JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 22, 1109-1123 (2006)

American Sign Language Recognition Using Multi-dimensional Hidden Markov Models*

HONGGANG WANG[1], MING C. LEU[2] AND CEMIL OZ[3]
[1]*Department of Industrial Engineering*
Purdue University
West Lafayette, IN 47907, U.S.A.
[2]*Department of Mechanical and Aerospace Engineering*
University of Missouri-Rolla
Rolla, MO 65409, U.S.A.
[3]*Department of Computer Engineering*
Sakarya University
54187 Sakarya, Turkey

An American Sign Language (ASL) recognition system developed based on multi-dimensional Hidden Markov Models (HMM) is presented in this paper. A Cyberglove™ sensory glove and a Flock of Birds® motion tracker are used to extract the features of ASL gestures. The data obtained from the strain gages in the glove defines the hand shape while the data from the motion tracker describes the trajectory of hand movement. Our objective is to continuously recognize ASL gestures using these input devices in real time. With the features extracted from the sensory data, we specify multi-dimensional states for ASL signs in the HMM processor. The system gives an average of 95% correct recognition for the 26 alphabets and 36 basic handshapes in the ASL after it has been trained with 8 samples. New gestures can be accommodated in the system with an interactive learning processor. The developed system forms a sound foundation for continuous recognition of ASL full signs.

Keywords: American sign language, ASL recognition, handshape gestures, hidden Markov model, data glove, motion tracker

1. INTRODUCTION

Sign language is a higly visual-spacial, linguistically complete and natural language. It is typically the first language and main means of communication for deaf individuals. The signers, however, still have serious problems of communicating with speaking persons, who are not sign users. The communication difficulty adversely affects the life and interpersonal relationships in the deaf community. Deaf individuals communicate with speaking people usually via interpreters or text writing. Although interpreters can help the communication between deaf and hearing persons, they are often expensive and have negative effect on independency and privacy. Note writing is used by many deaf people to communicate with someone who is seated nearby, but it is awkward while walking, standing at a distance, and when more than two persons are in a conversation.

Received August 16, 2005; accepted January 17, 2006.
Communicated by Jhing-Fa Wang, Pau-Choo Chung and Mark Billinghurst.
* This research was partially supported by the National Science Foundation award (DMI- 0079404) and the Ford Foundation grant, as well as by the Intelligent Systems Center at the University of Missouri-Rolla.

There is no universal sign language. Different countries use different sign languages. To help find a communication aid for deaf people, many researchers have been working on recognition of various sign languages, e.g. Australian, Japanese, Chinese, German, and American sign languages, etc. In general, based on how the features of gestures are extracted, the methodologies used in sign language recognition can be either video-based or device-based methods. In a video-based system, a video camera is usually used to acquire the image features of gestures, with an image processing system to classify and recognize those features. The main advantage of this approach is that the user does not need to wear any clumsy devices and facial expression can also be incorporated. However, the system may require complex image processing, which demands a large amount of data and slows the recognition rate. In a device-based system, some sensory devices are worn by a person to allow "measuring" the physical features of gestures, e.g. dimensions, angles, motions, and colors. Instrumented gloves, e.g. the Cyberglove [1, 2] – a glove equipped with strain gages for detecting finger bending, abductions and shape – have been conceived as a useful device for recognizing sign languages [1].

Research in sign language recognition started to appear in literature at the beginning of 1990s. Charahpayan and Marble [3] developed an image processing system to understand American Sign Language by interpreting the hand motion. Takahashi and Kishino [4] used a range classifier to recognize 46 Japanese Kana manual alphabets with a VPL Data Glove™. The hand gestures were simply encoded with data ranges for joint angles and hand orientations based on experiments. This system could recognize 30 out of 46 hand gestures correctly, but the remaining 16 signs could not be reliably recognized.

Artificial neural networks have been widely used in sign language recognition research. Murakami and Taguchi [5] investigated the use of recurrent neural nets for Japanese Sign Language recognition. Although it achieved a high accuracy of 96%, their system was limited only to 10 distinct signs. Kramer and Leifer [6, 7] developed an ASL fingerspelling system using a Cyberglove, with the use of neural networks for data segmentation, feature classifier, and sign recognition. Using a tree-structured neural classifying vector quantizer, a large neural network with 51 nodes was developed for the recognition of ASL alphabets. They claimed a recognition accuracy of 98.9% for the system. In the project Glove-Talk II, Fels and Hinton [8] used three neural networks and several input devices to translate hand gestures to speech. One neural network was used for the vowel/consonant decider, and two others were used for the individual vowel selector and the consonant selector. Their system is very extensive. However, the training time is as long as over 100 hours before the system is able to perform intelligibly. Waldron and Kim [9] used neural networks to recognize 14 ASL signs using different networks for handshapes and for hand orientation and position. The limited sign vocabulary was divided into a standard set of motions, which was recognized by another network. The overall accuracy of this system was 86%. Although the use of neural networks can provide reliable recognition of handshapes and a limited sign vocabulary, it is not a feasible method in the cases of a large sign vocabulary and recognition at the sentence level.

