A NEURAL-NETWORK-BASED APPROACH TO RECOGNIZING 3D ARM MOVEMENTS

MU-CHUN SU¹, YU-XIANG ZHAO², AND EUGENE LAI²

¹Department of Computer Science and Information Engineering,
National Central University, Taoyuan,

²Department of Electrical Engineering, Tamkang University, Taipei, Taiwan

ABSTRACT

Gesture recognition is needed for a variety of applications. One particular application of gesture-based systems is to implement a speaking aid for the deaf. Among several factors constituting a hand gesture, the arm movement pattern is one of the most challenging features to recognize. In this paper, we propose a neural-network-based approach to recognition of spatio-temporal patterns of nonlinear 3D arm movements. Compared to Hidden-Markov-Model-based methods, the most appealing property of the proposed method is its simplicity. The effectiveness of this method is evaluated by a database consisted of 10 persons.

Biomed Eng Appl Basis Comm, 2003 (February); 15: 17-26.

Keywords: assistive technology, human-computer interface, gesture recognition, neural networks.

1. INTRODUCTION

Gestures are an efficient communication modality in many everyday situations. The desire to provide a more convenient and natural way of interacting with computers has led to considerable interest in recognizing hand gestures. For example, it is a more intuitive and natural way to use a so-called finger mouse that enables a user to specify commands and additional parameters by drawing single intuitive gestures with his or her finger. A variety of gesture-based applications have been created so far. The use of hand gestures as input interfaces is widely addressed in several articles (to name just a few here) [1]-[15].

One particularly important application of gesture-based applications is to implement speaking aids for the deaf [16]-[23]. Fig. 1 illustrate a speaking aid. Successful rehabilitation of the deaf in a hearing world

must provide a mechanism by which meaningful communication between the deaf and hearing persons is possible. For this to happen, it requires a hand gesture recognition system to recognize sign language and a speech synthesizer to provide speech to the hearing person.

In general, hand gestures can be described by the following four attributes: (1) hand shape (i.e. posture); (2) palm orientation; (3) hand location; and (4) arm movement. The attributes can broadly be separated into static (spatial) and dynamic (spatio-temporal) attributes. Generally, the values of the static attributes will be calculated directly from the data supplied by the hardware, whereas the dynamic attribute values will be calculated from the values of a sequence of static attribute values. Since hand gestures are spatio-temporally varying, the automatic gesture recognition turns out to be very challenging. Many different approaches to recognition of spatio-temporal patterns of hand gestures have recently reported.

(1) Neural Networks: Multi-layer neural networks [17], recurrent neural networks [21], and self-organizing feature maps [22] are three popular types of neural networks that were used in gesture recognition.

(2) Dynamic Programming Matching: Dynamic programming matching (DP matching) provides the effect of a non-linear normalization process, allowing two

Received: Oct. 29, 2001; Accepted: Feb. 25, 2002

Correspondence: Mu-Chun Su, Professor
Department of Computer Science and Information
Engineering, National Central University, Taoyuan,
Taiwan

E-mail: muchun@csie.ncu.edu.tw

Fig. 1 A speaking aid for the deaf.

dynamic signals to be matched. This method allows classifying isolated dynamic gestures [23]. However, this method requires a great amount of time and memory.

(3) Hidden Markov Models: Recently, the property of Hidden Markov models (HMMs) to compensate time and amplitude variance of patterns makes them appear an ideal approach for hand gesture recognition [25]-[26]. HMMs are a type of statistical model [27]. Although several algorithms have been proposed to solve the parameter estimation problem, it still is not an easy job to build an HMM.

We have tackled the problem of recognizing hand gestures without involving arm movements [19]-[20]. In this paper, we focus on recognizing 3D arm movements. An approach based on HMMs to recognizing 3D arm movements was proposed in [28]. A decision-tree-based method was presented in [29]. In our previous work, we have proposed a fuzzy rule-base approach to recognizing 3D arm movements [30]. Each has its own merits and disadvantages. Here we propose a neural-network-based approach. The main aim of this research is to explore the possibility of using a method as simple as possible to solve the recognition of arm movements. We first introduce the special class of neural networks employed by us in the Section 2. The proposed recognition method is discussed in Section 3. In Section 4, we give the results obtained by applying the method to the database consisting of 8 types of 3D arm movements. Finally, Section 5 concludes the paper.

2. NEURAL NETWORKS

The most useful properties of neural networks are their "training" and "generalizing" abilities. These abilities are based, to a great extent, on the nature of the neural-type junctions and the type of activation functions. The most frequently encountered neurons in neural network applications to date are those that consist of a linearly summing device followed by an "activation function" of the sigmoidal type. However, it is

possible to use other types of activation functions endowing the networks with a variety of useful properties.

