional ring type growing SOM using asymmetric neighborhood function. As the topology of one dimensional ring type feature map is more suitable to increase or decrease the number of units, and the disorder of the map is available to be solved by the asymmetric neighborhood function, the proposed model gives priority of learning performance to the conventional two dimensional growing SOM. Additionally, One-D-R-A-G-SOM is introduced to a hand shape recognition and instruction learning system. Experiment results showed the effectiveness of the novel system comparing with systems using the conventional SOMs.

Keywords: SOM, PL-G-SOM, HMI, reinforcement learning, hand shape instruction.

1. Introduction

Kohonen's self-organizing map (SOM) is an artificial neural network model which can interpret how different information processing areas in the cerebral cortex of the brain are formed according to the stimulation of sensory input [1]-[4]. SOM has been applied in many information processing fields such as big data classification, dimensionality reduction, and pattern recognition. As we know, there were more than 10,000 publications concerning with SOM till 2011. The spread of SOM comes from its simple but solid structure and competitive learning rule:

(i) Units (neurons) on a low-dimensional grid (usually one or two dimensions) are connected to input vectors with weights;

(ii) The similar units in the means of nearby connection weights gather to each other according to the modification of connection weights using a topological neighborhood function;

(iii) Similar inputs, which similarity is difficult understood for their high dimensionality, are mapped to their area on the grid. So the grid is also called feature map.

However, there are some problems in the classical SOM:

(i) Exhaustion of units of the grid;

(ii) Adjustment of parameters such as learning rate, neighborhood function extent;

(iii) Disorder of topology happened in the learning process.

For the first problem mentioned above, growing SOM (GSOM) [5]-[7], growing hierarchical SOM (GHSOM) [8], transient-SOM (TSOM) [9] [10], etc, are proposed. In GSOM and GHSOM, the number of units is increased according the determinant threshold distance between the input and the connection weights.

Meanwhile, TSOM stores "matured" units in an additional memory space and reuses the initialized map iteratively.

For the second problem, Beglund and Sitte proposed a "parameter-less SOM (PL-SOM)" which reduced parameters, such as learning rate and rang of neighborhood function, using the distance between input and connection weights. Kuremoto et al. adopted PL-SOM into GSOM as a parameter-less growing SOM (PL-G-SOM) in 2010 [11] and applied it to a voice instruction learning system [11] –[13] and a hand shape instruction learning system [13] [14].

The third problem has been tackled by Aoki et al. since 2007 [15]-[17]. Different from the classical SOM using a symmetric neighborhood function, e.g., Gaussian function, an asymmetric neighborhood function is adopted into the competitive learning rule. Additionally, by changing the asymmetric direction iteratively, twist and disorder of map's topology are removed more effectively.

In this paper, we name Aoki et al.'s SOM as "A-SOM" and adopt it to our previous PL-G-SOM to improve the learning performance at first. Then, considering the unnecessary of increased units in 2D PL-G-SOM, we introduce a one dimensional ring type SOM [7] to PL-G-SOM instead of the conventional 2D map. The new structure also makes it easy to delete a unit on the ring type map to realize computational cost down. So the new PL-G-SOM with asymmetric neighborhood function and one dimensional ring type map is named as "One-D-R-A-G-SOM". Furthermore, the proposed SOMs are applied to the hand shape instruction learning system [9] [10] [13] [14] as a novel feature classifier.

The rest of this paper is organized as follows. In Section 2, Kohonen's original SOM and the conventional PL-G-SOM are described at first, then, the One-D-R-A-G-SOM is proposed. In Section 3, a hand shape recognition and instruction learning system is introduced, and the experiment results of the system with different SOMs are reported. Conclusions are in Section 4.

2. Self-Organizing Maps

2.1. *The Original SOM*

In Kohonen's SOM [1]-[4], the input n-dimensional data $\mathbf{x}(x_1, x_2 \ldots, x_n)$ is mapped to a low-dimensional space with connections $\mathbf{m}_i(m_1, m_2 \ldots, m_n)$ by a winner-takes-all rule:

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

where i=1, 2, …, m means the number of unit on a low-dimension map (1 or 2-dimension grids). $\boldsymbol{c}$ indicates the best-match-unit (*BMU*) on the map which has the shortest Euclidean distance with the input data $\mathbf{x}$. Initially, connection weight $\mathbf{m}_i$ is given by a random value, and following learning rule makes input of different class data separately to the different position on the feature map:

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

where α is a learning rate and h_{ci} is a neighborhood function usually as following:

$$h_{ci} = \exp(- \frac{\| \mathbf{r}_i - \mathbf{r}_c \|^2}{2\sigma^2}) . \quad (3)$$

Here, $\mathbf{r_i}, \mathbf{r_c}$ denotes the positions of an arbitrary unit i on the output map and *BMU* $\mathbf{c}$, respectively, $i=1, 2, \ldots, k \leq m= N \times M$, σ is a constant effecting the range of neighborhood units. Obviously, $h_{ci}(x) \geq 0$, $h_{ci}(0) = 1$, $h_{ci}(\infty) = 0$.