Since 1980s, Hidden Markov Models (HMM) have been widely used by the speech recognition research community [10]. With a well-founded mathematical basis and an efficient doubly stochastic process, impressive HMM-based recognizers have been developed for sign language recognition. With their intrinsic properties of implicit signal segmentation, HMMs provide a better solution in the case of continuous sign language recognition. Vogler and Metaxas [11] used HMMs for continuous ASL recognition with a vocabulary of 53 signs and a completely unconstrained sentence structure using videos. Wu *et al*. [12] recognized 26 words in Chinese Sign Language with 90% accuracy in an HMM-based recognition system. Grobel and Assan [13] used HMMs to recognize isolated signs. They extracted the features from recorded videos of signers wearing colored gloves and achieved 91.3% accuracy for a vocabulary of 262 signs. Lee and Xu [14] used HMMs to recognize ASL alphabets as a means of human-robot interface. They also tackled the issue of interactive learning using HMMs. They proposed a system using a one-dimensional HMM for recognition of 14 alphabets in ASL. The use of one-dimensional HMM limits the accuracy of the system. Also, their work did not consider the alphabets that depend on the hand orientation.

American Sign Language (ASL) is an efficient technique for communication among most of the 2 million deaf people in United States and Canada. ASL consists of about 6,000 signs for representing the commonly used words [15]. Wilbur [16] stated that most of signs in ASL could be considered as a combination of 36 basic handshapes. These 36 handshapes include most of ASL alphabets and their variations. Therefore, the recognition of ASL alphabets is not only important for spelling a person's name and the words which are not in the ASL vocabulary, but vital for further research on word and sentence recognition.

This paper presents a talking hand system using a Cyberglove and a Flock of Birds tracker as input devices. The ASL recognizer has been developed with a multi-dimensional HMM process. The system allows fast training and intelligent learning of new gestures. While the system can currently recognize 26 alphabets and 36 basic handshapes, it is extendable to developing a full-sign ASL recognizer. The paper is organized as follows. Section 2 describes the system setup. Section 3 discusses the data collection, processing and feature extraction of the system. Section 4 presents the details of the methodologies and techniques developed for the system. Testing results with the system are given in section 5.

2. SYSTEM SETUP

We use a right-hand Cyberglove™ (Fig. 1) to obtain the joint angle values. It has 18 sensors and the data recording frequency is up to 150 Hz. The data used in our ASL recognition system is from 15 sensors: 3 sensors for the thumb, 2 sensors for each of the other four fingers, and 4 sensors between each neighboring two of the five fingers. To track the position and orientation of the hand in 3-D space, the Flock of Birds® motion tracker (Fig. 2) mounted on the wrist is used. The receiver is located in a DC pulsed magnetic field with the effective range up to 8 feet around the transmitter. The measuring frequency is up to 144 Hz.

Open Inventor SDK is used as the software development tool for the 3-D scene rendering and interactive programming. Microsoft® Speech SDK is used for the programming of speech synthesis. The software is implemented using the object oriented programming technology, therefore it is well extendable.

Fig. 3 shows the overall structure of our system. The Cyberglove and the Flock of Birds tracker are connected to the computer system with two separate RS-232 serial ports.

Fig. 1. Cyberglove™ with 18 sensors.

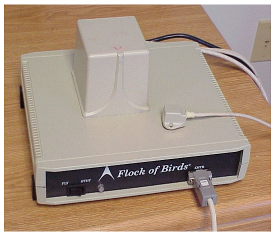

Fig. 2. Flock of Birds® 3-D motion tracker.

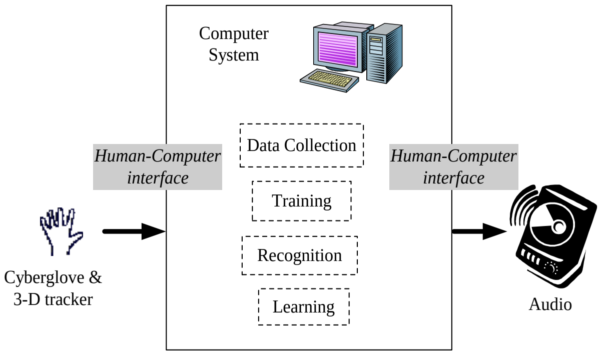

Fig. 3. The system schematic.

The data stream from these devices is retrieved and input to the software program. The software modules developed in the system include data collection, HMM training, recognition and learning processors. The recognition results can be output as text and audio speech.

