Feedback Nonlinear Adaptive Manifold SOM for Hand Gesture Classification

Hideaki Kawano
Graduate School of Computer Science
and Systems Engineering
Kyushu Institute of Technology
Iizuka, Fukuoka 820-8502
Email: kawano-hideaki@edu.brain.kyutech.ac.jp

Keiichi Horio
and Takeshi Yamakawa
Graduate School of Life Science
and Systems Engineering
Kyushu Institute of Technology
Wakamatsu, Kitakyushu 808-0196
Email: horio@brain.kyutech.ac.jp,
yamakawa@brain.kyutech.ac.jp

***Abstract*— The feedback self-organizing map (SOM) is one of the most effective methods for applying the SOM to spatio-temporal pattern classification. In order to improve classification ability, the hybrid system of the nonlinear adaptive manifold SOM and feedback SOM is proposed in this paper. In the proposed method, each unit on the competitive layer is represented by nonlinear manifold in the input space. Thus the robust pattern classification can be achieved. The effectiveness of the proposed method is verified by applying it to a hand gesture classification problem.**

I. Introduction

Recently many researches that SOM is applied to spatio-temporal pattern classification have been reported[1]-[4]. The feedback SOM (FSOM), in which the feedback loops from the competitive layer to the input layer are employed, to embed temporal information into SOM was proposed by the authors[4]. In the FSOM, the output of the winner unit is set to 1, and those of the other units are done to 0. The outputs are fed back to the input layer through the feedback layer. The function of the units in the feedback layer is a leaky integrator. The current winner unit is assigned in consideration of the current input vector and the history of the winner units. Thus the temporal information can be embedded without using the tapped-delay or the hierarchical structure. The structure of the FSOM is very clear and simple, and the learning algorithm is the same to Kohonen's one. The classification results show that FSOM can successfully classify temporally expanded or contracted spatio-temporal patterns. However, the patterns which are spatially transformed can not be classified correctly because of limitation of the classification ability of the SOM.

The nonlinear adaptive manifold SOM (NAMSOM) was proposed by the authors in order to improve the classification ability of the SOM[5]. In the SOM, each unit is represented by one weight vector. In the NAMSOM, on the other hand, each unit is represented by a nonlinear manifold characterized by one mean vector and some basis vectors. The patterns belonging to the same class can be correctly classified by the NAMSOM even if the patterns are spatially transformed.

In this paper, the hybrid model of FSOM and NAMSOM, named feedback nonlinear adaptive manifold SOM (FNAMSOM), is proposed. The proposed FNAMSOM is expected to facilitate robust pattern classification, even when the spatio-temporal patterns are temporally expanded or contracted and spatially transformed. The effectiveness of the proposed FNAMSOM is verified by applying it to a hand gesture classification problem.

II. Feedback SOM

Fig.1 shows the structure of the FSOM. It consists of the input layer, the competitive layer and the feedback layer, in which $m+n$, n and n units are included, respectively. The j-th unit in the competitive layer is connected to all the units in the input layer by the weight vector $\boldsymbol{w}_j = [w_{j1}, \cdots, w_{jm}, w_{j,m+1}, \cdots, w_{j,m+n}]$. The outputs of the units in the feedback layer are defined as the feedback vector $\boldsymbol{h}(t) = [h_1(t), \cdots, h_n(t)]$ and each element is calculated by:

$$h_j(t) = z_j(t) + \gamma h_j(t-1), \tag{1}$$

where γ is a constant representing retention of the past outputs and $z_j(t)$ represents the output of the j-th unit in the competitive layer. The vector $[x_1(t), \cdots, x_m(t), h_1(t), \cdots, h_n(t)]$ obtained by combining the input vector $\boldsymbol{x}(t) = [x_1, \cdots, x_i, \cdots, x_m]$ with the feedback vector is referred to as the learning vector $\boldsymbol{I}(t)$. When the learning vector is applied to the input layer, the distance between the learning vector and the weight vector of the unit j is calculated by:

$$D_j = \sqrt{\sum_{i=1}^{m}(x_i(t) - w_{ji})^2 + \sum_{i=1}^{n}\eta(h_i(t) - w_{j,m+i})^2}, \tag{2}$$

where η is a constant that determines the relative importance of the feedback vector with respect to the input vector. A small η causes that the input vector is to be the dominant factor, while a large η causes the feedback vector to be the dominant factor. The unit in the competitive layer which has the minimum D_j is referred to as the winner unit c. The output $z_c(t)$ of the winner unit is set to 1, and those of the other units are done to 0. The units that are located within the neighborhood of the winner unit are referred to as the neighboring units. The weight vectors of the winner

