2006 International Joint Conference on Neural Networks
Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada
July 16-21, 2006

Behavior of Fatigable SOM and its Application to Clustering

Masato Tomita, *Non-member, IEEE*, Haruna Matsushita, *Student member, IEEE* , and Yoshifumi Nishio, *Member, IEEE*

***Abstract*— The Self-Organizing Map (SOM) is popular algorithm for unsupervised learning and visualization introduced by Teuvo Kohonen. One of the most attractive applications of SOM is clustering and several algorithms for various kinds of clustering problems have been reported and investigated. In this study, we propose a new type of SOM algorithm, which is called Fatigable SOM (FSOM) algorithm. The important feature of FSOM is that the neurons are fatigable, namely, the neurons which have become a winner can not become a winner during a certain period of time. Because of this feature, FSOM tends to self-organize only in the area where input data are concentrated. We investigate the behavior of FSOM and apply FSOM to clustering problems. Further, we introduce the fatigue level to FSOM to increase its flexibility for various kinds of clustering problems. The efficiencies of FSOM and the fatigue level are confirmed by several simulation results.**

I. INTRODUCTION

In data mining, clustering is one of the most important issues and is also very useful for many applications, such as industrial engineering, image processing, biology and medicine. Then, the Self-Organizing Map (SOM) attracts attentions for clustering in recent years. SOM is popular tools for clustering and visualization of high-dimensional data [1]. SOM is unsupervised neural network introduced by Kohonen in 1982 [2] and is a model simplifying self-organization process of the brain. SOM maps multidimensional data onto a 2-dimensional grid. SOM can classify input data according to similarities and patterns which are obtained by the distance between neurons and is applied to wide fields of data classifications. Although many methods to extract clusters by using SOM have been proposed [3]-[10], it seems to be very difficult to construct a simple method using SOM for universal input data. On the one hand, in the world, the amount and complexity of data increase from year to year. Therefore, it is important to investigate various extraction methods of clusters from data including a lot of noises.

In this study, we propose a new type of SOM algorithm, which is called Fatigable SOM (FSOM) algorithm. The important feature of FSOM is that the neurons of FSOM are fatigable, namely, in the learning process of FSOM, the neurons which have become a winner can not become a winner during a certain period of time. Because of this feature, FSOM tends to self-organize the area where the input data are concentrated. We investigate the behavior of FSOM and apply FSOM to clustering problems. Further, we introduce the fatigue level to FSOM to increase the flexibility for various kinds of clustering problems. The efficiencies of FSOM and the fatigue level are confirmed by several simulation results.

In the Section II, the algorithm of the conventional SOM is introduced. In the Section III, we explain the learning algorithm of the proposed FSOM algorithm. In the Section IV, the behavior of FSOM is explained with some simulation results of clustering. In the Section V, we introduce the fatigue level to FSOM. In the Section VI, the efficiency of the fatigue level is confirmed with some simulation results.

Masato Tomita, Haruna Matsushita and Yoshifumi Nishio are with the Department of Electrical and Electronic Engineering, Tokushima University, Tokushima, Japan (phone: +81-88-656-7470; fax: +81-88-656-7471; email: {tomita, haruna, nishio}@ee.tokushima-u.ac.jp).

II. SELF-ORGANIZING MAP (SOM)

In this section we introduce the conventional SOM algorithm in order to make clear new points of the proposed algorithm.

SOM has two-layer structure of the input layer and the competitive layer. In the competitive layer, m neurons are arranged as a regular 2-dimensional grid. The range of the elements of d-dimensional input data $\boldsymbol{x}_j = (x_{j1}, x_{j2}, \cdots, x_{jd})$ $(j = 1, 2, \cdots, N)$ are assumed to be from 0 to 1.