3. CODING FEATURES OF ASL GESTURES

We define the features of ASL gestures with a 21-dimensional vector. The feature vector is extracted from the sensory data obtained from the Cyberglove and the Flock of Birds. The joint angles specify the hand shape and the data collected from the Birds' sensors describes the motion of hand. Thus the ASL gestures can be well defined with these sensory data.

In the field of speech signal processing, vector quantization (VQ) is a clustering method using the Euclidean distance measure. The goal of vector quantization is to find the set of quantization levels that minimizes the average deviation (also called distortion) over all samples. In 1980, Linde, Buzo, and Gray [17] proposed a VQ design algorithm (often called the LBG algorithm) based on a training sequence. The LBG algorithm is an iterative algorithm that splits the code vectors at each iteration, and finally chooses the optimal code vector. In our coding of geasture features, the codebook is generated using the LBG algorithm on the training samples of gesture data. A typical procedure in our system is as follows:

1. Assign a training sequence $\Psi = \{X_1, X_2, \ldots, X_M\}$ to a cluster that corresponds to an ASL sign. Each element of this sequence is a source vector $X_i = \{O_1, O_2, \ldots, O_k\}$; in our case, $k = 21$.
2. Compute the sample mean $\mu_0 = \frac{1}{M}\sum_{m=1}^{M} X_m$ of each cluster. (Note that μ_0 is a k-dimensional vector.) Compute the average distortion using mean squares, $D_{ave}^0 = \frac{1}{Mk}\sum_{m=1}^{M} \| X_m - \mu_0 \|^2$.
3. Split μ_0 to $\mu_1 = (1 + \varepsilon)\mu_0$ and $\mu_2 = (1 + \varepsilon)\mu_0$, where ε is a fixed small number, e.g. $0.01 < \varepsilon < 0.1$.
4. Calculate the average distortion of all samples in the cluster, with μ_1 and μ_2 respectively, i.e. $D_{ave}^1 = \frac{1}{Mk}\sum_{m=1}^{M} \| X_m - \mu_1 \|^2$ and $D_{ave}^2 = \frac{1}{Mk}\sum_{m=1}^{M} \| X_m - \mu_2 \|^2$.
5. Compare among the 3 average distortions D_{ave}^0, D_{ave}^1, D_{ave}^2 and choose the minimum one.
6. If D_{ave}^0 has the minimum value, stop. Otherwise do step 7.
7. Let $D_{ave}^0 = \min(D_{ave}^1, D_{ave}^2)$, and μ_0 equal the corresponding splitted values (i.e. μ_1 or μ_2). Repeat steps 3, 4, and 5 until the desired minimum distortion has been obtained.

Following the above procedure we get D_{ave}^0 as the code vector in the training database, which specifies the features of the trained ASL sign.

4. HIDDEN MARKOV MODEL APPROACH

A major challenge for an intelligent sign recognition system is isolating the boundaries within continuous signs. Hidden Markov models possess the ability to segment the data to its constituent signs implicitly and continuously. To achieve continuous recognition of ASL signs, we have developed a stochastic process based on the HMM algorithm. The data from the Cyberglove and Flock of Birds sensors is clustered for extracting the gesture features using the LBG algorithm. The parameters of the HMMs are defined with Gaussian distributions based on the training samples. The system can perform interactive training and online recognition of ASL gestures in real time. It can also intelligently learn new gestures, which can be used in recognition afterwards. This section describes how to design and build the HMM models. Fig. 4 shows the software architecture in our system.

4.1 Overview of Hidden Markov Models

Hidden Markov Models are a type of doubly stochastic models [10]. In developing our system we use the discrete HMM, where the observations are characterized as a finite set of symbols. A typical discrete HMM can be specified by N distinct states, M distinct observation symbols per state, and the probability matrices of state transitions, observation symbols, and the initial state of the HMM process. A discrete HMM can be presented as $\lambda = (A, B, \pi)$, where, $A = \{a_{ij}\}$ is a matrix of the state transition probability distribution, and a_{ij} specifies the probability that state S_i changes to state S_j ($1 \le i, j \le N$);

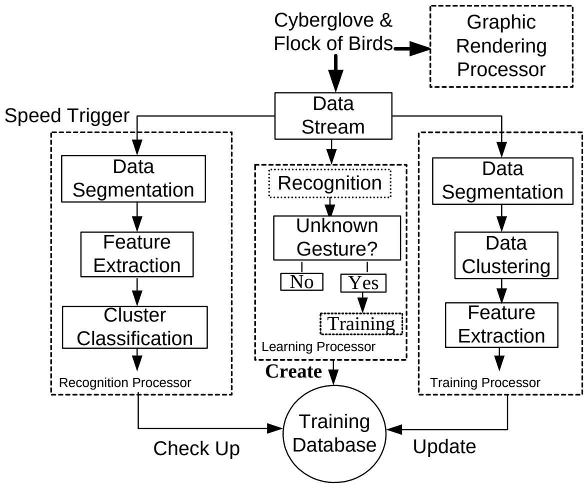

Fig. 4. The software architecture.