Here, we adopt the use of a linear activation function that allows the output of a neuron represents the similarity degree between an input vector and its synaptic weight vector. The neuron with a linear activation function is described by the following equations:

$$net_j(\underline{x}) = \sum_{i=1}^n w_{ji} x_i = \underline{w}_j^T \underline{x} \quad (1)$$

and

$$Out_j(\underline{x}) = f(net_j(\underline{x})) = \frac{1}{2}(net_j(\underline{x}) + 1) \quad (2)$$

where $\underline{x} = (x_1, x_2, \dots, x_n)^T$ is an input vector, $\underline{w}_j = (w_{j1}, w_{j2}, \dots, w_{jn})^T$ is the synaptic weight vector of neuron j , $f(\cdot)$ is the activation function, and $Out_j(\underline{x})$ is the output of the neuron.

If the vectors $\underline{x}$ and $\underline{w}_j$ are both normalized to have unit length, that is, $\|\underline{x}\| = \|\underline{w}_j\| = 1$ then Eq. (2) can be rewritten as:

$$Out_j(\underline{x}) = \frac{1}{2}(\|\underline{x}\| \cdot \|\underline{w}_j\| \cos(\theta) + 1) \\ = \frac{1}{2}(\cos(\theta) + 1) \quad (3)$$

where θ represents the angle subtended between the vectors $\underline{x}$ and $\underline{w}_j$. As a result, the value of $Out_j(\underline{x})$ is constrained to be in a range of values from 0 to 1. Furthermore, it measures the similarity between vectors $\underline{x}$ and $\underline{w}_j$. When $\underline{x}$ and $\underline{w}_j$ are oriented in the same direction with respect to the origin, the value of $Out_j(\underline{x})$ reaches its maximum. From the viewpoint of fuzzy logic, the output function of neuron j , $Out_j(\underline{x})$, can be regarded as a membership function of a fuzzy set D_j defining a particular direction characterized by vector $\underline{w}_j$.

Fig. 2 The 8 direction vectors used by the 8-direction chain code and their corresponding 8 synaptic weight vectors.

An important step (the details of this step will be presented in the next section) in our 3D arm movement recognition method is to employ the 8-directional chain coding scheme to code the arm movements in the vocabulary so as to generate templates for every typical arm movement. Therefore, 8 neural networks corresponding to the 8 direction vectors shown in Fig. 2 are required in our recognition system. There are two ways to train the 8 neural networks. The simplest way is to directly assign the 8 direction vectors to be the corresponding synaptic weight vectors of the corresponding neural networks. The other way is to ask some persons to sign the 8 directions for several times so as to create 8 training sets. Then we use the 8 training sets to train 8 neural networks. The updating formula is given as follows:

$$\underline{w}_j(k+1) = \frac{k}{k+1} \underline{w}_j(k) + \frac{1}{k+1} \underline{x}^{(j)}(k+1) \quad \text{for } j = 1, \dots, 8 \quad (4)$$

where k represents the iteration time step and $\underline{x}^{(j)}(k+1)$ represents the $(k+1)$ th data in the j th training set. Although the first method is very simple and straightforward, the neural networks trained by the second method may be more representative.

3. RECOGNITION METHOD

There are several factors making the problem of arm movement recognition become very challenging:

- Human variability causes the same type of arm movement to be conducted differently each time. Even for the same person, different instances of the same type of arm movement may not be identical.
- The same arm movement can be performed in dif-

Table I. The 8 types of arm movements used in the vocabulary.

	Movement primes	Template
1	—	$\{\underline{W}_1, \underline{W}_2, \underline{W}_3, \underline{W}_4, \underline{W}_5, \underline{W}_6, \underline{W}_7, \underline{W}_8\}$
2	⤿	$\{\underline{W}_2, \underline{W}_8, \underline{W}_8, \underline{W}_6, \underline{W}_6, \underline{W}_4, \underline{W}_4, \underline{W}_2\}$
3	└	$\{\underline{W}_7, \underline{W}_5, \underline{W}_7, \underline{W}_1\}$
4	┘	$\{\underline{W}_7, \underline{W}_4, \underline{W}_7, \underline{W}_2\}$
5	⌚	$\{\underline{W}_1, \underline{W}_7, \underline{W}_5, \underline{W}_5, \underline{W}_7, \underline{W}_1\}$
6	M	$\{\underline{W}_3, \underline{W}_8, \underline{W}_2, \underline{W}_7, \underline{W}_1, \underline{W}_6, \underline{W}_8, \underline{W}_5\}$
7	↶	$\{\underline{W}_6, \underline{W}_1, \underline{W}_6\}$
8	○	$\{\underline{W}_3, \underline{W}_2, \dots, \underline{W}_5, \underline{W}_4, \underline{W}_4, \underline{W}_3, \dots, \underline{W}_6, \underline{W}_5\}$

ferent orientations and / or positions.

Similar to the method proposed in [30], the method of recognizing 3D arm movements involves the following six steps. The comparisons and differences between these two methods will be discussed in the concluding section.

Step 1: Sampling:

A low-cost ultrasonic tracker developed by us is used to sense the 3D absolute position (x, y, z) of the dominant hand in space. The sampling rate is 24 samples/second.

