

A Real-time Continuous Alphabetic Sign Language to Speech Conversion VR System

Rung-Huei Liang

Ming Ouhyoung

Communications & Multimedia Lab., Computer Science and Information Engineering Dept.,
National Taiwan University, Taipei, Taiwan
email: ming@csie.ntu.edu.tw, FAX: 886-2-3628167

Abstract

Many ways of communications are used between human and computer, while using gesture is considered to be one of the most natural way in a virtual reality system. Because of its intuitiveness and its capability of helping the hearing impaired or speaking impaired, we develop a gesture recognition system. Considering the world-wide use of ASL (American Sign Language), this system focuses on the recognition of a continuous flow of alphabets in ASL to spell a word followed by the speech synthesis, and adopts a simple and efficient windowed template matching recognition strategy to achieve the goal of a real-time and continuous recognition. In addition to the abduction and the flex information in a gesture, we introduce a concept of contact-point into our system to solve the intrinsic ambiguities of some gestures in ASL. Five tact switches, served as contact-points and sensed by an analogue to digital board, are sewn on a glove cover to enhance the functions of a traditional data glove.

Keywords: gesture recognition, virtual reality applications

1. Introduction

The ways of communication have attracted much of interest for different reasons. Backtrack to the early 1900s, scientists had tried to solve the myth of communication among human beings and animals. In 1951, Keith and Hayes [1] had conducted an experiment in which they tried to teach a chimpanzee, named *Viki*, to speak English, however, only 4 words, *papa*, *mama*, *cup*, and *up*, could be uttered after more than six years of training. In the 1960s, Lieberman[1] discovered that a chimpanzee is incapable of human speech for anatomical reasons. Nevertheless, in 1976, Fouts, Chown, Kimble, and Couch[1] showed that a chimpanzee could respond correctly to signed commands by teaching the subject *Ali* to learn ASL. This project also showed that *Ali* could respond correctly to spatial arrangements and learn the grammar of the requested response consisting of an ordered sequence during training. The results above reveal the possibility to communicate with the chimpanzee and the human neonate by gestures rather than by other means.

Kunii [2] developed a system to translate natural language to sign language and then synthesised through corresponding computer animation, making much effort on the analysis of the grammar of natural

language and the synthesis of the sign language. This system aimed at the goal of the visualisation of ASL translated from English and facilitated communication between the hearing impaired and those with normal speaking capabilities. To control a virtual arm by gestures, Papper and Gigante[3] used a self-defined set of gestures to prove the usefulness in teleoperation in which interpreted gesture commands were used to control a robotic arm. It is reasonable to use a specifically defined set of gestures in this system since it was designed to facilitate experts to teleoperate. D. Thalmann[4] proposed a completely automatic grasping system for synthetic actors, including solving many problems in hand control such as kinematics of the hand and collision avoidance. Väänänen and Böhm[5] introduced GIVEN(Gesture driven Interaction as a human factor in Virtual Environment) structure and suggested the gesture recognition approach with neural networks for both static and dynamic gestures.

For the purpose of daily use of gestures, using ASL as Kunii did instead of a set of self-defined gestures is much reasonable. As our goal is to recognise gestures in ASL, not to synthesise animated sign language, the recognition approach using neural networks proposed by Väänänen and Böhm[5] seems to be a good candidate. However, the 10 input data(2x5 finger angles) of the input layer in the neural networks are not enough to solve the intrinsic ambiguities of several alphabetic gestures in ASL(fig. 2) which are also frequently used to compose a meaningful sentence. James Kramer[6] designed the Talking Glove with twenty-two sensors, including one abduction sensor per finger and thumb, to convert a finger-spelled word into synthesised speech. This system uses extremely thin strain gauges, instead of fiber-optic cables, to measure how much the fingers bend. This Talking Glove is connected to a neural-network gesture recogniser and the output is sent to a voice synthesiser for the hearing people. This award-winning product has received much attention and could help the disabled in communication.