2.2. *PL-G-SOM*

In fact, the size of the map i= 1, 2, .., $m = N \times M$ in the original SOM is fixed in advance, so it prevents additional learning when the number of categories of data is more than the number of units on the map (i.e. the size of map) m. To overcome this defects of exhaustion of units problem of the original SOM, Growing SOM (GSOM) [5]-[8] and Transient SOM (T-SOM) [9] [10] were proposed. GSOM suggests to set a few units (one or two) at the beginning of learning process of SOM, and then to increase the number of units when a new input cannot find a *BMU* for its too far distance to all units on the map. Meanwhile T-SOM uses a memory layer to store "matured" units which are trained enough and release the matured units on the map with initial random values. However, according to this forced processing, the topology of the map become to be disordered and this affects the learning performance of T-SOM. So in this study, we concentrate to the former one, GSOM.

When a new row/column needs to be insert to the neighbor of a *BMU* ***c***, for example, in the middle of unit ***c*** and the farthest unite ***f*** from ***c***, the weight of connection between input and the new unite ***e*** takes average values of ***c*** and ***f***,

$$\mathbf{m}_r = 0.5\ (\mathbf{m}_c + \mathbf{m}_e)\ , \tag{4}$$

and so do them of ***e***'s neighbors:

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

where $l=1, 2, ..., N$ or M. After this process, the map size changes to $N\times(M+1)$, or $(N+1)\times M$ (See Fig. 1).

To decide parameters α in Eq. (2) and σ in Eq. (3), Berglund & Sitte proposed a data-driven method in their PL-SOM [5]:

$$\alpha(t) = \frac{\| \mathbf{x(t)} - \mathbf{m}_c(t) \|^2}{r(t)}) \ . \tag{6}$$

$$r(t) = \max(\| \mathbf{x}(t) - \mathbf{m}_c(t) \|^2, r(t-1)) \ . \tag{7}$$

$$\sigma(t) = \sigma_{\max}\ \alpha(t) \geq \sigma_{\min} \ . \tag{8}$$

Where $\sigma_{\max}, \sigma_{\min}$ are positive parameters, for example, the value may be the size of the map and 1.0, respectively. t is the number of training time.

In our previous works [11]-[13], we fused PL-SOM and GSOM to be "PL-G-SOM", and applied it on voice instruction learning system [11] [13] and image (hand shape) instruction learning system [12] [13].

2.3. *One-D-R-A-G-SOM*

Because the initialization of the original SOM uses random values for connection weights of units, topological defect which shows the disorder (twist) of the map may happen and it affects learning performance of SOM and the categorization results. To tackle this problem, Aoki et al. [15]-[17] proposed to use an asymmetric neighborhood function Eq. (9)-Eq. (12) instead of conventional symmetric Gaussian function Eq. (3).

$$h_\beta(\tilde{r}_{ic}) = 2(\frac{1}{\beta_0} + \beta_0)^{-1} \exp(-\frac{\tilde{r}_{ic}^2}{2\sigma^2}), \tag{9}$$

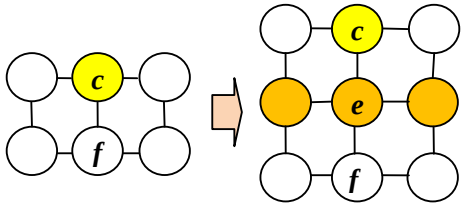


Fig. 1. GSOM: Insert a new row or column between the BMU c and the farthest unit f among neighbors of c.