Step 2: Preprocessing:

For further processing the raw data must be filtered. Two cascaded moving-average filters of the form

$$\dot{\underline{x}}_n = \frac{1}{5} \sum_{l=-2}^2 \underline{x}_{n+l} \quad (5)$$

were used to smooth the raw data in our experiments.

The vectors $\underline{x}_n$ and $\dot{\underline{x}}_n$ are the original and smoothed samples, respectively. Fig. 3 illustrates the results of applying the smoothing filters. In this example, a user signed a M-shaped arm movement.

Fig. 3 The results of preprocessing process: (a) the original trajectory; (b) the result of the first smoothing filter; (c) the result of the second smoothing filter.

Step3: Projection:

Usually, the whole trajectory of an isolated hand gesture is conducted in a single plane in space. Therefore, we first find the best fitting plane (either X-Y plane, Y-Z plane or the X-Z plane) for the preprocessed 3D data and then project the 3D data into the 2D coordinates on that plane. To find the best fitting plane, we simply compute the three variances of the 3D data along each coordinate axis and then the two axes that have the two largest variances are chosen. The projection can reduce the computational complexity for further processing the data.

Step4: Template Generating:

The 8-directional chain coding scheme is employed to code each typical type of arm movements in the vocabulary. In our experiments, 8 types of arm

movements shown in the second column of Table I were used. There are two ways to generate templates for each arm movement. The first way is to directly assign direction vectors to corresponding arm movement by just viewing the shape of that arm movement. By viewing each type of the arm movements in the vocabulary it is easy to find what kind of combination of direction vectors shown in the second column of Table I consists of what type of arm movement. These observations are tabulated in the last column of Table I. The second way is to generate templates from training data sets. First, we have to compute the difference vectors, $\underline{x}_i = \underline{x}_{i+1} - \underline{x}_i$ for $i = 1, \dots, N$ from the smoothed sequence. Then we have to find out the corresponding 8-direction chain code of the sequence by assigning the difference vectors to corresponding direction vectors. Suppose the 8-directional chain code of a sequence $\{\underline{x}_1, \dots, \underline{x}_i, \dots, \underline{x}_N\}$ is

$$\{1^*, \dots, i^*, \dots, N^*\}$$

where $i^* = \arg \max_{j=1, \dots, 8} Out_j(\underline{x}_i)$. By observing this

coded sequence, we would know this typical arm movement is consisted of what kind of combination of direction vectors. Then the direction vectors appearing in the corresponding coded sequence represent the template for that typical arm movement. For example, an L-shaped movement may result in the following coded sequence $\{77777788111111\}$. Then we will say a template of the L-shaped arm movement is $\{\underline{w}_7, \underline{w}_1\}$. Note that the reason why the vector $\underline{w}_8$ does not appear in the template is that the appearing frequency of this vector is much less than those of $\underline{w}_7$ and $\underline{w}_1$, therefore, it should be ignored. Since the former turns to have good classification performance and is much simpler, we adopt the former here. At last, each arm type may have multiple templates since the same trajectories may start from different directions. For example, the first type of arm movements has 8 different templates, $\{\underline{w}_1\}, \dots, \{\underline{w}_8\}$.

Step 5: Pattern Recognition:

When a sampling sequence of an unknown arm movement with total N difference vectors, $\underline{x}_i = \underline{x}_{i+1} - \underline{x}_i$ for $i = 1, \dots, N$, is to be classified, these N difference vectors are presented to trained neural networks and "accumulative similarities" are then computed accordingly. Suppose the k th type of arm movements in the vocabulary is consisted of D_k

Fig. 4 A finite state machine to compute S_k .

direction vectors, $\underline{w}_{d_1^k}, \underline{w}_{d_2^k}, \dots, \underline{w}_{d_{D_k}^k}$, where d_j^k is an integer between 1 and 8. In addition, these direction vectors are also assumed to appear in the k th type of arm movements in this order. For example, for the L-shaped arm movement is consisted of $\underline{w}_7$ and $\underline{w}_1$.

Then the accumulative similarity S_k between the sampling sequence and the k th type of arm movements is computed as follows:

$$S_k = \sum_{i=1}^{N_{d_1^k}} S(d_1^k, i) + \sum_{i=N_{d_1^k}+1}^{N_{d_1^k}+N_{d_2^k}} S(d_2^k, i) + \dots + \sum_{i=N_{d_1^k}+N_{d_2^k}+\dots+N_{d_{D_k-1}^k}+1}^{N_{d_1^k}+N_{d_2^k}+\dots+N_{d_{D_k}^k}} S(d_{D_k}^k, i) \quad (6)$$

where

$$S(d_j^k, i) = Out_{d_j^k}(\underline{x}_i), \quad (7)$$

$N_{d_j^k}$ denotes the number of difference vectors that should be presented to the neural network corresponding to the direction vector $\underline{w}_{d_j^k}$. The value of the local similarity, $S(d_j^k, i) = Out_{d_j^k}(\underline{x}_i)$, measures the degree of similarity between the pattern $\underline{x}_i$ and the direction vector $\underline{w}_{d_j^k}$.