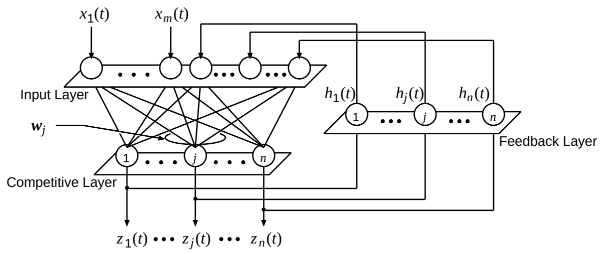

Fig. 1. The structure of the FSOM.

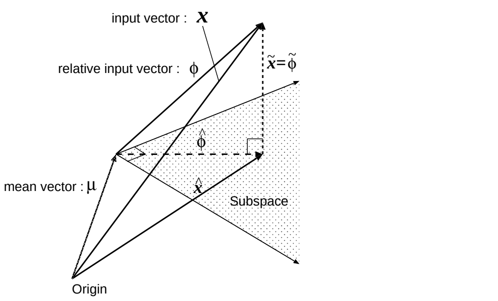

Fig. 2. A schematic interpretation for the orthogonal projection and the projection error of an input vector onto an affine subspace in a computational unit.

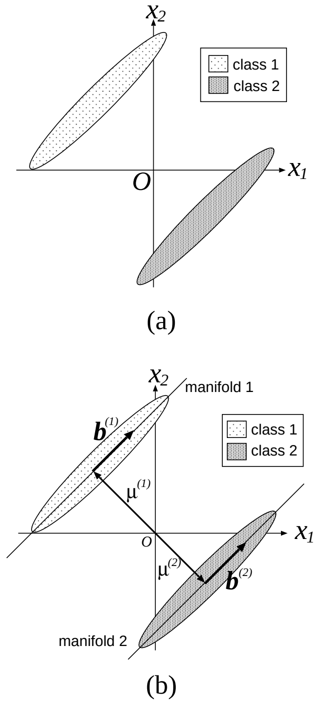

Fig. 3. (a) Clusters in 2-dimensional space: An example of the case which can not be separated without a mean value. (b) Two 1-dimensional affine subspaces to approximate and classify clusters.

and neighboring units are updated by:

$$\boldsymbol{w}_j(k+1) = \boldsymbol{w}_j(k) + \alpha_{\text{FSOM}}(k) h_{cj}(k)(\boldsymbol{I}(t) - \boldsymbol{w}_j(k)), \quad (3)$$

where $\boldsymbol{w}_j(k+1)$ and $\boldsymbol{w}_j(k)$ are weight vectors after and before the updating, respectively. $\alpha_{\text{FSOM}}(k)$ is a learning rate at the learning step k, which decreases with learning steps. Also $h_{cj}(k)$ is learning rate which is based on the distance between the winner unit and the unit to be updated. It is called neighboring function. The learning algorithm of the FSOM is the same to Kohonen's one.

After learning, sequences of input vectors $\{\boldsymbol{x}(t) | t = 0, \cdots, T\}$, *i.e.* spatio-temporal patterns, are applied to the input layer one after the other. The itinerancy of the winner units, which depends on the spatial and temporal information of the input vectors, can be observed. The winner unit at the end of the sequence represents the class of the input pattern.

It is confirmed that spatio-temporal patterns which are temporally expanded or contracted can be correctly classified by the FSOM. However the patterns which are spatially transformed, such as shifted, rotated and scaled, can not be classified correctly. The SOM can not also classify such patterns. In order to improve the classification ability of the FSOM we employ the NAMSOM.

