Regular Paper

Motion Feature Extraction Using Second-order Neural Network and Self-organizing Map for Gesture Recognition

Masato Aoba† and Yoshiyasu Takefuji††

We propose a neural preprocess approach for video-based gesture recognition system. Second-order neural network (SONN) and self-organizing map (SOM) are employed for extracting moving hand regions and for normalizing motion features respectively. The SONN is more robust to noise than frame difference technique. Obtained velocity feature vectors are translated into normalized feature space by the SOM with keeping their topology, and the transition of the activated node in the topological map is classified by DP matching. The topological nature of the SOM is quite suited to data normalization for the DP matching technique. Experimental results show that those neural networks effectively work on the gesture pattern recognition. The SONN shows its noise reduction ability for noisy backgrounds, and the SOM provides the robustness to spatial scaling of input images. The robustness of the SOM to spatial scaling is based on its robustness to velocity scaling.

1. Introduction

Recent innovation in the area of computers enables us to utilize more advanced electronic devices in our lives, and the importance of human-computer interaction (HCI) has been increasing. When we operate a computer with the keyboard and/or the mouse, they do not necessarily fulfill our demand. As we encounter to many scenes interacting with computers in our daily lives, various kinds of HCI devices such as remote control, touch-sensitive panel, voice recognition and motion recognition *etc.* have been developed [1),2)]. One of the effective ways for the motion recognition is to use hand gestures. Using hand gestures is a common way for communications between human and human, therefore the hand gesture recognition system has a potential to be a useful HCI tool.

The video-based gesture recognition includes time sequence analysis. Hidden Markov model (HMM) is a major method for recognizing gesture patterns [3)~5)]. Although the HMM is effective for recognizing sequential patterns, it needs many training data for the parameter tuning. Sagawa, et al. [6)] and Osaki, et al. [7)] employed dynamic programming (DP) matching for their gesture recognition systems. The DP matching shows a good performance to classify small scale sequential patterns and needs no complex algorithm to adjust its control parameters. On the other hands, artificial neural network models were embedded in some gesture recognition systems. Ng, et al. applied RBF network to classify hand shapes and used combination of recurrent neural network and HMM to recognize the changes in the hand shape [8)]. Lamar, et al. proposed T-CombNET for temporal series recognition and they applied it to hand gesture recognition system [9)].

In video based gesture recognition system, motion feature extraction is much effective on its recognition performance. Some researches have utilized frame difference or background subtraction for extracting moving objects [5),10)], and optical flow is also a popular method for motion segmentation [11)]. Extracting skin color regions is an effective way for hand gesture recognition [4),12)]. Some neural models have been proposed for motion extraction as prototypes [13),14)], however few real time software approaches have been proposed for video based gesture recognition system. Yoshiike, et al. reported that maximum neural network was effective to the noiseless motion extraction in gesture recognition [15)].

In this paper, we propose a neural preprocess approach for video-based gesture recognition system using two neural network models; second-order neural network (SONN) for extracting moving hand regions, and self-organizing map (SOM) for normalizing motion features. Time sequential motion feature pattern is classified by DP matching. Chashikawa, et al. reported that second-order neural network (SONN) has robustness to noise in ex-

† Takefuji Laboratory, Keio Research Institute at SFC
†† Faculty of Environmental Information, Keio University

tracting moving objects [16]. We employ this model for moving hand region extraction. The SOM is a well known neural network model introduced by Kohonen [17] and it translates feature vectors into another feature space with keeping its topology and data distribution. For motion recognition, obtained velocity feature vectors are translated into normalized feature space represented as the topological map, and a trajectory on this map is recognized as a time sequential pattern. This is quite suited to the DP matching technique since the distance on the topological map approximates the probabilistic distance in the original feature space.

We applied those ideas for recognizing twelve hand gestures. Experimental results show that the proposed system effectively works on the gesture pattern recognition. The SONN shows its noise reduction ability for noisy backgrounds, and the SOM provides the robustness to spatial scaling of input images. The robustness of the SOM to spatial scaling is based on its robustness to velocity scaling.

2. System Overview

We design a system to recognize velocity sequences of hand gestures. The overview of our system is shown in **Fig. 1**.

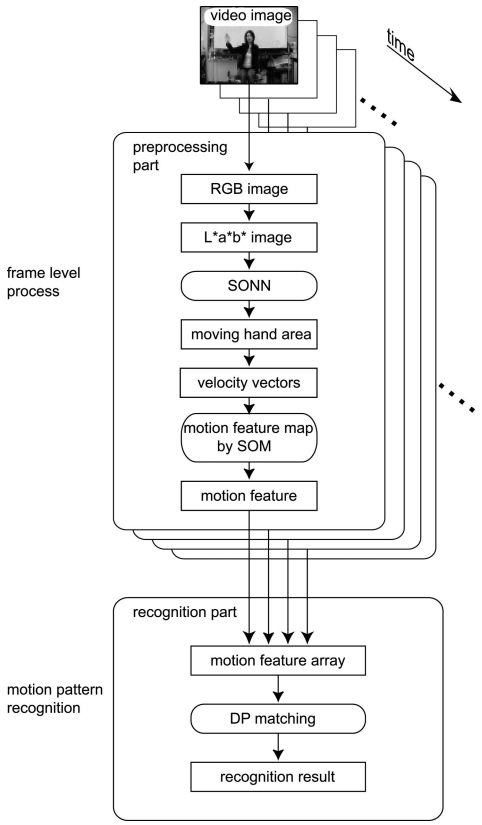

Fig. 1 Overview of the system.

In frame level process, preprocessing part extracts motion features from video images. RGB video images are captured by a video camera and are translated into $L^*a^*b^*$ images. Moving hand regions are extracted by SONN. Then velocity vector is calculated as the change of the gravity points on the moving hand region between two frames, and it is translated into motion feature by motion feature map (MFM) trained by SOM. The system feeds the motion features in time order as motion feature array throughout a gesture. The motion feature array is classified by DP matching and the system generates the recognition results.

3. Motion Feature Extraction Using SONN and SOM

3.1 Moving Hand Extraction

Input data to the preprocessing part are originally obtained by a video camera as a sequence of RGB images. The RGB colors in the video images are translated into $L^*a^*b^*$ color space in order to extract a moving hand-region. L^* represents the luminance while a^* and b^* represent hue. To reduce the effect of luminance, the preprocessing part should be more sensitive to a^* and b^* than L^*.

In order to track the moving hand, moving region is extracted from each frame. However, frame difference technique and background subtraction method [18),19] are simple and feasible, they are easily affected by noises. In order to overcome the noise problem, Chashikawa and Takefuji proposed that the second-order neural network (SONN) is effective for extracting moving object [16]. They demonstrated that the SONN is more robust to temporal Gaussian noise and generates more stable output for a blank wall problem [24] than the frame difference technique does. Their model has a similar structure to pulse coupled neural network (PCNN), which models a cat visual cortex and was applied to some static image processing [20),21]. Chashikawa has attached a feedforward shunting mechanism [22] and static threshold to PCNN structure for time sequential image processing. We improved that model in order to handle $L^*a^*b^*$ images for moving hand region extraction. The structure of the SONN for moving hand region extraction is shown in **Fig. 2**.