Instead of a spelling language, the ASL is constructed by some rules, iconicity, and symbolisation, to form a mentally different horizon of linguistic logic. However, the implementation of all vocabulary in ASL is very difficult and needs more study. Our current goal is to be able to recognise the set of alphabets continuously using a simple windowed template matching method by thresholding the flex angles and retrieving the abduction information in the recognition space. Meanwhile, in addition to the abduction and the flex information, we introduce a concept of contact-point into our data glove, reducing the total number of sensors to only fifteen.

2. Strategies for continuous ASL alphabets recognition

The VPL Data Glove™ can report on every finger what angle it bends, i.e., 0 to 90 degrees. However, the angle combination of five fingers in ASL is limited, making determining the arbitrary angle in every dimension unnecessary. Moreover, we notice that alphabets in ASL do not consist of those requiring a finger to bend a precise angle, for example, bending all flex of four fingers to 90 degrees, metacarpophalangeal joint of the thumb to 45 degrees and interphalangeal joint of the thumb to 90 degrees to produce alphabet ‘S’(fig. 2) in ASL. When we gesture, what we keep in mind is to bend some finger or relax it, and bend some finger toward a certain direction until stopped by some contact-point. According to this kind of behaviour, our strategy for gesture recognition is simply to assign a binary state for every flexion, ‘1’ representing the bent state and ‘0’ the relaxed state. A flex is judged to be ‘1’ if the corresponding angle is over some threshold, and ‘0’ otherwise. Contact-points are added to solve the abduction ambiguity like ‘U’ and ‘V’(fig. 2) in ASL, and intrinsic similarities such as ‘M’, ‘N’ and ‘S’, which are very difficult to distinct either using the original Data Glove or combined with the neural networks approach.

We simply divide ASL alphabets into three groups: the ones needing the abduction or the contact-point information, the ones involving motion detection or orientation, and the others.(fig. 2) All these gestures are recognised by a template matching method using vectors received from the sensing glove as input data.

1. The ones needing the abduction or contact-point information: Alphabets in this group can be easily recognised using the concept of contact-point since we have attached some tact switches on the Data Glove(photo 1). For example, ‘M’, ‘N’ and ‘S’ can be distinguished by determining whether the switch near the interphalangeal joint of the little or the ring finger is touched(photo 2, 3, 4). ‘U’ and ‘V’ are different only in the abduction between the index and the middle finger, while adding a switch close to the inter-phalangeal joint of the middle finger can meet the requirement(photo 5 and 6, fig. 2). Similarly, attaching a switch near the tip of the index finger can make ‘C’ and ‘O’ different in this dimension(photo 7 and 8, fig. 2).
2. The ones involving motion detection or orientation: Two sets of alphabets are found identical if the time-dependent information is not available. We simply add a motion detection module to see if the whole hand has moved as a differentiating factor for ‘D’ and ‘Z’ and for ‘I’ and ‘J’(fig. 2). Two sets of alphabets need the orientation information of hand: ‘P’ and ‘K’, ‘G’ and ‘Q’(fig. 2). This group needs additional 3D information obtained from a 3D tracker(Polhemus Co.).
3. The basic ones: Since alphabets of this group do not require any additional information, we simply use the binary outputs obtained from the flex for the recognition process.

The transit problem

Automatic and continuous gesture recognition is desirable, and the whole recognition process should not be interrupted by a confirm-signal like key-pressing. However, if we want to input ‘B’ and ‘O’ sequentially, a ‘C’ will probably be recognised between them, that is, resulting a sequence: ‘B’, ‘C’ and ‘O’; this is called the transit problem.

```

\ \ \ \ \ \ \ \ \ \ S S A A A A A A S \ \ \ A A A A A A S \ \ \ B B B B B B B \ \ \ \ B
B B B B B B B B B B B \ \ C C C C C \ \ \ \ \ \ \ \ \ \ \ D D D D D D D D D D
D D D \ \ E E E E E E E \ \ \ \ E \ \ \ E E E E E E E E E E \ \ F F F F F F
F F F F F F F F F F F F F F F F F \ \

```

Figure 1. *Recognised intermediate data from the input sequence: A, A, B, B, B, C, D, E, E, F, where “\” is an unknown symbol.*

Window concept to solve the transit problem

To achieve the goal of automatic and continuous recognition while one is gesturing, the transit problem must be solved first; that is, to know where is the start point, the end point and a transit state of a character. We use the following strategies based on the data in Figure 1.