III. Feedback Adaptive Nonlinear Manifold SOM

The NAMSOM is an extension of adaptive manifold SOM (AMSOM) by using kernel method. In this section the AMSOM and NAMSOM are explained at first, and the proposed FNAMSOM is described.

A. AMSOM

The AMSOM is an extension of the SOM in which representation of the the unit in the competitive layer is modified from one weight vector to affine subspace[6]. The affine subspace of the j-th unit in the competitive layer is composed of one mean vector $\boldsymbol{\mu}^{(j)}$ and a subspace spanned by H basis vectors $\boldsymbol{b}_h^{(j)}$, $h \in \{1, \cdots, H\}$.

The orthogonal projection of an input vector $\boldsymbol{x}$ onto the affine subspace of j-th unit is calculated by:

$$\hat{\boldsymbol{x}}^{(j)} = \boldsymbol{\mu}^{(j)} + \sum_{h=1}^{H} (\boldsymbol{\phi}^{(j)T} \boldsymbol{b}_h^{(j)}) \boldsymbol{b}_h^{(j)}, \quad (4)$$

where $\boldsymbol{\phi}^{(j)} = \boldsymbol{x} - \boldsymbol{\mu}^{(j)}$. Therefore the projection error is represented as

$$\tilde{\boldsymbol{x}}^{(j)} = \boldsymbol{\phi}^{(j)} - \sum_{h=1}^{H} (\boldsymbol{\phi}^{(j)T} \boldsymbol{b}_h^{(j)}) \boldsymbol{b}_h^{(j)}. \quad (5)$$

Fig.2 shows a schematic interpretation for the orthogonal projection and the projection error of an input vector onto the affine subspace defined in the j-th unit. The AMSOM is more general strategy than the adaptive subspace SOM (ASSOM), where each computational unit solely defines a subspace. To illustrate why this is so, let us consider a very simple case: Suppose two clusters as shown in Fig.3(a) are given. It is not possible to use one dimensional subspaces, that is lines intersecting the origin $\boldsymbol{O}$, to approximate the clusters. This is true even if the global mean is removed, so that the origin O is translated to the centroid of the two clusters. However, two one-dimensional affine subspaces can easily approximate the clusters as shown in Fig.3(b), since the basis vectors are aligned in the direction that minimizes the projection error.

In the AMSOM, the input vectors are grouped into some episodes in order to apply them to the network as an input sets. For pattern classification, an episode input is defined as a subset of training data belonging to the same category. Assume that the number of input vectors in the subset is E, then an episode input ω_q in the class q is denoted as $\omega_q = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_E\}, \omega_q \subseteq \Omega_q$, where Ω_q is a set of training patterns belonging to the class q. The set of input vectors of an episode has to be recognized as one class, such that any member of this set and even their arbitrary linear combination should have the same winning unit.

The training process in AMSOM has the following steps:

(a) Winner Lookup

The unit that gives the minimum projection error for an episode is selected. The unit is denoted as the winner, whose index is c. This decision criterion for the winner c is represented as

$$c = \arg\min_j \left\{ \sum_{e=1}^{E} ||\tilde{\boldsymbol{x}}_e^{(j)}||^2 \right\}, \tag{6}$$

where $j \in \{1, \cdots, n\}$.

(b) Updating

The mean vectors of the winner and neighboring units are updated by:

$$\begin{aligned} \boldsymbol{\mu}^{(j)}(k+1) &= \boldsymbol{\mu}^{(j)}(k) \\ &+ \lambda_m(k) h_{cj}(k) \left(\boldsymbol{x}_e - \boldsymbol{\mu}^{(j)}(k) \right), \end{aligned} \tag{7}$$