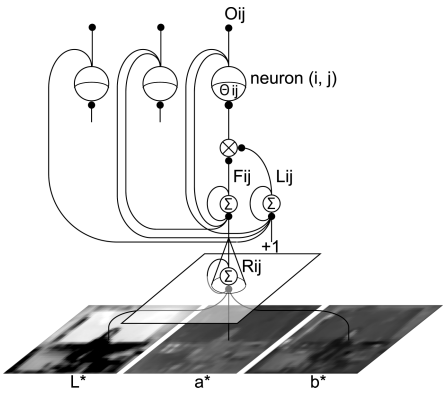

Fig. 2 Structure of SONN for moving hand region extraction.

The binary output $O_{ij}(t)$ at time t corresponding to the pixel (i, j) is calculated as follows,

$$O_{ij}(t) = \begin{cases} 1 & \text{if } U_{ij}(t) \geq \Theta_{ij}(t) \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where U_{ij} is the internal activity and Θ_{ij} is the dynamic threshold. The dynamic threshold Θ_{ij} is defined by,

$$\Theta_{ij} = \theta_o\left(1 + \xi \sum_{i,j} U_{ij}(t)/(l_h \times l_w)\right) \quad (2)$$

where θ_o and ξ are the constant parameters. l_h and l_w represent the image height and the image width respectively. The internal activity $U_{ij}(t)$ at time t is given by,

$$U_{ij}(t) = F_{ij}(t)\big(1 + \beta L_{ij}(t)\big) \quad (3)$$

where $F_{ij}(t)$ is the feeding signal, $L_{ij}(t)$ is the linking signal and β is the constant for the linking strength. The feeding signal $F_{ij}(t)$ and the linking signal $L_{ij}(t)$ are written by,

$$\begin{aligned} F_{ij}(t) = \gamma_F &\sum_{k,j} W^F_{ijkl} O_{kl}(t-1) \\ &+ \sum_{k,j} W^R_{ijkl} R_{kl}(t) \\ &+ \exp(-\tau_F) F_{ij}(t-1) \end{aligned} \quad (4)$$

$$\begin{aligned} L_{ij}(t) = \gamma_L &\sum_{k,j} W^L_{ijkl}\big(O_{kl}(t-1) - 1\big) \\ &+ \exp(-\tau_L) L_{ij}(t-1) \end{aligned} \quad (5)$$

where τ_F and τ_L are the attenuation constants for the feeding and the linking signals. W^F_{ijkl}, W^L_{ijkl} and W^R_{ijkl} are the connection weights, which are Gaussian kernels centered around pixel (i, j). γ_F and γ_L are the normalization constants for the connection weights. Note that $O_{kl}(t-1)$ is the feedback signal from the binary output. $R_{ij}(t)$ is the transient response which is described as,

$$R_{ij}(t) = \gamma_R\big(S_{ij}(t) + \exp(-\tau_R) R_{ij}(t-1)\big) \quad (6)$$

where $S_{ij}(t)$ is the input stimuli, τ_R is the attenuation constant for the input stimuli, and γ_R is the normalization constant. The input stimuli $S_{ij}(t)$ is given by,

$$S_{ij}(t) = \frac{D^{L^*}_{ij}(t) + D^{a^*}_{ij}(t) + D^{b^*}_{ij}(t)}{3} \quad (7)$$

$$D^{L^*}_{ij} = C_{L^*} |I^{L^*}_{ij}(t) - I^{L^*}_{ij}(t-1)| \quad (8)$$

$$\begin{aligned} D^{a^*}_{ij} = &\exp\left(-\frac{\big(I^{a^*}_{ij}(t) - m_{a^*}\big)^2}{\sigma^2_{a^*}}\right) \\ &\times |I^{a^*}_{ij}(t) - I^{a^*}_{ij}(t-1)| \end{aligned} \quad (9)$$

$$\begin{aligned} D^{b^*}_{ij} = &\exp\left(-\frac{\big(I^{b^*}_{ij}(t) - m_{b^*}\big)^2}{\sigma^2_{b^*}}\right) \\ &\times |I^{b^*}_{ij}(t) - I^{b^*}_{ij}(t-1)| \end{aligned} \quad (10)$$

where $I^{L^*}_{ij}(t)$, $I^{a^*}_{ij}(t)$ and $I^{b^*}_{ij}(t)$ are the input value at pixel (i, j) for L^*, a^* and b^* respectively. C_{L^*}, m_{a^*}, m_{b^*}, σ_{a^*} and σ_{b^*} are the constants to define the sensitivity to skin color.

Fig. 3 Example of moving hand region extraction.

An example of hand gestures is shown in **Fig. 3** and the white pixels indicate the region extracted by SONN. In this example, the parameters are given as follows: $C_{L^*} = 0.3$, $m_{a^*} = 0$, $m_{b^*} = 1.5$, $\sigma_{a^*} = 10$, $\sigma_{b^*} = 20$, $\theta_o = 0.1$, $\beta = 0.4$, $\tau_F = 5$, $\tau_L = 3.5$, $\tau_R = 20$, $\gamma_F = 0.2$, $\gamma_L = 2.5$, $\gamma_R = 7$, and the standard deviations for W^F, W^L and W^R are 5, 3 and 2 respectively.

A position of the moving hand is simply represented by a gravitational center of the moving hand region. The center of gravity $\mathbf{G}(t)$ is given by

$$\mathbf{G}(t) = \frac{1}{N_i N_j} \sum_{i}^{N_i} \sum_{j}^{N_j} O_{ij}(t)[i, j] \quad (11)$$

where N_i and N_j represent the size of the input

images.

3.2 Motion Feature Map

The system employs velocity transitions of $\mathbf{G}(t)$ as the key to classification of motion patterns. The velocity of the gravitational center at time t is defined as

$$\mathbf{v}(t) = \mathbf{G}(t) - \mathbf{G}(t - \Delta t) \quad (12)$$

where Δt is the unit time for the velocity. Then we define velocity array vector $\mathbf{V}(t)$ as an array of $\mathbf{v}(t)$ to $\mathbf{v}(t-n_v)$ where n_v is a positive integer constant.

$$\mathbf{V}(t) = [\mathbf{v}(t), \mathbf{v}(t-1), \cdots, \mathbf{v}(t-n_v)] \quad (13)$$

For the recognition part, the $\mathbf{V}(t)$ should be normalized. Kohonen reported that self-organizing map (SOM) is capable of mapping input feature vectors into different feature space [17]. The translation keeps topological relationships between input vectors on the original feature space, and also the feature map quantizes the data distribution. We utilize the topological map by SOM in order to normalize the velocity array vector $\mathbf{V}(t)$.