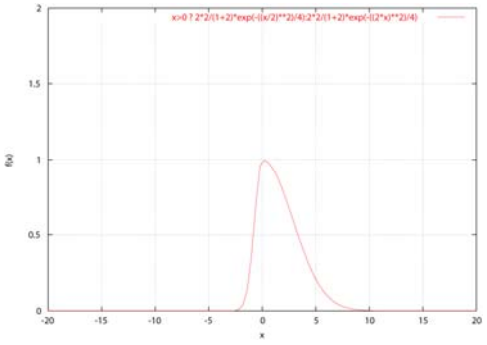


Fig. 2. A one-dimension asymmetric neighborhood function ($\beta_0 = 2.0; \sigma = 2.0$).

$$\tilde{r}_{ic} = \begin{cases} \sqrt{(\frac{\mathbf{r}_{ic}\cdot\mathbf{k}}{\beta_0})^2 + \|\mathbf{r}_\perp\|^2}, if\ \mathbf{r}_{ic}\cdot\mathbf{k} \geq 0, \\ \sqrt{(\beta_0\cdot\mathbf{r}_{ic}\cdot\mathbf{k})^2 + \|\mathbf{r}_\perp\|^2}, if\ \mathbf{r}_{ic}\cdot\mathbf{k} < 0. \end{cases} \tag{10}$$

$$\mathbf{r}_\perp \equiv \mathbf{r}_{ic} - (\mathbf{r}_{ic}\cdot\mathbf{k})\mathbf{k} \ , \tag{11}$$

$$\mathbf{r}_{ic} \equiv \| \mathbf{r}_i - \mathbf{r}_c \| \ . \tag{12}$$

Where i is an arbitrary unit on the map, ***c*** is the *BMU*, $\mathbf{k}$ indicates the asymmetry direction, $\mathbf{r}_\perp$ are the remaining components perpendicular to $\mathbf{k}$, and the asymmetric parameter $\beta_0 \geq 1$ gives the degree of asymmetric.

In Fig. 2, the shape of a one-dimension asymmetric neighborhood function is shown. It is supposed that BMU is on the origin of the dimension, and units on the right direction means the asymmetry direction.

Furthermore, an improved practical algorithm using asymmetric neighborhood function was also proposed by Aoki *et al.* [2]. The improved algorithm is during the training process, the direction of asymmetry is inversed in a certain period T. And the disorder of the map's

topology is reduced by an asymptotic adjustment of asymmetric parameter β_0:

$$\beta_0(t) \leftarrow \begin{cases} 1+(\beta_0-1)(1-\frac{t}{T_{total}}), & if \quad t < T_{total}, \\ 1, & if \quad t \geq T_{total}, \end{cases} \quad (13)$$

Here, we propose to adopt the asymmetric neighborhood function to PL-G-SOM, and to restrain excessive growing of units, one dimensional ring type SOM [7] is used. So the learning rule of a novel SOM named "One-D-R-A-G-SOM" is proposed as following:

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + 2(\frac{1}{\beta_0}+\beta_0)^{-1} \exp(-\frac{\tilde{r}_{ic}^2}{2\sigma(t)^2})(\mathbf{x}-\mathbf{m}_i(t)) \quad . \quad (14)$$

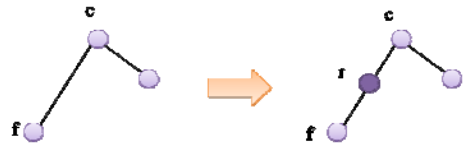
The timing to generate a new unit on the ring of one dimensional SOM is when a unit is selected as *BMU* too many times (i.e., setting a threshold of *BMU* counting). The new unit is inserted between the BMU and its furthest unit of neighborhoods. Additionally, it is easy to delete useless unit when a unit is not selected often. The counter is given by Eq. (15).

$$C_i(t+1) = \begin{cases} C_i(t)+1 & if\ i\ is\ \ BMU\ c \\ C_i(t)-0.005 & if\ i\ \ is\ the\ ne\ arest\ unit\ \ to\ BMU\ c \\ C_i(t)-0.008 & ot herwise \end{cases} \quad (15)$$

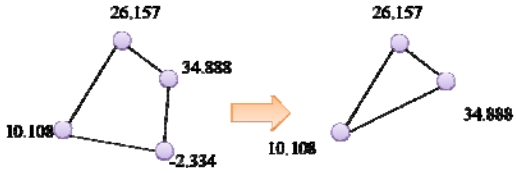
The generation and elimination of units of One-D-R-A-G-SOM is depicted in Fig. (3).

3. A Hand Shape Instruction Learning System

Because we have applied T-SOM, PL-T-SOM, and PL-G-SOM to a hand shape instruction learning system for partner robots in the previous works [9] [10] [13] [14], it is interesting to apply the One-D-R-A-G-SOM proposed in the former Section to the system. The action learning system was designed to realize human machine interaction (HMI) by natural interface: different shapes of a hand of instructor (user) is captured by the visual sensor (camera) of robot, and these labeled image classes play a role to control the output of a robot. The research of HMI has developed since the center of last century, various input information such as pose of body, gesture of hands, face expression, iris movement, etc, are used in the field. And the gesture of hand can be consider as one of the most convenient input signals to the robot [20]-[22].