1. Sliding window: A sliding window at every time instance is kept(fig. 1). This window can memorise several characters that have been recognised in the past.
2. Template match: Using template matching, a vector can be recognised as a character or identified as an unknown one. Every vector in a window is either a character or an unknown symbol.

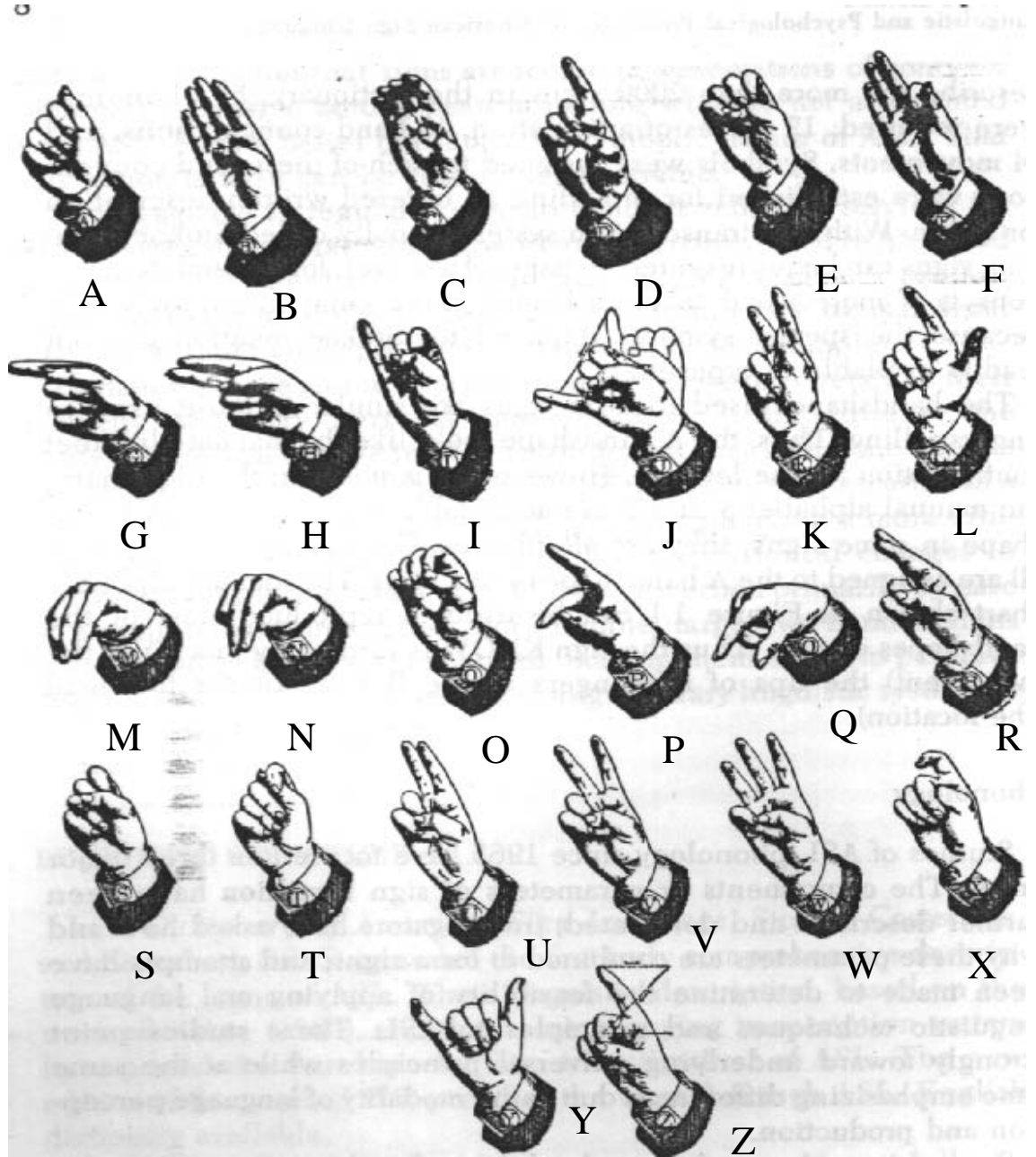


Figure 2. *The American Manual Alphabet*

3. Rule-based voting: Vote a most likely character according to the following rule: the majority one (may be a character or an unknown one) in a window wins. A confidence factor for this vector is kept and updated.

4. Accept when stable: A vector is accepted if it stably wins across several adjacent windows. The confidence factor is used to determine the stability.

The strategies described above can work because a classification can be done for every sample, resulting from template matching binary input data.

The situation described above is suitable for those completely matched vectors. When an input vector is not completely matched, a most likely gesture may be found. Here we choose the alphabet that maximise the inner product with the input vector.

3. System implementation