The following question is how to distribute the N patterns into D_k parts. First, we won't compute the accumulative similarity S_k until we find a pattern $\underline{x}_f$ in the N difference vectors which meets the following condition:

$$S(d_1^k, f) = \max_{j=1, \dots, 8} S(j, f). \quad (8)$$

That is, the first $f-1$ patterns, $\underline{x}_1, \dots, \underline{x}_{f-1}$, do not contribute to the computation of the accumulative similarity S_k (i.e. $S(d_1^k, i) = 0$ for $i = 1, \dots, f-1$.) After this, we then really begin to feed the input patterns starting from $\underline{x}_f$ into the neural network

corresponding to the direction vector $\underline{w}_{d_1^k}$ and compute the accumulative similarity S_k until a pattern $\underline{x}_i$ satisfies the following condition

$$S(d_1^k, i) \leq S(d_2^k, i) \text{ and } S(d_2^k, i) \geq \theta \quad (9)$$

where θ is a prespecified threshold. In our experiments, we set $\theta = 0.8$. In the following, we feed the input patterns starting from $\underline{x}_i$ into the neural network corresponding to the direction vector $\underline{w}_{d_2^k}$ until the following condition happens

$$S(d_2^k, r) \leq S(d_3^k, r) \text{ and } S(d_3^k, r) \geq \theta. \quad (10)$$

This procedure is repeated until all patterns are processed. Finally, it is easy to find that the number of patterns that feed into the neural network corresponding to the direction vector $\underline{w}_{d_j^k}$ is $N_{d_j^k}$ since the following two conditions are satisfied:

$$S(d_{j-1}^k, N_{d_1^k} + N_{d_2^k} + \dots + N_{d_{j-1}^k}) \leq S(d_j^k, N_{d_1^k} + N_{d_2^k} + \dots + N_{d_{j-1}^k} + 1) \quad (11)$$

and

$$S(d_j^k, N_{d_1^k} + N_{d_2^k} + \dots + N_{d_{j-1}^k} + N_{d_j^k}) \leq S(d_{j+1}^k, N_{d_1^k} + N_{d_2^k} + \dots + N_{d_j^k} + 1) \quad (12)$$

Finally, if $\sum_{j=1}^J N_{d_j^k} = N$ for some $J < D_k$ then

S_k has to be reset to be 0 because there is no patterns that match the remaining direction vectors, $\underline{w}_{J+1}^k, \underline{w}_{J+2}^k, \dots, \underline{w}_{D_k}^k$. Fig. 4 shows the schematic diagram of the pattern recognition procedure. Another thing should be emphasized is that since each type of arm movements has multiple templates the accumulated similarity should be computed by using the maximum operator to find the maximum value from the accumulated similarities between each template and the unknown sequence.

Step 6: Decision Making:

The sampling sequence is finally classified to be the type of arm movements with the largest accumulative similarity in the vocabulary. That is, the unknown arm movement will be classified as the k^* th type of arm movements in the vocabulary if the following condition is satisfied:

$$S_{k^*} \geq S_k \quad \text{for } k \neq k^* \quad (13)$$

The computation of S_k can be implemented by a finite state machine shown in Fig. 4. In addition, we use Table II to illustrate the recognition procedure. In

Table II. An example to illustrate how to compute S_k : Since $S_7 > S_k$ for $k \neq 7$ the sampling sequence is correctly classified to be the seventh type of arm movements.