2-dimensional topological SOM is a two layered competitive network as illustrated in **Fig. 4**. The competitive layer is a 2-dimensional $N_f \times N_f$ array of output neurons, which abides by winner-take-all rule. The output signal y_{ij} of the ijth output neuron is calculated as follows,

$$y_{ij} = \begin{cases} 1 & \text{if} \quad i = i_{win} \cap j = j_{win} \\ 0 & \text{otherwise} \end{cases} \quad (14)$$

$$||\mathbf{m}_{i_{win}j_{win}} - \mathbf{V}|| = \min_{i,j} ||\mathbf{m}_{ij} - \mathbf{V}|| \quad (15)$$

where i_{win} and j_{win} are the indices of the winner neuron, $\mathbf{m}_{ij}$ is the codebook vector. We define motion feature $\mathbf{x}(t)$ at time t as following equation.

$$\mathbf{x}(t) = [i_{win}, j_{win}] \quad (16)$$

The codebook vectors $\mathbf{m}_{ij}$ $(i = 1, 2, \cdots, N_f, j = 1, 2, \cdots, N_f)$ are adjusted by SOM learning rule.

$$\mathbf{m}_{ij}(s+1) = \mathbf{m}_{ij}(s) + \eta(s)\Phi(s)\mathbf{d}_{ij} \quad (17)$$

$$\Phi(s) = \exp\left(-\frac{|||[i,j] - [i_w(s), j_w(s)]||^2}{\sigma_n^2(s)}\right) \quad (18)$$

$$\mathbf{d}_{ij}(s) = \mathbf{V}_p(s) - \mathbf{m}_{ij}(s) \quad (19)$$

$$||\mathbf{m}_{i_w(s)j_w(s)}(s) - \mathbf{V}_p(s)|| = \min_{i,j} ||\mathbf{m}_{ij}(s) - \mathbf{V}_p(s)|| \quad (20)$$

where s is the iteration step for learning procedure, $i_w(s)$ and $j_w(s)$ are the indices of the winner neuron at step s, $\mathbf{V}_p(s)$ is the input pattern vector at step s, $\eta(s)$ is the learning rate and $\sigma_n(s)$ is the variable which defines the learning rate of neighborhoods. $\eta(s)$ and $\sigma_n(s)$ should have positive value respectively and decrease by degree to zero as the step s grows up.

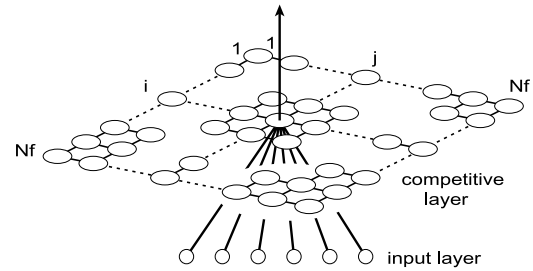

Fig. 4 Structure of the 2-D topological SOM.

4. Recognition

In the recognition part, dynamic programming (DP) matching is implemented. DP matching is able to compare sequential data to template pattern [23]. This algorithm has an ability to adjust distorted data to template. Each motion category has a template, and a similarity between the input motion pattern and its motion template. The motion pattern $\mathbf{X}$ is defined as a sequence of the input motion feature $\mathbf{x}(t)$ at time t,

$$\mathbf{X} = \{\mathbf{x}(1), \cdots, \mathbf{x}(t), \cdots, \mathbf{x}(t_{max})\} \quad (21)$$

where $t = 1$ is the first frame of the motion pattern and $t = t_{max}$ is the last frame of the motion pattern. The template $\mathbf{R}_q$ of the category q is also defined as a sequence of the motion feature $\mathbf{r}_q(u)$ at time u.

$$\mathbf{R}_q = \{\mathbf{r}_q(1), \cdots, \mathbf{r}_q(u), \cdots, \mathbf{r}_q(u_{max})\} \quad (22)$$

An accumulated cost $C_q(\mathbf{X}, t, u)$ is calculated by the DP matching rule,

$$C_q(\mathbf{X}, 1, 1) = ||\mathbf{r}_q(1) - \mathbf{x}(1)|| \quad (23)$$

$$L_q(\mathbf{X}, 1, 1) = 0 \quad (24)$$

$$C_q(\mathbf{X}, t, u) = \min \left\{ \begin{array}{c} C_{q1}(\mathbf{X}, t, u), \\ C_{q2}(\mathbf{X}, t, u), \\ C_{q3}(\mathbf{X}, t, u) \end{array} \right\} \quad (25)$$

where $C_{q1}(\mathbf{X}, t, u)$, $C_{q2}(\mathbf{X}, t, u)$ and $C_{q3}(\mathbf{X}, t, u)$ are described as follows.

$$C_{q1}(\mathbf{X}, t, u) = C_q(\mathbf{X}, t-1, u) + ||\mathbf{r}_q(u) - \mathbf{x}(t)|| \quad (26)$$

$$C_{q2}(\mathbf{X},t,u) = C_q(\mathbf{X},t-1,u-1) + 2||\mathbf{r}_q(u) - \mathbf{x}(t)|| \quad (27)$$

$$C_{q3}(\mathbf{X},t,u) = C_q(\mathbf{X},t,u-1) + ||\mathbf{r}_q(u) - \mathbf{x}(t)|| \quad (28)$$

A length of the path $L_q(\mathbf{X},t,u)$ is calculated by

$$L_q(\mathbf{X},t,u) = \begin{cases} L_q(\mathbf{X},t-1,u)+1 \\ \text{if } C_q(\mathbf{X},t,u) = C_{q1}(\mathbf{X},t,u) \\ L_q(\mathbf{X},t,u)+2 \\ \text{if } C_q(\mathbf{X},t,u) = C_{q2}(\mathbf{X},t,u) \\ L_q(\mathbf{X},t,u-1)+1 \\ \text{if } C_q(\mathbf{X},t,u) = C_{q3}(\mathbf{X},t,u) \end{cases} \quad (29)$$

Normalized accumulation cost $Z_q(\mathbf{X})$ is acquired by following.

$$Z_q(\mathbf{X}) = \frac{C_q(\mathbf{X},t_{max},u_{max})}{L_q(\mathbf{X},t_{max},u_{max})} \quad (30)$$

Recognition result q_{result} is obtained by finding the category with minimum $Z_q(\mathbf{X})$ among all q.

$$Z_{q_{result}}(\mathbf{X}) = \min_q Z_q(\mathbf{X}) \quad (31)$$

The template is figured out as averaged vectors of time normalized input patterns. We define the parameter u_{max} as a constant for the size of the templates. The pth input motion pattern $\mathbf{X}_p^q$ for the category q is normalized into the motion feature sequence $\mathbf{X'}_p^q$ which has u_{max} elements by linear interpolation. Then the template for the category q is written as,

$$\mathbf{R}_q = \frac{1}{N_q}\sum_q \mathbf{X'}_p^q \quad (32)$$

where N_q is the number of the input motion patterns for template $\mathbf{R}_q$.

5. Gesture Recognition Experiments

5.1 Training Conditions

The system is trained to recognize twelve hand-gesture patterns and the defined paths are shown in **Fig. 5**. Training data were obtained from three examinees at different backgrounds. We label them as scene A, B and C respectively. **Figure 6** shows some images from the training data movies. Three examinees performed all gesture patterns 3 times each with right and left arms. Thus 18 data were acquired for each category and the total number of the training data is 216. The movie data for the training were captured by a video camera with appropriate format: the frame rate is 15 [fps] and the image size is 80×60 [pixels]. The SONN has the same condition as described in Section 3.1. The parameters for the motion feature map (MFM) are $\Delta t = 3$, $n_v = 2$ and $N_f = 7$.

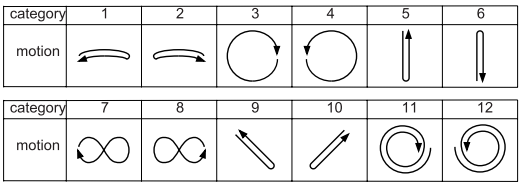

Fig. 5 Defined motion patterns for the simulation.

Fig. 6 Example images from the training data movies.