where $\lambda_m(k)$ is the learning rate for $\boldsymbol{\mu}^{(j)}$ at learning step k, and $h_{cj}(k)$ is the neighborhood function at learning step k with respect to distance between the winner unit c and the unit j. Both $\lambda_m(k)$ and $h_{cj}(k)$ are monotonic decreasing function with respect to k. In this paper, $\lambda_m(k) = \lambda_m^{ini}(\lambda_m^{fin}/\lambda_m^{fin})^{k/k_{max}}$ and $h_{cj}(k) = \exp(-|c-j|/\gamma(k))$ and used, where $\gamma(k) = \gamma^{ini}(\gamma^{fin}/\gamma^{fin})^{k/k_{max}}$. The basis vectors of the winner and neighboring units are updated by

$$\begin{aligned} &\boldsymbol{b}_h^{(j)}(k+1) = \boldsymbol{b}_h^{(j)}(k) \\ &+ \lambda_b(k) h_{cj}(k) \frac{\boldsymbol{\phi}_e^{(j)}(k)^T \boldsymbol{b}_h^{(j)}(k)}{||\hat{\boldsymbol{\phi}}_e^{(j)}(k)|| ||\boldsymbol{\phi}_e^{(j)}(k)||} \boldsymbol{\phi}_e^{(j)}(k), \end{aligned} \tag{8}$$

where $\boldsymbol{\phi}_e^{(j)}(k)$ is the relative input vector in the manifold j updated the mean vector, which is represented by $\boldsymbol{\phi}_e^{(j)}(k) = \boldsymbol{x}_e - \boldsymbol{\mu}^{(j)}(k+1)$, $\hat{\boldsymbol{\phi}}_e^{(j)}(k)$ is the orthogonal projection of the relative input vector, which is represented by $\hat{\boldsymbol{\phi}}_e^{(j)}(k) = \sum_{h=1}^{H}(\boldsymbol{\phi}^{(j)}(k)^T \boldsymbol{b}_h^{(j)}(k))\boldsymbol{b}_h^{(j)}(k)$ and $\lambda_b(k)$ is the learning rate for the basis vectors, which is also monotonic decreasing function with respect to k. In this paper, $\lambda_b(k) = \lambda_b^{ini}(\lambda_b^{fin}/\lambda_b^{fin})^{k/k_{max}}$ is used.

After the learning phase, a categorization phase to determine the class association of each unit has to be achieved. Input data used for learning is applied to the AMSOM again. Then the unit j is labeled by the most major class of the input data which defines the unit j as the winner. It could be considered that some neighboring units are associated with the same class.

Each unit is labeled by the class index for which is selected as the winner most frequently when the input data for learning are applied to the AMSOM again.

B. NAMSOM

1) Reproducing Kernels: Reproducing kernels are functions $k : \mathcal{X}^2 \mapsto \Re$ which for all pattern sets

$$\{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_l, \cdots, \boldsymbol{x}_L\} \subset \mathcal{X} \tag{9}$$

produce positive matrices $K_{pq} := k(\boldsymbol{x}_p, \boldsymbol{x}_q)$. Here, $\mathcal{X}$ is some compact set in which the data resides, typically a subset of $\Re^n$. In the field of Support Vector Machine (SVM), reproducing kernels are often referred to as *Mercer kernels*. They provide an elegant way of dealing with nonlinear algorithms by reducing them to linear ones in some feature space $\mathcal{F}$ nonlinearly related to input space: Using k instead of a dot product in $\Re^n$ corresponds to mapping the data into a possibly high-dimensional dot product space $\mathcal{F}$ by a (usually nonlinear) map $\Phi : \Re^n \mapsto \mathcal{F}$, and taking the dot product there, *i.e.*[7]

$$k(\boldsymbol{x}, \boldsymbol{y}) = (\Phi(\boldsymbol{x}), \Phi(\boldsymbol{y})). \tag{10}$$