$B = \{b_j(k)\}$ is the observation symbol probability distribution in state S_j, and it represents the probability that the system will output an observable symbol O_k at the state S_j ($1 \leq j \leq N$; $1 \leq k \leq M$); π is a vector representing the probability that each state is the initial state of the HMM process.

There are three fundamental tasks in the HMM design: (1) Given an observation sequence, compute the probability with which those observations can be generated by a given HMM model; (2) Determine the most likely sequence of internal states in a given model which will give rise to a given observation sequence; (3) Adjust the model parameters of an HMM to optimize the probability distribution matrices for a given set of observations. The details of these tasks are described in reference [10].

4.2 Multi-dimensional HMM

We use a multi-dimensional HMM for better recognition rates compared with one-dimensional HMM. In this model, each dimension of the HMM state corresponds to the data from each sensor channel. The multiple data streams from the sensory glove and 3-D motion tracker are the inputs to the HMM process, and this raw data corresponds to the sequence of observations in the HMM. There are 21 channels of raw data, so we define one 21-dimensional state for each ASL alphabet in the HMM model. The probability matrices of the HMM (A, B, Л) are specified by clustering Gaussian distribution. We use a 5-Bakis-state HMM in our system (Fig. 5), i.e. one state in an HMM can reach the same state or one of next two states. For instance, 5 consecutive

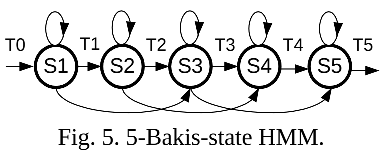

Fig. 5. 5-Bakis-state HMM.

readings obtained from time instant $T1$ to $T5$ are fed into the 5-state HMM process. The first reading at $T1$ is recognized as $S1$. The second reading data is then recognized as either the same state $S1$ (the sensory data does not change at time $T1$ and $T2$), or a new state $S2$ (the sensory data at $T2$ is different from the one at $T1$). In our case of alphabet recognition, these successive readings are typically classified as the same state if the user intends to gesture a certain ASL alphabet. Due to the large amount of words spelled by alphabets, we assume equal probabilities for the transition of internal states and the initial state of an HMM process. By clustering the Gaussian distribution, we define multiple B matrices for the dimensions of each HMM state.

Since we are using a discrete HMM, it is necessary to represent a gesture as a sequence of discrete symbols. We must preprocess the raw gesture data, which in our case are the values of the 15 bending and abduction angles, 3 position coordinates and 3 orientation angles, which are obtained from the Cyberglove and the Flock of Birds tracker. The retrieval rate of the raw data is 40Hz and the system segments the data stream based on a velocity trigger. The segmentation procedure starts when the hand is stationary, i.e. the speed of the hand of the user is below a preset threshold, and ends when the speed gets above a threshold.

4.3 Probability Function Calculation

A continuous Gaussian probability density function with mean μ and standard deviation σ can be defined as Eq. (1), where $x \in (-\infty, +\infty)$, μ is the clustered feature of a set of N observations (X_1, X_2, …, X_N), and σ is the distortion over the N samples. The cumulative Gaussian distribution function can be obtained as Eq. (2), where erf is the error function, $erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt.$ Since we cannot get an explicit solution based on Eq. (2), we use an approximation Eq. (3) to compute the integral of Gaussian distribution [18].

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma} \tag{1}$$

$$\int_{-\infty}^{x} P(t)dt = \int_{-\infty}^{0} \left(\frac{1}{\sqrt{2\pi}} e^{-\frac{T^2}{2}} \right) dT + \int_{0}^{x} \left(\frac{1}{\sqrt{2\pi}} e^{-\frac{T^2}{2}} \right) dT = \frac{1}{2}\left(1 + erf\left(\frac{x}{\sqrt{2}}\right)\right) \tag{2}$$

$$f(x) = \int_{-\infty}^{x} P(u)du = 1 - P(x) \times [C_1 \times t + C_2 \times t^2 + C_3 \times t^3] + \varepsilon(x) \tag{3}$$

where $P(u)$ is the standard Gaussian probability function, $t = \frac{1}{1 + 0.33267 \times x}$, C_1 = 0.4361836, C_2 = – 0.1201676, C_3 = 0.9372980, and $|\varepsilon(x)| < 1x10^{-5}$.