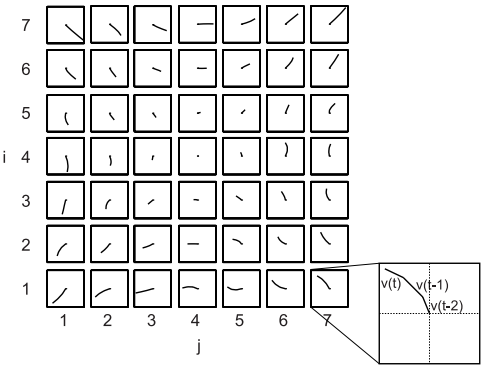

Fig. 7 Motion feature map.

5.2 Obtained Motion Feature Map

The obtained motion feature map (MFM) calculated by SOM is shown in **Fig. 7**. Each square corresponds to each codebook vector $\mathbf{m}_{ij}$ in the MFM. Actually, the dimension of the codebook vectors is 6, we divide the vector elements into three 2-dimensional velocity vectors in order to visualize them in Fig. 7.

5.3 Example of System Internal States

An example of internal states of the system for a test movie is shown in **Fig. 8**. The datum fed to the system was performed by the person at the scene A and belongs to the category 10. Figure 8 (a) shows extracted moving hand regions by SONN in some frames of the input movie. Figure 8 (b) is the transition of the activated node in the MFM. Figure 8 (c) is a 3-dimensional plot of Fig. 8 (b) with time axis. The normalized accumulation costs of the

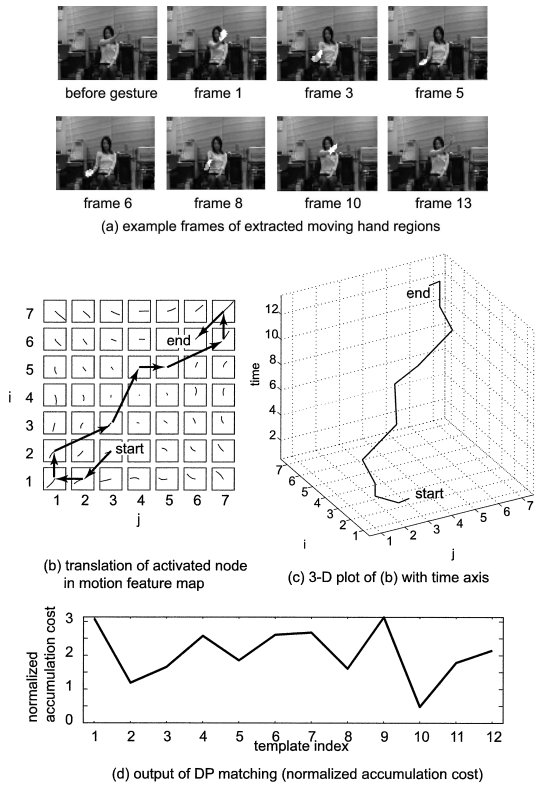

Fig. 8 Example of the calculation result.

DP matching templates are plotted in Fig. 8 (d), and the template which has the lowest cost corresponds to the category 10.

5.4 Experimental Results

5.4.1 Recognition Rates

At first, we have tested 360 untrained data to recognize the gestures in "known" situations. The test data were obtained in the same conditions as the training data, that is, the scene A, B and C. They performed all gesture patterns 5 times each with right and left arms. Thus 10 data were used for each category for each person, and 120 data were used for each scene. Then we have tested 720 untrained data to recognize the gestures at "unknown" situations. The test data were obtained from different six persons from the persons in the "known" situations. They were also captured with different background, and some of them were captured closely to the video camera. We label them as scene 1 to 6. **Figure 9** shows some images from the test data movies. The six examinees performed all gestures 5 times each with right and left arms. Thus 10 data were used for each category for each person, and 120 data were used for each scene. The results are shown in **Fig. 10**.

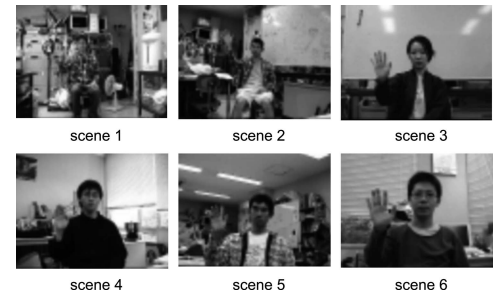

Fig. 9 Example images from "unknown" situation movies.

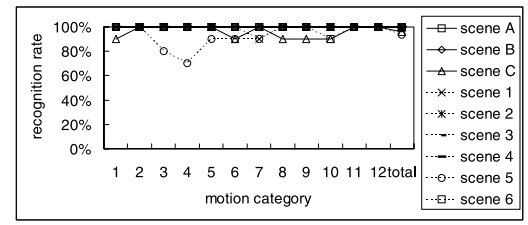

Fig. 10 Recognition rates.

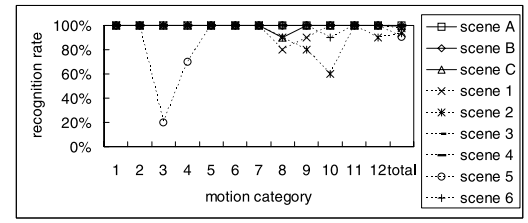

Fig. 11 Recognition rates of the system using frame difference technique.

5.4.2 Comparative Experiments

We also examine systems which have some different conditions from our original system; 1) a system using frame difference technique, 2) a system without the MFM.

At first, we replace the SONN in our system with frame difference technique. This system also uses skin-color regions extracted by thresholding in the $L^*a^*b^*$ color space. The moving hand region is calculated by "AND" operation between the frame difference and the skin-color regions. Its training and test conditions are the same as those of the previous experiments described in Section 5.4.1. The recognition results for this modification are shown in **Fig. 11**.

Above mentioned, the SONN has robustness to temporal Gaussian noise. This kind of noises sometimes appear in the real world as vibrations of clusters of small objects, for example, rustling leaves, riffles, waving sunblind, and so on. In order to verify the noise reduction ability of the SONN, we prepared additional test data

Fig. 12 Example images of the noisy background movies.

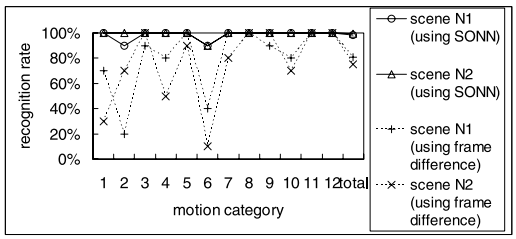

Fig. 13 Comparison of the recognition rates for scene N1 and N2.

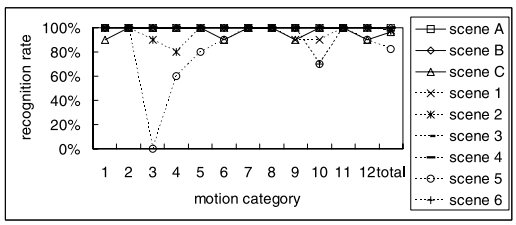

Fig. 14 Recognition rates of the system without MFM.

as scene N1 and N2. Example images in the scene N1 and N2 are shown in **Fig. 12**. The scene N1 and N2 contain an ornament waving by wind at each background. Note that the color of these ornament is close to skin-color, and they are not capable of ignoring them by utilizing skin-color regions. The recognition rates of the system using frame difference technique are compared with those of the system using SONN for the scene N1 and N2 in **Fig. 13**.