By virtue of this property, a map Φ is called as a *feature map associated with* k. Any linear algorithm which can be carried out in terms of dot products can be made nonlinear by substituting a priori chosen kernel. Examples of such algorithms include the potential function method[8], SVM [9][10] and kernel PCA[11]. The price that one has to pay for this elegance, however, is that the solutions are only obtained as expansions in terms of input patterns mapped into feature space. For instance, the normal vector of an SV hyperplane is expanded in terms of Support Vectors, just as the kernel PCA feature extractors are expressed in terms of training examples,

$$\Psi = \sum_{l=1}^{L} \alpha_l \Phi(\boldsymbol{x}_l). \tag{11}$$

2) AMSOM in the feature space: The AMSOM in the high-dimensional feature space $\mathcal{F}$ is considered. The method is referred to as Nonlinear Adaptive Manifold Self-Organizing Map (NAMSOM). In NAMSOM, an affine subspace defined by the unit j in the competitive layer take the nonlinear form

$$\mathcal{M}_j = \{\Phi(\boldsymbol{x}) | \Phi(\boldsymbol{x}) = \Phi(\boldsymbol{\mu}^{(j)}) + \sum_{h=1}^{H} \xi \Phi(\boldsymbol{b}_h^{(j)})\}, \tag{12}$$

where $\xi \in \Re$. Given training data set $\{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_L\}$, the mean vector and the basis vector in the unit j are represented by the following form

$$\Phi(\boldsymbol{\mu}^{(j)}) = \sum_{l=1}^{L} \alpha_l^{(j)} \Phi(\boldsymbol{x}_l), \tag{13}$$

$$\Phi(\boldsymbol{b}_h^{(j)}) = \sum_{l=1}^{L} \beta_{hl}^{(j)} \Phi(\boldsymbol{x}_l), \tag{14}$$

respectively. $\alpha_l^{(j)}$ in Eq.(13) and $\beta_{hl}^{(j)}$ in Eq.(14) are the parameters adjusted by learning.

The derivation of training procedure in NAMSOM is given as follows:

(a) Winner lookup

The norm of the orthogonal projection error onto the j-th nonlinear affine subspace with respect to present input $\boldsymbol{x}_p$ is calculated as follows:

$$\begin{aligned} ||\Phi(\tilde{\boldsymbol{x}}_p)^{(j)}||^2 &= k(\boldsymbol{x}_p,\boldsymbol{x}_p) + \sum_{h=1}^{H} {P_h^{(j)}}^2 \\ &+ \sum_{l_1=1}^{L}\sum_{l_2=1}^{L} \alpha_{l_1}^{(j)}\alpha_{l_2}^{(j)} k(\boldsymbol{x}_{l_1},\boldsymbol{x}_{l_2}) \\ &- 2\sum_{l=1}^{L}\alpha_l k(\boldsymbol{x},\boldsymbol{x}_l) \\ &+ 2\sum_{h=1}^{H}\sum_{l_1=1}^{L}\sum_{l_2=1}^{L} P_h^{(j)}\alpha_{l_1}\beta_{hl_2}k(\boldsymbol{x}_{l_1},\boldsymbol{x}_{l_2}) \\ &- 2\sum_{h=1}^{H}\sum_{l=1}^{L} P_h^{(j)}\beta_{hl}k(\boldsymbol{x},\boldsymbol{x}_l), \end{aligned} \quad (15)$$

where $P_h^{(j)}$ means the orthogonal projection component of present input $\boldsymbol{x}_p$ into the basis $\Phi(\boldsymbol{b}_h^{(j)})$ and it is calculated by

$$P_h^{(j)} = \sum_{l=1}^{N}\beta_{hl}^{(j)}k(\boldsymbol{x}_p,\boldsymbol{x}_l) - \sum_{l_1=1}^{N}\sum_{l_2=1}^{N}\alpha_{l_1}^{(j)}\beta_{hl_2}^{(j)}k(\boldsymbol{x}_{l_1},\boldsymbol{x}_{l_2}). \quad (16)$$

The reproducing kernels used in general applications are as follows:

$$k(\boldsymbol{x},\boldsymbol{y}) = (\boldsymbol{x}^T\boldsymbol{y})^d \qquad d \in \aleph, \quad (17)$$

$$k(\boldsymbol{x},\boldsymbol{y}) = (\boldsymbol{x}^T\boldsymbol{y}+1)^d \qquad d \in \aleph, \quad (18)$$

$$k(\boldsymbol{x},\boldsymbol{y}) = \exp\left(-\frac{||\boldsymbol{x}-\boldsymbol{y}||^2}{2\sigma^2}\right) \qquad \sigma \in \Re, \quad (19)$$

where $\aleph$ and $\Re$ are the set of natural numbers and the set of reals, respectively. Eq.(14), Eq.(15) and Eq.(16) are referred as to homogeneous polynomial kernels, non-homogeneous polynomial kernels and gaussian kernels, respectively.