Arm Type	1	2	3	4	5	6	7	8
Template	{1},{2},{3}	{2,8},{8,6}	{7,5},{7,1}	{7,4},{7,2}	{1,7,5},	{3,8,2,7},	{6,1,6},	{3,2,...,5,4}
Data	{,}{4},.....	,.....						,.....
$\underline{x}_1$	$S(7,1)=$ 0.894536	$S(0,1)=$ 0.00000	$S(0,1)=$ 0.00000	$S(0,1)=$ 0.00000	$S(0,1)=$ 0.00000	$S(0,1)=$ 0.00000	$S(6,1)=$ 0.996167	$S(6,1)=$ 0.996156
$\underline{x}_2$	$S(7,2)=$ 0.903059	$S(0,2)=$ 0.00000	$S(0,2)=$ 0.00000	$S(0,2)=$ 0.00000	$S(0,2)=$ 0.00000	$S(0,2)=$ 0.00000	$S(6,2)=$ 0.994223	$S(6,2)=$ 0.999571
$\underline{x}_3$	$S(7,3)=$ 0.909719	$S(0,3)=$ 0.00000	$S(0,3)=$ 0.00000	$S(0,3)=$ 0.00000	$S(0,3)=$ 0.00000	$S(0,3)=$ 0.00000	$S(6,3)=$ 0.992360	$S(6,3)=$ 0.992360
$\underline{x}_4$	$S(7,4)=$ 0.986545	$S(0,4)=$ 0.00000	$S(7,4)=$ 0.986545	$S(7,4)=$ 0.986545	$S(0,4)=$ 0.00000	$S(0,4)=$ 0.00000	$S(6,4)=$ 0.925508	$S(6,4)=$ 0.925508
$\underline{x}_5$	$S(7,5)=$ 0.640096	$S(0,5)=$ 0.00000	$S(7,5)=$ 0.640096	$S(7,5)=$ 0.640096	$S(1,5)=$ 0.979972	$S(0,5)=$ 0.00000	$S(1,5)=$ 0.979972	$S(6,5)=$ 0.259672
$\underline{x}_6$	$S(7,6)=$ 0.577868	$S(0,6)=$ 0.00000	$S(7,6)=$ 0.577868	$S(7,6)=$ 0.577868	$S(1,6)=$ 0.993899	$S(0,6)=$ 0.00000	$S(1,6)=$ 0.993899	$S(6,6)=$ 0.205822
$\underline{x}_7$	$S(7,7)=$ 0.561808	$S(0,7)=$ 0.00000	$S(7,7)=$ 0.561808	$S(7,7)=$ 0.561808	$S(1,7)=$ 0.996165	$S(0,7)=$ 0.00000	$S(1,7)=$ 0.996165	$S(6,7)=$ 0.192863
$\underline{x}_8$	$S(7,8)=$ 0.515891	$S(0,8)=$ 0.00000	$S(7,8)=$ 0.515891	$S(7,8)=$ 0.515891	$S(1,8)=$ 0.999747	$S(0,8)=$ 0.00000	$S(1,8)=$ 0.999747	$S(6,8)=$ 0.157862
$\underline{x}_9$	$S(7,9)=$ 0.999599	$S(0,9)=$ 0.00000	$S(7,9)=$ 0.999599	$S(7,9)=$ 0.999599	$S(7,9)=$ 0.999599	$S(0,9)=$ 0.00000	$S(6,9)=$ 0.867420	$S(7,9)=$ 0.999599
$\underline{x}_{10}$	$S(7,10)=$ 0.831645	$S(0,10)=$ 0.00000	$S(5,10)=$ 0.874181	$S(7,10)=$ 0.831645	$S(5,10)=$ 0.874181	$S(0,10)=$ 0.00000	$S(6,10)=$ 0.999095	$S(7,10)=$ 0.831645
$\underline{x}_{11}$	$S(7,11)=$ 0.919762	$S(0,11)=$ 0.00000	$S(5,11)=$ 0.771662	$S(7,11)=$ 0.919762	$S(5,11)=$ 0.771662	$S(0,11)=$ 0.00000	$S(6,11)=$ 0.988910	$S(7,11)=$ 0.919762
$\underline{x}_{12}$	$S(7,12)=$ 0.847637	$S(0,12)=$ 0.00000	$S(5,12)=$ 0.859372	$S(7,12)=$ 0.847637	$S(5,12)=$ 0.859372	$S(0,12)=$ 0.00000	$S(6,12)=$ 0.999931	$S(7,12)=$ 0.847637
Total	$S_1=$ 9.588165	$S_2=$ 0.00000	$S_3=$ 6.787022	$S_4=$ 0.00000	$S_5=$ 7.474597	$S_6=$ 0.00000	$S_7=$ 11.733397	$S_8=$ 0.00000

this example the user signed the L-shaped arm movements. Suppose the sampling sequence contains only 8 difference vectors. From Table II, we find that $S_7 > S_k$ for $k \neq 7$, therefore, the sampling sequence will be classified to be the seventh type of arm movements. One may ask why $S_k = 0$ for $k = 2, 4, 6, 8$. Without loss of generalization, we just show why $S_4 = 0$ here. There are two templates, $\{\underline{w}_7, \underline{w}_4\}$ and $\{\underline{w}_7, \underline{w}_2\}$, in the fourth type of arm movements. First, we find the first three patterns won't contribute to the computation of S_4 because $\max_{j=1,\dots,8} S(j,i) \neq S(7,i)$ for

$i = 1, \dots, 3$. In addition, since we can't find any pattern that meets the condition $S(7,i) \leq S(4,i)$ or $S(7,i) \leq S(2,i)$ the whole sampling sequence will only be presented to the neural network corresponding to $\underline{w}_7$ without the chance of presenting sampling pat-

terns to the neural network corresponding to $\underline{w}_4$ or $\underline{w}_2$. Therefore S_4 is finally reset to be zero. On the other hand, the reason why $S_1 = 9.588165$ is because $\sum_{i=1}^{12} s(7,i) = 9.588156 > \sum_{i=1}^{12} s(j,i)$ for $j \neq 7$.

4. EXPERIMENTAL RESULTS

We use the same data sets used in [29]-[30] to test the performance of the proposed method. We first briefly introduce the structure of Taiwanese Sign Language (TSL) that is the language most commonly used by the Taiwan deaf community for interpersonal communication. It uses hands, body and facial expressions to make signs. The manual components of TSL can be uniquely described by the shape of the hands, the location of the hands relative to the signer, and the orientations of the palms or the trajectory of the arms. There

Table III. The 10 typical types of arm movements involved in Taiwanese Sign Language: in the database used in the experiments, we merged the 6th and the 8th types into one single type and skipped the 9th type in the original database.