4.4 Gesture Recognition

After the system clusters the codebook in the training database, it is ready to recognize input signs. A recognition process dealing with the data acqusition, segmentation, feature extraction, and pattern recognition is implemented as follows. We have the sequence of observations, i.e. the raw data stream from the sensors as an input to the

HMM process. We need to determine the most likely sequence of internal states, $S_{opt} = \{S_{opt1}, S_{opt2}, \ldots, S_{optL}\}$for the given observation sequence $X = \{X_1, X_2, \ldots, X_L\}$ such that the probability $P[S_{opt}, X \mid \lambda] = \max(P[S, X \mid \lambda])$, where S is a random sequence of L internal states. Since we are using a discrete HMM model, we can find the state sequence S_{opt} by calculating the maximum probability, $\max(P[S, X_i \mid \lambda])$, where $S_{arb} \in \{S_1, S_2, \ldots, S_N\}$ and $X_i = \{X_1, X_2, \ldots, X_L\}$. In our case, for each of the internal states, we create the probability matrices B by inputting the sequence of observations (21-dimensional vectors of sensor readings) to the HMM system to calculate the integral of Gaussian function for those inputs based on the clustered codebook. The various readings of each sensor are then quantized as a single value that represents $b_j(k)$. Therefore, there are 21 observation probability matrices of B.

5. SYSTEM IMPLEMENTATION AND EVALUATION

To test the HMM-based gesture recognition system, we ask the user to test both the training and recognition processes. The testing is designed to quantitatively analyze the performance of the system. Tests are also done to verify the interactive learning feature of new gestures in this system.

The user signs the gestures of 26 ASL alphabets and 36 ASL handshapes by wearing a Cyberglove. The system performs data retrieval, feature extraction, and feature classification. A training database organizes the sample data and the statistical data of these samples.

5.1 ASL Recognition

To implement the continuous recogntion, the gesture features must be preprocessed in a continuous way. This system segments the raw data stream via a speed which tracks the hand speed using the motion tracker data. In reality, deaf people spell a word with ASL alphabets one by one. There is always some pauses or slow movements between two consecutive alphabets. It is applicable for the speed tracker to trigger the segmentation process. Whenever the hand motion is within a low speed zone the program will start to segment the data stream to extract the features of the current gesture. The gesture feature then is ready for further processing. Fig. 6 illustrates the data flow in the HMM recog- nition processor.

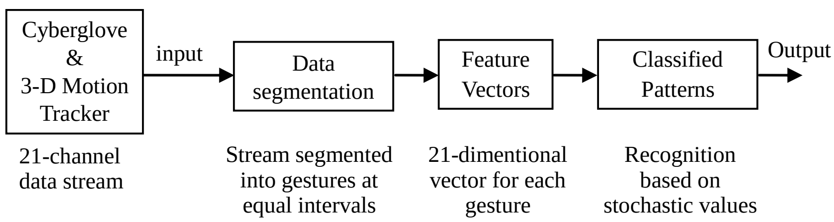

Fig. 6. Data flow in the HMM recognition processor.

5.2 Interactive Training/Learning

The training system adjusts the parameters of the HMM. This is the third major task in the HMM design. For each gesture, a series of trials cluster the features of the gesture with the sample readings, which are 21-dimensional vectors, and then get the Gaussian distributions as the probability B matrices in the HMM. We have developed a codebook using these clustered Gaussian distributions. The database of the codebook is a record-oriented file in that we store each HMM state as a record with a corresponding index. One record in the database generally includes two 21-dimensional vectors of mean values and standard deviations.

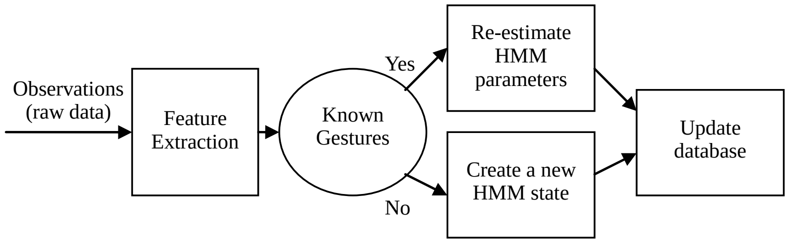

Fig. 7. The procedure of HMM training.

We have built a system that allows online training and learning of new gestures. As seen in Fig. 7, the typical procedure for the interactive, online training and learning is as follows:

(1) The user makes a certain gesture;
(2) The data stream is segmented and is given as an input to the HMM process, then it is classified as either a predefined gesture or an unknown gesture;
(3) If the gesture is a known gesture, the system recognizes it and updates the database simultaneously. If it is an unknown gesture, the system will add this record to the database and the user can define it as a new HMM state.

Several people were asked to test the training system with a few samples for each ASL alphabet and basic handshape. A preprocessing program handled the data acquisition and analysis before the data of features was input into the training processor. The training system then calculated the statistic values of the samples. Fig. 8 shows the clustering for the features of sign ‘A’. Each user trained the system with 8 samples for sign ‘A’. Each sample had 21 data that correspond to the data from different sensors. The training was done online. It took about 15 minutes for each user to train the HMM database with 8 samples.