(a) A new unit is generated between *BMU* **c** and unit **f**



(b) Unit with a low value of *BMU* time is eliminated by a threshold

Fig. 3. Insert / Delete a unit to / from a one dimensional ring type SOM.

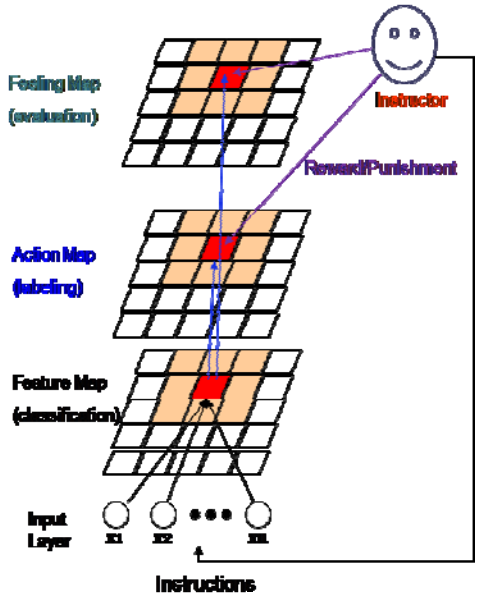


Fig. 4. A hand shape instruction learning system.

3.1. *The Structure of System*

Fig. 4 shows the architecture of a hand shape recognition and instruction learning system.
There are 4 layers in the system:

(i) Input layer which accept input data in *n* dimension space;

(ii) Feature map layer which is a One-D-R-A-G-SOM to classify the input data;

(iii) Action map which labels the input patterns to candidate actions;

(iv) Feeling map which expresses the success rate of instruction learning for evaluation.

The details of Action map and Feeling map are described in the next subsections.

3.2. *Action Map*

The instructor presents his/her instructions with the different shapes of his/her hand to a robot, and in the view of the robot, a hand shape which is observed means a state of the environment s_t, the robot intends to select a valuable action $a_t(i)$ adapting to the state, $i = 1, 2, \ldots, A$ is the number of candidate action, by a stochastic action policy π, which is according to Gibbs distribution (Boltzmann distribution) as given by Eq. (16).

$$\pi_t(a_t(i) \mid s_t) = \frac{e^{\frac{Q_t(s_t, a_t(i))}{T}}}{\sum_{j=1}^{A} e^{\frac{Q_t(s_t, a_t(j))}{T}}}, \qquad (16)$$

where T is a parameter named "temperature" which comes from the physical state description of a system (higher temperature lower possibility), t is the iteration time of learning, A is the number of available actions. When an action is selected according to Eq. (16) and performed by the robot, its instructor evaluates the action by giving a reward/punishment r to robot. The reward is accepted and used to modify the value of Q_t by Eq. (17), where Q_t is called "state-action value function" in reinforcement learning (RL) [18].

$$Q_{t+1}(s_t, a_t(i)) = Q_t(s_t, a_t(i)) + r \quad . \qquad (17)$$

Where s means the state of environment observed by the robot, a is the action selected by the learner, r is the reward (scalar) given by the instructor (user).

Table 1 shows a Q-value table which is used to match input image patterns to candidate actions. The value in the table is given by random number initially, and changed by Eq. (17). Q value is used in the probability distribution function Eq. (16) to select an adaptive action meeting to a hand image instruction decided by the user.

3.3. *Feeling Map*

To express the degree of how an instruction is learned by robot, a Feeling map which has the same number of units with Action map is designed as the output layer of the learning system as shown in Fig. 4. Feeling map expresses instruction recognition rate, i.e., the feeling of robot: more successful, happier it is. Feelings of partner robots, such as pet robots, entertainment robots, and so on, are important for human-machine interaction (HMI) when they are able to express vividly by their face expressions [19]. The distance between input pattern and units on 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 initial value of each unit here. The calculation of Feeling map is given by Eq. (18).

$$F_{t+1}(i) = F_t(i) \pm aC - bD_i, \qquad (18)$$

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 i on Feature Map and the input data, a, b are constants, and $0 < a < 1,\ 0 < b \ll 1$.