The second comparative system does not employ the MFM described in Section 3.2. The velocity vector array is directly fed to the DP matching process, therefore, the equation 16 is replaced with the following equation so that.

$$\mathbf{x}(t) = \mathbf{V}(t) \tag{33}$$

This omission of the MFM makes the system incapable of normalizing the velocities. Its training and test conditions are the same as those of the other experiments. The recognition results are shown in **Fig. 14**.

In addition, we have investigated the robustness of the MFM for image scaling. We translated the 80×60 [pixels] video images of the scene 3 into following sized images; 60×45 [pixels] as s-75%, 40×30 [pixels] as s-50%, 60×60 [pixels] as h-75%, 40×60 [pixels] as h-50%, 80×45 [pixels] as v-75%, 80×30 [pixels] as v-50%, 160×120 [pixels] as s-200%, 240×180 [pixels] as s-300%, 80×120 [pixels] as v-200%, 80×180 [pixels] as v-300%, 160×60 [pixels] as h-200% and 240×60 [pixels] as h-300%. **Figure 15** shows the distortion of the video images. Comparisons of the recognition rates for the image distortions are shown in **Figs. 16**, **17**, **18**, **19**, **20** and **21**. Figure 16 to Fig. 18 are the results for the scaling-down distortions,

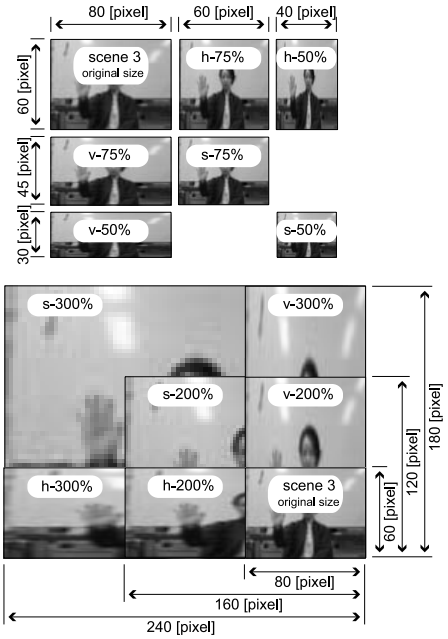

Fig. 15 Distorted images from test data movies.

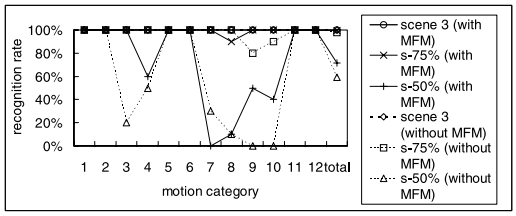

Fig. 16 Comparison of recognition rates for scaling-down distortion.

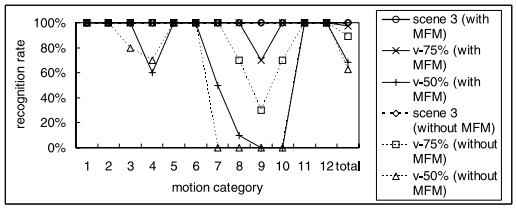

Fig. 17 Comparison of recognition rates for vertical scaling-down.

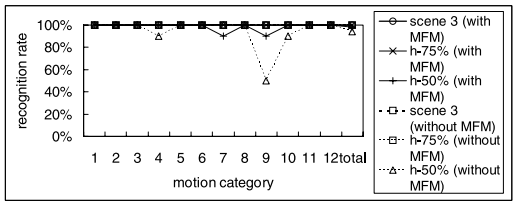

Fig. 18 Comparison of recognition rates for horizontal scaling-down.

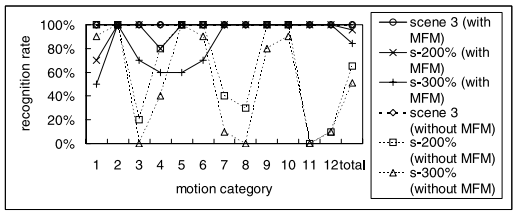

Fig. 19 Comparison of recognition rates for scaling-up distortion.

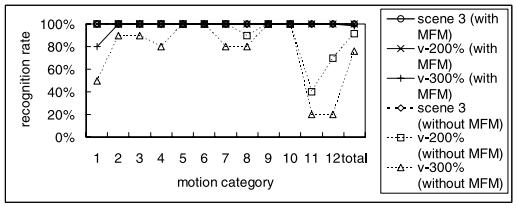

Fig. 20 Comparison of recognition rates for vertical scaling-up.

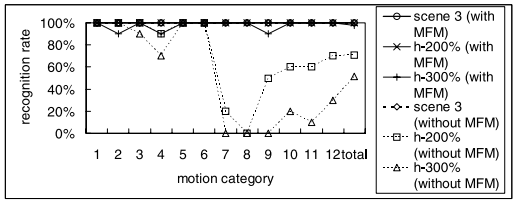

Fig. 21 Comparison of recognition rates for horizontal scaling-up.

and Fig. 19 to Fig. 21 are those for the scaling-up distortions, respectively.

6. Discussions

The recognition results of our system are shown in Fig. 10. The results show that the system has a high performance for recognizing gestures by various persons at various backgrounds. Recognition rates for the category 3 in the scene 5 are not high including comparative experiments; see Figs. 10 (80 [%]), 11 (20 [%]) and 14 (0 [%]). In this situation, the examinee tended to move his hand quite slowly and horizontally like an ellipse during the motion

Fig. 22 Motion trajectories of category 3 (clockwise circular rotation).

of the category 3 (clockwise circular rotation) while the ideal motion trajectory forms a circle (see **Fig. 22**), and it is difficult for the system to distinguish the input motion from the category 2 (waving hand). However, the SONN and the MFM alleviate the difficulty, and this is discussed later.

6.1 Moving Hand Extraction Using SONN

As illustrated in Figs. 3 and 8, SONN well extracts moving hand regions. Figures 10 and 11 show the recognition rates of our system using SONN and comparative system using frame difference technique respectively, and it seems that the SONN slightly improves the recognition performance. One of the characteristics of the SONN is output stability for a blank wall problem [16),24)], and it is especially effective on the recognition rate for the category 3 in the scene 5. In this situation, a hand of the examinee occupied rather large area and the motion was quite slow, therefore, a kind of blank wall problem sometimes occurred in the frame difference (see the bottom row of **Fig. 23**). This caused the instability of extracting velocity features and misclassification. On the other hand, the SONN extracted the hand regions more stable (see the top row of Fig. 23). Figure 13 shows the recognition rates of the both systems for noisy background. It indicates that the SONN is able to reduce more noises than the frame difference technique does. The comparative system sporadically detects noises of skin color like objects and returns a fallacious recognition result. On the other hand, the SONN eliminates some degree of noises even if the noise is skin colored (see **Fig. 24**). Therefore, the SONN acts on scenes at noisy backgrounds more appropriately than the frame difference technique. The drawback of the SONN lies in the difficulty in its parameter tuning, and the parameter tuning problem is a future work.

The input signals for the SONN are frame difference values in essentials. While the SONN

Fig. 23 Examples of blank wall problems.