(SOM1) The initial values of all the weight vectors $\boldsymbol{w}_i = (w_{i1}, w_{i2}, \cdots, w_{id})$ $(i = 1, 2, \cdots, m)$ of the neurons are given between 0 and 1 at random.
(SOM2) An input data $\boldsymbol{x}_j$ is inputted to all the neurons at the same time in parallel.
(SOM3) We find the winner neuron c by calculating the distances between $\boldsymbol{x}_j$ and $\boldsymbol{w}_i$ according to;

$$c = \arg\min_i \{\|\boldsymbol{w}_i - \boldsymbol{x}_j\|\}. \qquad (1)$$

In other words, the winner neuron c is the neuron with the weight vector nearest to the input vector $\boldsymbol{x}_j$. In this study, Euclidean distance is used for Eq. (1).
(SOM4) The weight vectors of all the neurons are updated as;

$$\boldsymbol{w}_i(t+1) = \boldsymbol{w}_i(t) + h_{c,i}(t)(\boldsymbol{x}_j - \boldsymbol{w}_i(t)) \qquad (2)$$

where t is the learning step. The function $h_{c,i}(t)$ is called the neighborhood function and it is described as follows;

$$h_{c,i}(t) = \alpha(t) \exp\left(-\frac{\|\boldsymbol{r}_i - \boldsymbol{r}_c\|^2}{2\sigma^2(t)}\right) \qquad (3)$$

where $\alpha(t)$ is the learning rate, $\boldsymbol{r}_i$ and $\boldsymbol{r}_c$ are the vectorial locations of the neurons on the display grid, and $\sigma(t)$ corresponds to the widths of the neighborhood function. $\alpha(t)$ and $\sigma(t)$ decrease with time according to the following equations;

$$\alpha(t) = \alpha(0)\,(1 - t/T)\,,$$
$$\sigma(t) = \sigma(0)\,(1 - t/T)\,, \tag{4}$$

where T is the maximum number of the learning.

(SOM5) The steps from (SOM2) to (SOM4) are repeated for all the input data, namely, from $j = 1$ to $j = N$.

III. Fatigable SOM (FSOM)

In this study, we propose Fatigable SOM (FSOM) algorithm. The important feature of FSOM is that the neurons are fatigable. Namely, in the FSOM algorithm, the neurons which have become a winner can not become a winner during a certain period of time. We call the neuron which has become a winner the fatigued neuron. We have very little difficulty in understanding why neurons are fatigued with becoming a winner. Because human-beings are fatigued with doing something, neurons also are fatigued.

A. Learning Algorithm

We explain the learning algorithm of FSOM. In FSOM, m neurons are arranged as a regular 2-dimensional grid. The range of the elements of d-dimensional input data $\boldsymbol{x}_j$ = $(x_{j1}, x_{j2}, \cdots, x_{jd})$ $(j = 1, 2, \cdots, N)$ are assumed to be from 0 to 1.

(FSOM1) The initial values of all the weight vectors $\boldsymbol{w}_{Fi}$ = $(w_{Fi1}, w_{Fi2}, \cdots, w_{Fid})$ $(i = 1, 2, \cdots, m)$ of the neurons are given between 0 and 1 at random.

(FSOM2) An input data $\boldsymbol{x}_j$ is inputted to all the neurons at the same time in parallel.

(FSOM3) We find the winner neuron c among all the neurons EXCEPT the neurons which have become winners (namely, the fatigued neurons) by calculating the distance between $\boldsymbol{x}_j$ and $\boldsymbol{w}_{Fi}$ according to;

$$c = \arg\min_i \{\|\boldsymbol{w}_{Fi} - \boldsymbol{x}_j\|\}. \tag{5}$$

In other words, the winner neuron is the neuron with the weight vector nearest to the input vector among the all neurons except the fatigued neurons. In this study, Euclidean distance is used for Eq. (5)