Table 1. The value of units $Q_t(s_t, a_t)$ on Action Map

Unit of Action Map (s_t)	**Action 1** ($a_t(1)$)	**Action 2** ($a_t(2)$)	…	**Action A** ($a_t(A)$)
1	6	2	…	0
2	10	1	…	1
…	…	…	…	…
p	-27	3	…	2
…	…	…	…	…
…	…	…	…	…
$N \times M$	0	2	…	2

3.1. *Experiments and Results*

Skin area in the image captured by a CCD camera needs to be extracted and regularized at first. For a frame of image in RGB format, it is transformed to HSV format at first, then, using the threshold values of Hue (H), and Saturation (S) [23] and Red (R) [9] [10] [13] [14] threshold in RGB, skin area is extracted as a binary image. Noise elimination and holes filling are also effective to segment a hand area from the binary

image. The thresholds for skin of a yellow race people in the room of fluorescent lights (around 500*lx*) are given as follows as we investigated:

1) When H, $S \in [0, 360]$ degree,

If $10 \leq S < 15$, then $H > 350$;

If $15 \leq S < 20$, then $H > 330$;

If $20 \leq S < 30$, then $H > 300$ or $H < 40$;

If $30 \leq S < 50$, then $H > 250$ or $H < 30$;

If $50 \leq S < 70$, then $H > 230$ or $H < 30$;

If $70 \leq S < 150$, then $H > 220$ or $H < 40$;

If $S < 10$ or $150 \leq S \leq 360$,

then $H > 300$ or $H < 40$;

2) When R, G, $B \in [0, 255]$,

$30 < R < 250$.

In Fig. 5, the processing of binary image of hand area is shown. Skin area is segmented according to the HSV & RGB method described above (Fig. 5 (a)), and normalized by centering, axis decision, rotation, and expansion (Fig. 5 (b)).

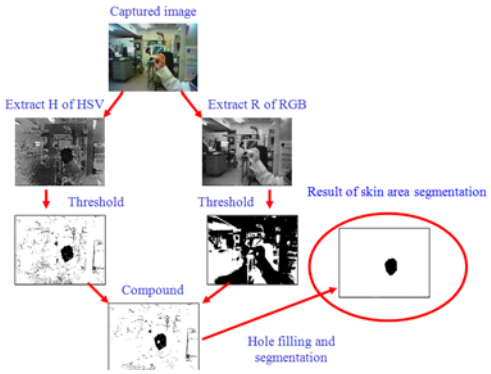
To distinguish the type of a hand shape, feature space definition is important to result high rate of pattern recognition. We discussed the methods of feature space construction in our previous works and proposed a useful feature vector space of hand shapes. The input images are analyzed by an 80-dimension vector space (See Fig. 6). From the origin of the space to the end of the hand area, the lengths in axes each 1.8-degree increased (80 axes) are the values of the feature vector, i.e., $(x_1, x_2, \ldots, x_{80})$.

8 kinds of hand shapes were used in the hand image instruction learning system experiment (Fig. 7). For each kind of instruction, we recorded 3 samples, so the total input data for SOM was 24 samples. Parameters used in the experiments are shown in Table 2.

In Fig. 7, original images are shown in the left column, binary images of interesting region are in the center column, and feature vectors are depicted in the right column.

In Fig. 8, learning convergence comparison between conventional PL-G-SOM and the proposed One-D-R-A-G-SOM is shown. The distance between input and units of map decreased according to the increase of the learning time in all cases, but One-D-R-A-G-SOM showed better performance than the conventional PL-G-SOM.

In Fig. 9, the growth of maps during learning process is shown. The initial size of each map was 9 units. The final number of units of conventional PL-G-SOM and the proposed One-D-R-A-G-SOM are 44, and 16, respectively. It suggests that the later one had less cost of computation.



(a) Hand area is extracted and binarized

(b) Binary hand area is normalized as an input image.

Fig. 5. Hand shape extraction.

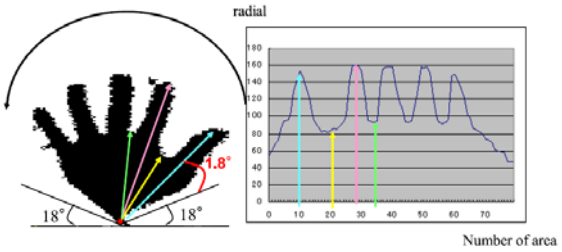


Fig. 6. Input vector with 80 dimensions for hand shape recognition [9] [10] [13] [14].

In Fig. 10, the change of adaptive learning rate $\alpha(t)$ (Eq. (6)) is shown. It can be confirmed that $\alpha(t)$ decreased according to the training time and this change provided better convergence of learning than using fixed learning rate.