Fig. 24 Examples of extracted moving hand regions at noisy background.

is able to adopt background subtraction values as its input signals, this technique has a problem such that it is difficult to determine and update a background since the background itself changes every second in the real world. In our system, the frame difference should be fed to the SONN since an object of interest is not a whole person but a moving hand. Some researchers proposed blob models for segmenting images as to color information [25)~27)]. The blob models aslo have the robustness for random noise and the stability for blank wall problem. However, they have restrictions for initializing the blobs. In most of the blob models, they use the background subtraction for extracting an object of interest during stable background in order to create the initial blobs [25),26)]. Starner, et al. utilized the blob model to gesture recognition system and their system simply generates hand regions from skin-color [27)]. However, their system cannot deal with the problem of the background image including at least one same skin-colored object.

For employing tracking algorithm, the system easily finds a region of interest by using the SONN since it generates a moving object as a connected region and reduces background noises. This alleviates the exceptional process for noises in tracking procedure. When expanding our system for two-hands gesture recognition, suitable tracking method should be tested in our future work. Wren, et al. [28)] and Bullock, et al. [29)] utilized Kalman filtering and Condensation algorithm respectively for blob based hand tracking. They reported these methods are efficient for occlusion, and their approaches are of reference to the expansion of our system.

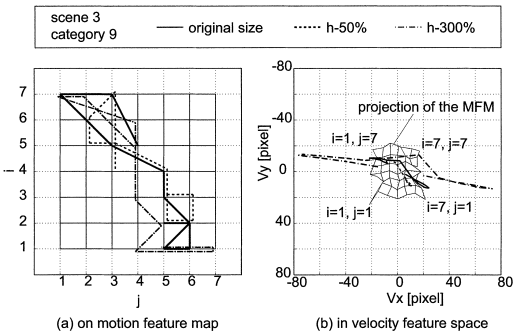

Fig. 25 Examples of feature vector trajectories for distorted movies.

6.2 Motion Feature Map

Comparing the results in Figs. 10 and 14, it seems that the absence of the MFM somewhat degrades the recognition rates. The most significant difference between the system with MFM and the system without MFM lies in the recognition rate of the category 3 in the scene 5; 80 [%] for the system with the MFM and 0 [%] for that without the MFM respectively. As mentioned above, the examinee moved his hand slowly and elliptically. This is a kind of distorted motion, and we can speculate that the MFM provides robustness to the distortion. The results of the comparative experiments in Fig. 16 to Fig. 21 show the robustness. The results in Fig. 16 to Fig. 18 show that the MFM alleviates the effects of scaling-down distortions, and those in Fig. 19 to Fig. 21 significantly indicate the robustness of the MFM to scaling-up distortions.

Figure 25 shows the motion feature trajectories of the category 9 in the scene 3 with transforming by horizontal scaling. Figure 25 (a) shows the trajectories on the MFM, and Fig. 25 (b) shows the trajectories in the velocity feature space. (In order to illustrate the trajectories in the 6-dimensional velocity feature space, we define 2-dimensional vectors $[V_x, V_y]$ which have the same Euclidean norms as the corresponding velocity array vectors and have the directions calculated by averaging velocity elements in the velocity array vectors. We will use the same representation method in the rest of figures when we illustrate the velocity feature space.) The projection of the MFM to the velocity feature space is also shown in

Fig. 25 (b). As described in Section 3.2, the MFM is figured out by topological SOM and the SOM quantizes and approximates data distribution with keeping their topology. The examples in Fig. 25 shows that the MFM alleviates the distortion of the input motion feature because of that trait of the SOM. This is suited to the DP matching due to the robustness of the DP matching to distortions in a certain range.

Here, DP matching templates on the MFM and those in the velocity feature space are shown in **Fig. 26** (a) and (b) respectively. The important point is that the templates on the MFM are defined not by mapping the templates in the velocity feature space but by averaging motion patterns after mapping the motion features. For this definition, motion features with extremely large norm do not directly affect the determination of templates, and it improves the recognition performance for scaled motions. In **Figs. 27** and **28**, the robustness of the MFM for the scaling-down is explained by comparing the recognition results of the system with MFM and that without MFM. Both results

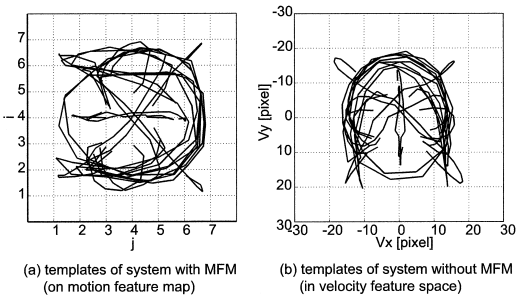

Fig. 26 Comparison of templates.

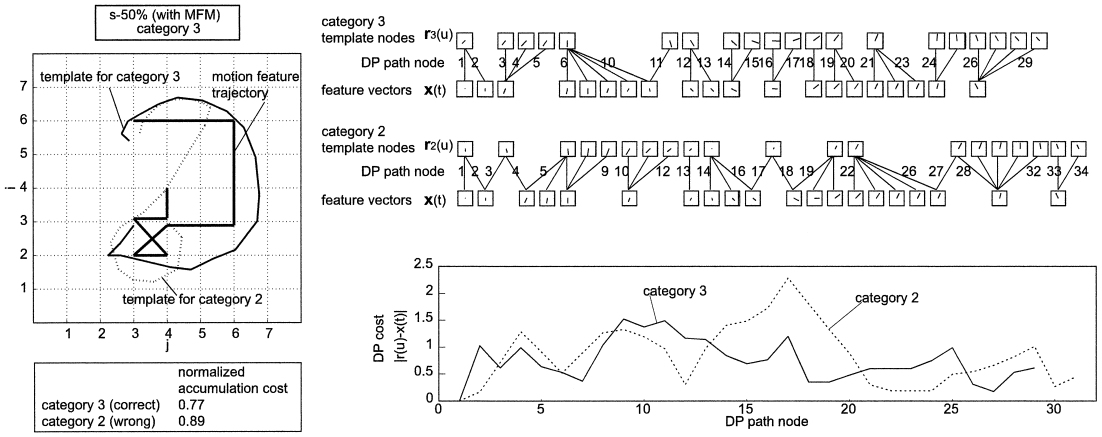

Fig. 27 Example of the motion feature trajectory and the history in the DP matching of the system using MFM.

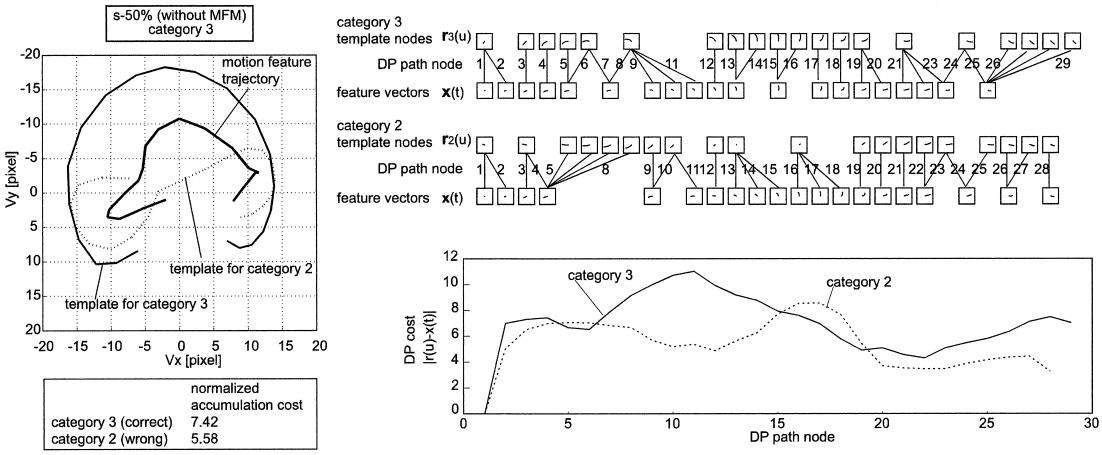

Fig. 28 Example of the motion feature trajectory and the history in the DP matching of the system without MFM.