5.3 Evaluation Results

Five users tested the recongtion performance of the system after training with several samples. Each user trained the system with 2, 3, 4, 5, 6, 7, and 8 samples, and then repeated the test of the recogntion processor after each training with those samples.

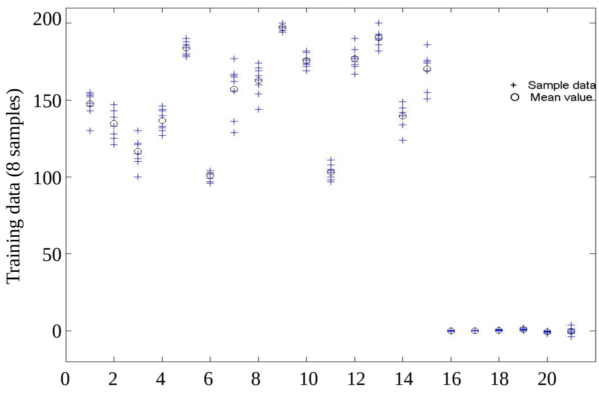

Fig. 8. Data clustering for the sign of alphabet ‘A’.

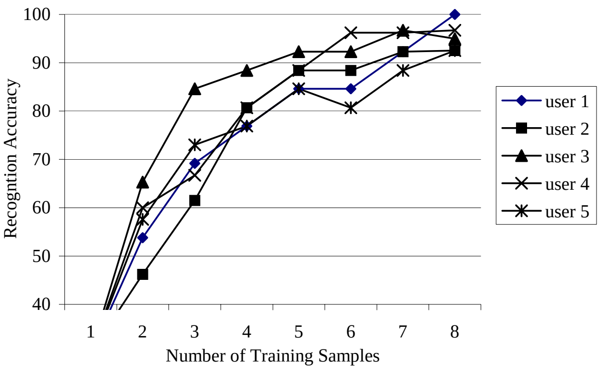

Fig. 9. Recognition accuracy vs. number of training samples.

Table 1. Recognition probabilities for ASL alphabet ‘A’.

%	A	B	C	D	E	F	G
Prob.	90.2	11.2	3.9	52.0	83.0	12.1	40.2
%	H	I	J	K	L	M	
Prob.	41.9	17.6	36.9	21.0	76.4	46.0	
%	N	O	P	Q	R	S	T
Prob.	10.7	19.1	14.5	35.7	31.8	17.6	3.6
%	U	V	W	X	Y	Z	
Prob.	11.8	15.4	4.1	17.2	31.1	37.0	

Fig. 9 shows the testing results for the 5 users. The recognition accuray was defined as the ratio of number of gestures correctly recognized to the total number of gestures tested. It is clear that the recognition accuracy increases when the system is trained with more samples. After being trained with 8 samples, the system can recognize ASL alphabets and handshapes by all users with an accuracy above 90%. Table 1 shows the recognition result of the gesture 'A' signed by a user with different probabilities of similarity to all alphabets. For this gesture, the system recognized it as the trained ASL sign 'A' in the database with 90.2% probability.

The system trained by different users performed in the gesture recognition with different accuracies. In Fig. 9, after user 1 trained the system with 8 samples, it could recognize all the 26 ASL alphabets and the 36 basic handshapes. After user 3 trained the system with 7 samples, the system recognized 96.7% of tested signs of alphabets and handshapes; after training with 8 samples, however, the recognition accuracy decreased to 95%. This was because the 8th sample data from user 3 had a relatively big variation, which influenced the overall distributions of some gestures and as a result the system misrecognized some signs. Table 2 shows the detailed results of the training and recogniton performance of the system for user 1.

Table 2. Training vs. recognition of the 26 ASL alphabets for user 1.

	Recognition Results with Different Number of Training Samples						
Alphabets	2	3	4	5	6	7	8
A	X	/	/	/	/	/	/
B	X	/	/	/	/	/	/
C	/	/	/	/	/	/	/
D	X	/	/	/	/	/	/
E	/	/	/	/	/	/	/
F	/	/	/	/	/	/	/
G	/	X	X	X	/	/	/
H	/	/	/	/	/	/	/
I	X	X	/	/	/	/	/
J	/	/	/	/	/	/	/
K	X	/	/	/	/	/	/
L	/	X	X	/	/	/	/
M	/	/	/	/	/	/	/
N	/	/	/	/	/	/	/
O	/	X	/	/	/	/	/
P	X	/	/	/	/	/	/
Q	/	/	/	/	/	/	/
R	X	X	X	/	/	/	/
S	/	/	/	/	X	/	/
T	X	X	X	X	/	/	/
U	X	X	X	X	X	X	/
V	X	/	/	/	/	/	/
W	X	X	X	X	/	/	/
X	X	/	/	/	/	/	/
Y	/	/	/	/	/	/	/
Z	/	/	/	/	/	/	/

NOTE: X: Unrecognized; /: Recognized.