(FSOM4) The weight vectors of all the neurons are updated as;

$$\boldsymbol{w}_{Fi}(t+1) = \boldsymbol{w}_{Fi}(t) + h_{Fc,i}(t)(\boldsymbol{x}_j - \boldsymbol{w}_{Fi}(t)) \tag{6}$$

where t is the learning step. The function $h_{Fc,i}(t)$ is called the neighborhood function and it is described as follows;

$$h_{Fc,i}(t) = p_c(t) \exp\left(-\frac{\|\boldsymbol{r}_i - \boldsymbol{r}_c\|^2}{2\sigma^2(t)}\right), \tag{7}$$

where $p_c(t)$ is the learning function, $\boldsymbol{r}_i$ and $\boldsymbol{r}_c$ are the vectorial locations of the neurons on the display grid, and $\sigma(t)$ corresponds to the widths of the neighborhood function. $\sigma(t)$ decrease with time according to the following equation;

$$\sigma(t) = \sigma(0)(1 - t/T), \tag{8}$$

where T is the maximum number of the learning. The learning function $p_c(t)$ is explained in the next subsection.

(FSOM5) The winner neuron c is labeled as a fatigued neuron.

(FSOM6) If FSOM has n_f fatigued neurons, all the fatigued neurons are recovered and turn to be normal neurons.

(FSOM7) The steps from (FSOM2) to (FSOM6) are repeated for all the input data, namely, from $j = 1$ to $j = N$.

B. Learning function

We use the learning function proposed in our past study [10]. The value of the learning function is determined by the distance between the input vector $\boldsymbol{x}_j$ and the weight vector $\boldsymbol{w}_{Fc}$ of the winner neuron c according to;

$$p_c(t) = \beta(t) \exp\left(-\frac{\|\boldsymbol{w}_{Fc} - \boldsymbol{x}_j\|^2}{2\sigma_P^2}\right), \tag{9}$$

where σ_P corresponds to the width of the learning function, and $\beta(t)$ corresponds to the maximum value of the learning function. $\beta(t)$ decrease with time according to the following equation;

$$\beta(t) = \beta(0)(1 - t/T). \tag{10}$$

IV. Simulation Results of FSOM

A. Behavior of FSOM

We consider the 2-dimensional input data of 400 points whose distribution is non-uniform as Fig. 1(a). 200 points are distributed within a small range from 0.3 to 0.5 horizontally and from 0.3 to 0.5 vertically. The remaining 200 points are uniformly distributed between 0 and 1 at random. We consider the conventional SOM and the proposed FSOM with 100 neurons (10×10). The parameters for the learning of SOM and FSOM are chosen as follows;

$$\alpha(0) = 0.7,\ \sigma(0) = 3.0,$$
$$\beta(0) = 0.7,\ \sigma_P = 1/16,\ n_f = 100. \tag{11}$$

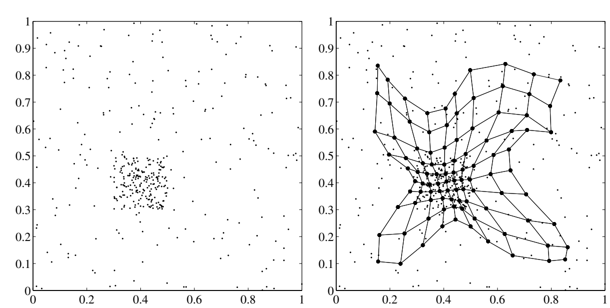


Fig. 1. Conventional SOM. (a) 2-dimensional input data. (b) Learning results.

We execute the learning for all input data once. The simulation results of the conventional SOM is shown in Fig. 1(b). We can see that the conventional SOM self-organizes all over the input data. Figure 2 shows the learning process of the proposed FSOM. The final result is shown in Fig. 2(f). FSOM self-organizes only the area where the input