The winner unit for an episode input $\omega_q = \{\Phi(\boldsymbol{x}_1),\cdots,\Phi(\boldsymbol{x}_E)\}$ is decided by the same manner as the AMSOM as follows:

$$c = \arg\min_j \left\{ \sum_{e=1}^{E} ||\Phi(\tilde{\boldsymbol{x}}_e)^{(j)}||^2 \right\},\ j \in \{1,\cdots,n\}. \quad (20)$$

(b) Updating

The learning rule for $\alpha_l^{(j)}$ and $\beta_{hl}^{(j)}$ are as follows:

$$\Delta\alpha_l^{(j)} = \begin{cases} -\alpha_l^{(j)}(t)\lambda_m(k)h_{ci}(k) & for\ \ l \neq e \\ -\alpha_l^{(j)}(k)\lambda_m(k)h_{cj}(k) + \lambda_m(k)h_{cj}(k) & for\ \ l = e \end{cases}, \quad (21)$$

$$\Delta\beta_{hl}^{(j)} = \begin{cases} -\alpha_l^{(j)}(k+1)\lambda_b(k)h_{cj}(k)T_h^{(j)}(k) & for\ \ l \neq e \\ -\alpha_l^{(j)}(k+1)\lambda_b(k)h_{cj}(k)T(k) + \lambda_b(k)h_{cj}(k)T(k) & for\ \ l = e \end{cases}, \quad (22)$$

where

$$T(t) = \frac{\Phi(\boldsymbol{\phi}_e^{(j)}(k))^T\Phi(\boldsymbol{b}_h^{(j)}(k))}{||\Phi(\hat{\boldsymbol{\phi}}_e^{(j)}(k))||\,||\Phi(\boldsymbol{\phi}_e^{(j)}(k))||}, \quad (23)$$

$$\|\Phi(\hat{\boldsymbol{\phi}}_e^{(j)}(k)))\| = \left\{ \sum_{h=1}^{H}\Big[\sum_{l=1}^{L}\beta_{hl}^{(j)}k(\boldsymbol{x}_e,\boldsymbol{x}_l) - \sum_{l_1=1}^{L}\sum_{l_2=1}^{L}\alpha_{l_1}\beta_{kl_2}k(\boldsymbol{x}_{l_1},\boldsymbol{x}_{l_2})\Big]^2 \right\}^{\frac{1}{2}}, \quad (24)$$

$$\|\Phi(\boldsymbol{\phi}_e^{(j)})(k)\| = \left\{ k(\boldsymbol{x}_e,\boldsymbol{x}_e) - 2\sum_{l=1}^{L}\alpha_l^{(j)}k(\boldsymbol{x}_e,\boldsymbol{x}_l) + \sum_{l_1=1}^{L}\sum_{l_2=1}^{L}\alpha_{l_1}^{(j)}\alpha_{l_2}^{(j)}k(\boldsymbol{x}_{l_1},\boldsymbol{x}_{l_2}) \right\}^{\frac{1}{2}}, \quad (25)$$

$$\Phi(\boldsymbol{\phi}_e^{(j)}(k))^T\Phi(\boldsymbol{b}_h^{(j)}(k)) = \sum_{l=1}^{L}\beta_{hl}k(\boldsymbol{x}_e,\boldsymbol{x}_l) - \sum_{l_1=1}^{L}\sum_{l_2=1}^{L}\alpha_{l_1}^{(j)}\beta_{hl_2}^{(j)}k(\boldsymbol{x}_{l_1},\boldsymbol{x}_{l_2}). \quad (26)$$

In Eqs.(21) and (22), $\lambda_m(k)$, $\lambda_b(k)$ and $h_{cj}(k)$ are the same parameters as mentioned in the AMSOM training process.