	Move- ment primes	Taiwane Sign Language		Move- ment primes	Taiwane Sign Language	
1	—	Boats,Stars Bug,Fast,...	merged into one single type	1	—	Boats,Stars Bug,Fast,...
2	⤿	Cobra,Rainbow Child,Famale,...		2	⤿	Cobra,Rainbow Child,Famale,...
3	└	Desk,Angle,...		3	└	Desk,Angle,...
4	↓	Enough,Ding Yu,...		4	↓	Enough,Ding Yu,...
5	▭	Paper,Chang, Rectangle...		5	▭	Paper,Chang, Rectangle...
6	⋈	Thunder,...		6	⋈	Mountain, Thunder,...
7	↶	Electric,...		7	↶	Electric,...
8	⋈	Mountain,Frog Grantmather,...		8	○	Fire,Coffee, Beautiful,Love,...
9	3	Ten thousand,...	skipped			
10	○	Fire,Coffee, Beautiful,Love,...				

are 51 fundamental hand shapes in TSL. Most sign words consist of either one or two hand shape. There are 22 typical locations and 6 different orientations of the palms in TSL. The semantics of two sign words can be different even if the above three parameters are quite the same. Motion trajectories play an important rule to make the classifications. By viewing TSL textbooks, one may observe that there are 10 typical types (as shown in Table III) of arm movements involved in TSL. However, we skip the 9th type of arm movements because among thousands of sign words, only one particular sign word involves this type of arm movements. In addition, the 6th and the 8th types are merged into one single type because a latter version of textbooks no longer amplifies the difference between these two types of arm movements. Therefore, our experiments were carried out for the recognition of the remaining 8 types of arm movements.

An ultrasonic 3D tracker was used to sense the 3D absolute position (x, y, z) in space. Ten persons were asked to sign the 8 typical arm movements shown in Table I. Each arm movement was signed for ten times for each person. The templates for each typical arm movement are tabulated in the third column of Table I. Since the templates of the O-type arm movement may consist of all 8 directions and it may start from either one direction we decided to use the following heuristic rules to recognize the O-type arm movement so as to avoid unnecessary computational load. The heuristic rules are as follows:

1. The angle subtended from the two vectors $(\underline{x}_2 - \underline{x}_1)$ and $(\underline{x}_N - \underline{x}_{N-1})$ must be smaller than a prespecified threshold.
2. The curvature of the trajectory must be larger than another prespecified threshold.

In our experiments we adopted the simplest method to generate templates for the ten arm movements. That is, the 8 synaptic weight vectors were set to be the corresponding 8 direction vectors. The database consisted of 800 sampling sequences. The final recognition results are tabulated in Table IV. The average correct recognition rate was 98.3%.

By further examining our experimental results, we found that the fourth type of arm movements sometimes may be misclassified to be the third type of arm movements. It is not a surprising result because the template of the fourth type of arm movements, $\{\underline{w}_7, \underline{w}_4\}$ (or $\{\underline{w}_7, \underline{w}_2\}$), is similar to the template of the third type of arm movements, $\{\underline{w}_7, \underline{w}_5\}$ (or $\{\underline{w}_7, \underline{w}_1\}$). It is probable that a user actually intends to sign the fourth type of arm movements, however, the second half of the stroke is too flat. Therefore the difference vectors in the second half of the sampling sequence may match the direction vector $\underline{w}_5$ (or $\underline{w}_1$) instead of $\underline{w}_4$ (or $\underline{w}_2$). Thus the arm movement will be recognized as the third type of arm movements because S_3 is larger than S_4 . To make it clearer, we

Table IV: The recognition performance: the average recognition rate is 98.3%.

Person/ Type	1	2	3	4	5	6	7	8	9	10	Average
1	100 %	100 %	100 %	90 %	100 %	100 %	100 %	100 %	100 %	100 %	99 %
2	90 %	100 %	100 %	100 %	90 %	100 %	100 %	100 %	100 %	100 %	98 %
3	100 %	100 %	100 %	100 %	100 %	100 %	100 %	100 %	100 %	100 %	100 %
4	100 %	90 %	80 %	100 %	100 %	100 %	90 %	100 %	100 %	100 %	96 %
5	100 %	100 %	100 %	100 %	100 %	100 %	100 %	100 %	100 %	100 %	100 %
6	100 %	100 %	100 %	100 %	100 %	100 %	100 %	100 %	100 %	100 %	100 %
7	100 %	90 %	100 %	100 %	100 %	100 %	100 %	90 %	100 %	100 %	98 %
8	100 %	100 %	100 %	80 %	100 %	100 %	100 %	80 %	100 %	100 %	96 %
Average	98.7 %	97.5 %	97.5 %	96.2 %	98.7 %	100 %	98.7 %	96.2 %	100 %	100 %	98.3 %

use Table V to explain why it is easy to confuse these two types of arm movements. In this case, $S_3 = 7.876590 > S_4 = 6.875818$ although the user actually signed the fourth type of arm movements. In addition, it is interesting to find the second largest value

of S_k is $S_1 = \sum_{i=1}^8 S(6, i) = 7.104775$ instead of S_4 .