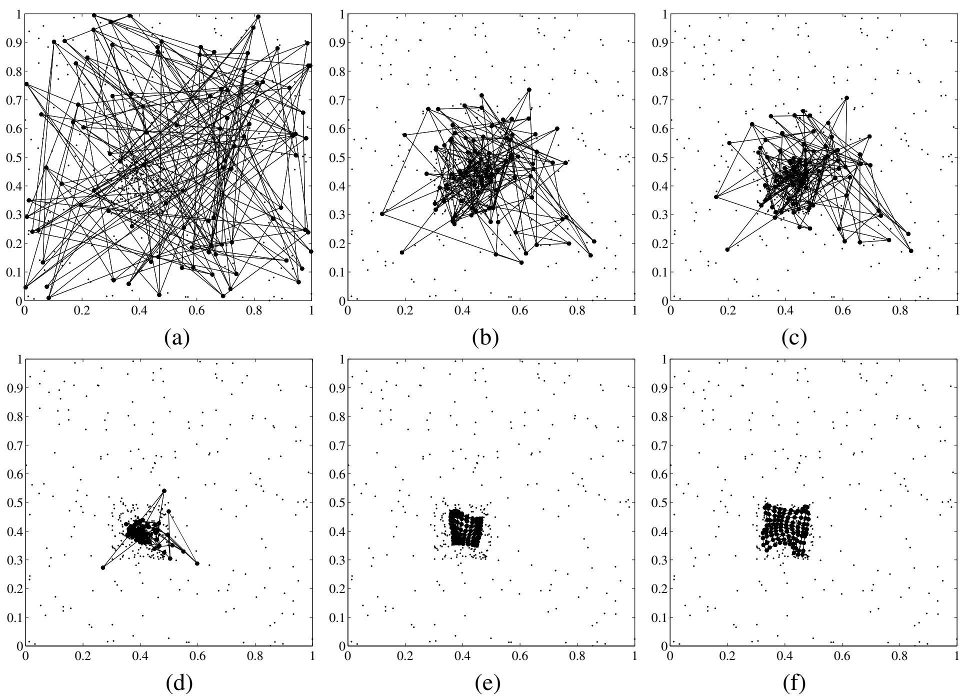


Fig. 2. Learning of proposed FSOM. (a) Initial state (t = 0). (b) t = 5. (c) t = 10. (d) t = 20. (e) t = 100. (f) Learning results (t = 400).

data are concentrated. Let us examine the behavior of FSOM in more detail.

(Early behavior) In the early stage of the learning process, all the neurons of FSOM gather in the area where the input data are concentrated as shown in Figs. 2(a) to (d). This behavior can be explained by the most important feature of FSOM, namely the neurons which have become a winner can not become a winner during a certain period of time as shown in Fig. 3. When we use the conventional SOM, it is highly possible that the same neurons become the winner many times for the concentrated input data. This prevents the neurons except the winner neurons from gathering in a cluster. However, neurons in FSOM are fatigable and cannot be a winner twice. This means that many different neurons can be winners for the concentrated input data.

(Later behavior) In the later stage of the learning process, all the neurons of FSOM move to fit the area where the input data are concentrated as shown in Figs. 2(e) and (f). This behavior can be explained by the learning function Eq. (9). This learning function enhances the movement of the neurons if the winner neuron is close to the input data. Hence, FSOM tends to move to fit the area. On the other hand, this learning function prevents the neurons to make a large move for input data far apart from the neurons in the later stage. Hence, FSOM does not spread out of the area where the input data are concentrated.

Because of these two different features, FSOM tends to self-organize only the area where the input data are concentrated.

B. Application to Clustering

The concept using FSOM can be exploited to extract the data only in clusters of the input data including a lot of noises, because FSOM can find such areas by itself. We carry out the extraction of cluster after Fig. 1(b) and Fig. 2(f). The extraction method is a relatively simple as follows. After learning, the input data which is within a radius of R from all neurons on the map are classified into the cluster.

The simulation results of the conventional SOM and FSOM are shown in Figs. 4(a) and (b), respectively (R = 0.05). As we can see from these figures, the cluster obtained by the conventional SOM includes a lot of noises. However, FSOM can successfully extract only the cluster.