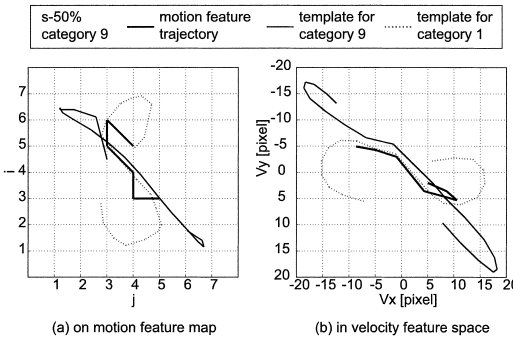

Fig. 29 Example of the recognition failure due to small velocity.

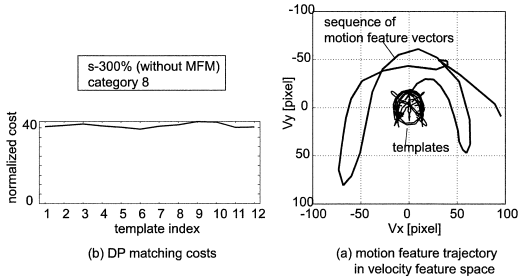

Fig. 30 Example of the motion feature trajectory and the recognition result of the system without MFM for scaling-up distorted movie.

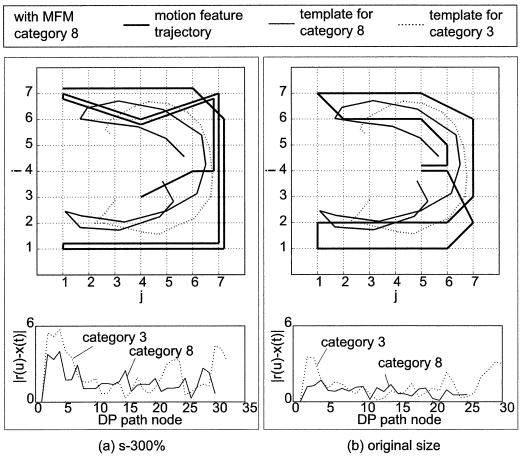

Fig. 31 Comparison of the motion feature trajectories for the scaling-up distorted movie with that for the original size movie.

are for the same movie datum (the category 3 in the s-50%), and the two systems return different recognition results; our system recognized it correctly, and the other recognized it as the category 2. Comparing the size of the templates (see the left of the Fig. 27 and the Fig. 28), the size difference between two templates in velocity feature space is larger than that of the MFM. Thus, the size of the motion feature trajectory influences the recognition by the system without MFM than the shape does. For example, the DP cost of the category 3 around the 11th DP path node (see the right of the Fig. 28) is too high and it causes the misclassification. On the other hand, the size of the template for the category 3 on the MFM is normalized (see the left of the Fig. 27). In Fig. 27, the influence of the trajectory size is alleviated and the DP matching works well for comparison of the trajectory shape among templates. While the MFM gives the robustness to some degree of scaling-down distortion, it has a limitation since obtained velocities are not normalized before mapping to the MFM. **Figure 29** shows an example of recognition failure when the velocities are too small to classify (the category 9 of the s-50%). For solving this problem, we should investigate a new scheme to normalize the velocities before mapping to the MFM for future work.

Next we discuss about the robustness to the scaling-up distortion. **Figure 30** shows the motion feature trajectory in the velocity feature space for the category 8 in the s-300%. It is clear that the system without MFM is not able to classify the motion when velocity is quite large. **Figure 31** (a) shows the motion feature trajectory on the MFM and the DP costs for the same datum as shown in Fig. 30. Figure 31 (b) shows those for the original size video images. Most of the motion features are assigned to the edge of the MFM. Then, the DP matching classifies the shape of the trajectory because that all templates are normalized.

Then we shall observe the real datum. **Figure 32** shows the motion feature trajectories of the category 3 in the scene 5. The similar phenomenon to that in Figs. 27 and 28 occurs since the examinee moved his hand quite slowly, and a partial distortion also occurs. The system using the MFM correctly classifies it because of the normalized templates and the quantization of the MFM, and the system without MFM classifies it as wrong category.

The robustness of our system to the image scaling results from the robustness of the MFM to the velocity scaling. On the contrary, the MFM is not sensible to the velocity transformation and it might be hard for our system to distinguish patterns when moving to the same direction with different speed. We may note about trade-off between the memory capacity

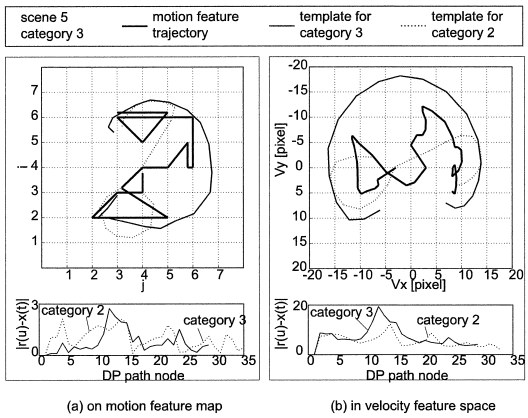

Fig. 32 The motion feature trajectories and the DP cost histories for a movie belongs to category 3 in scene 5.

of the MFM and the normalization ability. The larger the number of the nodes, the larger the number of motion patterns for gesture recognition. However, the increase of the number of the nodes degrades the normalization ability of the MFM for outliers. We should investigate a new scheme to optimize the size of the MFM for various conditions in our future work.

Our system employs the DP matching technique for classifying time sequential patterns. The DP matching is a quite efficient method for classifying small scale sequential patterns, and the strong point of the DP matching is its usability. Hidden Markov model (HMM) is quite popular method for classifying time sequential patterns, while the HMM needs many data for the parameter tuning since it is intrinsically a statistic method. For the DP matching technique, its templates can be created by few reference data, actually, our experiments show good recognition performance even if it has not been trained by so many data. Some existing systems using HMM employ k-mean algorithm to quantize its input vectors [4),5)], however such method is not available for data normalization in the DP matching since it needs a feature vector space with topological information. Thus the topology preservation of the SOM is suited to data normalization for the DP matching. In addition, when large quantities of training data are obtained, each node in the MFM by SOM is able to correspond to discrete symbols for HMM since each node quantizes the feature space. However, the topological property of the SOM is nullified in that case.

7. Conclusion

We propose a neural approach for video-based gesture recognition. We employ two types of neural networks for gesture recognition; 1) SONN for extracting moving hand regions, 2) SOM for normalizing motion features. Input time sequence pattern is classified by DP matching. Our experimental results show that the system has a good performance to classify twelve hand gesture patterns by various persons at various backgrounds. By comparing experimental results, we indicate that SONN and SOM improve the performance of the system. For situations with noisy backgrounds, the SONN performs better than the frame difference technique does. The SOM provides the robustness to spatial scaling distortion of input video images, and this is based on its robustness to velocity scaling. The topological property of SOM is quite suitable to normalizing feature vectors for DP matching technique.