Fig. 10 shows how a speed tracker monitors the hand motion while the system is running. When the tracker finds the velocity of the hand is below a threshold, in this case 0.05 (unit/second), it will send a message to the main program to request the recognition

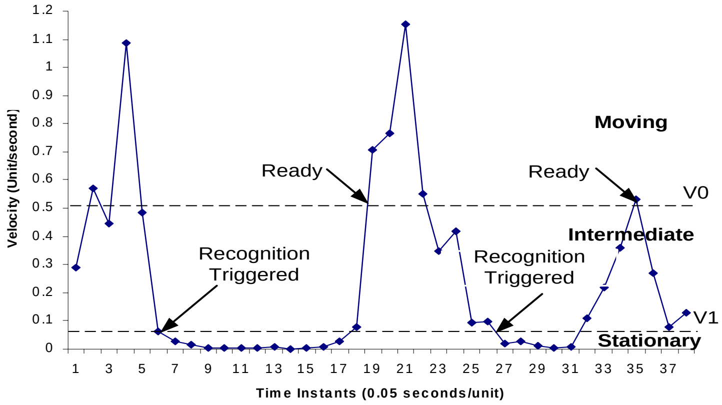

Fig. 10. Tracking hand motion by the system.

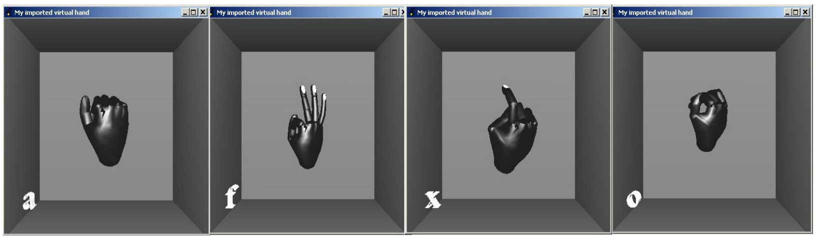

Fig. 11. Samples of recognized ASL alphabets.

process. Once the recognition is triggered, the communication between the speed tracker and the recongition processor will be turned off until a high speed reactivates it. Fig. 10 shows the reactivation velocity set at 0.5. Fig. 11 illustrates 4 recognized ASL alphabets, 'A', 'F', 'X', and 'O'.

5.4 Discussion

We visited the Missouri School for the Deaf to evaluate our system with the teaching staff there. It took about 15 minutes for a user to do testing including both the training and recognition parts. The results showed that 12, 19 and 23 of the 26 letters were recognized correctly after the system had been trained with 2, 3, and 4 samples, respectively, for the user. These results are close to those obtained from the testing with students in our research laboratory (see Fig. 9). The multi-dimensional HMMs as described offer better recognition performance than the one-dimensional models [14]. The online training and learning capabilities enable the system to be fast and intelligent. These capabilities also shorten the training time and improve the recognition performance via real-time update of the training database.

With the intrinsic time-varying process, HMMs are suitable for the full-sign recognition of ASL. Since most of the ASL signs can be gestured by a sequence of some of the 36 basic handshapes. The continuous signs can be segmented, with the basic handshapes in these signs extracted as the input to the HMM processor. Then the basic handshapes can be recognized and chained as the output of ASL words. The system is extendable to such a full-sign recognition system with the techniques described in this paper.

6. CONCLUSION

An ASL recognition system developed using a multi-dimensional HMM based method is described. The system can perform online training and real-time recognition of ASL alphabets and basic handshapes. The evaluation results show that the proposed method allows fast training and online learning of new gestures, and reliable recognition of the trained gestures afterwards. Future work will include extending the developed method to full-sign ASL recognition.

REFERENCES

1. J. Kramer, "Communication system for deaf, deaf-blind and non-vocal individuals using instrumented gloves," United States Patent No. 5047952, Virtual Technologies Co., 1991.
2. R. Carmel, C. Ullrich, and J. Silver, "VirtualHand v2.5-user's guide," Virtual Technologies Inc., 2001.
3. C. Charayaphan and A. Marble, "Image processing system for interpreting motion in American sign language," *Journal of Biomedical Engineering*, Vol. 14, 1992, pp. 419-425.
4. T. Takahashi and F. Kishino, "Gesture coding based in experiments with a hand gesture interface device," *ACM SIGCHI Bulletin*, Vol. 23, 1991, pp. 67-73.
5. K. Murakami and H. Taguchi, "Gesture recognition using recurrent neural networks," in *Proceedings of the Conference on Human Factors and Computing Systems*, 1991, pp. 237-242.
6. J. Kramer and L. J. Leifer, "A 'talking glove' for nonverbal deaf individuals," Technical Report No. CDR 19900312, Center for Design Research, Stanford University, U.S.A., 1990.
7. J. Kramer, "The TalkingGlove (RTM): hand-gesture-to-speech using an instrumented glove and a tree-structured neural classifying vector quantizer," Ph.D. Thesis, Department of Mechanical Engneering, Stanford University, U.S.A., 1996.
8. S. S. Fels and G. Hinton, "Glove-talk II: a neural-network interface which maps gestures to a parallel formant speech synthesizer controls," *IEEE Transactions on Neural Networks*, Vol. 8, 1997, pp. 977-984.
9. M. B. Waldron and S. Kim, "Isolated ASL recognition system for deaf persons," *IEEE Transactions on Rehabilitation Engineering*, Vol. 3, 1995, pp. 261-271.
10. L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," in *Proceedings of the IEEE*, Vol. 77, 1989, pp. 257-286.
11. C. Vogler and D. Metaxas, "Adapting hidden Markov models for ASL recognition by using three dimensional computer vision methods," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, Vol. 1, 1997, pp. 156-161.
12. J. Wu, W. Gao, Y. Song, W. Liu, and B. Pang, "A simple sign language recognition