Figure 3 shows the relationship of several modules. The recognition module gets input data from two modules of device driver, the Data Glove and tact switches, and output the recognised alphabet to the speech module simultaneously. Not only a single alphabet but also a complete word can be pronounced by means of accumulating alphabets until a “complete” sign being recognised, then the word formed from these alphabets is sent to the speech module by the recognition module.

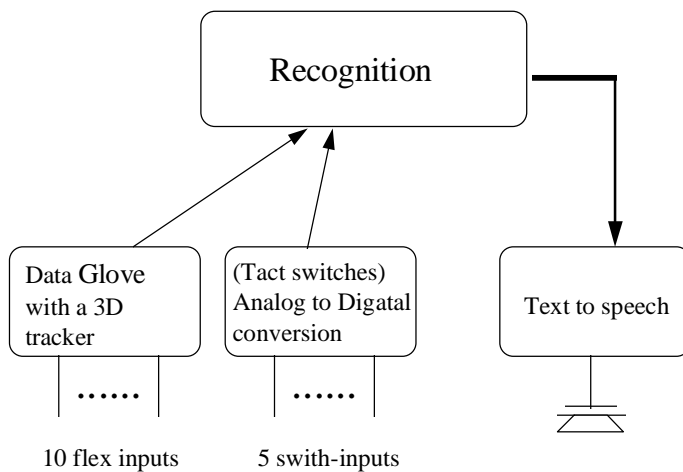


Figure 3. The system diagram

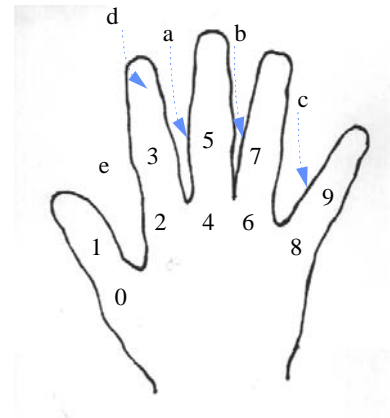


Figure 4. The ordering of the gesture vector

- Data Glove: Ten flex inputs are received from the Data Glove(VPL Co., DataGlove™ Model 4 System) module. It sends the first ten components of a gesture vector (Table 1.) by translating every flexion's value into a binary number according to some threshold which can be set at the registration phase. This module also captures the information of motion with a Polhemus 3D tracker.
- Tact switches: Five tact switches are sewn on another glove to put on the Data Glove. Each switch is supplied with +5 voltage and grounded on the other end, and connected to the Analogue to Digital board. Output is grounded as the switch is pressed, and high voltage otherwise. This module can

know whether a switch is triggered or not, representing whether a particular point on a finger is touched.