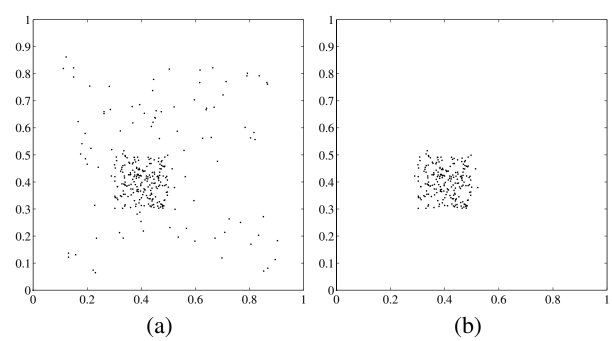


Fig. 4. Extraction of cluster. (a) Cluster extracted by conventional SOM. (b) Cluster extracted by FSOM.

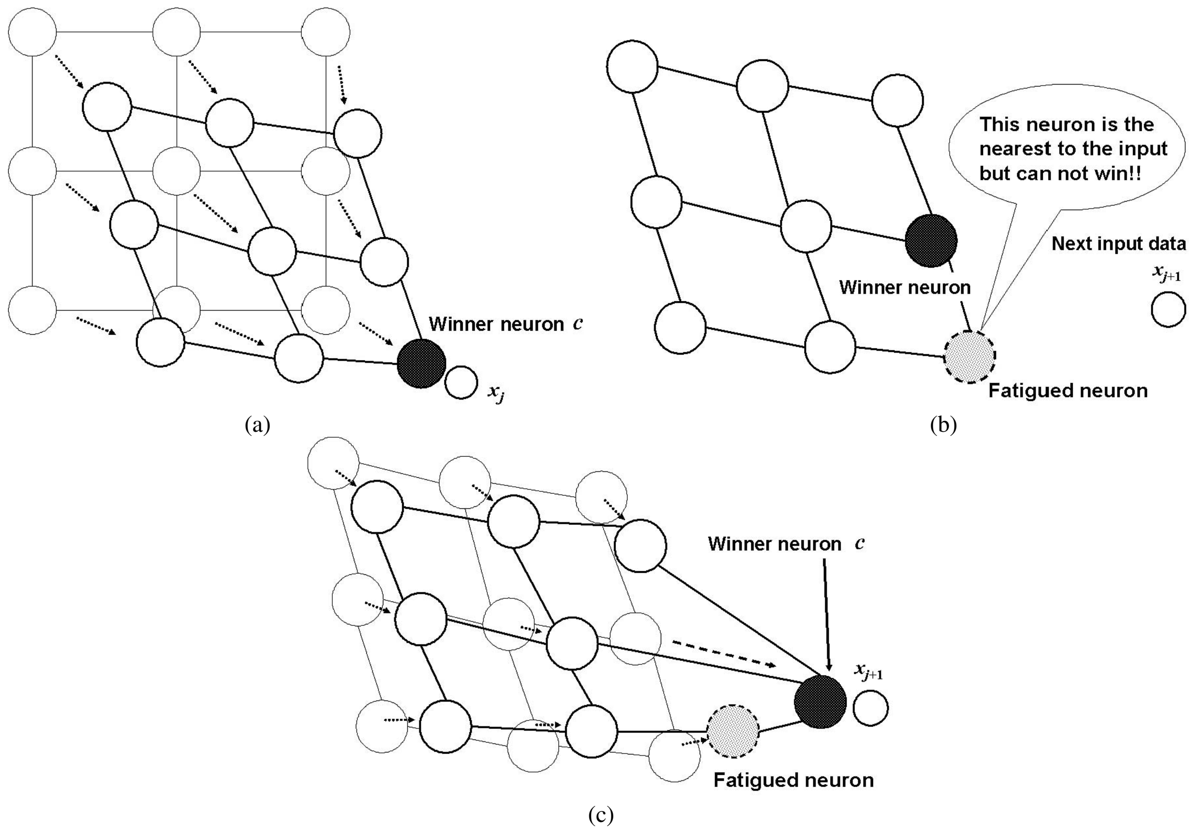


Fig. 3. How to determine a winner neuron c. (a) t = 1. (b) t = 2. (c) t = 3.