system based on data glove," in *Proceedings of the 4th International Conference on Signal Processing*, Vol. 2, 1998, pp. 1257-1260.
13. K. Grobel and M. Assan, "Isolated sign language recognition using hidden Markov models," in *Proceedings of the IEEE International Conference on System, Man and Cybernetics*, 1996, pp. 162-167.
14. C. Lee and Y. Xu, "Online, interactive learning of gestures for human/robot interface," in *Proceedings of the IEEE International Conference on Robotics And Automation*, Vol. 4, 1996, pp. 2982-2987.
15. L. A. Martin and E. D. Sternberg, *American Sign Language Dictionary*, Revised edition, HarperCollins Publisher, New York, 1994.
16. R. B. Wilbur, *American Sign Language: Linguistic and Applied Dimensions*, 2nd ed., College Hill Publication, Boston, 1987.
17. Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Transactions on Communications*, Vol. 28, 1980, pp. 84-95.
18. M. Abramowitz and I. A. Stegum, *Handbook of Mathematical Functions: with Formulas, Graphs, and Tables*, National Bureau of Standards Applied Mathematics Series, Washington, Vol. 55, 1964, pp. 932.

Honggang Wang is a Ph.D. student in School of Industrial Engineering at Purdue University. He got his M.S. degree of Manufacturing Engineering from University of Missouri-Rolla in 2004 and his B.S. in Power Engineering from Shanghai Jiao Tong University, China in 1996.

Ming C. Leu is the Keith and Pat Bailey Missouri Distinguished Professor in Integrated Product Development, in the Department of Mechanical and Aerospace Engineering, University of Missouri-Rolla (UMR), where he holds the leadership positions of the Director of Intelligent Systems Center and also Director of Center for Aerospace Manufacturing Technologies. Prior to joining UMR, he was the Program Director for Manufacturing Machines and Equipment at the National Science Foundation (NSF) for three years until August 1999. For the NSF position he took a leave from the New Jersey Institute of Technology, where he had been the State Chair Professor in Manufacturing Productivity in the Department of Mechanical Engineering since his initial appointment in 1987. Before that he was on the faculty of the Sibley School of Mechanical and Aerospace Engineering, Cornell University. Professor Leu obtained his Ph.D. degree in 1981 from the University of California at Berkeley, his M.S. degree in 1977 from the Pennsylvania State

University, and his B.S. degree in 1972 from the National Taiwan University, all in Mechanical Engineering. He has published over 200 papers in refereed scientific and engineering journals and conference proceedings, and three United States patents (plus two pending). He has received several professional awards, including the Distinguished Service Award (ASME, 2004), AMAE Faculty Excellence Award (UMR, 2001 & 2004), Harlan J. Perlis Research Award (NJIT, 1993), Presidential Young Investigator Award (NSF, 1985), Ralph R. Teetor Education Award (SAE, 1985), and Wood Paper Award (FPRS, 1981). Also, he was elected to ASME Fellow in 1993 and is a member of the Sigma Xi, Tau Beta Pi, and Phi Kappa Phi honor societies.

Cemil Oz was born in Cankiri, Turkey in 1967. He received his B.S. degree in Electronic and Communication Engineering in 1989 from Yildiz Tecnical University and his M.S. degree in Electronics and Computer Education in 1993 from Marmara University, Istanbul. During the M.S. studies, he worked as a lecturer in Istanbul Technical University. In 1994, he begun his Ph.D. study in Electronics Engineering in Sakarya University. He has completed his Ph.D. in 1998. He worked as a research fellow in University of Missourı-Rolla, MO, U.S.A. He has been working as an Assistant Professor in Engineering Faculty, Department of Computer Engineering in Sakarya University. His research interests include robotics, vision, artificial intelligence, virtual reality and pattern recognition.