- Text to speech: Here we use a Sound Blaster Card(from Creative Lab.) to synthesise speech.
- Recognition: This module gets input data from the Data Glove, 3D tracker and tact switches, and followed by pattern-matching according to Table 1. The ordering of the gesture vector is shown in figure 4.

Table 1. The vectors of 26 alphabets, the numbers(in Hex.) in the first row indicate the ordering of the corresponding vector component. The components of the vector are defined by values from fig. 4.

0123456789abcde	0123456789abcde	0123456789abcde	0123456789abcde
A11111111100010	B11000000000000	C010101010110000	D110011111100000
E110101011100000	F110100000000000	G101011111110000	H100000111110000
I111111110000000	J111111110000000	K100000111100000	L000011111100000
M11111111100100	N11111111101000	O111101010110000	P110010111100000
Q00101111110000	R111000111100001	S111111111100000	T111111111100000
U110000111110000	V110000111100000	W110000001100000	X110111111100000
Y001111110000000	Z110011111100000		

This system uses an IBM compatible PC 486 as a platform. An analogue to digital board of 14-bit precision is used and the text-to-speech is done by the Sound Blaster board. Five tact switches, sensed by the AD board, are sewn on a glove cover to put on Data Glove(photo 1).

People observed that a Data Glove is not really necessary as a function control device in many virtual reality systems because of the cost of this product, and also because of its associated computation complexity of a neural-network recogniser or other recognition approach. In general, many tasks can be performed well and efficiently using other methods like clicking buttons of a 3D mouse, since the amount of commands to control a typical VR system is often limited to a small number. However, gesture recognition has its own advantage that can not simply be replaced by a mouse or keyboard in a task such as ASL recognition. The whole system can be made low cost if commercialisation is needed.

4. Results

Using the approach we have proposed, a continuous gesture recognition can be achieved, 15 to 20 samples can be recognised per second, under 19200 baud rate from input devices via RS232 interface. To solve the transit problem, we have tried windows of size 3, 4, and 5. It turns out that a window of size 4 is both stable and not tardy for gesture response, therefore, the upper bond for accepted rate is 3 to 4 characters per second. We found that the bottle-neck of input speed is not the sampling rate, but is limited by human gesture speed. Because our approach is simple and efficient, the gesture recognition rate can meet the normal speed of gesturing.

5. Future work

As we have mentioned in Section 1, alphabetic sign language is not frequently used among those hearing disabled. Therefore, the capability to recognise the common used vocabulary in ASL should be the final solution. Realisation of the vocabulary in ASL is a very challenging goal and our next step toward this goal is to recognise the iconic gestures which are the very first and the most natural way of gesture composition.

The difficulty of dynamic gesture recognition, which needs time-dependent information, involves not only the complexity of windowed method, but also, again, the transit problem. A window of much larger size is needed and the simple template matching method is not enough. Only if the transit problem can be solved efficiently that the dynamic gestures can be solved to a certain degree.

Moreover, a recognition system for the Chinese Sign Language is certainly our future work. There are estimated two million hearing impaired people in the US and according to this percentage, there may be more than ten million in mainland China. An efficient and low cost sign language translation system is indeed needed.

Currently our system has not been fully tested by a hearing disabled person, and part of the reason is to wait for the recognition of dynamic gestures.

Short term wish list for a successful ASL to speech conversion system: (a) A predictive spelling approach (e.g. typing “stude”, “student” will appear predictively) can be adopted that is very popular in a contemporary electronic dictionary. (b) Simple word processing commands should be implemented such as back space, delete, etc.

Acknowledgement: This project was partially supported by National Science Council under grant NSC-830425E002 140 and NSC-830408E002 006.