After the learning phase, a categorization phase to determine the class association of each unit has to be achieved. The procedure of the categorization phase is done in the same manner as mentioned in previous section.

C. Proposed Method

In order to improve the classification ability of the FSOM, the NAMSOM is embedded to the FSOM. Basic algorithm of the FSOM is not changed, however only the method for definition of the winner unit is modified. In the FNAMSOM, the winner c units is assigned by:

$$c = \arg\min_j \left\{ \|\Phi(\tilde{\boldsymbol{x}}(t))^{(j)}\|^2 + \sum_{i=1}^{n}\eta(h_i(t) - w_{j,m+i})^2 \right\}. \quad (27)$$

where, $\Phi(\tilde{\boldsymbol{x}}(t))^{(j)}$ is the orthogonal projection error onto the j-th nonlinear affine subspace with respect to present input $\boldsymbol{x}(t)$.

IV. Experimental Results

The proposed FNAMSOM is applied to the hand gesture classification problem in order to verify its effectiveness. Fig.4 shows the examples of the hand gestures used in the experiments. These data is obtained from Sebastien Marcel's Gesture Database Web Page[12]. 3 spatio-temporal patterns are included in each class. The lengths of all the sequences t are adjusted to 30. Each image is down-sampled to 20×20. Four spatio-temporal patterns (one from each class) are used for learning of the FNAMSOM. In the learning, The number of the units is 15 and each unit is represented by 2-dimensional nonlinear affine subspace. Gaussian kernel shown in Eq.(19) is used with $\sigma = 0.01$. $\gamma = 0.5$, $\eta = 0.01$. These four spatio-temporal patterns are applied to the FNAMSOM after learning

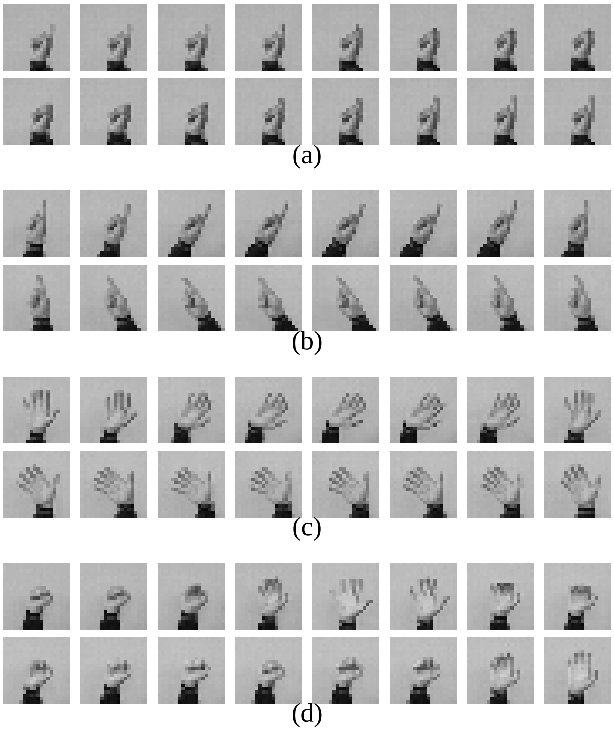

Fig. 4. Some examples of hand gestures used in the experiments. (a) Class 1: 'Clic', (b) Class 2: 'No', (c) Class 3: 'Rotate', and (d) Class 4: 'StopGraspOk'.