V. FSOM WITH FATIGUE LEVEL

In this section, in order to increase the flexibility of FSOM, we introduce a fatigue level to FSOM. The basic algorithm is the same as the FSOM learning algorithm and we add an additional index controlling update of the weight vectors.

We assign a variable p_i ($i = 1, 2, \cdots, m$) indicating the fatigue level to each neuron i of FSOM. This fatigue level p_i is updated according to the following rule. We set the initial value of p_i as 3.0. If a neuron i becomes a winner, p_i decreases 0.5. Further, p_i of all the neurons decrease according to;

$$p_i(t+1) = p_i(t) - 0.5\ h_{Fc,i}(t)(\boldsymbol{x}_j - \boldsymbol{w}_{F\,i}(t)). \qquad (12)$$

The value of p_i is reset to 3.0 when the fatigued neurons are recovered by (FSOM6).

The fatigue level influences the update of the weight vectors. This rule is added to realize the movement such that the neurons with lower fatigue levels move slower.

(Case 1) For $2 < p_i \leq 3$:
The weight vector of neuron i is updated according to;

$$\boldsymbol{w}_{\boldsymbol{F}\,i}(t+1) = \boldsymbol{w}_{\boldsymbol{F}\,i}(t) + 1.5\ h_{Fc,i}(t)(\boldsymbol{x}_j - \boldsymbol{w}_{\boldsymbol{F}\,i}(t)), \quad (13)$$

instead of Eq. (6).

(Case 2) For $1 < p_i \leq 2$:
The weight vector of neuron i is updated according to;

$$\boldsymbol{w}_{\boldsymbol{F}\,i}(t+1) = \boldsymbol{w}_{\boldsymbol{F}\,i}(t) + h_{Fc,i}(t)(\boldsymbol{x}_j - \boldsymbol{w}_{\boldsymbol{F}\,i}(t)), \quad (14)$$

which is the same update equation as Eq. (6).

(Case 3) For $0 < p_i \leq 1$:
The weight vector of neuron i is updated according to;

$$\boldsymbol{w}_{\boldsymbol{F}\,i}(t+1) = \boldsymbol{w}_{\boldsymbol{F}\,i}(t) + 0.5\ h_{Fc,i}(t)(\boldsymbol{x}_j - \boldsymbol{w}_{\boldsymbol{F}\,i}(t)), \quad (15)$$

instead of Eq. (6).

(Case 4) For $p_i \leq 0$:
The weight vector of neuron i is updated according to Eq. (14), where the initial values of β and σ are reset to $\beta(0)$=0.7 and $\sigma(0)$=4.0.

VI. SIMULATION RESULTS OF FSOM WITH FATIGUE LEVEL

In this section, we explain the behavior of FSOM with the fatigue level with some simulation results.

A. Example 1

Input data is 2-dimensional random data of 400 points whose distribution is non-uniform as Fig. 5. 200 points are distributed within a small range from 0.2 to 0.4 horizontally and from 0.2 to 0.4 vertically. The remaining 200 points are uniformly distributed within a small range from 0.6 to 0.8 horizontally and from 0.6 to 0.8 vertically. The parameters of SOM are the same as those in the simulation in the Section IV.

We execute the learning for all input data once. The simulation results of FSOM and FSOM with the fatigue level

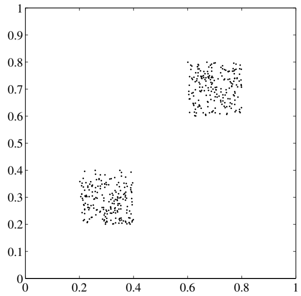


Fig. 5. 2-dimensional input data for Example 1.