In Fig. 11, the changed of feeling values of conventional PL-G-SOM (gray lines) and proposed One-D-R-A-G-SOM (dark lines) feeling maps are plotted. It can be observed that the proposed method results the better convergence than the conventional system though both feeling values reached 1.0, the highest feeling value, i.e., 100% success rate of instruction learning.

Asymmetric neighborhood function can be adopted in to PL-G-SOM naturally as "A-PL-G-SOM". Meanwhile, a one dimensional ring type SOM is also available to be

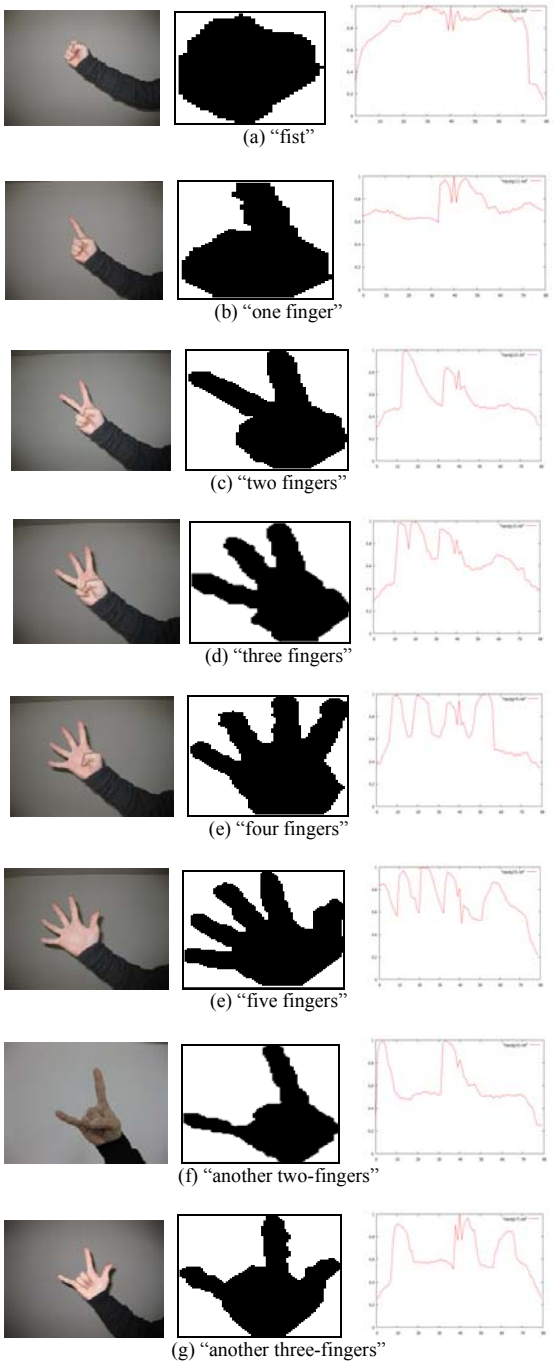


Fig. 7. Hand shapes for instruction learning of robot. Left: original image; Center: binary image; Right: normalized feature (input of SOM).

used in the hand image instruction learning system as a feature map (See Fig. 4).

Table 3 shows a comparison of learning performance between different SOMs used in the hand image instruction learning experiments. On index of the

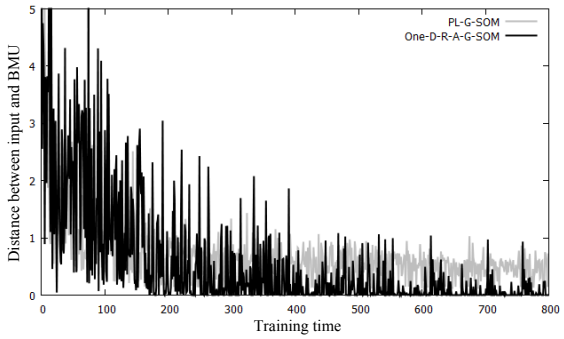


Fig. 8. Comparison of learning convergence: Gray lines for conventional PL-G-M; Black lines for One-D-R-A-G-SOM.

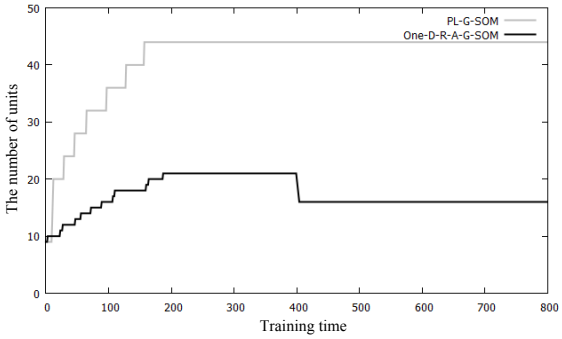


Fig. 9. Comparison of the change of number of units during training between different SOMs.