References

1) Rebman, C.M., Aiken, M.W. and Cegielski, C.G.: Speech recognition in the human-computer interface, *Information and Management*, Vol.40, Issue 6, pp.509–519 (2003).
2) Wang, L., Hu, W. and Tan, T.: Recent developments in human motion analysis, *Pattern Recognition*, Vol.36, Issue 3, pp.585–601 (2003).
3) Yamamoto, J., Ohya, J. and Ishii, K. : Recognizing Human Action in Time-Sequential Images Using Hidden Markov Models, *IEICE* (*D-II*), Vol.J76-D-II, No.12, pp.2556–2563 (1993).
4) Min, B., Yoon, H., Soh, J., Ohashi, T. and Ejima, T.: Gesture-based editing system for graphic primitives and alphanumeric characters, *Engineering Applications of Artificial Intelligence*, Vol.12, pp.429–441 (1999).
5) Chen, F., Fu, C. and Huang, C.: Hand gesture recognition using a real-time tracking method and hidden Markov models, *Image and Vision Computing*, Vol.21, pp.745–758 (2003).
6) Sagawa, H., Ohki, M., Sakiyama, T., Ohara, E., Ikeda, H. and Fujisawa, H.: Pattern Recognition and Synthesis for Sign Language Translation System, *Journal of Visual Languages and Computing*, Vol.7, pp.109–127 (1996).
7) Osaki, R., Shimada, M. and Uehara, K.: Extraction of primitive motions by using clustering and segmentation of motion-captured data, *JSAI*, Vol.15, No.5 pp.878–886 (2000).
8) Ng, C.W. and Ranganath, S.: Real-time gesture recognition system and application, *Image and Vision Computing*, Vol.20, pp.993–

1007 (2002).
9) Lamar, M.V., Bhuiyan, M.S. and Iwata, A.: Temporal Series Recognition Using a New Neural Network Structure T-CombNET, *6th International Conference on Neural Information Processing*, Vol.III, Perth, Australia, pp.1112–1117 (1999).
10) Henry, C.C. and Liyanage, C.D.S.: Human Activity Recognition by Head Movement using Elman Network and Neuro-Markovian Hybrids, *Proc. Image and Vision Computing New Zealand 2003* (*IVCNZ2003*), pp.320–326 (2003).
11) Freeman, W.T., Anderson, D.B., Beardsley, P.A., Dodge, C.N., Roth, M., Weissman, C.D., Yerazunis, W.S., Kage, H., Kyuma, K., Miyake, Y. and Tanaka, K.: Computer Vision for Interactive Computer Graphics, *IEEE Computer Graphics and Applications*, Vol.18, Issue 3, pp.42–53 (1998).
12) Shin, M.C., Tsap, L.V. and Goldgof D.B.: Gesture recognition using Bezier curves for visualization navigation from registered 3-D data, *Pattern Recognition*, Vol.37, pp.1011–1024 (2004).
13) Kubota, T.: Massively parallel networks for edge localization and contour integration — adaptable relaxation approach, *Neural Networks*, Vol.17, pp.411–425 (2004).
14) Katayama, K., Ando, M. and Horiguchi, T.: Models of MT and MST areas using wake-sleep algorithm, *Neural Networks*, Vol.17, pp.339–351 (2004).
15) Yoshiike, N. and Takefuji, Y.: Object segmentation using maximum neural networks for the gesture recognition system, *Neurocomputing*, Vol.51, pp.213–224 (2003).
16) Chashikawa, T. and Takefuji, Y.: Extracting Moving Object Areas Based on Second-order Neural Network, *IPSJ*, Vol.44, No.SIG 14(TOM 9), pp.31–47 (2003).
17) Kohonen, T.: *Self-Organizing Maps*, Springer-Verlag, Berlin (1995).
18) Elgammal, A., Harwood, D. and Davis, L.S.: Non-parametric Model for Background Subtraction, *Proc. IEEE ICCV'99 FRAME= RATE Workshop* (1999).
19) Jain, R., Martin, W.H. and Aggarwal, J.K.: Segmentation through the detection of changes due to motion, *Computer Vision, Graphics and Image Processing*, Vol.11, pp.13–34 (1979).
20) Eckhorn, R., Reitboeck, H.J., Arndt, M. and Dicke, P.: Feature Linking via Synchronization among Distributed Assemblies: Simulations of Results from Cat Visual Cortex, *Neural Computing*, Vol.2, pp.293–307 (1990).
21) Kinser, J.M. and Johnson, J.L.: Stabilized Input with a Feedback Pulse-Coupled Neural Network, *Optical Engineering*, Vol.35, No.8, pp.2158–2161 (1996).
22) Lopez, L.R.: Feedforward shunting: A simple second-order neural network motion sensor, *Proc. International Society for Optical Engineering* (*SPIE*), Vol.1297, pp.350–358 (1990).
23) Bellman, R.: *Dynamic Programming*, Princeton Univ. Press, New Jersey (1957).
24) Simoncelli, E.P.: Local Analysis of Visual Motion, *The Visual Neurosciences*, Chapter 109, MIT Press (2003).
25) Wren, C.R., Azarbayejani, A., Darrell, T. and Pentland, A.P.: Pfinder: Real-Time Tracking of the Human Body, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.19, No.7, pp.780–785 (1997).
26) Takashima, A., Futsuhara, H., Ohashi, T., Noma, T. and Ejima, T.: Ptracker: Real-Time Tracking of the Human Motion, *Technical Report of IEICE*, PRMU99-33, pp.25–32 (1999).
27) Starner, T., Weaver, J. and Pentland, A.: Real-Time American Sign Language Recognition Using Desk and Wearable Computer Based Video, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.20, No.12, pp.1371–1375 (1998).
28) Wren, C.R., Clarkson, B.P. and Pentland, A.P.: Understanding Purposeful Human Motion, *Proc. 4th IEEE International Conference on Automatic Face and Gesture Recognition*, pp.378–383 (2000).
29) Bullock, D.J. and Zelek, J.S.: Real-time tracking for visual interface applications in cluttered and occluding situations, *Image and Vision Computing*, Vol.22, pp.1083–1091 (2004).

(Received November 22, 2004)
(Revised January 11, 2005)
(Accepted January 24, 2005)
(Released July 13, 2005)
(Paper version of this article can be found in the IPSJ Transactions on Mathematical Modeling and Its Applications, Vol.46 No.SIG10(TOM12), pp.124–137.)

Masato Aoba is a visiting researcher of Keio Research Institute at Shonan Fujisawa Campus. He received his M.S. degree in Mechanical Engineering from Waseda University in 1995. He worked at TOYO Communication Equipment Co., Ltd. from 1995 to 2002. He received his Ph.D. degree in Media and Governance from Keio University in 2005. His research interests focus on neural computing and image pattern recognition.

Yoshiyasu Takefuji is a tenured professor on faculty of environmental information at Keio University since April 1992 and was on tenured faculty of Electrical Engineering at Case Western Reserve University since 1988. Before joining Case, he taught at the University of South Florida and the University of South Carolina. He received his BS (1978), MS (1980), and Ph.D. (1983) in Electrical Engineering from Keio University. His research interests focus on neural computing and hyperspectral computing.