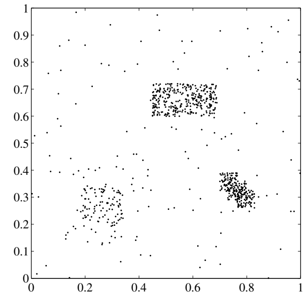
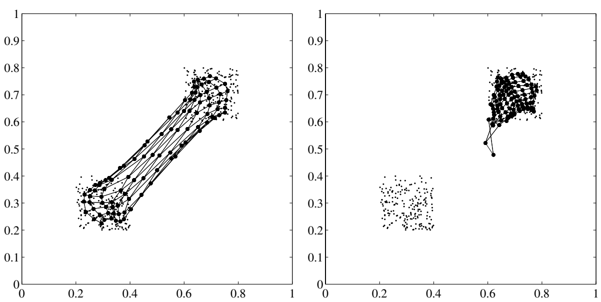


Fig. 7. 2-dimensional input data for Example 2.

are shown in Figs. 6(a) and (b), respectively. From the results, we explain the difference between FSOM and FSOM with the fatigue level. In the Section IV, we can see that FSOM successfully extracts only cluster. However, when input data is containing two similar clusters as Fig. 5, FSOM spread over the two clusters. However, by introducing the fatigue level, the neurons gather in one of the two clusters.



(a) (b)

Fig. 6. Learning results. (a) FSOM. (b) FSOM with the fatigue level.

B. Example 2

Next, we consider more complicated case that input data is 2-dimensional random data of 800 points whose distribution is non-uniform as Fig. 7. 320 points are distributed within a small range from 0.4 to 0.7 horizontally and from 0.6 to 0.75 vertically. 240 points are distributed within a small range from 0.7 to 0.85 horizontally and from 0.25 to 0.4 vertically. 160 points are distributed within a small range from 0.15 to 0.35 horizontally and from 0.2 to 0.35 vertically. The remaining 80 points are uniformly distributed between 0 and 1 at random. The parameters of SOM are the same as those in the simulation in the Section IV.

We execute the learning for all input data once. The simulation results are shown in Fig. 8. From the results, we can say that the conventional SOM tends to self-organize all the input data including a lot of noises and FSOM self-organizes some of the clusters. On the other hand, FSOM with the fatigue level self-organizes only one cluster and is not influenced by noises.

Because of the fatigue level, many neurons in the cluster become fatigued neurons easily. Further, the learning function of the neurons whose fatigue levels become negative is reset according to (Case 4). This means that the fatigued neurons in the cluster tend to tug other neurons. When plural clusters exist, the cluster in which more fatigued neurons exist can win the tug-of-war. Consequently, all of the neurons gather to the cluster where the number of the input data is the largest.

C. Application to Clustering

We carry out the extraction of cluster after Figs. 9(a), (b) and (c). The extraction method is the same as the Section IV.

The simulation results of the conventional SOM, FSOM and FSOM with the fatigue level are shown in Figs. 9(a), (b) and (c), respectively (R = 0.05). As we can see from the figures, the cluster obtained by the conventional SOM includes a lot of noises. In addition, FSOM obtains two clusters including similar number of input data and a few noises between two clusters. However, FSOM with the fatigue level can successfully extract only one cluster including the most input data. As a result, by using FSOM with the fatigue level for application to clustering, we can attained good results. The algorithm of FSOM with the fatigue level is a little bit complicated and it may be difficult to find good parameters for various kinds of input data. However, we consider that this can increase the flexibility of FSOM.

VII. Conclusions

In this study, we have proposed the Fatigable SOM (FSOM). We have explained the differences between SOM and FSOM and FSOM with the fatigue level with learning algorithm and have investigated its behavior. Furthermore, we have applied the proposed FSOM to extract only one of some clusters including a lot of noises and have confirmed its efficiency.