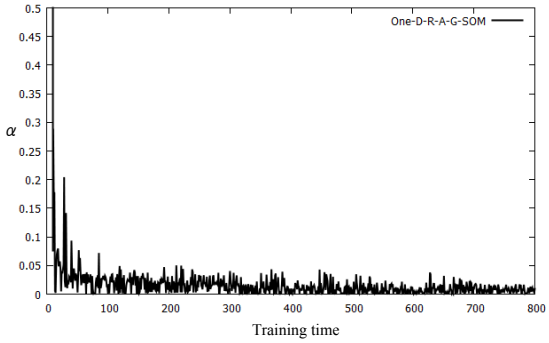


Fig.10. Parameter value in Eq. (6) changed during training (One-D-R-A-G-SOM).

learning error "MSE", A-PL-G-SOM records the best learning performance. All of systems learned 4 kinds of instructions with 100% success rate whereas Feeling value reached the highest 1.0. One-D-R-A-G-SOM proposed in this paper marked the shortest computational time 7.46 seconds, with the least number of units 16.

Table 2. Parameters used in the experiments.

Description	**Symbol**	**Quantity**
Size of initial PL-G-SOM, A-PL-G-SOM, One-D-R-SOM, One-D-R-A-G-SOM	$N \times M$	3×3 , 9
Number of images	-	24
Iteration times for learning	t	800
Temperature in the Bolzmann distribution	T	1.0
Number of instructions (actions)	$a(i)$	8
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
Range of neighborhood in AGSOM and IAGSOM	σ	4.0
Learning rate of neighborhood of AGSOM and IAGSOM	α	0.04
Expansion/Compress rate of asymmetric function	β_0	3.0
Threshold to eliminate a unit	-	$F_t(i)<0.5$

Table 3 The value of units $Q_t(s_t, a_t)$ on Action Map

Item	PL-G-SOM	A-PL-G-SOM	One-D-R-SOM	One-D-R-A-G-SOM
MSE	0.729	**0.141**	1.534	0.552
Feeling value	**1.0**	**1.0**	**1.0**	**1.0**
Number of units	44	90	18	**16**
Running time (sec.)	140.40	280.50	8.10	**7.46**

4. Conclusions

To improve the learning performance of growing self-organizing map (GSOM), Aoki et al.'s asymmetric neighborhood function is adopted in parameterless growing SOM (PL-G-SOM). And to reduce the number of units of PL-G-SOM, one dimensional ring type grid

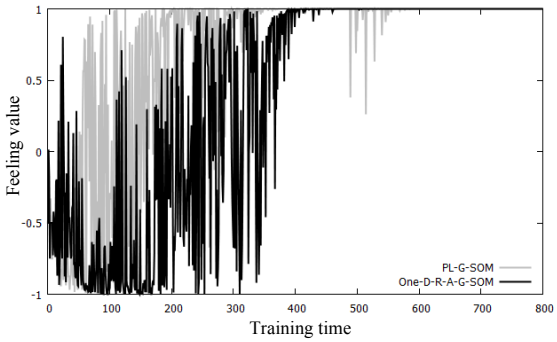


Fig. 11. Comparison of success rates between the conventional PL-G-SOM and proposed One-D-R-A-G-SOM.

is used to instead of the conventional map used in PL-G-SOM.

The proposed "One-D-R-A-G-SOM" was verified its prior learning performance in a hand shape recognition and instruction learning system which was designed to be a human-robot interaction system. Comparing with conventional SOMs, A-PL-G-SOM and One-D-R-A-G-SOM showed their high learning convergence and low learning costs.

Not only image instruction learning, but also voice instruction learning system, or brain wave instruction learning system are considerable to be designed, using the proposed One-D-R-A-G-SOM, and we leave these works in the future.

Acknowledgements

This work was supported in part by KAKENHI (No. 26330254, No.23500181, and No.25330287) from JSPS, Japan.