Actually we also are not surprised about this result since the direction pointed by $\underline{w}_6$ is between the directions pointed by $\underline{w}_5$ (or $\underline{w}_4$) and $\underline{w}_7$.

5. CONCLUSIONS

In this paper, we solved the problem of recognizing 3D nonlinear arm movements by decomposing a complex spatio-temporal pattern recognition problem into simple 2D pattern matching problem. We first project 3D sequence onto 2D planes and then use a special pattern matching method without involving dynamic time warping procedure to compute the similarities between the unknown sequence and the templates in the vocabulary. At last the unknown arm movement is classified to be the corresponding arm movement with the highest accumulative similarity. By employing the 8-directional chain coding scheme we can easily generate the templates of the 8 types of arm movements. In addition, neural networks with linear activation functions are used to measure the local similarities between the difference vectors and the 8 direction vectors. In this manner we obviate substantial computation for time alignment that is usually required by the dynamic time warping algorithm. Compared to the fuzzy rule-based method proposed in [30], the neural-

network-based method outperforms the fuzzy rule-based method based on the comparisons of training procedures involved in step 4 and computational loads involved in step 5. In addition, 98.3% recognition rate on average can be achieved by the neural-network-based method.

ACKNOWLEDGEMENT

This work was supported by the MOE Program for Promoting Academic Excellent of Universities under the grant number EX-91-E-FA06-4-4.

REFERENCE

1. Azarbayejani A, Starner T, Horowitz B and Pentland A: Visually Controlled Graphics. IEEE Transactions on Pattern Recognition and Machine Intelligence, 1993; 15: 602-605.
2. Sawada H, Hata S and Hashimoto S: Gesture recognition for human-friendly interface in designer-consumer cooperate design system. International Workshop on Robot and Human Interaction, 1999; 400-405.
3. Segen J and Kumar S: Human-computer Interaction using gesture recognition and 3D hand tracking. IEEE, 1998; 188-192.
4. Kjeldsen R and Kender J: Visual Hand Recognition for Window System Control. Proc. International Conference on Automatic Face and Gesture Recognition, June, 1995; 184-188.
5. Kjeldsen R and Kender J: Towards the use of Gesture in Traditional User Interfaces. Proc. International Conference on Automatic Face and Gesture

Table V. An example to illustrate why the fourth type of arm movements may be misclassified to be the third type of arm movements: Since $\hat{S}_3 > \hat{S}_4$ the sampling sequence is misclassified to be the third type of arm movements although the user originally signed the fourth type of arm movements.

Arm Type	1	2	3	4	5	6	7	8
Template	{1},{2},{3}, {4},.....	{2,8},{8,6}, ,.....	{7,5},{7,1},	{7,4},{7,2},	{1,7,5},	{3,8,2,7},	{6,1,6},	{3,2,...,5,4}, ,.....
Data								
$\underline{x}_1$	S(6,1)= 0.807080	S(0,1)= 0.000000	S(7,1)= 0.996156	S(7,1)= 0.996156	S(0,1)= 0.000000	S(0,1)= 0.000000	S(0,1)= 0.000000	S(7,1)= 0.996156
$\underline{x}_2$	S(6,2)= 0.867890	S(0,2)= 0.000000	S(7,2)= 0.999571	S(7,2)= 0.999571	S(0,2)= 0.000000	S(0,2)= 0.000000	S(0,2)= 0.000000	S(7,2)= 0.999571
$\underline{x}_3$	S(6,3)= 0.938571	S(0,3)= 0.000000	S(7,3)= 0.979904	S(7,3)= 0.979904	S(0,3)= 0.000000	S(0,3)= 0.000000	S(0,3)= 0.000000	S(7,3)= 0.979904
$\underline{x}_4$	S(6,4)= 0.992706	S(0,4)= 0.000000	S(5,4)= 0.908564	S(7,4)= 0.788228	S(0,4)= 0.000000	S(0,4)= 0.000000	S(6,4)= 0.992706	S(6,4)= 0.992706
$\underline{x}_5$	S(6,5)= 0.907265	S(0,5)= 0.000000	S(5,5)= 0.993084	S(7,5)= 0.582875	S(0,5)= 0.000000	S(0,5)= 0.000000	S(6,5)= 0.907265	S(5,5)= 0.993084
$\underline{x}_6$	S(6,6)= 0.867441	S(0,6)= 0.000000	S(5,6)= 0.999598	S(4,6)= 0.839098	S(0,6)= 0.000000	S(0,6)= 0.000000	S(6,6)= 0.8674410	S(5,6)= 0.999598
$\underline{x}_7$	S(6,7)= 0.861598	S(0,7)= 0.000000	S(5,7)= 0.999867	S(4,7)= 0.845321	S(0,7)= 0.000000	S(0,7)= 0.000000	S(6,7)= 0.861598	S(5,7)= 0.999867
$\underline{x}_8$	S(6,8)= 0.862224	S(0,8)= 0.000000	S(5,8)= 0.999846	S(4,8)= 0.844665	S(0,8)= 0.000000	S(0,8)= 0.000000	S(6,8)= 0.862224	S(5,8)= 0.999846
Total	$S_1 =$ 7.104775	$S_2 =$ 0.000000	$S_3 =$ 7.876590	$S_4 =$ 6.875818	$S_5 =$ 0.000000	$S_6 =$ 0.000000	$S_7 =$ 0.000000	$S_8 =$ 0.000000