6. References

- [1] Roger Fouts, Gary Shapiro, Charity O’neil, “Studies of Linguistic Behaviour in Apes and Children,” pp. 163-185, *Understanding Language through Sign Language Research*, Academic Press, 1978.
- [2] Jintae Lee, Toshiyasu L. Kunii, “Computer Animated Visual Translation From Natural Language to Sign Language,” pp. 63-78, Vol. 4, *The Journal of Visualisation and Computer Animation*, 1993.
- [3] Michael J. Papper, Michael A. Gigante, “Using Gestures to Control a Virtual Arm,” pp. 237-246, *Virtual Reality System*, Academic Press, 1993.
- [4] R. M. Sanso, Daniel Thalmann, “A Hand Control and Automatic Grasping System for System Actors,” pp. C168-C177, Vol. 13, number 3, *Eurographics* 1994.
- [5] Kaisa Väänänen and Klaus Böhm, “Gesture Driven Interaction as a Human Factor in Virtual Environments - An Approach with Neural Networks,” pp. 93-106, *Virtual Reality System, Academic Press*, 1993.

- [6] J. Kramer, L. Leifer, "The Talking Glove: An Expressive and Receptive 'Verbal' Communication Aid for the Deaf, Dead-Blind, and Nonvocal," *Proceeding of Third Annual Conference on computer Technology/ Special Education/ Rehabilitation*, pp. 335-340, Northridge, CA, October, 1987.
- [7] J. Kramer, L. Leifer, "The Talking Glove: A Speaking Aid for Nonvocal Deaf and Deaf-Blind Individuals," *Proceeding of the RESNA 12th Annual Conference*, New Orleans, Louisiana, pp. 471-472, 1989.
- [8] Grigore Burdea, Philippe Coiffet, *Virtual Reality Technology*, pp. 276-278, John Wiley & Sons, 1994.



Photo 1. *Sensing Glove with five tact switches.*

Photo 2. *Gesture “M” with switch c pressed.*



Photo 3. *Gesture “N” with switch b pressed.*

Photo 4. *Gesture “S”.*

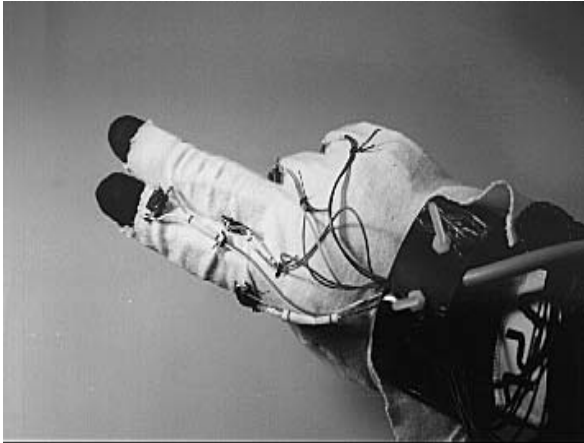


Photo 5. *Gesture “U” with switch a pressed.*

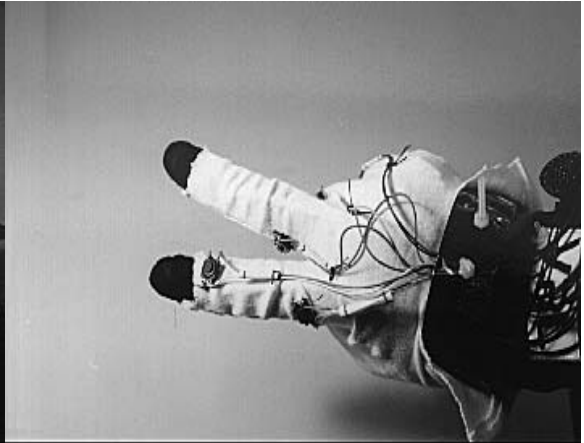


Photo 6. *Gesture “V”.*



Photo 7. *Gesture “C”.*



Photo 8. *Gesture “O” with switch d pressed.*