References

1. T. Kohonen, Self-organized formation of topologically correct feature maps, *Biological Cybernetics* 43(1) (1982) 59-69.
2. T. Kohonen, T., *Self-Organizing Maps* (Springer, Berlin; Heidelberg; New-York, Series in Information Sciences, 1995).
3. T. Kohonen, The self-organizing map, *Neurocomputing*, 21 (1998) 1-6.
4. T. Kohonen, Essentials of the self-organizing map, *Neural Networks*, 37 (2013) 52-65.
5. B. Fritzke, Growing grid – a self-organizing network with constant neighborhood range and adaption strength, *Neural Processing Letters*, 2(5) (1995) 9-13.

6. H.-U. Bauer and Th. Villmann, Growing a hypercubical output space in a self-organizing feature map, *IEEE Transaction on Neural Networks*, 8(2) (1997) 218-226.
7. R. Ohta and T. Saito, A growing self-organizing algorithm for dynamic clustering, In *Proceedings of International Joint Conference on Neural Networks (IJCNN '01)*, (Washington, DC, 2001) pp. 469-473.
8. M. Dittenbach, D. Merkl and A. Rauber, The growing hierarchical self-organizing map, In *Proceedings of IEEE International Joint Conference on Neural Network (IJCNN '00)*, (Como, Italy, 2000) pp. 15-19.
9. 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 (ICNC '06-FSKD'06)* (Xi'an, China, 2006) pp. 403-414.
10. T. Hano, T. Kuremoto, K. Kobayashi, and M. Obayashi, A hand image instruction learning system using Transient-SOM, *Transactions on SICE (Society of Instrument and Control Engineering)* 43(11) (2007) 1004-1006 (in Japanese)
11. T. Kuremoto, T. Komoto, K. Kobayashi and M. Obayashi, Parameterless-Growing-SOM and its application to a voice instruction learning system, *Robotics* (2010) 9 pages (MDPI, Open Access).
12. T. Kuremoto, T. Yamane, L.-B. Feng, K. Kobayashi and M. Obayashi, A human-machine interaction system: A voice command learning system using PL-G-SOM", *Proceedings of International Conference on Industrial Engineering and Management Special Session in MASS (IEEE-IEM 2011)*, (Zhengzhou, China, 2011) pp. 83-86.
13. T. Kuremoto, M. Obayashi, K. Kobayashi and L.-B. Feng, Instruction learning systems for partner robots, *Advances in Robotics-Modeling, Control, and Applications*, eds. C. Ciufudean and L. Garcia, (iConcept, Hong Kong, 2012) pp. 149-170.
14. T. Kuremoto, T. Otani, L.-B. Feng, K. Kobayashi and M. Obayashi, A hand image instruction learning system using PL-G-SOM, In *Proceedings of the 12th International Conference on Artificial Intelligence (ICAI 2012)*, (Las Vegas, U.S.A., 2012).
15. T. Aoki and T. Aoyagi, Self-organizing maps with asymmetric neighborhood function, *Neural Computation*, 19 (2007) 2515-2535.
16. T. Aoki, K. Ota, K. Kurata and T. Aoyagi, Ordering process of self-organizing maps improved by asymmetric neighborhood function, Cognitive Neurodyamics, 3 (2009) 9-15.
17. K. Ota, T. Aoki, K. Kurata and T. Aoyagi, Asymmetric neighborhood functions accelerate ordering process of self-organizing maps, *Physical Review E*, 83 (2011) 021903.
18. S.S. Sutton and A.G. Barto, *Reinforcement Learning: An Instruction*, (The MIT Press, London, 1998).
19. T. Kuremoto, T. Hano, K. Kobayashi and M. Obayashi, Robot feeling formation based on image features, In *Proceedings of International Conference on Control, Automation and Systems (ICCAS 2007)*, (Seoul, Korean, 2007) 758-761.
20. V.I. Pavlovic, R. Sharma and T.S. Huang, Visual interpretation of hand gesture for human-computer interaction: a review. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 19 (7) (1997) 667–693.
21. T. Kuremoto, Y. Kinoshita, L.-B. Feng, S.Watanabe, K. Kobayashi and M. Obayashi, A gesture recognition system using One-Pass DP method. In *Proceedings of the 7th International Conference on Intelligent Computing (ICIC 2011)*, *Lecture Note in Artificial Intelligence (LNAI)*, 6839 (Zhengzhou, China, 2011) 581-587.
22. T. Kuremoto, Y. Kinoshita, L.-B. Feng, S.Watanabe, K. Kobayashi and M. Obayashi, A gesture recognition system with retina-V1 model and one-pass dynamic programming. *Neurocomputing*, 116 (2012) 291-300.
23. S. Sherrah, S. Gong, *Skin Colour Analysis*, online URL: http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/GONG1/cvOnline-skinColourAnalysis.html, 2001.