- Recognition, October, 1996; 151-156.
- Krueger MW: Artificial Reality II. Addison Wesley, 1991.
 - Maggioni C: GestureComputer – New Ways of Operating a Computer. Proc. International Conference on Automatic Face and Gesture Recognition, June, 1995; 166-171.
 - Rehg JM and Kanade T: DigitalEyes: Vision Based Human Hand Tracking. CMU Tech Report CMU-CS-93-220, 1993.
 - Segen J: Controlling Computers with Gloveless Gestures. Proceedings of Virtual Reality Systems, 1993.
 - Vincent VJ: Dwelving in the depth of the mind. Proc. Interface to Real and Virtual Worlds, 1991.
 - Weimer D and Ganapathy SK: Interaction Techniques using Hand Tracking and Speech Recognition. Multimedia Interface Design, ed. Blettner M and Dannenberg R, Addison Wesley, 1992; 109-126.
 - Wellner P: The DigitalDesk Calculator: Tangible Manipulation on a Desktop Display. Proc. ACM Symposium on User Interface Software and Technology, November, 1991.
 - Rubine D: Specifying gestures by example. ACM Computer Graphics, 1991; 25: 329-337.
 - Byong KK and Yang HS: Finger mouse and gesture recognition system as a new human computer interface. Computer & Graphics, 1997; 21: 555-561.
 - Minsky MR: Manipulating simulated objects with real-world gesture using a force and position sensitive screen. ACM Computer Graphics, 1984; 18: 195-203.
 - Kramer J and Leifer L: The talking glove: an expressive and receptive verbal communication aid for the deaf, deaf-blind, and non-vocal. Proc. of the third Annual Conf. On Computer Technology / Special Education / Rehabilitation, CA, Northridge, Oct, 1987; 335-340.
 - Fels SS and Hinton GE: Glove-Talk: a neural network interface between a Data-Glove and a speech synthesizer. IEEE Trans. On Neural Networks, 1993; 1: 2-8.
 - Kramer J and Leifer L: The talking glove: a speaking aid for non-vocal deaf and deaf-blind individuals. Proc. of the RESNA12th annual Conf., New Orleans, Louisiana, 1993; 471-472.
 - Su MC: A speaking aid for the deaf using neural networks for the deaf. Biomedical Engineering – Applications, Basis & Communications, 1996; 8: 33-39.
 - Su MC: A fuzzy rule-based approach to spatio-temporal hand gesture recognition. IEEE Trans. on Systems, Man, and Cybernetics-Part C: Applications and Reviews, May, 2000; 30: 276-281.
 - Waldron MB and Kim S: Isolated ASL sign recognition.

21. nition system for deaf persons. *IEEE trans. On Rehabilitation Engineering*, 1995; 2: 261-271.
22. Murakami K and Taguchi H: Gesture recognition using recurrent neural networks. *CHI'91 Proceedings*, 1991; 237-242.
23. Sagawa H, Sakou H and Abe M: Sign language translation using continuous DP matching. In *MVA'92- IAPR Workshop on Machine Vision Applications*, Tokyo, 1992.
24. Liong RA and Ouhyoung M: A real-time continuous gesture recognition system for sign language. *Third IEEE Int. Cnf. on Automatic Face and Gesture Recognition*, 1998; 558-567.
25. Liang RH and Ouhyoung M: A sign language recognition using hidden Markov model and context sensitive search. *ACM symposium on Virtual Reality Software and Technology*, 1996.
26. Nam Y and Wohn KY: Recognition of space-time hand-gestures using hidden Markov model. *ACM Symposium on Virtual Reality Software and Technology*, 1996; 51-58.
27. Rabiner R: A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. of the IEEE*, 1989; 77: 257-285.
28. Nam Y and Wohn KY: Recognition of hand gestures with 3D nonlinear arm movement. *Pattern Recognition Letters*, 1997; 18: 105-113.
29. Su MC, Chen YY, Wang KH, Tew CY and Huang H: 3D arm movement recognition using syntactic pattern recognition. *Artificial Intelligence in Engineering*, 2000; 14: 113-118.
30. Su MC, Zhao YX, Huang H and Chen HF: A Fuzzy Rule-Based Approach to Recognizing 3D Arm Movements. *IEEE Trans. on Neural Systems and Rehabilitation Engineering*, June, 2001; 9: 191-201.