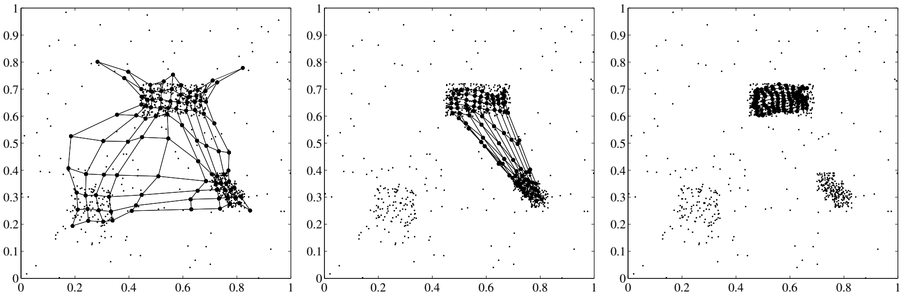
In the future, we try to discover new applications of FSOM in diverse fields such as sound data processing.

References

[1] T. Kohonen, "The Self-Organizing Maps," *Neurocomputing*, vol. 21, pp. 1-6, 1998.

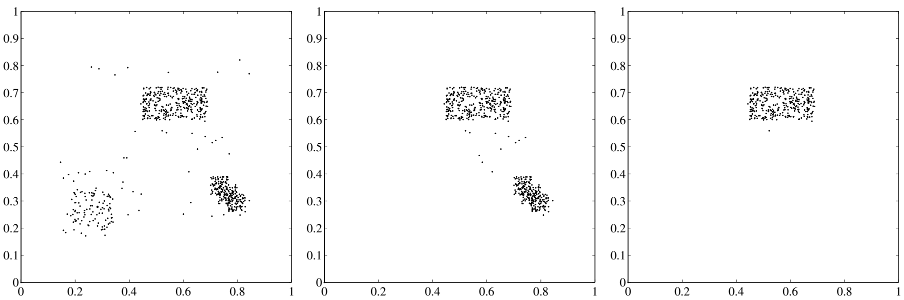
[2] T. Kohonen, *Self-Organizing Maps*, Berlin, Springer, vol. 30, 1995.

[3] Y. Cheng, "Clustering with Competing Self-Organizing Maps," *Proc. of IJCNN'92*, vol. IV, pp. 785-790, 1992.



(a) (b) (c)

Fig. 8. Learning results. (a) Conventional SOM. (b) FSOM. (c) FSOM with the fatigue level.



(a) (b) (c)

Fig. 9. Extraction of clusters. (a) Conventional SOM. (b) FSOM. (c) FSOM with the fatigue level.

[4] W. Wan and D. Fraser, “M2dSOMAP: Clustering and Classification of Remotely Sensed Imagery by Combining Multiple Kohonen Self-Organizing Maps and Associative Memory,” *Proc. of IJCNN’93*, vol. III, pp. 2464-2467, 1993.

[5] J. Vesanto and E. Alhoniemi, “Clustering of the Self-Organizing Map,” *IEEE Transactions on Neural Networks*, vol. 11, no. 3, pp. 586-600, 2000.

[6] C. Derek and E. Adrian, “Finding Curvilinear Features in Spatial Point Patterns: Principal Curve Clustering with Noise,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 6, pp. 601-609, 2000.

[7] P. Doucette, P. Agouris and A. Stefanidis, “Self-Organized Clustering for Road Extraction in Classified Imagery,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 55, Issues 5-6, pp. 347-358, 2001.

[8] M. Gervey and S. Lek, “Patterning and Clustering Ecological Assemblages,” *Proc. of WSOM’05*, pp. 9-16, 2005.

[9] H. Matsushita and Y. Nishio, “Competing Behavior of Two Kinds of SOMs and its Application to Clustering,” *Proc. of WSOM’05*, pp. 355-362, 2005.

[10] H. Matsushita and Y. Nishio, “Competing and Accommodating Behaviors of Peace SOM,” *Proc. of ISCAS’06*, 2006 (to appear).