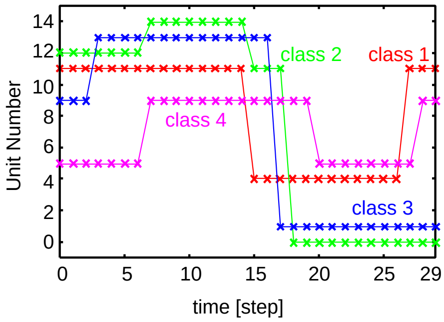

Fig. 5. The sequence of the winner units for four learning patterns.

again. Fig.5 shows the sequences of the winner units for the patterns. The four patterns are successfully classified (unit 1 for class 1, unit 5 for class 2, unit 10 for class 3, and unit 15 for class 4). The final postures of the class 1 and 2 are very similar but the different units are defined as the winner units, because the winner units for the final postures are decided in consideration of current postures and the sequences of the past winner units. Fig.6 shows the sequences of the winner units for the testing patterns. The sequences for the patterns belonging to the same classes is very similar and the classification is correctly achieved. When the ordinary FSOM and the FSOM with AMSOM are applied to this hand gesture problem, the patterns can not be classified. It means that the classification ability of the FSOM can be improved by employing the concept of the NAMSOM.

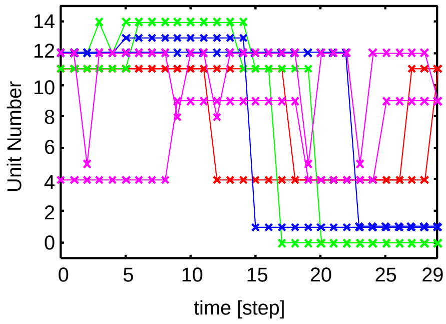

Fig. 6. The sequence of the winner units for testing patterns.

V. Conclusions

In this paper, the FNAMSOM which is the hybrid system of the FSOM and NAMSOM is proposed. The proposed NAMSOM has an ability of robust classification even when the spatio-temporal patterns are temporally expanded or contracted and spatially transformed. The FNAMSOM is applied to a hand gesture problem and the improvement of the classification ability can be verified.

References

[1] N. R. Euliano and J. C. Principe, "Spatio-temporal self-organizing feature maps," *Proc. of the IEEE Int. Conf. on Neural Network '96*, 4, 1900-1905, 1996.

[2] M. Varsta, J. Milan and J. Heikkonen, "A recurrent self-organizing map for temporal sequence processing," *Proc. of the Int. Conf. on Artificial Neural Networks*, 421-426, 1997.

[3] T. Koskela, M. Varsta, J. Heikkonen and K. Kaski, "Temporal sequence processing using recurrent SOM," *Proc. of the Second Int. Conf. on Knowledge Based Intelligent Electronics Systems*, 290-297, 1998.

[4] K. Horio and T. Yamakawa, "Feedback Self-Organizing Map and Its Application to Spatio-Temporal Pattern Classification," *Int. J. of Computational Intelligence and Applications*, 1, 1-18, 2001.

[5] H. Kawano, T. Yamakawa and K. Horio, "Kernel-Based Adaptive-Subspace Self-Organizing Map as a Nonlinear Subspace Pattern Recognition," *Proc. of the World Automation Congress*, in CD-ROM, 2004.

[6] Z. Q. Liu, "Adaptive Subspace Self-Organizing Map and Its Application in Face Recognition", *International Journal of Image and Graphics*, Vol.2, No.4, pp.519-540, (2002)

[7] B.E. Boser, I.M. Guyon and V.N. Vapnik, "A Training Algorithm for Optimal Margin Classifier," *Proc. of the 5th Annual ACM Workshop on Computational Learning Theory*, 144-152, 1992.

[8] M. Aizerman, E. Braverman and L. Rozonoer, "Theoretical Foundations of the Potential Function Method in Pattern Recognition Learning," *Automation and Remote Control*, 25, 821-837, 1964.

[9] C. Cortes and V. Vapnik, "Support Vector Networks," *Machine Learning*, 20, 273-297, 1995.

[10] V. Vapnik, *The Nature of Statistical Learning Theory*, Spring-Verlag, 1995.

[11] B. Scholkopf, A. J. Smola and K. R. Muler,"Nonlinear Component Analysis as a Kernel Eigenvalue Problem," *Technical Report 44, Max-Planck-Institut fur biologische Kybernetik*, 1996.

[12] Sebastien Marcel's Gesture Database Web Page, http://www.idiap.ch/˜marcel/Databases/main.html